V Bibliographie

III Applications

IV Troubleshooting et sécurité

Table des matières

I Protocole TCP/IP

II Configuration

V Bibliographie

[1] TCP/IP Network administration
Graig HUNT (O'Reilly).

[2] TCP/IP Architecture, Protocoles, Applications
Douglas COMER (Inter-éditions).

[3] Practical Unix Security
Simson GARFINKEL & Gene SPAFFORD (O'Reilly).

[4] Conseil de Sécurité sur l'Administration de Machines Unix sur un Réseau TCP/IP
Jean-Luc ARCHIMBAUD (ftp anonyme sur ftp.urec.fr).

rassemble les services Internet (*Telnet, ftp, mail,...*) pour tout le réseau qu'elle protège. Ceci nécessite de créer des comptes dédiés à ces services, ces comptes sont accessibles par un certain nombre de personnes du réseau. Dans le cas de secteur très sensibles, c'est une possibilité de trier les personnes de confiances qui peuvent accéder à ces services qui ont tout l'intérêt d'Internet.

Le crontrôle de routage

Les firewalls ont l'avantage d'être très sûrs, mais le désavantage pour des secteurs moins sensibles la question de la sécurité de limiter considérablement l'accès au réseau Internet. Une solution logicielle consiste en l'utilisation d'un avantage du protocol IP. Un site peut être isolé de tout le réseau en ne désignant dans sa table de routage qu'un seul site de sortie. Donc aucun autre site que celui désigné ne connaîtra le réseau ainsi protégé. Toute fois ceci nécessite que toutes les machines du site soient configurées de la même manière.

Le contrôle d'accès

Limiter la table de routage est une solution mais elle ne convient pas une utilisation régulière du réseau. C'est pourquoi le contrôle d'accès peut être une bonne alternative. Le contrôled'accès consiste en un fichier qui est consulté par les machines et les routeurs, l'accès est accordé uniquement lorsque le mot de passe est bon. Le daemon TCPD permet de faire ce contrôle chaque demande d'accès un server (*ftp, rlogin, Telnet,...*) Il suffit de spécifier le path de chaque daemon comme tant celui de TCPD dans le fichier */etc/inetd.conf*.

IV-2.6 Conclusion

La sécurité réseau rejoind dans de nombreux cas la sécurité du système. On rappellera que la sécurité réseau tient tout d'abord l'établissement d'une politique et se poursuit par une surveillance régulière des fichiers de configuration.

NFS

Le système de gestion montage d'arborescences de fichiers à travers le réseau conçu par SUN se réfère à un fichier de configuration */etc/export* Chaque système peut préciser les machines auxquelles il permets de parcourir son arborescence. Il est donc conseillé de ne donner ces droits qu'à des machines de confiance. D'autre part, il est aussi important de se protéger contre les fichiers exécutables SUID ou SGID qui pourraient s'y trouver.

Les terminaux

Les terminaux sont souvent nombreux autour d'une machine. les fichiers */etc/ttys* ou */etc/tttab* contiennent la liste des écrans en précisant si le root peut se logger directement. En supprimant "secure" de toutes les lignes, l'administrateur devra donc se logger d'abord sous son nom avant de passer root par la commande su.

Des logiciels de trop

Les applications systat, tftp, link sont à désactiver en les supprimant du fichier */etc/inetd.conf*. Les commandes rx peuvent propager la brèche dans tout le réseau local. Si le niveau de confidentialité est élevé, alors il est conseillé de désactiver ces services.

IV-2.5 Les protections

Umask, une protection par défaut

Les fichiers privés par défaut sont un moyen de se préserver contre les intrusions. On a vu la commande umask qui permet de préciser la protection par défaut des fichiers et des répertoires créés.

Le cryptage

Le cryptage est une solution la confidentialité des données. Deux programmes de cryptage sont disponibles avec le système UNIX : "des" et "crypt". "*Des*" est un un programme propre UNIX dont l'algorithme a été conçu dans les années 70. "*Crypt*" est un programme dont l'algotithme est celui de la machine Enigma, il n'est donc pas très fiable car le mécanisme de décryptage est connu de tous.

Les firewalls

Face à ces nombreuses menaces, il peut sembler nécessaire d'isoler les réseaux locaux du réseau internationnal. Une solution efficace est la machine "firewalls". C'est une machine qui est placée à la place d'un routeur IP qui sépare deux réseaux ou le réseau local de Internet. On distingue donc les firewalls internes et les firewalls externes. La machine firewall a la fonction de serveur de noms et

Les commandes rx

Unix permet aux utilisateurs d'exécuter des commandes sur des machines distantes lorsque les fichiers hosts.equiv et *.rhosts* contiennent les coordonnées des machines et des utilisateurs de confiance. Dans ce cas, les utilisateurs de confiance n'ont pas besoin de préciser un mot de passe.

> */etc/hosts.equiv* doit contenir le nom des machines ou des groupes de machines qui peuvent accéder aux services rx sur la machine.
>
> *.rhosts* est un fichier qui se trouve dans la racine de chaque compte qui contient les comptes qui peuvent entrer sans mot de passe.

Le cas particulier de *rexec* est qu'elle demande le mot de passe pour qu'une commande soit exécute localement et elle renvoie le résultat du test. Ceci peut être une indication de base pour un programme recherchant un mot de passe. Il est donc vivement conseillé de supprimer rexec du fichier */etc/inetd.conf.*

Finger

Cette commande renvoie des informations stockées dans des fichiers réservés à l'utilisateur. Ces informations telles que le nom, l'adresse, le numéro de téléphone, ... Ces indications peuvent être très utiles pour un programme de décriptage de mots de passe. Dans des cas sensibles, il est conseillé de désactiver fingerd, le serveur répondant la demande *finger.*

Simple Mail Transfert Protocol

Les applications permettant de transférer du courier électronique sont nombreuses. Sendmail est la plus répendue. Elle fait référence un fichier d'alias : /usr/lib/aliases. Dans sa configuration par défaut, sendmail peut dérouler des commandes ou ouvrir des shells chez le destinataire sans préciser le mot de passe. Trois commandes sont dangereuses: *debug*, *wiz* et *kill*, il faut donc vérifier qu'elles ne sont pas valides par une session Telnet sur localhost smtp. si ces commandes sont valides alors il faut changer de version de sendmail. Vérifier que des alias decode faisant référence uudecode ne sont pas dans le fichier aliases.

Quelques lignes de */usr/lib/aliases* :

```
Francois.Borderies:borderie@isis
Borderies:borderie@isis
Jean-Christophe.Denis:denis@isis
Denis:denis@isis
```

Vérifier que qu'il n'y a pas de mot de passe pour un éventuel «magicien» dans le fichier *sendmail.cf* :

```
# let the wizarddo what he wants
OWstir68ods
```

est a remplacer par :

```
#do not let the wizard do anything
OW*
```

IV-2.4 Applications

UUCP et UUX

Ce programme est le premier utilitaire qui fut disponible sous UNIX pour que des machines distantes puissent communiquer. UUCP est Unix to Unix copy et UUX est Unix to Unix eXecute.

```
uucp /file imag!/file
```

Pour la copie d'un fichier sur une machine distante

```
uucp ensisun!/file imag!/file
```

Pour la copie de fichier entre deux machines distantes.

Soulignons que en Cshell ! est une commande permettant de rappeler la dernière commande exécutée, il faut donc préciser le caractére par \!.

Uux permet d'exécuter une commande sur une machine distante en lui précisant le fichier qu'elle doit prendre en entrée.

```
uux -system!commande<inputfile
```

Les ordres et l'adresse du fichier d'entrée sont tout d'abord stockés dans un Spooler en attendant d'être effectivement executés sur la machine.

Le programme uucico (Unix to Unix Copy in copy out) est chargé de relever le login du uucp et de le comparer celui du compte uucp de la machine distante qui se trouve dans */etc/passwd*. Le fichier exécutable uucp est SUID uucp (nom de l'utilisateur particulier) ceci limite l'accès au compte uucp et aux fichiers world writable ou world readable.

Dans la version 2, il existe des fichiers de configuration, on en retiendra trois : USERFILE, l.cmds et l.sys.

L.sys : contient les coordonnées des machines et des personnes qui peuvent accéder au service uucp d'une machine distante.

L.cmds : contient le PATH local au compte uucp ce qui permet de limiter les exécutables accessibles, suivit des applications qui peuvent avoir accès aux services de uucp, on inclut souvent *rmail, rnews, lpr, who, finger ...*

USERFILE : spécifie les répertoires qui peuvent être ouverts par uucp, si la machine distante doit rappeler son identité et quels fichiers peuvent être transférés.

Précautions : ces fichiers sont a protéger pour que personne puisse les lire à part le root. Le compte uucp doit contenir le moins de répertoires possibles, il faut donner un login par machine distante et limiter les commandes utilisables.

IV-2.3 Les fichiers

Pour avoir une vue correcte de la vulnérabilité des fichiers, il est important de rappeler comment un fichier est représente dans le système UNIX. Un fichier est stock sur disque avec un certain nombres d'informations gérées par UNIX : l'emplacement sur le disque, le type, le taille, ctime, mtime, atime, owner, group. La commande *ls* permet d'accéder à ces informations.

```
-rwx------ 1 denis 245 Apr 2 14:26 .xsession~
drwxr-xr-x 6 denis 512 Nov 13 1992 ALGO

- : plain file
d : directory
c : device (printer tty,...)
b : lock device (disk, tape,...)
l : link (BSD)
s : socket (BSD)
= : FIFO (Sys V)
r : read
w : write
x : execute
s : set mode
```

Chmod et Umask sont des commandes UNIX permettant de spécifier ou de modifier le mode des fichiers. Umask est la commande qui précise le mode par défaut lors de la création d'un fichier ou d'un répertoire. Ceci est spécifié dans le fichier .login ou .profile ou .cshrc. On soulignera que cette commande s'exécute comme cd sous le shell courant. Umask fait un & avec le masque où tous les bits sont à un.

SUID et SGID

L'intérêt d'UNIX est que tout a une représentation de fichier. On a pu remarquer que certain fichiers peuvent être exécutables et donc provoquer la naissance d'un process fils du shell qui a ouvert le fichier exécutable. On peut donc imaginer que des commandes (donc des fichiers excutables) aient plus de privilèges que les utilisateurs normaux, ils sont SUID ou SGID. C'est le cas de "*passwd*" qui a le droit de modifier le le fichier */etc/passwd*. Tout fichier peut devenir SUID ou SGID. Ces fichiers sont représentés par le masque du type : -rwsr-s-r-t.

La dernière lettre signifie que le fichier est "sticky", après sont exécution, il ne sera pas enlevé de la zone mémoire qui lui est affecté ce qui permet d'y accéder très rapidement. Les premières versions d'UNIX permettaient de faire une copie du shell avec le privilège SUID au nom du root, cette brêche a été rapidement colmatée et on ne peut plus faire de copie SUID de shell si l'on est pas déjà root. Il est donc important pour l'administrateur de connaître tous les fichiers SUID présents.

```
#find /-perm -002000 -o -perm -004000 -type f -print
```

Lors du montage des fichiers distants, il est aussi important qu'aucun fichier ne soit exécutables SUID ou SGID en BSD, la commande est :

```
#mount -o -nosuid imag:/athena /usr/athena.
```

nombre de fichiers. Ils peuvent ainsi travailler en commun sur ceux-ci sans se soucier des droits d'accès. La correspondance entre le numéro de GID, le mot de passe du groupe et les utilisateurs appartenant ce groupe est précise dans le fichier */etc/group*.

```
install:*:63:cassagne,eudes,laforgue,richier,jean,nicollin,lenne,martinet,
santana,challier,delaunay,rouverol
```

Des comptes spéciaux

Le compte superuser, appelé aussi root, permet l'accès à tous les fichiers. Ce compte est celui partir duquel l'administrateur configure et contrôle le système et le réseau. Un group est réservé aux personnes ayant accès à ce compte, il a pour GID 0 et porte souvent le nom de "wheel". On peut comprendre que ce compte soit la première cible des intrus qui essaient d'accéder à ce privilège.

```
wheel:*:0:cassagne,waille,laforgue,richier,root,eudes,martinet,challier,de
launay,jean
```

Le compte UUCP, est un compte lié à un programme que nous détaillerons plus tard du point de vue de la scurité.

Chaque démon est un utilisateur particulier, il possède un UID. Les démons sont des programmes lancés au moment du boot, ils sont souvent des serveurs.

La commande «su» est celle qui permet de devenir super-user. Elle demande une validation de mot de passe. UNIX system V n'admet qu'un mot de passe pour devenir superuser, ceci oblige donc une circulation d'un mot de passe sensible ce qui peut compromettre la sécurité du réseau. UNIX BSD propose chaque root de devenir superuser en utilisant son propre mot de passe. SU rend compte de toute tentative de login erronée dans le fichier /usr/adm/messages ce qui permet à l'administrateur d'être averti de toute tentative d'intrusion.

La protection passe par une surveillance, les administrateurs doivent scruter les fichiers clés et les fichiers relevant les erreurs comme celui vu précédement, ou les fichiers où toutes le actions ont été enregistrées comme */etc/wtmp*.

Le problème des comptes publiques

On comprend l'utilité des comptes de démonstration ou les comptes à usage publique. Toutefois ceux-ci représentent une brêche dans la sécurité d'un système. Une solution consiste à imposer une configuration restrictive. Un interperteur shell restrictif permet de réduire le nombre de commandes accessibles par ces comptes et une sortie automatique du shell lors de tentatives illicites. Le fichier .profile de ce compte permet de le configurer en précisant les terminaux qui permettent de d'utiliser ce compte, le shell restrictif, l'interdiction des ports. Cette solution est toutefois fragile car un utilisateur déjà loggé sous un autre compte pourra y accéder avec un shell normal.

- Tout compte doit comporter un password mis à part quelques uns qui sont alors très restrcifs dans leur utilisation.
- Un mot de passe ne doit être en aucun cas inspiré des informations disponibles par la commande vue précédement.
- Un mot de passe ne doit pas être tiré d'un dictionnaire surtout de la langue anglaise.
- Choisir une phrase de huit mot ou plus (Unix ne crypt que huit lettres du mot de passe mais il est possible d'en taper plus).
- Insérer des caractères spéciaux et des majuscules.

Des outils existent comme "*npasswd*" ou "*passwd+*" qui permettent de vérifier la validité d'un mot de passe.

Npasswd : ce programme permet à l'utilisateur de choisir son mot de passe mais il doit respecter certains critères testés par le programme. Il élimine les répétitions de caractères, les caractères impropres, les mots en minuscules, les mots en majuscules, tous les mots se rapportant aux informations comtenues dans les fichiers lus par *"finger*" et les mots appartenant à un dictionnaire. "*Passwd+*" propose un jeu de tests plus important et peut être configuré par un langage assez complet.

• Les mots de passe doivent être changés régulièrement, Une procédure de surveillance des mots de passe est à établir. En voici une : copier le fichier */etc/passwd* sur une bande, 30 jours plus tard, comparer le fichier courant avec celui sauvé, avertir les utilisateurs qu'ils doivent changer leur mot de passe dans les trente jours, vérifier 21 jours après, prévoir un deuxième avertissement, au terme de la période des 60 jours, faire une dernière vérification et supprimer les comptes rebelles. (L' utilisateur pourra toujours retrouver son compte sur une bande de backup.) Cette méthode élimine rapidement les comptes dormants, les plus compromettants pour la sécurité d'une machine.

IV-2.2 UID, GID

Les comptes usuels

Les utilisateurs ont accès à la machine si ils ont un numéros de 16 bit qui les identifie dans le noyau. Ce numéro est le User Identification (UID). La correspondance entre nom, le mot de passe et le UID est précise dans le fichier /etc/passwd.

```
denis:iORT/teYQQlko:15033:10510:Jean-Christophe Denis,,,:/h/isis/ensimag/
students/denis:/bin/tcsh
```

15033 est le numéro d'utilisateur et 10510 est le numéro de groupe.

Il se peut que deux utilisateurs peuvent avoir le même numro d'identifcation, dans ce cas le noyau les voit comme la même personne. Ceci peut poser des problèmes si un intrus veut se faire passer pour quelqu'un. Les UID ont des valeurs réserves, les entiers entre 0 et 9 identifient des fonctions système, les utilisateurs sont identifiés par des valeurs supérieures 20.

Les utilisateurs sont regroupés, chaque groupe est identifié par un numéro : le Group Identification (GID). L'intérêt de grouper les utilisateurs est de leur donner les mêmes droits sur un certain

Les primitives sont :

dst destination ------------------ to destination host
src source ------------------------ from source host
host host ------------------------- to or from host
between host1 host2 ----------- to or from host1 and host2
dstnet destination --------------- to destination network
srcnet source -------------------- to source network
dstport destination -------------- to destination port
srcport source ------------------- to source port
proto protocol ------------------- of protocol type (icmp, udp, tcp)

Chaque primitive est un filtre que l'on peut combiner pour obtenir des filtres plus sophistiqués.

IV-1.9 Conclusion

La résolution des problèmes de réseau nécessitent des connaissances de base sur le fonctionnement du protocol TCP/IP et plus généralement du modèle OSI. On a vu que les erreurs sont souvent localisées dans les fichiers de configuration des applications, des tables de routage, des serveurs de noms. Il peut arriver que le protocol lui même soit en cause, et dans ce cas des outils spécifiques tels que Tcpdump ou Etherfind permettent de visualiser en détail les paquets transmis entre deux machines.

IV-2 Sécurité

« It was not designed from the start to be secure. It was designed with the necessary characteristics to make security serviceable. »

Dennis Ritchie.

Les réseaux sont basés sur le principe de l'autoroute, tout le monde y a accès et c'est à chacun de se protéger. Pour que tout soit clair, l'administrateur doit prévoir une politique de sécurité précisant les droits d'accès, les services réseau disponibles, les précautions à prendre, les procédures à suivre lorsqu'une faille a été décelée dans la protection du réseau et des méthodes de restauration de données. Le rôle de l'administrateur consiste aussi en la diffusion des information relatives au réseau par les "mailing lists". Des informations intéressantes sont diffusées régulièrement par le CERT et la DDN. Via les mails list "CERT advisories" et "DDN security bulletin".

La protection des données et des applications dépend de leur niveau de confidentialité. La protection la plus sûr est l'isolation physique du réseau. Toutefois les versions successives d'UNIX ont proposées des solutions aux problèmes d'accès et aux menaces logicielles.

IV-2.1 Passwords

Les mots de passes sont à la base de la sécurité d'une machine. Il est donc important de se préserver contre les mauvaises âmes qui cherchent à ouvrir les portes du système. Des indices importants sur les utilisateurs sont accessibles par la commande «*finger*», il est donc vivement recommandé de suivre des précautions élémentaires.

```
Authoritative answers can be found from:
PRINCETON.EDU nameserver = PRINCETON.EDU
PRINCETON.EDU nameserver = NS.CWRU.EDU
PRINCETON.EDU nameserver = NISC.JVNC.NET
PRINCETON.EDU internet address = 128.112.128.1
NS.CWRU.EDU internet address = 129.22.4.1
NISC.JVNC.NET internet address = 128.121.50.7
> server PRINCETON.EDU
Default Server: PRINCETON.EDU
Address: 128.112.128.1

> set type=ANY
> ls
Server: PRINCETON.EDU
Address: 128.112.128.1

ls.imag.fr preference = 10, mail exchanger = imag.imag.fr
ls.imag.fr preference = 50, mail exchanger = babbage.imag.fr
ls.imag.fr preference = 80, mail exchanger = brahma.imag.fr
ls.imag.fr preference = 100, mail exchanger = sophia.inria.fr
imag.imag.fr internet address = 129.88.32.1
babbage.imag.fr internet address = 129.88.31.3
babbage.imag.fr internet address = 192.33.172.34
brahma.imag.fr internet address = 129.88.32.41
sophia.inria.fr internet address = 138.96.32.20

Nslookup fonctionne selon plusieurs modes, ci-dessus on a vu l'utilisation de
deux d'entre-elles. Le changement de mode se fait en affectant à la varible
type les valeurs suivantes :

NS : pour obtenir les enrégistrements correspondants aux Servuers de Noms.
ANY : pour obtenir tous les enrégistrements à propos d'une machine.
SOA : pour obtenir les enrégistrements à propos des domaine de «start of
authority».
HINFO : pour obtenir les informations à propos d'une machine précise.
MINFO : pour obtenir les informations à propos des mail list.
UINFO : pour obtenir les informations à propos des utilisateurs.
MX : pour afficher les enrégistrements à propos des échanges de mail.
...
```

Les autres modes sont détaillés dans le RFC-1035.

Par rapport à nslookup, dig a l'avantage de pouvoir être utilisé pour convertir les adresses IP en noms et inversement en spécifiant l'option -x.

IV-1.8 Les problèmes de protocols

TCP/IP pause souvant des problèmes dans sa configuration et les outils présentés jusqu'à maintenant suffisent pour trouver une solution. Plus rarement, le protocol lui-même peut être en cause, pour cela, il est nécessaire d'analyser les paquets transmis d'une machine à l'autre. Dans un environnement UNIX, les analyseurs de protocoles les plus utilisés sont tcpdump (UNIX BSD) et etherfind (Sun-OS).

Les analyseurs de protocoles proposent de filtrer les paquets spécifiés par l'utilisateur. Le principe est de permettre aux paquets de remonter jusqu'à la plus haute couche du protocole pour qu'une application puisse visualiser le contenu des paquets (entête et données).

```
~> /local/etc/traceroute sumex-aim.stanford.edu
traceroute to sumex-aim.stanford.edu (36.44.0.6), 30 hops max, 40 byte
packets
 1 aramis-campus (129.88.32.254) 3 ms 4 ms 3 ms
 2 aramis-ensl (192.42.102.5) 18 ms 23 ms 18 ms
 3 * * *
 4 * * *
 5 * * *
 6 * * *
 7 * * *
 8 * * *
 9 * * *
10 * * *
11 * * *
12 * * *
13 * * *
14 * * *
15 * * *
16 * * *
17 * * t3-0.New-York-cnss32.t3.ans.net (140.222.32.1) 131 ms
18 t3-1.Cleveland-cnss40.t3.ans.net (140.222.40.2) 153 ms 165 ms 136 ms
19 * t3-2.Chicago-cnss24.t3.ans.net (140.222.24.3) 169 ms 204 ms
20 t3-1.San-Francisco-cnss8.t3.ans.net (140.222.8.2) 189 ms 204 ms 199 ms
21 t3-0.San-Francisco-cnss9.t3.ans.net (140.222.9.1) 210 ms 231 ms 206 ms
22 t3-0.enss128.t3.ans.net (140.222.128.1) 206 ms 187 ms 213 ms
23 SU-CM.BARRNET.NET (192.31.48.200) 226 ms 187 ms 194 ms
24 s101-gateway.Stanford.EDU (36.56.0.41) 222 ms 190 ms 227 ms
25 SUMEX-AIM.Stanford.EDU (36.44.0.6) 257 ms 189 ms 190 ms
~>
```

Lorsque les 30 paquets se voient retourner des astérisques c'est que les tables de routages ne sont pas bien configurées. Il s'agit alors de contacter les responsables de la dernière machine qui a répondu pour réviser la configuration de leur routeur.

IV-1.7 Tester le serveur de nom

Les commandes *"nslookup*" et "*dig*" servent à consulter l'état de la table du serveur de nom. Ce dernier est en cause lorsque le message "unknown host" est retourné lors d'une tentative de connexion. Cette commande est très utile lorsque l'administrateur veut vérifier que la machine distante est bien configurée.

```
~> nslookup
Default Server: imag.imag.fr
Address: 129.88.32.1

> set type=NS
> princeton.edu
Server: imag.imag.fr
Address: 129.88.32.1

Non-authoritative answer:
princeton.edu nameserver = PRINCETON.EDU
princeton.edu nameserver = NS.CWRU.EDU
princeton.edu nameserver = NISC.JVNC.NET
```

```
129.88.32.0 129.88.32.1 U 376 2182362 le0
129.88.33.0 129.88.32.254 UG 10 59322 le0
129.88.41.0 129.88.32.55 UG 3 37240 le0
129.88.34.0 129.88.32.254 UG 2 45202 le0
129.88.2.0 129.88.32.254 UG 0 10028 le0
129.88.42.0 129.88.32.19 UG 4 72445 le0
129.88.59.0 129.88.32.159 UG 2 34573 le0
129.88.51.0 129.88.32.151 UG 2 14318 le0
129.88.100.0 129.88.32.254 UG 7 252082 le0
152.77.0.0 129.88.32.254 UG 2 15459 le0
192.33.174.0 129.88.32.254 UG 12 64083 le0
130.190.0.0 129.88.32.254 UG 5 70716 le0
129.88.38.0 129.88.32.254 UG 29 3773129 le0
129.88.110.0 129.88.32.254 UG 0 9767 le0
192.33.175.0 129.88.32.254 UG 6 23663 le0
129.88.31.0 129.88.32.254 UG 1 3411 le0
129.88.39.0 129.88.32.254 UG 16 124978 le0
129.88.111.0 129.88.32.254 UG 0 60 le0
```

L'option -r permet de lire les tables de routage.

On peut lire la destination, le gateway qu'il faut emprunter pour atteindre le réseau désigné dans le premier champ.

IV-1.5 Tester les mises à jour

Le démon de routage en service sur toute machine en principe est en attente des informations de mise à jour envoyées par les autres machines du réseau local. Les paquets de message de mise à jour sont des RIP. Pour consulter ces messages, on utilise la commande *"ripquery*".

```
~> /local/etc/ripquery aramis-campus
444 bytes from aramis-campus(129.88.32.254):
 imag-geta(129.88.120.0), metric 3
 rfmq-domain(129.88.111.0), metric 3
 imag-rfmq(129.88.110.0), metric 3
 net-bullimag0(129.88.100.0), metric 3
 imag-igei(129.88.31.0), metric 1
 imag-cicg(129.88.2.0), metric 1
...
```

Ce sont les routes qui sont valides au moment ou la commande a été exécutée. Si elles diffèrent de celles contenues dans la configuration des tables de routage.

IV-1.6 Tester les routes ouvertes par une connexion

La commande *"traceroute*" décrit la route empruntée par les UDP pour aller d'une machine à une autre. Traceroute fonctionne en envoyant des paquets UDP avec une durée de vie incrémentée de un a chaque fois jusqu'à ce que la machine distante soit atteinte. Pour chaque paquet la machine recevant le paquet dont la durée de vie est nulle retourne le temps qu'il a mis pour l'atteindre et son adresse à la machine émettrice

```
zss1: flags=51<UP,POINTOPOINT,RUNNING>
xpkt0: flags=51<UP,POINTOPOINT,RUNNING>
ip0: flags=0<>
std0: flags=0<>
osixpkt0: flags=0<>
hdlc0: flags=51<UP,POINTOPOINT,RUNNING>
snit_xpkt0: flags=41<UP,RUNNING>
lo0: flags=49<UP,LOOPBACK,RUNNING>
 inet 127.0.0.1 netmask ff000000
```

Arp : cette commande est utile pour analyser les problèmes dus à la traduction des adresses IP. Trois options sont utiles, -a donne toutes les entrées, -d efface une entrée de la table, -s ajoute une entrée dans la table. Les deux dernières options sont réservées au root. Arp est à utiliser lorsqu'une mauvaise machine répond. Ce genre d'anomalies sont dûes lorsque deux machines ont la même adresse IP.

```
~> /usr/etc/arp -a
floyd (129.88.32.32) at 8:0:20:a:e5:d7
celsius-251 (129.88.32.64) at 0:0:a7:0:7b:9b
esperanza-1 (129.88.32.65) at 0:0:a7:10:9f:62
farenheit-451 (129.88.32.33) at 0:0:a7:11:90:5
media (129.88.32.17) at 8:0:20:b:f9:2c
esperanza-3 (129.88.32.66) at 0:0:a7:10:a0:b1
alexandrie (129.88.32.34) at 0:0:a7:11:8f:e2
cap-ferret (129.88.32.18) at 0:0 ...
```

Les trois premiers octets de l'adresse physique indique la marque des machines, par exemple 8:0:20 correspond à un SUN. Les références sont répertoriées dans «Assigned Numbers RFC».

Netstat : suivant les options utilisées, netstat permet de de visualiser trois types d'informations, la première délivre les sockets valides utilisés par les différents protocoles, la seconde est une des nombreuses structures de données du réseau, la troisième sont des statistiques sur la transmission de paquets.

```
~> netstat -i
Name Mtu  Net/Dest  Address   Ipkts   Ierrs Opkts   Oerrs Collis Queue
le0  1500 imag-batb imag      8864760 6     8418838 2     258930 0
lo0  1536 loopback  localhost 765048  0     765048  0     0      0
~>
```

L'option -i permet de visualiser l'interface avec le réseau et les statistiques sur les paquets transmis. Si il y a des paquets dans la rubrique «queue» c'est que l'interface est à changer. Si les erreurs Ierrs et Oerrs ne sont pas proches de zéro, cela signifie que le réseau local est saturé. Les collisions sont a prendre en compte en rapport avec le nombre de paquets transmis et recus (ipkts+opkts) si ce pourcentage est élevé sur toute les machines de votre réseau local c'est qu'un sous-réseau serait le bien venu.

```
~> netstat -nr
Routing tables
Destination Gateway Flags Refcnt Use Interface
127.0.0.1 127.0.0.1 UH 17 188811 lo0
129.88.56.0 129.88.32.156 UG 2 7866 le0
default 129.88.32.254 UG 40 579209 le0
129.88.120.0 129.88.32.254 UG 0 15984 le0
129.88.40.0 129.88.32.29 UG 2 42546 le0
```

Quelques détails à propos de Ping

Ping host[packetsize][count]

host : est le nom de la machine distante.

packetsize : est la taille des paquets ICMP que Ping envoie pour avoir un echo de la machine distante, par défaut, ping envoie des paquets de 64 bytes.

count : est le nombre de paquets envoyés par la Ping à la machine distante.

l'option -s permet d'afficher les paquets echo retournés par la machine contactée :

```
~> /usr/etc/ping -s princeton.edu 56 5
PING princeton.edu: 56 data bytes
64 bytes from Princeton.EDU (128.112.128.1): icmp_seq=0. time=150. ms
64 bytes from Princeton.EDU (128.112.128.1): icmp_seq=1. time=260. ms
64 bytes from Princeton.EDU (128.112.128.1): icmp_seq=2. time=155. ms
64 bytes from Princeton.EDU (128.112.128.1): icmp_seq=3. time=608. ms
64 bytes from Princeton.EDU (128.112.128.1): icmp_seq=4. time=149. ms

----Princeton.EDU PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms) min/avg/max = 149/264/608
~>
```

Si le pourcentage de paquets perdus est élevé et le temps de réponse est bas ou que les paquets arrivent dans le désordre alors il pourrait y avoir un problème matériel sur le médium. Il n'y a pas d'inquiétude à avoir car TCP/IP est conçu pour résoudre les erreurs. Mais si ces problèmes concernent un réseau local alors il faut faire un contrôle du médium, en effet sur un réseau local le temps de parcours doit être presque nul et il ne devrait pas y avoir de paquets perdus.

IV-1.4 Tester l'accès au réseau

Les messages «no answer» et «cannot connect» permettent de conclure que le problème vient des couches basses du protocol TCP/IP. Trois commandes UNIX permettent de tester la couche d'accès au réseau.

Ifconfig : sans options cette commande permet de vérifier la configuration et les paramètres de d'une interface réseau d'une machine. Avec l'option -a, elle renvoie les configurations de toutes les interfaces d'un système.

```
~> /etc/ifconfig le0
le0: flags=63<UP,BROADCAST,NOTRAILERS,RUNNING>
 inet 129.88.32.1 netmask ffffff00 broadcast 129.88.32.255
```

En spécifiant une interface on obtient deux lignes, la première contient le nom et les caractéristiques, la deuxième donne l'adresse IP, le "subnet mask" et l'adresse de diffusion.

```
~> /etc/ifconfig -a
le0: flags=63<UP,BROADCAST,NOTRAILERS,RUNNING>
 inet 129.88.32.1 netmask ffffff00 broadcast 129.88.32.255
le1: flags=40<RUNNING>
zss0: flags=10<POINTOPOINT>
```

- ifconfig : est utilisé pour attribuer les adresses a un réseau et pour configurer les parametres de l'interface avec le réseau. Utilisée sans options cette commande délivre la configuration courante de l'interface du réseau désigné. Les paramètres utiles à observer sont les «subnet mask», et les adresses IP.
- arp : cette commande délivre les correspondances entre réseaux internets, utilisée sans options elle donne l'état courant des tables de la machine donnée.
- netstat : donne de nombreuses informations sur les interfaces, les sockets et les tables de routage.
- ping : cette commande lance un programme qui utilise le protocol ICMP, utilisée sans options elle permet de savoir si une machine distante est accessible, avec l'option -v (réservée aux administrateurs) la commande affiche les packets ICMP qu'elle voit passer sur le réseau.
- nslookup : lance des requêtes aux «Internet domain name server» pour avoir la liste des machines (hosts) connues sur le réseau. (Un programme similaire est disponoble sur le réseau : dig.)
- ripquery : affiche le contenu des packets RIP différés qui ont été envoyés à une machine distante.
- traceroute : affiche les différents réseaux traversés par les paquets entre deux machines distantes.
- etherfind : est un analyseur du protocol TCP/IP il permet d'observer le contenu des paquets (les entêtes et les donnés).

IV-1.3 Tester la présence sur le réseau

Ping est la commande qui permet de vérifier que la machine est accessible à partir de votre machine. Ceci permet de déterminer si on doit orienter la recherche vers le réseau lui-même ou bien vers les couches supérieures.

Si ping renvoie une réponse positive, les paquets peuvent traverser le réseau dans les deux sens et le problème doit se situer dans les couches supérieures. Si, par contre les paquets n'atteignent pas la machine distante, les couches basses du protocole de communication peuvent être en cause.

La commande Ping peut être exécutée à partir d'autres comptes ou d'autres machines. Si Ping échoue uniquement à partir de l'utilisateur en question, vous pouvez orienter votre analyse sur la configuration du système utilisateur. Si Ping ne fonctionne à partir d'aucun sites, alors les messages d'erreur peuvent vous aider.

- **Unknown host** : la convertion des noms en adresses ne fonctionne pas correctement. Essayez alors d'effectuer la commande ping avec l'adresse IP de la machine distante concernée, si Ping l'atteint de cette manière c'est que le «name service» sur votre machine ou sur l'autre est défectueux, poursuivez avec *"Nslookup*" ou "*Dig"* pour tester les serveur de noms local et distant.
- **Network unreachable** : ceci signifie que le protocol n'a pas de route établie pour atteindre la machine désignée, vérifiez la table de routage et réinstallée la. Si la route statique par défaut a été utilisée alors réinstallez la. Si tout semble correcte alors vérifiez les tables de routage du «gateway» par défaut spécifié sur votre machine.
- **No answer** : la route pour atteindre le système distant existe mais la machine ne répond pas. Les raisons peuvent être multiples à cela. La machine distante est peut être mal configurée ou des «gateways» entre les deux machines n'ont pas des tables de routage correctes ou encore il y a un problème de connexion. Cette situation vous impose de contacter l'administrateur du réseau où est connectée la machine distante.

IV Troubleshooting et sécurité

La mission de l'administrateur réseau peut être divisée en trois tâches biens distinctes : la configuration du réseau, la résolution des problèmes de fonctionnement (Troubleshooting) et la sécurité. Si la première tâche nécessite des connaissances détaillées des scripts d'installation et de configuration, la résolution des disfonctionnements du réseau est confrontée a des situations imprévues. La sécurité du réseau est a mi-chemin entre les deux premières tâches, la sécurité d'un site se prévoit lors de l'installation des différents systèmes sur le réseau et se porsuit par une surveillance et une information régulières des utilisateurs.

IV-1 Troubleshooting

Cette partie présente les méthodes et les outils disponibles pour la résolution des problèmes du réseau. Toutefois ceci nécessite d'avoir une vision claire du fonctionnement du réseau, ceci a été vu dans les premiers chapitres. Nous allons tout d'abord présenter une approche méthodologique.

IV-1.1 Analyse du problème

Les problèmes rencontrés avec TCP/IP sont très variés et nécessitent souvent des méthodes assez différentes les unes des autres mais l'analyse conduisant à la compréhension du problème est assez systématique. La difficulté principale est de pouvoir visualiser l'état du réseau et des protocoles à travers de nombreuses couches logicielles.

Les tests doivent permettre de savoir :

- si le problème est localisé à un utilisateur, une machine, une application
- si il concerne un groupe d'utilisateurs, plusieurs machines ou plusieurs applications
- si il concerne une ou plusieurs machines distantes

Toutefois la qualité de l'analyse dépend fortement de l'expérience de l'administrateur. voici quelques éléments qui peuvent vous éclairer dans votre démarche.

- un test doit être poursuivit tant que vous estimez qu'il peut vous apporter de nouvelles informations même si elles ne vous semblent pas directement liées au problème.
- établir un compte rendu de vos tests avec les conclusions obtenues au terme de chacun.
- ne vous concentrez pas sur des hypothèses trop hâtives, les messages d'erreurs contiennent beaucoup de détails utiles.
- dupliquez le problème sur d'autres machines.
- n'oubliez pas d'informer les utilisateurs et les autres administrateurs.
- vérifiez les connexions.

IV-1.2 Les programmes utiles

Analyser le problème en détail donne parfois une solution évidente. Mais dans des cas plus compliqués il est nécessaire de recourir a des outils de diagnostique. Regardons ici les commandes que UNIX fournit et quelques programmes disponibles par ftp anonyme.

En effet, du fait de son fonctionnement, où les par librairies sont chargées dynamiquement, SunOS 4.x a besoin de quelques composants supplémentaires : "le multiloader", les librairies C partagées, et le fichier *dev/zero*. Ce qui ce fait par exemple par :

```
# cd /usr/ftp
# mkdir usr
# mkdir usr/lib
# cp /usr/lib/ld.so usr/lib
# cp /usr/lib/libc.so.* usr/lib
# chmod 555 usr/lib/libc.so.* usr/lib usr
# cd /usr/ftp
# mkdir dev
# cd dev
# mknod zero c 3 12
# cd ..
# chmod 555 dev
```

Maintenant, vous n'avez plus qu'à mettre en place les fichiers que vous voulez rendre accessible par FTP anonyme dans le répertoire */usr/ftp/pub* (si vous voulez éviter que vos fichiers ne disparaissent, assurez vous qu'ils n'appartiennent pas à ftp et que leurs droits d'accès sont fixés à 644).

Enfin, il faut être conscient qu'un serveur FTP représente quand même un risque potentiel pour la sécurité, de sorte qu'il est à recommander de limiter le nombre de machine offrant un tel service dans un même réseau, et de surtout de bien vérifier que l'installation est correcte.

```
ensuntx:/users                          /users          nfs  rw  0  0
ensuntx:/users.ext                      /users.ext      nfs  rw  0  0
ensuntx:/var/spool/mail                 /var/spool/mail nfs  rw  0  0
ensuntx:/export/share/sunos.4.1.2       /usr/share      nfs  rw  0  0
ensuntx:/usr/local                      /usr/local      nfs  rw  0  0
```

III-6 Mise en place d'un serveur FTP anonyme

La mise en place d'un serveur FTP anonyme en 5 étapes :

1 - Ajouter l'utilisateur ftp au */etc/passwd.*

2 - Créer un répertoire racine du compte ftp, et en interdire l'accès en écriture.

3 - Y créer un sous répertoire *bin* appartenant au root; placer-y une copie de la commande *ls*, dont vous ne laisserez que le droit d'exécution, et enfin, enlevez le droit d'écriture au répertoire *bin*.

4 - Créer un sous répertoire *etc* appartenant au root, y placer un fichier *passwd* et un fichier *group* spécial, ne laisser que le droit de lecture pour ces deux fichiers, et enfin, interdire l'écriture dans le répertoire.

5 - Créer pour finir un sous répertoire *pub* appartenant à ftp avec tout les droits d'accès (mode 777). C'est dans ce dernier répertoire que les anonymes pourront déposer / prendre des fichiers.

Voyons en détail un exemple de mise en pratique :

```
# mkdir /usr/ftp
# cd /usr/ftp
# mkdir bin
# mkdir etc
# mkdir pub
# cp /bin/ls bin
# chmod 111 bin/ls
# cat > etc/group                       # attention : etc et non /etc !
anonymous:*:15:
^D
# cat > etc/passwd
ftp:*:15:15:acces au ftp anonyme:/usr/ftp:
^D
# chmod 444 etc/group etc/passwd
# cat etc/group >> /etc/group           # attention : >> et non > !
# cat etc/passwd >> /etc/passwd
# chown ftp pub
# chmod 777 pub
# chmod 555 bin etc
# cd ..
# chown ftp ftp
# chmod 555 ftp
```

C'est tout pour tout ceux qui utilisent un système différents du SunOS 4.x !

un accès en lecture. Si aucun nom de machine n'est spécifié, tout client NFS a accès en lecture / écriture.

access=*machine*[:*machine*]

Précise quelles sont les machines ayant droit de monter le répertoire (on utilise en général cette option en conjonction avec l'option ro)

Exemple :

```
/usr
/users
/users.nfs
/users.ext
/usr/local
/var/spool/mail
/var/spool/pcnfs
/export/exec/kvm/sun4c.sunos.4.1.2
/export/root/ensisun1 -access=ensisun1,root=ensisun1
/export/swap/ensisun1 -access=ensisun1,root=ensisun1
/export/root/ensisun2 -access=ensisun2,root=ensisun2
/export/swap/ensisun2 -access=ensisun2,root=ensisun2
/export/root/ensisun3 -access=ensisun3,root=ensisun3
/export/swap/ensisun3 -access=ensisun3,root=ensisun3
/export/root/ensisun4 -access=ensisun4,root=ensisun4
/export/swap/ensisun4 -access=ensisun4,root=ensisun4
```

III-5.3 Coté client ...

Pour monter un répertoire exporté par un serveur NFS il suffit (sous root) de faire un

mount nom_de_serveur_NFS:*nom_de_répertoire nom_de_répertoire_locale*

Par exemple :

```
# mkdir nfsusers
# mount ensisun:/users nfsusers
```

Cependant, si l'on veut que ces répertoires soient montés à chaque boot, il faut créer un fichier */etc/fstab* qui sera utilisé par la commande '*mount -vat nfs*' dans un script de démarrage pour remettre en place tous les répertoires requis.

Pour créer ce */etc/fstab*, le plus simple est de monter les répertoires à la main en s'aidant éventuellement de la commande '*showmount -e nom_de_serveur*' qui permet de lister les répertoires exportés par un serveur NFS; puis, de générer le fichier par la commande :

```
# mount -p > /etc/fstab
```

Ce qui donne par exemple :

```
ensuntx:/export/root/ensisun2             /              nfs rw 0 0
ensuntx:/export/exec/sun4.sunos.4.1.2     /usr           nfs ro 0 0
ensuntx:/export/exec/kvm/sun4c.sunos.4.1.2/usr/kvm       nfs ro 0 0
```

nfsd [nservers]	Le démon des serveurs NFS (le paramètre précise le nombre de démon à démarrer, en général on met 8).
biod [nservers]	Le démon des clients NFS (nservers à la même signification).
rpc.lockd	Le démon de verrouillage de fichiers tourne sur les clients comme sur les serveurs.
rpc.statd	Le démon de contrôle d'état, indispensable à rpc.lockd (en particulier pour récupérer d'un crash).
rpc.mountd	Le démon de mount (tourne du coté serveur) sont rôle est de gérer les demandes de mount des clients.

Ce qui donne par exemple les portions de scripts[1] :

```
# démarrage d'un client NFS
if [ -f /usr/etc/biod -a -f /usr/etc/rpc.statd -a -f /usr/etc//rpc.lockd ];
then
                biod 8 ;        echo -n ' biod'
                rpc.statd &     echo -n ' statd'
                rpc.lockd &     echo -n ' lockd'#
fi

# démarrage d'un serveur NFS
if [ -f /etc/exports ] ; then > /etc/xtab
                exportfs -a
                nfsd 8 &        echo -n ' nfsd'
                rpc.mountd
fi
```

On notera que dans ces scripts, on a testé l'existences des fichiers nécessaire avant de démarrer les démons.

III-5.2 Coté serveur ...

Le fichier */etc/exports* contient les informations décrivant les répertoires à exporter, que la commande '*exportfs -a*' utilise pour générer les informations nécessaires à mountd dans le fichier */etc/ xtab* .

La syntaxe de ce fichier est :

répertoire [-*option*][,*option*]...

où option précise les droits d'accès[2].

Les options de bases sont (pour plus d'informations faire '*man exports*') :

ro	"Read Only" tout client NFS peut lire mais aucun n'a le droit d'écrire dans le répertoire.
rw [=*machine*][:*machine*]...	"Read Write" si un (ou plusieurs) nom de machine est précisé[3], seul les machines spécifiées ont un accès en lecture / écriture les autres ont

---

[1] Ces programmes ne sont pas forcément dans */usr/etc/*, et ne sont pas forcément préfixés par *rpc.* ... consultez donc la documentation de vôtre système pour les détails ...

[2] Par défaut, tout client est autorisé à monter les répertoires pour des accès en lecture et en écriture !

[3] Dans ce fichier on peut utiliser des noms de groupe de machines (voir */etc/netgroup*)

staff (ensisun, smith,) admin

On peut alors utiliser ces nom de groupes partout où des noms de machine ou des noms d'utilisateur sont requis, en particulier, dans les *.rhosts* ou le */etc/hosts.equiv* il suffit de précéder le nom de groupe d'un caractère @ pour que le système le comprennent comme un nom de groupe et en extrait les noms d'utilisateur ou les noms de machine suivant ce qui est requis

Vous avez pu noter que le nom de domaine à été omis dans les exemples, c'est en général le cas, car on sort rarement du domaine dans ce genre de fichier. Une autre pratique courante est de séparer les groupe de personnes et les groupes de machine pour simplifier l'administration; ce qui donne par

exemple :

admin (-, dupond,) (-, durand,)
staff (-, smith,) admin
enseignement (ensisun, -,) (ensibull, -,)
machines enseignement (imag, -,)

III-5 NFS : Network File System

NFS à pour but de faire partager à plusieurs machines leurs systèmes de mémoires de masse. C'est une application transparente pour l'utilisateur, on peut très bien travailler sans être au courant de l'existence de ce système cependant je pense qu'il est intéressant de savoir de quoi il retourne 'pour la culture personnel'.

Les avantages d'NFS sont :

- Une réduction notable des besoin en espace disques : comme on peut avoir des stations de travail sans disques qui travail sur les fichiers d'une autre machine => on peut faire en sorte que tout les fichier (en particulier tout le système, les applications) sont en un seul exemplaire.
- Permet aux utilisateurs d'avoir le même environnement,et d'acceder aux mêmes fichiers qu'elle que soit le poste sur le quelle ils travaillent.
- Simplifie les tâche d'administration en centralisant les fichiers qui restent pourtant accessible sur tout le réseau.

NFS est basé sur un système client / serveur, le client utilise les fichiers du serveur comme s'ils faisaient partie des disques locaux. Lorsque que l'on s'attache une arborescence d'un disque d'une autre machine, on dit que l'on monte un répertoire ('to mount a directory' en anglais). Alors que rendre accessible une arborescence aux autres machines se dit exporter un répertoire.

III-5.1 Les démons NFS : mise en place d'NFS

Les démons nécessaires à faire tourner NFS sont lancés dans les scripts de démarrage des machines clientes et des machines serveurs :

Tableau 9 : Fichiers Unix et NIS maps

Fichier	maps	description
/etc/networks	*networks.byaddr* *networks.byname*	Donne les conversions d’ adresses IP de réseaux : numérique <--> par nom
/etc/netmasks	*netmasks.byaddr*	Donne les subnet-masks des réseaux du domaine
/etc/protocols	*protocols.byaddr* *protocols.byname*	Nom de protocole <--> port/protocole
/etc/services	*services.byname*	Services TCP/IP
/etc/aliases	*mail.byaddr* *mail.aliases*	Définit des alias pour joindre des personnes ou des groupes de personnes par mail.
/etc/netgroup	*netgroup.byuser* *netgroup.byhost*	Définit des groupes d’utilisateur et des groupes de machines.

III-4.1 Mise en place de NIS

NIS à besoin de connaître le nom du domaine pour maintenir sa base de données car elle réside dans un sous répertoire de */var/yp* dont le nom dérive du nom du domaine : si le nom du domaine est *cheops.imag.fr* le répertoire de la base NIS est */var/yp/cheops.imag.fr* . Il faut donc lui indiquer ce nom de domaine au démarrage ce qui est fait par la commande *domainname* dans un des fichiers de boot.

NIS peut être utilisé comme alternative à DNS pour un réseau fermé non connecté à Internet, mais pour tout ceux qui utilisent Internet il faut DNS, cependant NIS apporte des informations qui ne sont pas accessibles par DNS de sorte qu’il est courant d’utiliser les deux[1]. Pour ce faire il est nécessaire de faire une petite modification au fichier */var/yp/Makefile* : il faut enlever le caractère # qui est au début de la ligne ’B=-b’ et en mettre un devant ’B=’.

Pour lancer le serveur et reconstruire les ’NIS maps’ il faut d’abord exécuter *’ypinit -m’*, démarrer le serveur par *’ypserv’*, et enfin, démarrer le démon de transfert des maps par *’ypxfrd’* [2] sur la machine serveur principal. Quand aux machines clientes, elles se contentent de lancer *’ypbind’*.

III-4.2 Le fichier /etc/netgroup

Le fichier /etc/netgroup est un fichier qui n’est utilisé que par NIS et qui définit des groupes de machines ou d’utilisateur. Sa syntaxe est la suivante :

nom_du_groupe membre [membre] . . .

où membre est soit le nom d’un autre groupe soit la définition d’une entité suivant le format :

(nom_de_machine, nom_d’utilisateur, nom_de_domaine)

où le nom de domaine est optionnel. Ce qui donne par exemple :

```
admin (ensisun, dupond, ) (ensibull, durand, )
```

---

[1] Il est recommandé si l’on utilise à la fois NIS et DNS de leur spécifier le même nom de domaine.

[2] Retirez le # devant *ypxfrd* dans votre rc.local pour que le démon démarre automatiquement au boot.

```
    Salut, ma poule, tu sais qu'ta d'bosieux ?

  .
  250 Ok
  quit
  221 ensibull.imag.fr closing connection
  Connection closed by foreign host.
...
  ensisun~> mail
  Mail version SMI 4.0 Wed Oct 23 10:38:28 PDT 1991 Type ? for help.
  "/usr/spool/mail/dupond": 1 messages 1 new 1 unread
   N 1 le_grand_duduche@ensibull.imag.fr Sat Jun 26 10:23 13/465
  & 1
  Message 1:
  From le_grand_duduche@ensibull.imag.fr Sat Jun 26 10:23:45 1993
  Received: from ensibull.imag.fr by ensisun.imag.fr (4.1/5.17)
   id AA13434; Sat, 26 Jun 93 10:23:45 +0200
  Received: from ensisun by ensibull.imag.fr (5.61/5.17)
   id AA19417; Sat, 26 Jun 93 10:20:41 GMT
  Date: Sat, 26 Jun 93 10:20:41 GMT
  From: le_grand_duduche@ensibull.imag.fr
  Message-Id: <9306261020.AA19417@ensibull.imag.fr>
  Apparently-To: dupond@ensisun
  Status: R

    Salut, ma poule, tu sais qu'ta d'bosieux ?

  & x
```

Si cette technique permet sur des systèmes qui ne sont pas très stricts sur le plan de la sécurité[1], de faire quelle bonne farces, nous vous conseillons de ne pas en abuser car vous pourriez en arriver a vous méfier sans arrêt, et a ne plus savoir distinguer le vrai du faux ... (sans parler des ennuis que vous pourriez vous attirer si l'on prend vos faux courriers au sérieu !).

III-4 NIS : Network Information Service

Le NIS est une base de données contenant des informations sur le réseau, elle assure une distribution de l'information tout en gardant l'avantage de la simplicité de mise à jour des information d'un système centralisé. Les informations aux quelles NIS permet d'accéder, sont groupées dans des sortes de tables distribuées sur le réseau (appelées 'NIS maps'), mais leur contenu est en fait tiré de quelques fichiers Unix centralisés (dont certains ont déjà été décrits dans ce document) :

Tableau 9 : Fichiers Unix et NIS maps

Fichier	maps	description
/etc/ethers	*ethers.byaddr* *ethers.byname*	Donne les adresses ethernet à partir des adresses IP (utilisé par RARP)
/etc/hosts	*hosts.byaddr* *hosts.byname*	Donne les conversions d' adresses IP de machine: numérique <--> par nom

---

[1] Ce n'est pas la peine d'essayer de faire un faux courrier de *bill-clinton@whitehouse.gov* : leur démon *sendmail* est un peu plus sécurisé, et refusera de passer le courrier ...

III-3.3 Sendmail le centre de tri

Quand il faut déterminer la route que doit prendre un courrier, c'est encore *sendmail* qui est mis à contribution. Il s'appui pour ce faire sur le fichier de configuration *sendmail.cf*. Ce dernier a trois fonctions principales :

- Définir l'environnement de *senmail*.
- Donner des règles de ré-écriture des adresses dans une syntaxe appropriée au programme qui recevra le courrier.
- Etablir les instructions a exécuter pour faire parvenir le courrier en fonction de l'adresse.

La syntaxe de ce fichier étant fort complexe, (c'est un véritable langage) et très rébarbative (chaque mot, variable de ce langage n'est constitué que d'une seule lettre!); on se contentera d'indiquer comment ce procurer un fichier modèle, ainsi que des guides qui vous aiderons a l'adapter a vos besoins.

Le fichiers *ftp.uu.net:mail/sendmail/sendmail-5.65.tar.Z* contient tout ce dont vous aurez besoin pour établir votre *sendmail.cf*, en particulier, vous y trouverez :

• *tcpuucpproto.cf*	modèle pour les systèmes ayant un accès direct au réseaux TCP et UUCP.
• *tcpproto.cf*	modèle pour les systèmes n'ayant un accès direct qu'aux réseaux TCP.
• *uucpproto.cf*	modèle pour les systèmes n'ayant un accès direct qu'aux réseaux UUCP.
• *doc/07.sendmailop/*	Sendmail Installation and Operation Guide ...
• *doc/16.sendmail/*	Sendmail: An Internetwork Mail Router ...

III-3.4 Au rayon farces et attrapes : sendmail

En faisant un *telnet* sur le port 25 d'une machine, on peu rentrer en communication directe avec sendmail (le démon) et de cette façon, lui faire croire qu'il a reçu un courrier SMTP. Le point intéressant étant que l'on peut spécifier soit même le nom de la personne à l'origine du courrier !!!

exemple :

```
ensisun~> whoami
dupond
ensisun~> telnet ensibull 25
Trying 192.33.174.35 ...
Connected to ensibull.
Escape character is '^]'.
220 ensibull.imag.fr Sendmail 5.61/5.17 ready at Sat, 26 Jun 93 10:19:47 GMT
help
214-Commands:
214- HELO MAIL RCPT DATA RSET
214- NOOP QUIT HELP VRFY EXPN
214-For more info use "HELP <topic>".
214-To report bugs in the implementation contact eric@Berkeley.ARPA
214-or eric@UCB-ARPA.ARPA.
214-For local information contact postmaster at this site.
214 End of HELP info
MAIL FROM : le_grand_duduche
250 le_grand_duduche... Sender ok
RCPT TO: dupond@ensisun
250 dupond@ensisun... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
```

machine c'est crashé alors que des courriers étaient en cours de traitement. Enfin on peu démarrer le démon avec les options[1] :

-q*intervale* => précise la fréquence à la quelle on doit traiter la queue (1h => toute les heures; 15m => tout les quarts d'heure ...)

-bd => précise que sendmail doit fonctionner comme un démon, et doit écouter le port 25 pour réceptionner les courriers arrivant.

III-3.2 Sendmail le pourvoyeur d'alias

Les aliases que reconnaît sendmail sont définis dans */etc/aliases* avec le format :

alias:*recipient*[,*recipient*]... ou owner-*aliasname*:*address*

Ces aliases peuvent servir trois causes :

- donner des surnoms aux utilisateurs (ou admettre des noms complets, avec majuscule ...)
- faire suivre le courrier (cas des comptes déplacé ou supprimés)
- établir des listes de mailing

exemple :

```
##
# Aliases can have any mix of upper and lower case on the left-hand side,
# but the right-hand side should be proper case (usually lower)
#
# >>>>>>>>>> The program "newaliases" will need to be run after
# >> NOTE >> this file is updated for any changes to
# >>>>>>>>>> show through to sendmail.
#
# @(#)aliases 1.10 89/01/20 SMI
##

# Following alias is required by the mail protocol, RFC 822
# Set it to the address of a HUMAN who deals with this system's mail problems.
Postmaster: root

# Alias for mailer daemon; returned messages from our MAILER-DAEMON
# should be routed to our local Postmaster.
MAILER-DAEMON: postmaster

# Une alternative pour joindre le compte cathy :
cassagne: cathy

# Mailing-liste pour joindre les responsable ...
admin: cassagne, dupond, durand@ensibull, smith@imag
owner-admin: cathy
```

Il est noter que tout changement dans le fichier */etc/aliases* ne sera pas pris en compte tant que l'on aura pas executer la commande *newaliases* pour que sendmail remette à jours ses données .

---

[1] Pour les autres options et leurs significations, consultez le man ... !

```
230- it was last modified on Tue Jun 8 08:30:26 1993 - 9 days ago
230 Guest login ok, access restrictions apply.
ftp> bin
200 Type set to I.
ftp> hash
Hash mark printing on (8192 bytes/hash mark).
ftp> get network/ftp.servers.Z |uncompress>tst
200 PORT command successful.
150 Opening BINARY mode data connection for ftp.servers.Z (47919 bytes).
#######################################################################
226 Transfer complete.
local: |uncompress>tst remote: ftp.servers.Z
47919 bytes received in 3.3 seconds (14 Kbytes/s)
ftp> bye
221 Goodbye.
```

III-3 Le mail : courrier électronique

Une discussion détaillée du courrier electronique pourait suffire à faire l'objet d'un livre entier, c'est pourquoi nous nous contenterons ici de ne donner qu'un descriptif assez bref.

Lorsque vous envoyez un courrier par la commande *mail*[1], il se passe tout un enchainement d'opérations, mais l'important est la commande *sendmail* qui est le centre nerveux du système de courrier électronique. On peut distinguer trois tâches de *sendmail* :

- Recevoir le courrier électronique Internet : SMTP
- Permettre l'utilisation d'alias et listes de mailing
- Faire parvenir le courrier à destination en analysant l'adresse destinataire (ce dernier point masque aussi le fait qu'il y a pas moins de trois programmes de traitement du courrier électronique suivant qu'il s'agisse d'un courrier Internet, d'un courrier entre systèmes UUCP, ou, d'un courrier local entre deux utilisateurs d'une même machine !)

III-3.1 Sendmail le démon SMTP

Le démon sendmail, prend le courrier Internet qui arrive sur le port TCP 25 et le traite ...

Pour comprendre ce qu'il fait, regardons les lignes qui démarrent sendmail en tant que démon lors du boot :

```
if [ -f /usr/lib/sendmail -a -f /etc/sendmail.cf ]; then
        (cd /var/spool/mqueue; rm -f nf* lf*)
        /usr/lib/sendmail -bd -q1h ; echo -n ' sendmail'
fi
```

On voit d'abord le classique test d'existence des fichiers nécessaires, puis la ligne *rm* qui est chargée d'éliminer tous les verrous qui auraient pu rester dans le répertoire */var/spool/mqueue* si la

---

[1] Pour plus de renseignements lancer la comande *mail* puis demander l'aide en ligne en tapant *help* ...
(il y a trop de version trés differantes de mail)

```
Commands may be abbreviated. Commands are:

!               cr              macdef          proxy           send
$               delete          mdelete         sendport        status
account         debug           mdir            put             struct
append          dir             mget            pwd             sunique
ascii           disconnect      mkdir           quit            tenex
bell            form            mls             quote           trace
binary          get             mode            recv            type
bye             glob            mput            remotehelp      user
case            hash            nmap            rename          verbose
cd              help            ntrans          reset           ?
cdup            lcd             open            rmdir
close           ls              prompt          runique
ftp> help dir bin hash prompt mget
dir             list contents of remote directory
binary          set binary transfer type
hash            toggle printing '#' for each buffer transferred
prompt          force interactive prompting on multiple commands
mget            get multiple files
ftp> cd pub/ftp-list
250 CWD command successful.
ftp> dir ftp*
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
-rw-r--r-- 1 30750 21060 7701    Jan 3 1992 ftp.help
-rw-r--r-- 1 30750 21060 148620  Jan 3 1992 ftp.list
-rw-r--r-- 1 30750 21060 2762    Jan 3 1992 ftpmailservers
-rw-r--r-- 1 30750 21060 33890   Jan 3 1992 ftpserv.tar.uu
226 Transfer complete.
remote: ftp*
268 bytes received in 0.032 seconds (8.1 Kbytes/s)
ftp> prompt
Interactive mode off.
ftp> mget ftp.*
200 PORT command successful.
150 Opening ASCII mode data connection for ftp.help (7701 bytes).
226 Transfer complete.
local: ftp.help remote: ftp.help
7897 bytes received in 0.91 seconds (8.5 Kbytes/s)
200 PORT command successful.
150 Opening ASCII mode data connection for ftp.list (148620 bytes).
226 Transfer complete.
local: ftp.list remote: ftp.list
150683 bytes received in 11 seconds (13 Kbytes/s)
ftp> close
221 Goodbye.
ftp> open ftp.inria.fr
Connected to ftp.inria.fr.
220 ftp FTP server (Version 5.60) ready.
Name (ftp.inria.fr:dupond): anonymous
331 Guest login ok, send e-mail address as password.
Password:
230-
230- *************************************
230- * WELCOME to the INRIA FTP server *
230- *************************************
230-Please read the file README
```

III-2 Applications de base : telnet, ftp

La configuration du système ayant été réalisée (/etc/inetd.conf), il ne devrait, pas y avoir de problème pour ce servir de ces deux utilitaires (sauf si leur usage a été volontairement restreint).

• telnet	permet de se connecter a une machine distante qu'elle que soit son type, pour plus d'information utiliser le man.
• ftp	permet de transférer des fichiers entre des machines, même si ce sont des systèmes très différents (sous FTP faire help).

Pour ce qui est de FTP quelques informations complémentaires sont tout de même très utiles pour le débutant. Tout d'abord, il existe des serveurs FTP anonymes, cela signifie que sur de tels serveurs il est possible de se connecter sous le nom d'utilisateur *anonymous* en donnant sa propre adresse en guise de password.

Le fait de donner votre adresse en tant que password n'est pas une contrainte stricte : il suffit en général de taper un caractère @ dans la ligne pour que le serveur soit satisfait, cependant nous vous conseillons de mettre votre véritable adresse dans votre propre intérêt : avoir accès au serveurs FTP anonymes n'est pas un droit, c'est un privilège que l'on vous accorde, et il est normal (et même souhaitable) que le responsable d'un serveur anonyme soit au courant du public qu'il touche, de plus il peut arriver que l'on prenne un fichier dangereux (bug désastreux, cheval de Troie, virus, ...) on ne peut alors être prévenu du problème qu'à condition d'avoir donner son adresse !

Ensuite, il faut connaître quelques pièges d'FTP :

- Tout fichier qui n'est pas un fichier texte, ne doit être transféré qu'en mode binaire (par exemple, un fichier *.ZIP pour les PC ou un *.hqx pour les MAC ...)
- A l'inverse, tout fichier texte doit être transféré en mode ASCII car les marques de fin de ligne, de fin de fichier, et même le jeux de caractère, sont différent d'un constructeur à l'autre ... FTP réalise la conversion pour vous !
- Enfin, bien que l'on puisse utiliser des pipes dans les commandes d'FTP, il faut bien prendre garde de ne pas mettre de blanc entre le pipe et la commande qui le suit pour que FTP puisse l'interpréter correctement.

De plus, il n'est pas inutile de savoir que l'on peut se procurer sur le réseau des variantes d'FTP qui sont plus pratiques d'emploi (essayer sur *ftp.inria.fr* dans */system/user* le fichier *ncftp.tar.Z*).

Quelques adresses :

pilot.njin.net	=>	liste de serveurs anonymes : *pub/ftp-list/ftp.list*
ftp.inria.fr	=>	Sources, X11, TeX, GNU, Games, ...
tsx-11.mit.edu	=>	Linux (un Unix domaine public pour PC), GNUs, ...
nic.switch.ch	=>	Programmes domaines public pour PC, archives, ...

Exemples :

```
imag{10} ftp pilot.njin.net
Connected to pilot.njin.net.
220 pilot.njin.net FTP server ready.
Name (pilot.njin.net:dupond): anonymous
331 Guest login ok, send e-mail address as password.
Password:
230 Guest login ok, access restrictions apply.
ftp> help
```

III-1.1 Le fichier */etc/hosts.equiv*

Pour l'administrateur le fichier important est */etc/hosts.equiv*; ce dernier est composé de zéro ou plusieurs lignes de la forme [1] :

[+|-][*nom_de_machine*] [*nom_d'utilisateur*]

Ce qui donnerait par exemple dans le */etc/hosts.equiv* de la machine *ensisun* :

ensibull	=> Autorise toute personne ayant un compte sur *ensibull* et un sur *ensisun* sous le même nom d'utilisateur, à accéder à *ensisun* depuis *ensibull* sans password !
-*ensibull* dupond	=> Force l'utilisateur dupond@*ensibull* à donner son password lorsque qu'il tente d'accéder à *ensisun* depuis *ensibull* (utile quand dupond@*ensibull* et dupond@*ensisun* ne sont pas la même personne).
+ durand	=> Brèche dans la sécurité : Autorise toute personne ayant un compte sur une machine distante sous le nom durand à accéder à tous les comptes utilisateur d'*ensisun* sans password ! (le compte root excepté)
+ *ensibull* durand	=> ATTENTION : il y a un blanc entre + et *ensibull* (faute d'inattention?) => Brèche dans la sécurité : Autorise toute personne ayant un compte sur une machine distante sous le nom ensibull ou le nom durand à accéder à tous les comptes d'ensisun sans password ! (le compte root excepté)

III-1.2 Les fichiers *.rhosts*

Le *.rhosts* est un fichier utilisateur qui a la même syntaxe que le fichier */etc/hosts.equiv* mais qui ne donne accès qu'au compte de l'utilisateur qui l'a crée et placé dans la racine de sont répertoire utilisateur (le fichier .rhosts ne peut pas prendre le pas sur le fichier */etc/hosts.equiv*).

Exemple :

```
  ensisun{10} pwd
  /users/durand
  ensisun{11} cat > .rhosts
  ensibull dupond
  ^D
  ensisun{12} chmod go-w .rhosts
...
  ensibull{10} whoami
  dupond
  ensibull{11} rlogin -l durand ensisun
  Last login: Thu Jun 17 12:19:37 from imag
  SunOS Release 4.1.2 (ENSIMAG_SNC) #2: Fri Nov 20 16:26:13 MET 1992
  ********************************************************************
  ensisun{10} ...
```

---

[1] attention à la position du blanc : si vous tapez un blanc entre le plus et le nom_de_machine, ce dernier sera en fait interprété comme un nom d'utilisateur ce qui est fort dangereu ...

III Applications

Dans ce chapitre nous allons présenter les applications les plus courantes que l’on puisse trouver au-dessus de TCP/IP. Nous évoquerons leurs installations, leurs utilisations, leurs fichiers de configurations (utilisateur ou système) ainsi que certains ‘trucs & astuces’ intéressant.

III-1 La famille des commandes ‘r’

Pour les systèmes Unix, la famille des commandes ‘r’ comprend toutes les commandes utiles pour travailler à partir d’un serveur local sur un ou des serveurs distants; pourvu que les deux systèmes soient de type Unix (et que les deux machines puissent se joindre l’une l’autre!).

Ces commandes sont très pratiques car elles permettent de passer d’une machine à l’autre sans avoir à donner son mot de passe à chaque fois ... !

Elles représentent aussi par conséquent un danger considérable sur le plan de la sécurité si l’on ne sait pas les maîtriser[1].

Les trois commandes ‘remotes’ qui nous intéressent nécessitent la mise en place par l’opérateur ou par l’utilisateur de fichiers de configurations donnant des droits d’accès, nominatifs ou par machine. Ces commandes sont :

- *rlogin* : ‘remote login’ permet de se connecter sur une machine distante Unix.
- *rcp* : ‘remote copy’ permet de copier des fichiers d’une machine Unix a une autre.
- *rsh* : ‘remote shell’ permet d’exécuter une commande sur une machine distante.

Exemples (pour comprendre aidez-vous du man ...) :

```
imag{22} whoami
durand
imag{23} rlogin -l dupond ensisun
Last login: Wed Jun 16 08:11:02 from ensisun
SunOS Release 4.1.2 (ENSIMAG_SNC) #2: Fri Nov 20 16:26:13 MET 1992
**********************************************************************
You have mail.
Terminal recognized as vt100 (ANSI/VT100 Clone)
ensisun{10} whoami
dupond
ensisun{11} exit
Connection closed.
imag{24} rsh -l dupond ensisun ls -lag /etc/passwd
-rw-r--r-- 1 root staff 29592 Jun 15 15:02 /etc/passwd
imag{25} rcp dupond@ensisun:/etc/passwd pswd
imag{26} ls -lag pswd
-rw-r----- 1 durand students 29592 Jun 16 14:26 pswd
imag{27} wich rsh
/usr/ucb/rsh
imag{28} ln -s /usr/ucb/rsh ensisun
imag{29} ensisun 'echo "qui suis-je ? ... "`whoami`"@"$HOST'
qui suis-je ? ... durand@ensisun
imag{30} ensisun -l dupond 'echo "qui suis-je ? ... "`whoami`"@"$HOST'
qui suis-je ? ... dupond@ensisun
```

---

[1] le responsable système est en mesure de désactiver ces commandes en transformant les lignes correspondantes du fichier */etc/inetd.conf* en lignes de commentaires.

```
####################################################################
Received 6251 records.
> view temp
 [imag.imag.fr]
 imag.fr. server = imag.imag.fr
 imag 129.88.32.1
 imag.fr. server = hal.imag.fr
 hal 129.88.32.24
 imag.fr. server = layon.inria.fr
 imag.fr. server = mirsa.inria.fr
 imag.fr. server = archi.imag.fr
 archi 147.171.129.1
 imag.fr. 129.88.32.1
 saint-eynard 129.88.38.27
 notos 147.171.149.30
 maceudes 129.88.32.49
 lys 147.171.150.51
 durga 129.88.32.24
 oahu 129.88.100.64
 lion 129.88.33.32
 ensitty1 192.33.174.40
 ensitty2 192.33.174.41
 ensitty3 192.33.174.42
 ensitty4 192.33.174.43
 ensitty5 192.33.174.44
 gimli 129.88.33.21
 athena 129.88.40.3
 knuth1 192.33.172.51
 knuth2 192.33.172.52
 mac_archi-10 147.171.129.230
 knuth3 192.33.172.53
 mac_archi-11 147.171.129.231
 knuth4 192.33.172.54
 meltemi 147.171.149.20
 mac_archi-12 147.171.129.232
 knuth5 192.33.172.55
 aragorn 129.88.33.23
 mac_archi-13 147.171.129.233
 mare 129.88.100.18
 mac_archi-15 147.171.129.235
 mac_archi-16 147.171.129.236
 mac_archi-17 147.171.129.237
 mac_archi-20 147.171.129.240
 mac_archi-18 147.171.129.238
 mac_b2-1 129.88.59.2
 devoluy 129.88.38.14
 mac_archi-21 147.171.129.241
 mac_archi-19 147.171.129.239
 mac_b2-2 129.88.59.3
 mac_archi-22 147.171.129.242
 mac_b2-3 129.88.59.4
 mac_archi-23 147.171.129.243
 mac_b2-4 129.88.59.5
 mac_archi-24 147.171.129.244
 mac_b2-5 129.88.59.6
 aramis-campus 129.88.31.254
... ( to be continued)
```

ii-4.3e. Le fichier *named.host*

Ce fichier contient la plupart des informations sur le domaine. Ce fichier convertit les noms en adresse IP. Cela correspond aux enregistrements A. Ce fichier, comme *named.rev* est seulement crée pour les serveurs primaires. Les autres serveurs tirent ces informations des serveurs primaires. Voici le contenu d'un fichier exemple :

```
; Authoritative data for Berkeley.EDU (ORIGIN assumed Berkeley.EDU)
;
@        IN      SOA    ucbvax.berkeley.edu kjd.ucbvax.berkeley.edu (
                          1.1      ; Serial
                          10800    ; Refresh 3 hours
                          3600     ; Retry    1 hour
                          3600000  ; Expire   1000 hours
                          86400 )  ; Minimum 24 hours
         IN      MX       ucbvax 10
         IN      NS       monet
ucb-arpa IN      A        10.0.0.78
         IN      A        128.32.0.4
         IN      HINFO    VAX-11/780 UNIX
arpa     IN      CNAME    ucbarpa
ucb-vax 9999IN   A        10.2.0.78
         IN      A        128.32.0.10
         IN      HINFO    VAX-11/750 UNIX
ucbvax   IN      CNAME    ucbvax
monet    IN      A        128.32.0.7
         IN      HINFO    VAX-11/750 UNIX
ucbmonet IN      CNAME    monet
kjd      IN      MB       ucbarpa
dunlap   IN      MR       kjd
group    IN      MINFO    kjd kjd.Berkeley.EDU.
         IN      MG       name1
         IN      MG       name2
```

II-4.4 Utilisation de *nslookup*

nslookup est un outil de debuggage fourni avec le BIND. Il permet de faire des requêtes directement à un serveur de nom et de retrouver les informations connues du DNS. C'est très utile pour savoir si un serveur fonctionne correctement et est correctement configuré.

Cette commande est expliquée plus en détail dans un chapitre suivant.

Voici la manipulation à faire pour les données d'un domaine à partir d'un serveur primaire:

```
% nslookup
Default Server: ensisun.imag.fr
Addresses: 192.33.174.34, 192.33.174.65

> server imag.imag.fr
Default Server: imag.imag.fr
Address: 129.88.32.1

> ls imag.fr > temp
[imag.imag.fr]
```

• L'entrée SOA (Start of authority) identifie *ucbvax.Berkeley.EDU* comme le serveur initialisant cette zone, et l'adresse e-mail *kjd.ucbvax.Berkeley.EDU* est l'adresse ou poser des questions sur la zone.

• L'entré NS contient le nom de la machine hôte.

• Les entrées PTR convertissent les adresses en nom de machine.

• Les fichiers *named.boot*, *root.cache* et *named.local* sont les seuls fichiers nécessaires à la configuration des serveurs en catching-only et secondary. La plupart des serveurs n'utiliseront que ces fichiers.

ii-4.3d. Le fichier de domaine renversé *named.rev*

Ce fichier a la même structure que *named.local* car son but est de traduire des adresses IP en nom. Voici le contenu d'un fichier exemple cherché à Berkeley. :

```
;
;        @(#)named.rev    1.1      (Berkeley)       86/02/05
;

@       IN      SOA     ucbvax.berkeley.edu kjd.ucbvax.berkeley.edu (
                          1.2      ; Serial
                          10800    ; Refresh  3 hours
                          3600     ; Retry    1 hour
                          3600000  ; Expire   1000 hours
                          86400 )  ; Minimum  24 hours
         IN      NS       ucbvax.Berkeley.EDU.
0.0      IN      PTR      Berkeley-net.Berkeley.EDU.
         IN       A       255.255.255.0
0.130    IN      PTR      csdiv-net.Berkeley.EDU.
2.129    IN      PTR      monet.Berkeley.EDU.
2.140    IN      PTR      ucbarpa.Berkeley.EDU.
3.132    IN      PTR      cad.Berkeley.EDU.
4.0      IN      PTR      ucbarpa.Berkeley.EDU.
5.0      IN      PTR      cad.Berkeley.EDU.
6.0      IN      PTR      ernie.Berkeley.EDU.
6.130    IN      PTR      monet-cs.Berkeley.EDU.
7.0      IN      PTR      monet.Berkeley.EDU.
7.130    IN      PTR      kim.Berkeley.EDU.
9.0      IN      PTR      esvax.Berkeley.EDU.
10.0     IN      PTR      ucbvax.Berkeley.EDU.
11.0     IN      PTR      kim.Berkeley.EDU.
11.156   IN      PTR      esvax-156.Berkeley.EDU.
38.131   IN      PTR      monet.Berkeley.EDU.
```

- L'entrée SOA définit le domaine pour le fichier zone. On retrouve le même SOA sur tous les fichiers zone de ensisun.

- L'entrée NS définit le serveur de noms pour le domaine.

- Les entrées PTR traduisent des adresses en nom. Dans notre cas, les enregistrements PTR fournissent les noms de 0.0, 130.0, 129.2,... sur le réseau 128.66.

Le fichier cache pour ensisun est : */var/spool/named/root.cache*.

Il contient le nom et les adresses des serveurs root . Voici son contenu :

```
;
; Hints for root nameservers
;
.                   99999999    IN      NS      c.nyser.net.
                    99999999    IN      NS      kava.nisc.sri.com.
                    99999999    IN      NS      ns.nasa.gov.
                    99999999    IN      NS      aos.brl.mil.
                    99999999    IN      NS      ns.nic.ddn.mil.
                    99999999    IN      NS      terp.umd.edu.
                    99999999    IN      NS      nic.nordu.net.
c.nyser.net.        99999999    IN      A       192.33.4.12
kava.nisc.sri.com.  99999999    IN      A       192.33.33.24
ns.nasa.gov.        99999999    IN      A       192.52.195.10
                    99999999    IN      A       128.102.16.10
aos.brl.mil.        99999999    IN      A       192.5.25.82
ns.nic.ddn.mil.     99999999    IN      A       192.112.36.4
terp.umd.edu.       99999999    IN      A       128.8.10.90
nic.nordu.net.      99999999    IN      A       192.36.148.17
```

99999999 est le ttl (time to live). ttl représente la durée en seconde pendant laquelle l’information doit être conservée dans le cache. La valeur 99999999 - la plus grande possible - indique que le serveur root n’est jamais enlevé du cache.

Une liste de serveur root est disponible par ftp anonyme à *nic.ddn.mil* dans le fichier *netinfo/root/root-servers.txt*.

Si le réseau n’est pas connecté à Internet, il est inutile d’initialiser le cache avec les serveurs root ci-dessus, vu que vous ne pourrez pas les atteindre. Initialisez votre cache avec des entrées qui pointent sur le serveur de nom local.

ii-4.3c. Le fichier *named.local*

Ce fichier est utilisé pour convertir l’adresse 127.0.0.1 (l’adresse loopback) en le nom localhost. C’est le fichier zone du domaine renversé 0.0.127.in-addr.arpa. Ce fichier est presque chaque fois identique sur tous les serveurs. Voici le contenu du fichier mis à disposition à l’adresse de ftp ci-dessus :

```
;
; @(#)named.local 1.1 (Berkeley) 86/01/21
;
@ IN SOA ucbvax.Berkeley.EDU. kjd.ucbvax.Berkeley.EDU. (
 1.2 ; Serial
 3600 ; Refresh
 300 ; Retry
 3600000 ; Expire
 14400 ) ; Minimum
 IN NS ucbvax.Berkeley.EDU.
0 IN PTR loopback.ucbvax.Berkeley.EDU.
1 IN PTR localhost.
```

```
primary 0.0.127.IN-ADDR.ARPA named.local
cache root.cache
```

Tous les fichiers sont relatifs au répertoire /usr/local/domain

L'entrée cache dit à *named* de maintenir un cache aux réponses du serveur de noms et de l'initialiser avec le fichier *root.cache* . Ce nom est choisi par l'administrateur. Nous verrons son contenu dans le paragraphe suivant. L'entrée primary définit le serveur local comme serveur primaire de son propre domaine en loopback. Ce domaine est un in-addr.arpa domaine qui donne comme adresse à la machine locale 127.0.0.1.

• Configuration primary et secondary server

Voici le début du fichier qui définit imag comme serveur primaire sur le domaine imag.fr :

```
; ;  @(#)named.boot  1.3 (Berkeley)  86/01/30 IMAG JUIN 1990
;  boot file for primary and secondary name server
;
directory         /usr/spool/named
sortlist          129.88.0.0 147.171.0.0 130.190.0.0
;
; type            domain                    source file or host
;
primary           0.0.127.in-addr.arpa      rev.127.0.0
primary           imag.fr.                  imag.fr.zone
```

La deuxième entrée dit que la machine imag est le serveur primaire pour le domaine imag.fr et que les données pour ce domaine sont à lire dans le fichier *imag.fr.zone*.

Pour une configuration de serveurs secondaires, les entrées secondary n'indiquent plus des fichiers locaux mais des serveurs distants comme source d'information sur les domaines. Les entrées secondaires donnent le nom du domaine, l'adresse du serveur primaire pour de domaine et l'adresse le nom du fichier local ou l'information reçue du serveur primaire doit être stockée.

Voici ce fichier qui définit ensisun comme serveur secondaire:

```
directory /var/spool/named
sortlist 129.88.0.0 147.171.0.0 130.190.0.0
secondary imag.fr 129.88.32.1 imag.fr.bk
secondary 88.129.in-addr.arpa 129.88.32.1 129.88.bk
cache . root.cache
forwarders 129.88.32.1
slave
```

ii-4.3b. Le fichier d'initialisation cache *root.cache*

Une entrée dans chaque fichier *named.boot* indique quel est le fichier d'initialisation cache. Chaque serveur à ce fichier.

Si le fichier *resolv.conf* n'existe pas, le resolver enlève la première partie de l'affichage de hostname, i.e ensisun et utilise le reste, i.e imag.fr comme nom de domaine. Cela marche correctement si le nom de la machine est «*fully qualified domain name FQDN*». Si *hostname* retourne simplement ensisun, un fichier *resolv.conf* avec le nom de domaine par défaut est nécessaire.

ii-4.2b. Fichier de configuration *resolv.conf*

Si le système local ne fait pas tourner *named* ou si le nom du domaine ne peut pas être tiré de hostname, on doit utiliser le fichier *resolv.conf*. La configuration avec ce fichier a quelques avantages sur la configuration par défaut. La configuration est définie clairement et elle permet de choisir un serveur de nom autre que celui par défaut au cas ou ce dernier ne réponde plus. Ce fichier a deux entrées :

- **nameserver** address : identifie le ou les serveuri(s) de nom par son adresse IP. Si cette entrée n'existe pas dans le fichier, le serveur de nom est supposé être la machine locale. Sur une machine configurée resolver-only, le fichier *resolv.conf* contient des noms de serveurs mais qui ne sont jamais la machine locale.

- **domain** name : définit le nom du domaine par défaut, par exemple imag.fr.

II-4.3 Configuration de named

Alors que la configuration du resolver nécessite au plus un fichier, plusieurs fichiers sont nécessaires pour configurer *named*.

ii-4.3a. Le fichier *named.boot*

Ce fichier indique les sources de l'information DNS. Ces sources sont soit des fichiers locaux, soit des serveurs distants. Les entrées de ce fichier sont les suivantes :

• **directory** :	définit un répertoire de référence des fichiers.
• **primary** :	déclare que ce serveur est primaire dans la zone spécifiée.
• **secondary** :	déclare que ce serveur est secondaire dans la zone spécifiée.
• **cache** :	indique le fichier cache.
• **forwaders** :	liste des serveurs ou les requêtes sont expédiées.
• **slave** :	force le serveur à seulement utiliser des forwaders.

La façon dont on configure *named.boot* indique si on utilise le serveur comme primary server, secondary server ou catching-only server.

- **Configuration catching-only server**

Le contenu du fichier *named.boot* ressemble à ce qui suit :

```
;
; a catching-only server configuration
;
directory /usr/local/domain
```

- **catching-only** : ce type de serveur ne gère pas de fichiers de données. Il determine les réponses aux requêtes à partir d'autres serveurs distants. Une fois qu'il connaît la réponse, le serveur conserve l'information pour d'autres demandes futures de la même information. Ce type de serveur n'est pas autoritaire, vu qu'il dépend d'autres serveurs. Seul un fichier cache est nécessaire pour conserver les informations temporairement. Ce type de configuration de serveur est sûrement la plus répandue et la plus simple à mettre en place avec la configuration précédente.

- **primary** : le serveur de nom primaire est maître sur tout le domaine qu'il gère. Il connaît les informations concernant le domaine à partir d'un fichier local fait par l'administrateur réseau. Ce fichier de zone contient les informations précises sur le domaine ou le serveur est maître. La configuration de ce serveur demande un ensemble de fichiers : le fichier de zone pour le domaine (*named.hosts*) et le domaine à l'envers (*named.rev*),le fichier boot(*named.boot*), le fichier cache (*named.ca*) et le fichier de loopback (*named.local*).

- **secondary** : un serveur secondaire transfére les informations d'un serveur primaire chez lui. Le fichier de zone est ainsi transféré et est stocké dans un fichier local. Ce type de serveur a une copie complète des informations du domaine; on le considère par conséquent comme un serveur maître.

La configuration d'un serveur est une de celles ci-dessus mais peut aussi en regrouper plusieurs. Cependant, tous les systèmes doivent faire tourner un resolver. Commençons à regarder la configuration de la partie cliente du DNS.

II-4.2 Configuration du resolver

Il y a deux manières de procéder à la configuration de resolver : utilisation de la configuration par défaut ou utilisation du fichier *resolv.conf*.

Le resolver n'est pas un processus distinct; c'est une librairie de routines. Si le fichier *resolv.conf* existe, il est lu chaque fois qu'un processus utilisant le resolver commence. Ce fichier n'est pas demandé par les systèmes qui font tourner *named*. Tous ces systèmes peuvent utiliser la configuration par défaut.

ii-4.2a. Configuration par défaut

Pour connaître le domaine par défaut, le resolver utilise la configuration par défaut. Il utilise la machine locale comme serveur de nom par défaut et tire le nom du domaine par défaut de la sortie de la commande *hostname*.

Pour que la configuration par défaut fonctionne, il faut que la machine locale fasse tourner *named*.

hostname est une commande UNIX qui permet de vérifier ou d'installer le nom de la machine. Seul, le root peut en faire l'installation. Par contre, tout le monde peut en faire la vérification :

```
% hostname
ensisun.imag.fr
```

utilisé sur certains sites. **gated** est un seul logiciel qui combine à la fois RIP, Hello, BGP et EGP. Les avantages de **gated** sont :

- sur les systèmes qui utilisent plus d'un protocole de routage, **gated** combine les informations des différents protocoles et en tire la meilleure route.
- les routes apprises d'un protocole intérieur peuvent être annoncées via un protocole extérieur. Les informations entre systèmes autonomes s'adaptent aux changements de routes intérieur.
- **gated** simplifie la configuration. Tout tient dans un seul fichier : /etc/gated.conf.

ii-3.4c. Choix d'un protocole de routage

Bien qu'il y ait beaucoup de protocoles existant, en choisir un est relativement facile. Pour des réseaux locaux, RIP est le plus courant. OSPF n'est pas encore largement disponible et Hello n'a jamais été largement utilisé.

Pour un protocole extérieur, on a rarement le choix du protocole. Deux systèmes autonomes qui échangent des informations doivent utiliser le même protocole. Si l'autre système autonome fonctionne déjà, il faut alors utiliser le même protocole. Ce choix est souvent EGP même si BGP se diffuse de plus en plus.

II-4 Configuration du DNS[1]

Le service de nom DNS n'est pas vraiment indispensable pour la communication entre machines. Comme son nom l'indique, c'est un service dont le but est de rendre le réseau plus convivial. Le réseau fonctionne très bien avec les adresses IP mais l'utilisateur préfère utiliser des noms.

II-4.1 BIND

Au sein d'UNIX, DNS est implémenté par le **BIND** (Berkeley Internet Name Domain). C'est un programme qui repose sur une architecture client/serveur. La partie client du BIND est appelée le **resolver**. Il génère les demandes qui sont envoyées au serveur. Le serveur DNS répond alors aux requêtes du resolver. La partie serveur du BIND est un démon appelé *named*. Ce chapitre couvre les trois opérations à faire sur le BIND :

- configuration du **resolver**
- configuration du serveur de nom *named*
- construction des fichiers de données du serveur de nom, appelés **fichiers de zone**

BIND peut être configuré de plusieurs manières. Ces différentes configurations sont :

- **resolver-only systems** : sur les systèmes UNIX, le resolver n'est pas un client séparé mais plutôt une librairie. Certains systèmes utilisent seulement le resolver. Ils sont faciles à configurer: il suffit d'initialiser le fichier */etc/resolv.conf*. Ce type de configuration n'est quand même pas très courant. Elle est utilisée quand il y a une limitation technique qui empêche de faire tourner le serveur.

  (Les trois autres configurations sont toutes pour le serveur *named*).

---

[1] DNS = Domain Name Server.

La plus grande longueur pour RIP est de 15. Au dessus de 15, RIP suppose que la destination n'est pas joignable. En supposant que la meilleure route est la plus courte, RIP ne prend donc pas en compte les problèmes de congestion.

Inclus dans UNIX, RIP tourne grâce au routing daemon, *routed*. Le routing daemon construit les tables de routages avec les informations de mise à jour RIP. Les systèmes configurés pour traiter RIP échangent périodiquement ou sur demande ces informations de mise à jour.

D'autres protocoles ont été développés pour pallier à cette limitation de congestion.

Hello est un protocole qui utilise le délais[1] pour décider de la meilleure route. Le délais est le temps que met un paquet pour aller de la source à la destination et revenir ensuite à la source. Hello n'est pas largement utilisé. On le trouve peu en dehors de NSFNET.

ii-3.4b. Protocole de routage extérieur

Les protocoles de routage extérieur sont utilisés pour échanger des informations entre système autonomes. Les informations de routage qui passent entre des systèmes autonomes sont appelées *reachability information*. Ces informations indiquent quels réseaux peuvent être atteints à travers un système autonome spécifique.

Exterior Gateway Protocol (EGP) est le plus utilisé des protocoles extérieurs. Une passerelle qui utilise EGP annonce qu'elle peut atteindre les réseaux qui font partie de son système autonome.

Contrairement aux protocoles intérieurs, EGP n'essaie pas de choisir la meilleure route. EGP met à jour les informations de distance mais n'évalue pas ces informations. Ces informations de distance ne sont pas directement comparables parce que chaque système autonome utilise des critères différents pour évaluer ces valeurs.

Une structure de routage qui dépend d'un groupe de passerelles centralisées ne peut pas convenir à un accroissement rapide de Internet. C'est une des raisons pour lesquelles Internet tend vers une architecture distribuée ou un processus de routage tourne sur chaque système autonome.

Un autre protocole, **Border Gateway Protocole** (BGP) commence à remplacer EGP. Comme EGP, BGP échangent des informations entre systèmes autonomes mais BGP peut fournir plus d'informations pour chaque route et peut utiliser ces informations pour sélectionner la meilleure route.

Une remarque importante à se rappeler est que la plupart des systèmes ne font pas tourner de protocoles extérieurs. Ces protocoles ne sont utiles que pour des systèmes autonomes qui échangent des informations avec d'autres systèmes autonomes. La plupart des machines appartenant à un système autonome font tourner RIP. Seules les passerelles reliant deux systèmes autonomes font tourner un protocole extérieur.

EGP tourne soit comme processus séparé (**egpup**), soit comme partie du *Gateway Routing Daemon* (**gated**). Il est préférable d'utiliser **gated**. On utilise encore **egpup** car il est encore

---

[1] Round Trip Time

• L’argument suivant est l’adresse de la passerelle. Cette adresse doit être celle d’une passerelle directement connectée au réseau.

• Le dernier argument est la mesure de routage (routing metric). *Route* utilise cette mesure pour décider s’il s’agit d’une route à travers une interface locale ou bien à travers une passerelle externe. Si cette mesure vaut 0, la route est supposée passer par une interface locale et le flag G n’est pas mis en place. Si c’est supérieur à 0, le flag G est mis en place car la route est supposée passer par une passerelle externe.

La table de routage statique d’ensisun après avoir ajouter différentes routes est la suivante :

```
% netstat -nr
Routing tables
Destination       Gateway          Flags    Refcnt  Use        Interface
127.0.0.1         127.0.0.1        UH       5       2017       lo0
192.33.174.160    192.33.174.33    UG       0       0          le0
192.33.174.128    192.33.174.33    UG       21      13874      le0
129.88.0.0        192.33.174.62    UG       17      438296     le0
default           192.33.174.62    UG       0       161657     le0
192.33.174.96     192.33.174.35    UG       4       3509       le0
192.33.174.64     192.33.174.65    U        52      1273321    ne0
192.33.174.32     192.33.174.34    U        133     887308     le0
152.77.0.0        192.33.174.62    UG       0       0          le0
192.33.175.0      192.33.174.36    UG       0       28450      le0
```

Pour comprendre les différentes destinations de la table de routage, il suffit de se reporter au fichier */etc/networks* :

129.88.0.0 est le réseau net-imag

192.33.175.0 est le réseau net-ensimag23

Les autres réseaux 192.33.174.xxx sont les sous-réseaux du réseau net-ensimag01. Ce sont les réseaux : net-ensipc, net-ensuntx, net-ensibmtx, net-ensibultx,... Rapellons que ensisun est directement connecté sur net-ensuntx (192.33.174.64) et sur net_ensimag (192.33.174.32)

II-3.4 Les différents protocoles de routage

Tous les protocoles de routage assurent les mêmes fonctions de base : ils déterminent la meilleure route pour chaque destination et échangent les informations de routages entre les différents systèmes du réseau. Par contre, leur manière de procéder est différente.

ii-3.4a. Protocole de routage intérieur

Un protocole intérieur est utilisé sur un réseau indépendant. Ces réseaux indépendants sont aussi appelés système autonome en terminologie TCP/IP. Au sein d’un système autonome, l’information de routage est échangée en utilisant un protocole intérieur.

Routing Information Protocole (RIP) est le plus utilisé des protocoles intérieurs car il est inclus dans UNIX. RIP sélectionne la route dont la longueur est la plus faible comme étant la meilleure route. La longueur d’une route pour RIP est le nombre de passerelles que les données doivent franchir pour atteindre leur destination. RIP suppose que la meilleure route est celle qui utilise le moins de passerelles.

```
% ping -s ensibm
PING ensibm: 56 data bytes
64 bytes from ensibm (192.33.174.36): icmp_seq=0. time=3. ms
64 bytes from ensibm (192.33.174.36): icmp_seq=1. time=4. ms
64 bytes from ensibm (192.33.174.36): icmp_seq=2. time=4. ms
64 bytes from ensibm (192.33.174.36): icmp_seq=3. time=6. ms
^C
----ensibm PING Statistics----
4 packets transmitted, 4 packets received, 0% packet loss
round-trip (ms) min/avg/max = 3/4/6

% ping -s ensibull
PING ensibull: 56 data bytes
64 bytes from ensibull (192.33.174.35): icmp_seq=0. time=4. ms
64 bytes from ensibull (192.33.174.35): icmp_seq=1. time=15. ms
64 bytes from ensibull (192.33.174.35): icmp_seq=2. time=81. ms
^C
----ensibull PING Statistics----
3 packets transmitted, 3 packets received, 0% packet loss
round-trip (ms) min/avg/max = 4/33/81
```

ping utilise le protocole ICMP pour forcer la machine destinatrice à renvoyer un écho vers la machine locale. Si les paquets circulent entre les deux machines, cela veut dire que le routage est bon.

Si on fait un *ping* sur une machine qui n'est pas sur le réseau local, les résultats sont différents.

```
% ping 26.40.0.17
sendto: Network is unreachable
```

Ici, le message «sendto: Network is unreachable» montre que ensisun avec sa table de routage minimale ne sait pas envoyer des données au réseau ou se trouve la machine 26.40.0.17.

Ce test montre que la table de routage crée par *ifconfig* permet seulement la communication sur des machines sur le réseau local.

II-3.3 Construction d'une table de routage statique

Comme nous l'avons vu, la table de routage minimale ne marche qu'avec les machines directement connectées sur le réseau local. Grâce à la commande *route*, on rajoute des routes à travers des passerelles externes jusqu'à des machines distantes.

```
% route -n add 152.77.0.0 192.33.174.62 1
add net 152.77.0.0: gateway 192.33.174.62
```

- Le premier argument après *route* est soit **add** pour rajouter, soit **del** pour effacer une route.
- L'argument suivant 152.77.0.0 est l'adresse destination qui est l'adresse atteinte par cette route. Si le mot-clé **default** est utilisé, *route* crée une route par défaut. La route par défaut est utilisée quand il n'y a pas de routes spécifiques à une destination donnée.

- **routage statique** : un réseau avec un nombre limité de passerelles peut se configurer avec un routage statique. Une table de routage statique est construite manuellement par l'administrateur système qui utilise la commande *route*. Ces tables sont à utiliser seulement quand les routes ne sont pas modifiées, vu qu'elles ne s'auto-adaptent pas aux changements de réseaux. Quand une machine éloignée ne peut être atteinte qu'avec une seule route, le routage statique est la meilleure solution.

- **routage dynamique** : un réseau ou une même destination peut être atteinte par plusieurs routes devrait utiliser le routage dynamique. Une table de routage est construite à partir des informations échangées par les protocoles de routage. Ces protocoles de routage ont pour but de mettre à jour les tables de routage et de choisir quelle est la meilleure route vers une destination.

  Les routes sont construites à coup d'*ifconfig*, dans un script écrit par l'administrateur ou, dynamiquement par les protocoles de routage.

II-3.2 La table de routage minimale

Une fois la configuration de l'interface réalisée, la commande *netstat -nr* permet de connaître la table de routage construite par *ifconfig*, . Imaginons ensisun sur un réseau local coupé du monde.

```
% netstat -nr
Routing tables
Destination       Gateway          Flags    Refcnt  Use        Interface
127.0.0.1         127.0.0.1        UH       5       2017       lo0
192.33.174.64     192.33.174.65    U        52      1273321    ne0
192.33.174.32     192.33.174.34    U        133     887308     le0
```

La signification des Flags est la suivante :

- **U** indique l'état de la route (U si up)
- **G** indique si la route se dirige vers une passerelle (G si gateway). Dans le cas d'une table de routage minimale, le flag G n'apparait pas car toutes les routes sont directes à travers les interfaces locales, sans utiliser de passerelles extérieures.
- **H** indique que la route se dirige vers une machine (H si host). L'adresse destination est une adresse de machine et non pas une adresse de réseau. L'adresse de loopback est 127.0.0.0. L'adresse destination (127.0.0.1) est l'adresse de la machine locale. Cette route particulière se retrouve dans chaque table de routage.

La plupart des routes sont des routes vers des réseaux et non pas des routes vers des machines. C'est pour réduire la taille des tables. En effet une organisation peut avoir un seul réseau mais des centaines de machines sur ce réseau. Une table de routage avec une route pour chaque machine aurait une taille énorme et est donc inconcevable.

La colonne **refcnt** donne le nombre courant d'utilisateurs actifs par route. Les protocoles en mode connecté comptent pour une seule route durant la connexion alors que les protocoles en mode déconnecté obtiennent chaque fois une route pour les données envoyées vers la même destination.

La colonne **use** indique le nombre de paquets envoyés par route.

La colonne **interface** indique l'interface utilisée pour la route.

Pour vérifier la table de routage d'ensisun et ainsi le bon fonctionnement du routage, on peut faire un *ping* sur la machine ensibm ou ensibull.

Par conséquent , PPP est plus robuste que SLIP mais il est aussi plus difficile à implementer et n'est pas disponible aussi facilement que SLIP. Cependant, ses avantages en font le protocole de ligne série du futur.

ii-2.4b. choix d'un protocole série

Le choix ne doit pas porter sur le meilleur protocole dans l'absolu mais plutôt sur le protocole le plus adapte à une situation spécifique. Si vous avez un réseau étendu, vous utiliserez plutôt PPP.

PPP est préfère car c'est un standard Internet. Il offre donc plus de compatibilité entre les systèmes. PPP est aussi plus robuste que SLIP. Ces caractéristiques en font un bon protocole pour la connexion de routeurs sur une ligne série. Cependant, comme SLIP à été le premier protocole série pour IP largement diffusé, et comme il est simple à implementer, SLIP est disponible sur plus de matériels que PPP.

SLIP et PPP sont deux protocoles complètement différents et par conséquent incompatibles. Si votre serveur à seulement SLIP, la machine distante ou vous vous connectez doit aussi avoir SLIP. A cause de son installation plus ancienne, SLIP continuera à être largement utilisé dans le futur.

Donc, quel protocole utiliser ? Les deux ! PPP est le protocole du futur. Cependant il faut continuer a utiliser SLIP car c'est souvent le seul protocole série utilisable sur certains types de matériels.

Utilisez PPP quand vous le pouvez et SLIP quand vous le devez !

II-3 Configuration du routage

Internet repose sur le principe du routage. Sans le routage, le trafic serait limité à un seul câble physique. Le routage permet à votre machine d'atteindre une autre machine qui n'est pas sur le même réseau local que le votre. La communication peut très bien traverser une succession de réseaux intermédiaires afin de s'établir entre les deux machines.

En tant qu'administrateur système, on doit s'assurer que le routage du système est bien configuré. C'est le propos de ce chapitre.

II-3.1 Les différentes configurations de routage

Les trois configurations de routage les plus courantes sont :

- **routage minimal** : s'applique aux réseaux complètement isolés des autres réseaux. Une table de routage minimale est construite par *ifconfig* une fois l'interface réseau configurée. Si votre réseau n'a pas d'accès à d'autres réseaux TCP/IP et si vous n'utilisez pas de sous-réseaux, il se peut que ce soit la seule table utilisée.

II-2.4 TCP/IP sur une ligne série

TCP/IP traite une grande variété de réseaux physiques. Le support physique peut être du câble Ethernet comme dans notre réseau local, des fibres optiques (dans notre cas seulement entre les batiments) ou bien des lignes téléphoniques comme dans les WAN[1]. Presque toutes les communications se font via des interfaces séries. Une interface série envoie les données en un flot de bits sur un simple câble. Ce type d'interface correspond à presque toutes les interfaces de communications. y compris Ethernet, mais on emploie habituellement ce terme pour une interface connectée au réseau téléphonique via un modem. Une ligne téléphonique est souvent appelée une ligne série.

ii-2.4a. les protocoles séries

Le premier protocole reseau sur ligne série à avoir été crée est le **SLIP** (Serial Line IP) . Ce protocole permet aux machines isolées de se connecter au réseau téléphonique, via TCP/IP. SLIP envoie les datagram à travers une ligne série comme une série d'octets et utilise des caractères spéciaux pour indiquer quand une série d'octets doit être regroupée pour former un datagram.

SLIP définit deux caractères spéciaux :

- le caractère END : marque la fin d'un datagram. Quand SLIP reçoit un caractère END, il sait qu'un datagram complet peut être envoyé à IP.
- le caractère ESC : permet de différencier les caractères de contrôle.

Mais SLIP a quelques inconvénients :

- SLIP n'est pas un standard Internet.
- SLIP ne définit pas d'information de contrôle qui pourraient contrôler dynamiquement les caractéristiques de la connexion. Par conséquent, SLIP doit supposer certaines caractéristiques. Donc, à cause de cette limitation, SLIP est seulement utilise quand les deux machines connaissent mutuellement leurs adresses.
- SLIP ne corrige pas les effets du bruit des lignes téléphone. Le protocole ne fait pas de correction d'erreurs ni de compression de données.

Pour beaucoup d'applications sur des machines isolées, ces problèmes ne sont pas importants. Cependant dans un environnement dynamique comme celui des WAN, ces problèmes rendent le protocole inadéquat pour connecter les routeurs.

Pour répondre à cette faiblesse, **PPP** (Point to Point Protocole) a été développé comme un standard Internet. C'est un protocole a trois couches :

- **Data Link layer Protocol** : assure un envoi fiable des données sur n'importe quel type de ligne série.
- **Link Control Protocol** : fournit les informations de contrôle sur la ligne série. Cette couche est utilisée pour établir la connexion, négocier les paramètres de configuration, vérifier la qualité de liaison et clore la connexion.
- **Network Control Protocol** : Cette couche fournit les informations de configuration et de contrôle nécessaires à LCP. PPP est destine a faire transiter les données pour une grande variété de protocoles réseau. Cette couche permet à PPP de faire cela.

---

[1] WAN = Wide Area Network.

La commande *ifconfig* est normallement exécutée au moment du boot par un fichier de startup. Sur les systèmes BSD UNIX, les commandes *ifconfig* sont normalement placées dans les fichiers */etc/rc.boot* et */etc/rc.local*. Le script de /etc/rc.boot est exécuté à la procédure de startup alors que le *rc.local* est exécuté à la fin. Editez le fichier */etc/rc.boot* pour lire les lignes suivantes :

```
ifconfig le0 192.33.174.34 netmask 255.255.255.224 broadcast 192.33.174.63 -
trailers up
ifconfig ne0 192.33.174.65 netmask 255.255.255.224 broadcast 192.33.174.95 -
trailers up
ifconfig lo0 127.0.0.1 up
```

II-2.3 Autres options de *ifconfig*

On a utilisé i*fconfig* pour installer l'adresse IP de l'interface, son masque de sous-réseau et son adresse de diffusion. Ce sont certainement les fonctions les plus importantes de *ifconfig* mais il existe d'autres fonctions aussi qui sont :

- **activer et desactiver l'interface** :

  up active l'interface et **down** la désactive. Pour reconfigurer une interface, il faut d'abord la désactiver. Par exemple, pour changer l'adresse IP de le0 :

```
% ifconfig le0 down
% ifconfig le0 192.33.174.50 up
```

- **ARP et trailer**[1] :

  L'option **trailer** autorise l'encapsulation des datagrammes IP, **-trailer** l'inhibe. La plupart des systèmes autorisent l'encapsulation par défaut.

  L'option **arp** autorise le protocole ARP (Address Resolution Protocol), **-arp** l'inhibe. Ce protocole traduit les adresses IP en adresses Ethernet. A de rares exceptions près, ce protocole est toujours autorisé.

- **Metric** :

  le protocole de routage RIP (Routing Information Protocol) choisit une route en fonction de son coût. Ce coût est déterminé par une mesure associe a la route. Plus ce nombre est petit, plus le coût est faible. Quand il construit sa table de routage, RIP privilégie les routes de faibles coûts. Par défaut, la mesure d'une route entre une interface et un réseau qui lui est directement attaché est 0. Pour augmenter par exemple le coût d'une interface a 3, afin que RIP privilégie les routes de valeur 0,1 ou 2 on écrit :

```
% ifconfig le0 192.33.174.34 metric 3
```

On utilise cette option seulement s'il existe une autre route qui conduit à la même destination et qu'on désire l'utiliser comme route principale.

---

[1] Ces options sont utilisables seulement sur les interfaces Ethernet.

- **Net/Dest** : Indique le réseau ou la machine auxquels l’interface a accès.
- **Address** : adresse IP de l’interface
- **Ipkts** : Input Packets indique le nombre de trames reçues par l’interface.
- **Ierrs** : Input Errors indique le nombre de trames avec erreurs reçues par l’interface.
- **Opkts** : Output Packets indique le nombre de trames émises par l’interface.
- **Oerrs** : Output Errors indique le nombre de trames avec erreurs émises par l’interface.
- **Collis** : Collisions indique le nombre de collisions détectées par l’interface.
- **Queue** : Indique le nombre de trames en file d’attente d’émission de l’interface.

Le résultat de la commande *netstat -ain* sur la machine ensisun est le suivant :

```
%netstat -ain
Name Mtu Net/Dest Address Ipkts Ierrs Opkts Oerrs Collis Queue
le0 1500 192.33.174.32 192.33.174.34 1242702 1 985946 0 10284 0
le1* 1500 none none 0 0 0 0 0 0
ne0 1500 192.33.174.64 192.33.174.65 1009315 131 985120 0 419 0
lo0 1536 127.0.0.0 127.0.0.1 178672 0 178672 0 0 0
```

où :

- L’interface lo0 est l’interface de «loopback», que tous les systèmes possèdent.
- L’interface le0 est configurée pour Ethernet
- Pour information, l’interface std0 (DDN Standard X25), que l’on ne trouve pas ici, est configurée pour Milnet.

II-2.2 Vérification de l’interface avec *ifconfig*

Un script d’installation UNIX permet de configurer le réseau à la place de l’administrateur. Cependant, cette configuration peut ne pas convenir. On peut vérifier les caractéristiques d’une interface grâce a *ifconfig*.

```
% ifconfig lo0
lo0: flags=49<UP,LOOPBACK,RUNNING>
 inet 127.0.0.1 netmask ff000000
% ifconfig le0
le0: flags=63<UP,BROADCAST,NOTRAILERS,RUNNING>
 inet 192.33.174.34 netmask ffffffe0 broadcast 192.33.174.63
% ifconfig ne0
ne0: flags=63<UP,BROADCAST,NOTRAILERS,RUNNING>
 inet 192.33.174.65 netmask ffffffe0 broadcast 192.33.174.95
```

où, la première ligne indique le nom et les flags qui caractérisent l’interface. Les flags sont décrits par un nombre dont les bits sont traduits en un ensemble de noms. Ainsi 63 correspond a :

- **UP** : l’interface est disponible.
- **BROADCAST** : l’interface traite le diffusion, i.e elle est connectée a un réseau qui traite la diffusion comme Ethernet.
- **NOTRAILERS** : l’interface ne traite pas l’encapsulation.
- **RUNNING** : l’interface est utilisée

La seconde ligne affichant des informations attachées à TCP/IP, i.e adresse IP, masque de sous-réseau et adresse de diffusion.

```
Hostname :                   txsun01
IP adress :                  192.33.174.70
Subnet mask :                FF.FF.FF.E0
Default gateway :            192.33.174.65 (ensisun)
Broadcast address :          192.33.174.95
Domain Name :                imag.fr
Primary Name Server :        192.33.74.34
Secondary Name Server :      0.0.0.0
Routing Protocol :
```

II-2 Configuration de l'interface

Une des forces du protocole TCP/IP est d'être indépendant du support physique. Cet avantage augmente en fait la charge de travail de l'administrateur car il doit alors indiquer au protocole TCP/IP quelles interfaces il utiliser et pour chaque interface, il doit donner ses caractéristiques. Contrairement aux adresses Ethernet qui sont implémentées en hard, l'administrateur système attribue à chaque interface réseau une adresse IP, un masque de sous réseau et une adresse de diffusion.

II-2.1 La commande *ifconfig*

Cette commande permet d'installer ou de vérifier les attributs associés à chaque interface. L'exemple suivant est celui de la configuration de l'interface le0 de ensisun et de sa vérification ensuite :

```
% ifconfig le0 192.33.174.34 netmask ffffffe0 broadcast 192.33.174.63
% ifconfig le0
le0: flags=63<UP,BROADCAST,NOTRAILERS,RUNNING>
        inet 192.33.174.34 netmask ffffffe0 broadcast 192.33.174.63
```

où :

- le0 est le nom de l'interface d'ensisun que l'on configure.
- 192.33.174.34 est l'adresse IP de cette interface.
- ffffffe0 est le masque de sous réseau. On utilise un masque car le réseau net-ensimag01 d'adresse 192.33.174 est divisé en sous-réseaux. La subdivision est faite sur 3 bits (e0).
- 192.33.174.63 est l'adresse de diffusion, i.e tous les bits de la partie adresse de la machine sont à 1.

L'administrateur réseau fournit les valeurs des adresses, du masque de sous-réseau et l'adresse de diffusion. Ces valeurs sont directement tirées de la feuille de planning. Par contre, le nom de l'interface , qui est le premier argument de chaque ligne de commande *ifconfig*, est tiré de la documentation système.

Il existe une commande, *netstat*, dont une des fonctions est d'indiquer quelles sont les interfaces disponibles sur le système. La commande *netstat -ain* fournit en sortie les champs suivant pour chaque interface :

• **Name** :	Nom de l'interface. * indique que l'interface n'est pas disponible, i.e que l'interface n'est pas «UP».
• **Mtu** :	Maximum Transmission Unit montre la plus grande longueur de trame transmise par l'interface sans qu'il y ait de fragmentation.

L'administrateur réseau décide du type de routage utilisé ainsi que du choix de la passerelle par défaut de chaque machine.

Les protocoles de routages EGP et BGP demandent que les passerelles aient un «autonomous système number». Si votre réseau est connecté à un autre réseau qui utilise EGP ou BGP, il faut alors faire la demande de ces numéros auprès du NIC.

II-1.5 Définition d'un masque de sous-réseau

Les raisons pour lesquelles on divise un réseau en sous-réseau sont d'ordre topologique ou d'ordre organisationnel.

Les raisons d'ordre topologiques sont :

- **limitation en distance** : un réseau local sur Ethernet épais est limité en distance à un tronçon de 500m. On peut relier les câbles grâce à des routeurs IP pour augmenter la distance. A chaque câble est associe un sous-réseau.
- **connexions de réseaux de supports différents** : cela consiste par exemple, à relier de l'Ethernet avec du Token Ring en utilisant un routeur IP. On définir alors deux sous-réseaux.
- **filtrage du trafic** : les trames ne sortent du sous-réseau local que si elles sont destinées à un autre sous-réseau. Ce dernier n'est donc pas encombré par des trames qui ne lui sont pas destinées. Ce filtrage est réalisé dans les passerelles entre sous-réseaux.

Les raisons d'ordre organisationel sont :

- **simplification de l'administration du réseau** : on délègue le travail d'administration au niveau de chaque sous-réseau.
- **isolation du trafic** : pour des problèmes de sécurité, un département peut souhaiter que ses trames ne circulent pas sur tout le réseau. La subdivision du réseau en sous-réseaux apporte un début de solution.

II-1.6 Spécification de l'adresse de diffusion

C'est une adresse ou tous les bits de la partie adresse de la machine sont mis à 1. L'adresse de diffusion sur le réseau 192.33.174.0 est par exemple 192.33.174.255. L'adresse de diffusion est mise en place en utilisant la commande *ifconfig*.

II-1.7 Feuilles de planning

Apres avoir regroupé les différentes informations nécessaires au réseau, l'administrateur réseau distribue à chaque administrateur système de chaque machine les informations dont il dispose. L'administrateur système des terminaux X connectés à ensisun recevra par exemple la feuille suivante pour chaque terminal :

II-1.2 Obtention d'un nom de domaine

Un nom de domaine est obtenu comme pour l'adresse IP après avoir fait une demande auprès du NIC. Les demandes sont à envoyer aux mêmes adresses que celles du paragraphe précédent. Même si le réseau n'est pas connecté à Internet, il est quand même conseillé de faire appel au NIC. Il y a deux raisons à cela. La première est celle donnée au paragraphe précédent : on ne connaît pas l'avenir et on voudra peut être un jour connecter le réseau à Internet. La deuxième est plus immédiate : de nombreux réseaux non connectés à Internet ont des passerelles e-mail jusqu'à Internet et quelques uns de ces réseaux permettent à des machines sur Internet de vous adresser du mail avec un nom de domaine du style Internet. Ainsi UUNET et Bitnet offrent ce service.

Quand on demande un nom de domaine Internet, on devrait aussi demander un domaine in-addr.arpa. C'est ce qu'on appelle le domaine renversé. Ce domaine convertit les adresses IP en nom. C'est le processus normal de conversion à l'envers ! Supposons que votre réseau est 128.66. Son nom de domaine renversé est 66.128.in-addr.arpa.

II-1.3 Choix d'un nom de machine

Une fois le nom de domaine obtenu, l'administrateur est libre de choisir le nom de la machine à l'intérieur de ce domaine. Il faut s'assurer que le nom de la machine est unique a travers le domaine ou sous-domaine, de la même manière que l'adresse IP est unique a travers le réseau ou sous-réseau.

Le choix d'un nom de machine peut s'avérer délicat. Voici quelques suggestions pour choisir :

- utiliser des noms courts, facile à épeler et à mémoriser.
- utiliser des noms de thème, de personnages connus...
- éviter d'utiliser des noms personnels, de projets ou venant du jargon technique.

II-1.4 Planning du routage

Si des machines d'un réseau communiquent avec des machines d'un autre réseau, on a besoin de passerelles entre ces réseaux. Une route à travers des passerelles doit alors être définie. Il y a deux manières de le réaliser :

- **table de routage statique** : elle sont construites par l'administrateur système. Leurs mises à jour sont faites manuellement. Elles sont utilisées quand le nombre de passerelles est limité.
- **table de routage dynamique** : elles sont construites par les protocoles de routage. Les protocoles échangent des informations qu'ils utilisent pour la mise à jour des tables. Elles sont utilisées quand il y a beaucoup de passerelles sur le réseau ou quand on peut atteindre la même destination en empruntant plusieurs chemins.

En général, les réseaux utilisent les deux types de table simultanément : les tables de routages statiques sont plus appropriées pour les machines tandis que les passerelles font appels aux protocoles de routage et aux tables de routages dynamiques.

II Configuration

II-1 Démarrage

Avant d'entreprendre la configuration d'une machine on doit avoir certaines informations. Au minimum chaque machine doit avoir une adresse IP unique ainsi qu'un nom. On doit aussi avoir fait son sur certaines options avant de procéder à la configuration.

• **default gateway address** :	si le système communique avec des machines qui ne sont pas sur le réseau local, une adresse de passerelle par défaut est nécessaire.
• **routing protocol** :	chaque machine doit connaitre le protocole de routage utilisé sur le réseau.
• **name server address** :	pour convertir les noms de machines en adresse IP, chaque machine doit connaître les adresses des serveurs DNS.
• **domain name** :	une machine utilisant le service DNS doit connaître correctement son nom de domaine.
• **subnet mask** :	pour que la communication soit propre, chaque machine d'un réseau doit utiliser le même masque de sous-réseau.
• **broadcast address** :	pour éviter les problèmes de diffusion, les adresses de diffusion de chaque machine sur un réseau doivent être identiques.

II-1.1 Obtention d'une adresse

Chaque interface doit avoir une adresse IP unique sur un réseau TCP/IP. Si la machine est sur un réseau connecté à Internet, son adresse doit être unique (sur tout le réseau Internet). Pour un réseau non connecté, cette adresse doit être unique à l'échelle du réseau local. A cause de cela les administrateurs de réseaux non connectés à Internet choisissent souvent des adresses sans consulter le NIC (Network Information Center). Cependant, cela n'est pas recommandé. Si a l'avenir le réseau local a besoin de se connecter à Internet , l'administrateur n'aura pas besoin dans ce cas de changer toutes les adresses et de reconfigurer chaque machine du réseau.

Une demande d'adresse réseau peut se faire de deux manières :

- soit par courrier postal : envoyez votre demande au NIC à l'adresse suivante :

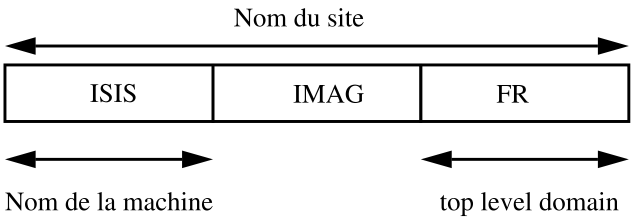
  `DDN Network Information Center`
  `14200 Park Meadow Drive`
  `Suite 200`
  `Chantilly, VA 22021`

- soit par e-mail : adressez votre mail a `hostmater@nic.ddn.mil`

Le NIC vous attribuera un numéro de réseau gratuitement. Supposons par exemple que l'on vous a attribué le numéro 128.66. L'administrateur peut alors utiliser librement les deux derniers octets pour adresser ses machines à l'exception de deux adresses réservées (tous les bits à 0 et tous les bits à 1).

I-3.3 Domain Name Service

DNS est une base de données distribuée qui permet de traduire les noms des sites connectés à Internet en adresses IP. Dans le système DNS, il n'y a pas de base centrale, les informations sont disséminées dans des milliers de serveurs de noms organisés hiérarchiquement. En haut de la hiérarchie se trouve le **root domain** desservi par des **root servers**, juste en dessous on trouve les **top level domains** qui représentent des réseaux de types géographiques ou organisationnels.



Les domaines géographiques correspondent à la distinction de l'ensemble des pays, en dehors des U.S.A., et sont caractérisés par deux lettres.

par exemple :

- fr France
- uk Angleterre

Aux Etats Unis, les top level domains sont du type organisationnel. Ainsi, on trouve :

• COM	organisations commerciales
• EDU	éducation
• GOV	agences gouvernementales
• MIL	organisations militaires
• NET	réseaux
• ORG	tout le reste ...

DNS se décompose en deux applications complémentaires. Le **resolver**, qui constitue la partie client et qui compose les requêtes auquel on ajoute le **name server** qui répond aux questions.

En conclusion DNS permet d'éviter de posséder une host table trop importante et toute nouvelle machine ajoutée est immédiatement connu sur le réseau. De part l'allure de l'accroissement spectaculaire du nombre de machines reliées a Internet, le DNS devient indispensable.

I-3.4 Network Information Service

Sun microsystems ont développé leur propre base de données distribuée NIS qui permet la traduction des noms en adresses IP. Les différences avec DNS sont que NIS ne connaît que le réseau local et non pas Internet dans son ensemble. De plus, NIS fournit un plus grand nombre d'informations que DNS.

I-3 Le serveur de noms

Dans ce chapitre nous allons évoquer comment on peut caractériser une machine par un nom à la place d'une adresse numérique.

I-3.1 Noms et adresses

Comme on l'a vu précédemment, chaque machine est adressée sur le réseau Internet par une adresse numérique IP. Il est cependant plus facile d'utiliser un nom pour désigner une machine ou un réseau plutôt qu'un nombre de 32 bits. Ainsi, par exemple, on peut tout aussi bien faire :

```
ftp isis.imag.fr
```

ou :

```
ftp 129.88.32.1
```

Dans le premier cas, il est nécessaire de convertir le nom de la machine appelée en une adresse numérique. Pour cela, il existe deux méthodes, la plus ancienne utilise la **host table** et la deuxième est basée sur l'interrogation d'une base de données distribuée **Domain Name Service**.

I-3.2 La Host Table

C'est la table qui sert à la conversion des noms en des adresses IP. Cette table se trouve dans le fichier /etc/hosts qui spécifie les adresses IP et les noms de machines correspondant.

```
cat /etc/hosts
# ENSIMAG
#:labo:ensimag          # LIGNE NECESSAIRE POUR LA FACTURATION ET HOST2NAMED
# Adresse routeur entree site
#192.33.174.62  aramis-campus   cisco-ensimag   # Interface ensi-batd
R
# Batiment D 2eme etage
# 129.88.34.2   ensipc18               # PC HP QS16S D204 G.Veillon 4676     R
#
# Batiment D                                      Enseignement ENSIMAG
##129.88.32.16  ensigata        ensigw-a0      # PC BM60 gate vers adminis.  R
##129.88.32.34  ensigate        ensigw-e0      # PC BM60 Passerelle ENSIMAG  R
192.33.174.33   ensigata        ensigw-a0      # PC BM60 gate vers adminis.  R
#192.33.174.34  ensigate        ensigw-e0      # PC BM60 Passerelle ENSIMAG  R
#                                            ##SOUS-RESEAU GROS ############
192.33.174.34   ensisun                      # SUN SPARC 670                C
192.33.174.35   ensibull                     # BULL DPX2 MIPS R6000         C
192.33.174.36   ensibm                       # IBM RS6000                   C
192.33.174.37   ensidpx1                     # BULL DPX2000                 C
192.33.174.38   ensidpx2                     # BULL DPX2000                 C
192.33.174.40   ensitty1                       # Serv.Term. ANNEX           R
192.33.174.41   ensitty2                       # Serv.Term. ANNEX           R
192.33.174.42   ensitty3                       # Serv.Term. ANNEX           R
192.33.174.43   ensitty4                       # Serv.Term. ANNEX           R
192.33.174.44   ensitty5                       # Serv.Term. ANNEX           R
```

Tableau 8 : Wellknown Ports

```
imag(6) cat /etc/services
#
# @(#)services 1.16 90/01/03 SMI
#
# Network services, Internet style
# This file is never consulted when the NIS are running
#
tcpmux          1/tcp                           # rfc-1078
echo            7/tcp
echo            7/udp
discard         9/tcp           sink null
discard         9/udp           sink null
systat          11/tcp          users
daytime         13/tcp
daytime         13/udp
netstat         15/tcp
chargen         19/tcp          ttytst source
chargen         19/udp          ttytst source
ftp-data        20/tcp
ftp             21/tcp
telnet          23/tcp
smtp            25/tcp          mail
time            37/tcp          timserver
time            37/udp          timserver
name            42/udp          nameserver
whois           43/tcp          nicname         # usually to sri-nic
domain          53/udp
domain          53/tcp
hostnames       101/tcp         hostname        # usually to sri-nic
sunrpc          111/udp
sunrpc          111/tcp
#
# Host specific functions
#
tftp            69/udp
rje             77/tcp
finger          79/tcp
link            87/tcp          ttylink
supdup          95/tcp
iso-tsap        102/tcp
x400            103/tcp                         # ISO Mail
x400-snd        104/tcp
csnet-ns        105/tcp
pop-2           109/tcp                         # Post Office
uucp-path       117/tcp
nntp            119/tcp         usenet          # Network News Transfer
erpc            121/udp                         # Annex rpc listener
ntp             123/tcp                         # Network Time Protocol
NeWS            144/tcp         news            # Window System
```

I-2.5 Protocoles et Ports

Sur le réseau, l'ensemble des informations échangées provenant d'applications différentes voyagent toutes sur un même médium physique, on fait du multiplexage. Le mécanisme de demultiplexage doit savoir, à la réception des données, quel protocole il doit appliquer et l'application ciblée.

i-2.5a. Numéros de protocoles

Ce numéro se trouve dans le troisième mot de l'entête du datagramme IP. Lorsque IP reçoit des données, il identifie le protocole à utiliser grâce au fichier */etc/protocols*.

```
cat /etc/protocols
#
# @(#)protocols 1.9 90/01/03 SMI
#
# Internet (IP) protocols
# This file is never consulted when the NIS are running
#
ip      0       IP      # internet protocol, pseudo protocol number
icmp    1       ICMP    # internet control message protocol
igmp    2       IGMP    # internet group multicast protocol
ggp     3       GGP     # gateway-gateway protocol
tcp     6       TCP     # transmission control protocol
pup     12      PUP     # PARC universal packet protocol
udp     17      UDP     # user datagram protocol
```

i-2.5b. Numéros de ports

Une fois que les données sont transmises au bon protocole de communication, il faut qu'elles atteignent à l'application qui les demande. En fait, les processus utilisant des ressources réseaux sont identifiés grâce à leur numéro de port qui est unique. Ainsi, tout comme il y-a une application au départ et à l'arrivée des données, il existe un *source port number* et un *destination port number.*

Les numéros de ports connus **wellknown services** dans le fichier */etc/services*. C'est le couple (Protocole;Port) qui détermine l'acheminement des données dans votre machine. Ainsi, on peut très bien avoir un même numéro de port assigné plusieurs fois avec des protocoles différents.

i-2.5c. Sockets

En plus des wellknown ports (voir page 12) facilitant la connecxion entre deux machines pour des applications standards, il existe des numéros de ports alloués dynamiquement. Par exemple, lors d'un *telnet*,à la connecxion, les machines source et destinataire vont s'échanger sur le port 23 les numéros de ports (alloués dynamiquement) qu'ils vont utiliser pendant la communication. C'est la combinaison d'une adresse IP et d'un numéro de port qui définie une ***socket***.

I-2.4 La résolution d'adresse

L'adresse IP, telle quelle, n'est pas utilisable par la couche physique pour envoyer des données sur le médium de transport. En effet, il existe autant de protocole d'adressage que de types de réseaux physiques.

Le protocole le plus utilisé est l' ARP (Address Resolution Protocol). Il permet la traduction des adresses IP en adresses Ethernet. ARP se charge de construire une table de traduction, dynamiquement, en interrogeant les autres machines reliées sur le réseau ethernet. Pour avoir une idée du contenu de cette table on utilise la commande *arp -a*.

```
/etc/arp -a
floyd            (129.88.32.32) at 8:0:20:a:e5:d7
esperanza-1      (129.88.32.65) at 0:0:a7:10:9f:62
farenheit-451    (129.88.32.33) at 0:0:a7:11:90:5
esperanza-3      (129.88.32.66) at 0:0:a7:10:a0:b1
cap-ferret       (129.88.32.18) at 0:0:a7:0:2f:1d
esperanza-2      (129.88.32.67) at 0:0:a7:10:a0:e
snoopy           (129.88.32.68) at 0:0:a7:10:1d:b1
penduick         (129.88.32.37) at 8:0:20:b:d4:4e
pastorius        (129.88.32.21) at 0:0:a7:0:2f:1
romeo            (129.88.32.22) at 0:0:a7:0:2f:2f
cheops           (129.88.32.54) at 0:0:a7:0:28:6b
igei-test        (129.88.32.39) at 0:0:a7:10:a0:60
sahara           (129.88.32.55) at 8:0:38:42:10:ff
lgitty1          (129.88.32.56) at 0:80:2d:0:3:d3
sphinx           (129.88.32.72) at 0:0:a7:12:3b:5e
durga            (129.88.32.24) at 8:0:20:12:42:b1
chephren         (129.88.32.57) at 0:0:a7:11:a3:da
brahma           (129.88.32.41) at 8:0:20:c:77:3a
dingo            (129.88.32.29) at 8:0:20:7:ac:98
callimaque       (129.88.32.61) at 0:0:a7:11:c0:53
aramis-campus    (129.88.32.254) at 0:0:c:0:9f:21
vigenere         (129.88.32.15) at 8:0:11:1:62:d3
```

La traduction inverse est faite par le *Reverse Address Resolution Protocol RARPA*. Ce protocole est utilisé lors de la configuration TCP/IP des machines sans disque *diskless*. En effet, lors de la mise en route, ces machines ne connaissent même pas leur adresse IP. Elles ont seulement leur adresse Ethernet qui est inscrite dans la ROM de la carte de communication. La machine diskless demande, donc, au travers du réseau, que l'on lui renvoie son adresse IP. Pour cela, certaines machines possèdent un fichier /etc/ethers , éditable manuellement, qui permet de déclarer les couples @Internet et @Ethernet. Par exemple sur notre machine :

```
cat ethers
0:0:c0:6c:28:17         ANONYME
8:0:2b:15:d4:a7         LACAN
8:0:2b:15:d4:a1         MORENO
8:0:2b:15:98:77         PAVLOV
8:0:2b:15:d4:a0         PIAGET
0:0:c0:6c:28:17         SOPHY
aa:0:4:0:1:14           TIMB
```

Tableau 7 : Table de routage (netstat -nr)

Destination	Gateway	Flags	Refcnt	Use	Interface
127.0.0.1	127.0.0.1	UH	18	163175	lo0
129.88.56.2	129.88.32.156	UGHD	0	915	le0
129.88.56.0	129.88.32.156	UG	0	1166	le0
default	129.88.32.254	UG	26	353782	le0
129.88.120.0	129.88.32.254	UG	1	3764	le0
129.88.40.0	129.88.32.29	UG	3	12200	le0
129.88.32.0	129.88.32.1	U	245	823571	le0
129.88.33.0	129.88.32.254	UG	3	51716	le0
129.88.41.0	129.88.32.55	UG	1	6854	le0
129.88.34.0	129.88.32.254	UG	3	23127	le0
129.88.2.0	129.88.32.254	UG	0	1631	le0
129.88.42.0	129.88.32.19	UG	0	186	le0
129.88.59.0	129.88.32.159	UG	0	4290	le0
129.88.51.0	129.88.32.151	UG	0	13622	le0
129.88.100.0	129.88.32.254	UG	3	38934	le0
152.77.0.0	129.88.32.254	UG	0	8399	le0
192.33.174.0	**129.88.32.254**	**UG**	**3**	**214231**	**le0**
129.88.38.0	129.88.32.254	UG	14	1382339	le0
129.88.110.0	129.88.32.254	UG	0	8027	le0
192.33.175.0	129.88.32.254	UG	2	29703	le0
129.88.39.0	129.88.32.254	UG	6	73946	le0
129.88.31.0	129.88.32.254	UG	0	2068	le0
129.88.111.0	129.88.32.254	UG	0	0	le0

Voici la table de routage de la machine **Isis** du réseau imag 129.88.32.1. Lorsque, de Isis, on envoie un datagramme vers la machine **Ensisun** du réseau ensimag 192.33.174.34, la couche IP applique le subnet mask 255.255.255.0 pour déterminer l'adresse du sous réseau 192.33.174.0. Cette adresse permet alors d'aller lire directement dans la table de routage l'adresse du routeur 129.88.32.254 auquel on doit envoyer les datagrammes.

I-2.2 Les sous-réseaux

Parfois, il convient de subdiviser un réseau en sous-réseaux (*subnets*) afin de mieux s'adapter à l'organisation du travail et du personnel. Cette subdivision est faite localement sur une adresse IP en y appliquant un masque (*subnet mask*) qui a pour effet de déplacer la frontière entre l'adresse réseau et l'adresse machine. Dans le cas d'un réseau de classe B, de la forme 255.255.0.0, l'un des masques le plus utilisé est 255.255.255.0. En effet, l'administrateur possède alors la possibilité de séparer son réseau de classe B en 255 sous-réseaux.

I-2.3 La table de routage

Toute machine reliée sur un réseau se charge d'aiguiller les paquets à émettre en fonction de leur destination :

- Si le destinataire est sur le réseau local, le datagramme est directement envoyé à la machine cible.
- Si le destinataire est sur un réseau différent, les données sont envoyées au routeur du réseau local.

La couche IP détermine donc si l'adresse de destination est sur le réseau local, auquel cas elle applique le subnet mask afin de déterminer le sous-réseau puis elle interroge la *table de routage* pour savoir à qui envoyer les données. Si jamais la destination se situe sur un réseau éloigné, la table de routage donne un routeur par défaut de sortie du réseau local.

Cette table peut être obtenu grâce à la commande *netsat -nr* (voir page 10), l'option -r permet d'avoir la table de routage, alors que l'option -n donne les adresses IP sous leur forme numérique. La commande netstat -nr donne pour résultat un tableau avec les champs :

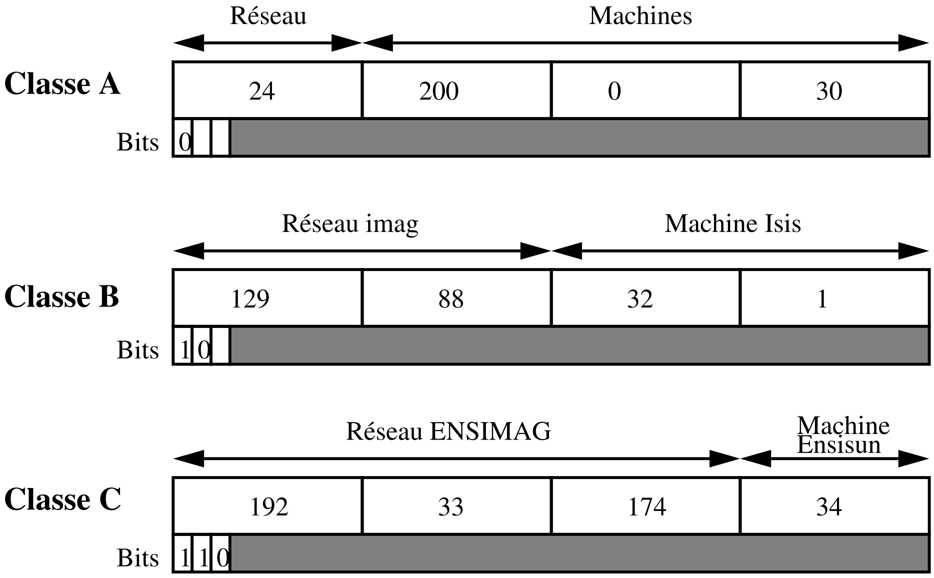
Destination	Adresse du destinataire
Gateway	Le routeur à joindre pour atteindre la destination
Flags	U la ligne est opérationnelle
	H Cette route est vers un site particulier (non pas un réseau)
	G Cette route utilise un routeur ou gateway
	D Cette route est rajoutée car la route normale n'accepte pas le transport (ICMP redirect)
Refcnt	Montre le nombre de fois que la route a été utilisée pour établir une connecxion
Use	Montre le nombre de datagrammes transmis
Interface	Nom de l'interface utilisée pour cette route

La première ligne du tableau correspond à la *loopback route* qui est utilisée lorsque la machine s'envoie des datagrammes à elle même. On distingue aussi une ligne spéciale avec le mot *default* qui correspond (default gateway) à la route utilisée lorsque l'adresse de destination n'est pas sur le réseau local.

I-2 Le transfert de données

I-2.1 L’adressage IP

Dans la couche IP, chaque datagramme est acheminé vers une machine unique connectée sur un réseau local. L’adresse IP de destination, mot 5 de l’entête, permet de définir un couple (réseau;machine) et un seul. Cette adresse est codée sur 32 bits, où les trois premiers bits servent à identifier la classe de celle-ci. En effet, IP distingue 3 classes principales qui sont la Classe A, Classe B et Classe C.



Toutes les adresses disponibles ne sont pas utilisables. En effet, il existe certaines restrictions sur des familles d’adresses. Les deux adresses de classe A des réseaux 0 et 127 correspondent respectivement au *default route* (utilisé pour le routage IP) et *loopback address* (permet d’accéder à sa machine comme à tout autre machine du réseau). De plus, quelle que soit la classe du réseau, les adresses machines 0 et 255 sont réservées. L’adresse machine 0 représente l’ensemble du réseau local, et dans l’exemple ci-dessus 129.88.0.0 représente le réseau imag. Pour adresser un message à l’ensemble des machines sur un réseau, tous les bits du champ de l’adresse machine doivent être à un. Ainsi, sur imag, l’adresse 129.88.255.255 fait référence à la totalité des machines (*broadcast address*).

Dans le cas d’un routeur, par exemple entre les deux réseaux **imag** et **ensimag**, celui ci possédera une adresse IP différente sur chacun des réseaux locaux auxquels il est connecté.

i-1.6b. Transmission Control Protocol

A l'inverse de UDP, le protocole TCP fonctionne en mode connecté et s'assure que les données ont bien été transmises. En effet, après vérification du **Checksum**, si il s'avère que le segment est endommagé, le récepteur n'envoie pas d'acquittement de bonne réception. Ainsi, après un certain temps, l'émetteur réemet le segment sur le réseau.

Tableau 6 : Entête TCP

<table>
<tr><td colspan="2">0</td><td colspan="2">4</td><td colspan="2">8</td><td colspan="2">12</td><td colspan="2">16</td><td colspan="2">20</td><td colspan="2">24</td><td colspan="2">28 31</td></tr>
<tr><td colspan="8">Source Port</td><td colspan="8">Destination Port</td></tr>
<tr><td colspan="16">Numéro de séquence</td></tr>
<tr><td colspan="16">Numéro d'acquittement</td></tr>
<tr><td colspan="2">Offset</td><td colspan="3">Réservé</td><td colspan="3">Flags</td><td colspan="8">Fenêtre</td></tr>
<tr><td colspan="8">Checksum</td><td colspan="8">Urgent Pointer</td></tr>
<tr><td colspan="12">Options</td><td colspan="4">Padding</td></tr>
<tr><td colspan="16">Données</td></tr>
</table>

Comme TCP fonctionne en mode connecté, il établit une connecxion logique, bout à bout, entre les deux intervenants. Au départ, avant tout transfert de données, TCP demande l'ouverture d'une connecxion à la machine cible qui renvoie un acquittement signifiant son accord. De même, lorsque l'ensemble des données ont été échangées, TCP demande la fermeture de la connecxion et un acquittement de fermeture est alors envoyé sur le réseau. Lors du transfert, à chaque datagramme, un acquittement de bonne réception est émis par le destinataire. L'application qui va récupérer les données provenant de la couche TCP est identifiée grâce au numéro du port de destination (Destination Port) de l'entête.

I-1.7 La couche Application

Cette couche rassemble l'ensemble des applications qui utilisent TCP/IP pour échanger des données. On dénombre de plus en plus de services différents, les derniers comme *WAIS* ou *WWW* étant de plus en plus performants et souples d'utilisation. Les applications les plus courantes sont :

• **TELNET**	Network Terminal Protocol
• **FTP**	File Transfer Protocol
• **SMTP**	Simple Mail Transfer Protocol
• **DNS**	Domain Name Service
• **RIP**	Routing Information Protocol
• **NFS**	Network file system

Nous verrons une description plus complète de ces services dans la suite de cet ouvrage. Il faut savoir que la majorité des applications fonctionnent au-dessus de TCP ou UDP, il existe toutefois des services, comme Extension Gateway Protocol (EGP), qui utilisent directement la couche IP.

i-1.5d. Internet Control Message Protocol

L' I.C.M.P. fait partie intégrante de la couche IP, c'est elle qui se charge d'envoyer des messages sous la forme de datagramme qui ont pour fonction :

- Le contrôle du flux des données
- Avertir si un destinataire n'est pas atteignable
- Le réaiguillage d'un paquet sur un autre gateway
- Interroger l'état de fonctionnement d'une machine éloignée

I-1.6 La couche transport

La couche transport fait le relais entre la couche IP et les applications utilisant les ressources du réseau. On discerne deux protocoles différents :

- **Transmission Control Protocol**
  TCP fonctionne en mode connecté
  Effectue la détection et le contrôle des erreurs

- **User Datagram Protocol**
  UDP fonctionne en mode non connecté
  Pas de contrôle d'erreur

Le choix entre ces deux protocoles dépend du type d'application utilisée dans la couche supérieure.

i-1.6a. User Datagram Protocol

Le protocole UDP fonctionne en mode non connecté et ne possède pas de moyen de détecter si un datagramme est bien parvenu à son destinataire. UDP utilise simplement les champs *SOURCE PORT* et *DESTINATION PORT*, de l'entête, pour identifier les applications utilisées.

Tableau 5 : Entête UDP

0	16 31
Source Port	Destination Port
Longueur	Checksum
Données	

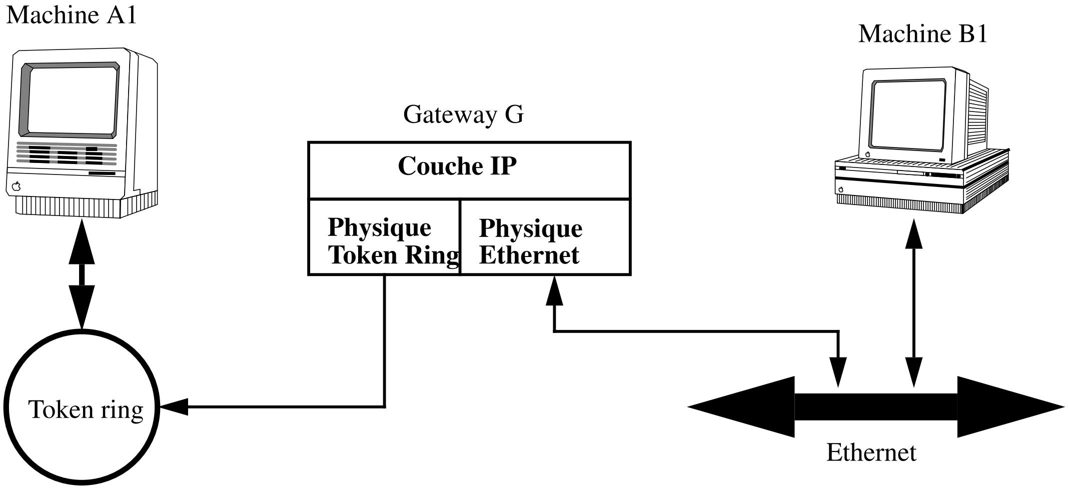
Le choix d'utiliser UDP comme protocole de la couche transport est justifié par plusieurs bonnes raisons. En effet, le fait d'utiliser une entête de taille très réduite procure un gain de place assez considérable. De plus, avec UDP on évite l'ensemble des opérations de connecxion, détection d'erreur et déconnecxion, dans ce cas le gain de temps peu être très appréciable, surtout pour de petits transferts.

Le **datagramme** est l'unité de base du transfert de données avec un protocole TCP/IP. Les premiers 5 ou 6 mots de 32 bits sont l'entête de ce datagramme. C'est elle qui donne l'ensemble des informations nécessaires au transfert du paquet. En fait, le nombre de mots constituant l'entête est variable car on donne la longueur de celle-ci dans le champ **IHL** (Internet Header Length). La couche IP fonctionne de telle manière que si l'adresse de destination ne se situe pas sur le même réseau local que l'adresse source, le datagramme est passé directement à un **routeur** (**gateway**). C'est ce que l'on appelle l'aiguillage (**routing**) d'un paquet. Cette action est réalisée individuellement pour chaque paquet à transmettre.

i-1.5a. Le routing

On appelle routeur une machine connectée sur deux réseaux locaux différents qui se charge de faire passer les données de l'un à l'autre. Ainsi, dans un routeur IP, le datagramme ne remonte jamais au delà de la couche 2. Par exemple, si une machine A1 sur un réseau A veuille atteindre une machine B1 sur un réseau différent, l'utilisation d'un routeur G est obligatoire.

ROUTEUR IP



i-1.5b. La fragmentation des données

La fragmentation d'un datagramme, par la couche IP, en segments intervient lors de son passage d'un réseau à un autre ne supportant pas la même taille de datagrammes (Taux maximum de transmission). Dans ce cas, c'est le deuxième mot de l'entête qui sert à identifier à quel datagramme appartient le segment.

- Identification : Donne le numéro du datagramme
- Offset : Donne le numéro du segment dans le datagramme
- Flags : Indique si il reste des segments à transmettre

i-1.5c. Passage aux couches supérieures

Lorsque la couche IP reçoit un datagramme, elle identifie la couche supérieure visée grâce au champ PROTOCOLE de l'entête.

Tout comme dans le modèle OSI, les données sont transférées verticalement d'une couche à un autre en y rajoutant une entête (header). Cette entête permet de rajouter des informations identifiant le type de données, le service demandé, le destinataire, l' adresse source etc...

Tableau 3 : Encapsulation des données

Couche Application				Données
Couche Transport			Entête	Données
Couche IP		Entête	Entête	Données
Couche Physique	Entête	Entête	Entête	Données

I-1.4 La couche Physique

Cette couche a pour fonction l'encapsulation des datagrammes provenant de la couche IP et la traduction des adresses en adresses physiques utilisées sur le réseau. Il y a donc autant de versions de la couche physique qu'il y a de type de moyen de transport des données. Ainsi, par exemple, la couche physique est différente suivant que l'on est sur un réseau X25 ou FDDI ou bien même TOKEN RING.

I-1.5 La couche IP (Internet Protocol)

La couche IP se situe directement au dessus de la couche physique chargée de placer les données sur le médium. IP est un protocole qui n'est pas connecté, donc il n'y a pas d'établissement de connecxion et de vérification de la validité des datagrammes. Ses principales fonctions sont :

- définir des datagrammes (unité de base de la transmission TCP/IP)
- aiguiller les datagrammes jusqu'à leur adresse de destination
- transférer les données entre la couche physique et la couche transport
- fragmenter et réassembler les datagrammes

Tableau 4 : Datagramme IP

BITS

<table>
<tr><td>0</td><td>4</td><td>8</td><td>12</td><td>16</td><td>20</td><td>24</td><td>28 31</td></tr>
<tr><td>Version</td><td>I.H.L.</td><td colspan="2">Type de service</td><td colspan="4">Longueur totale</td></tr>
<tr><td colspan="4">Identification</td><td>flags</td><td colspan="3">Offset de fragmentation</td></tr>
<tr><td colspan="2">Durée de vie</td><td colspan="2">Protocole</td><td colspan="4">Checksum de l'entête</td></tr>
<tr><td colspan="8">Adresse source</td></tr>
<tr><td colspan="8">Adresse destinataire</td></tr>
<tr><td colspan="6">Options</td><td colspan="2">Padding</td></tr>
<tr><td colspan="8">DONNEES</td></tr>
</table>

I-1.3 Architecture

Le modèle de référence pour l’échange de données informatiques est le modèle OSI (Open Systems Interconnect) adopté par l’ ISO (International Standards Organisation). Cette norme de communication repose sur l’empilement de 7 couches pouvant communiquer verticalement entre elles.

Tableau 1 : Le modèle de référence OSI

7 Couche Application Applications utilisant le réseau
6 Couche Présentation Formate les données en fonction de l’application
5 Couche Session Répartit les données suivant les applications
4 Couche Transport Détection et correcxion des erreurs
3 Couche Réseau S’occupe de la connecxion sur le réseau
2 Couche Liaison Transfert de données fiable sur le lien physique
1 Couche Physique Définie les caractéristiques physiques du média

Ce tableau représente l’empilement des sept couches du modèle OSI avec leurs noms et fonctions respectives. En Comparaison avec ce modèle, on peut ramener l’architecture de communication de données utilisant TCP/IP à un ensemble de quatre couches superposées.

Tableau 2 : Architecture utilisant le protocole TCP/IP

1 Couche Application Applications utilisées sur le réseau
2 Couche Transport Assure le transfert d’un site à un autre
3 Couche Internet Définie les datagrammes et leur routage
4 Couche Physique Ensemble de routines d’accès au média

I Protocole TCP/IP

I-1 Introduction à TCP/IP

I-1.1 Un peu d'histoire

C'est en 1969 que l'agence américaine **D.A.R.P.A.** (Defense Advanced Research Projects Agency) lança le projet de développer un réseau expérimental, à commutation de paquets : **ARPANET**. Ce réseau eut tellement de succès que la majeure partie des organisations qui y étaient rattachées débutèrent à l'utiliser quotidiennement. Ainsi en 1972 on pouvait assister à une démonstration d'**ARPANET** reliant 50 sites, utilisant 20 commutateurs, basé sur **NCP**, ancêtre de **TCP**. Cette même année commença le début des spécifications du protocole **TCP/IP** pour ARPANET. Dés 1980, **UNIX BSD 4.1** inclut TCP/IP comme protocole standard de communication, mais ce n'est qu'en 1983 que TCP remplaça officiellement NCP pour ARPANET. En même temps le nom d' **Internet** passa dans le langage courant pour désigner la totalité du réseau **ARPANET** et **MILNET** du **DDN** (Defence Data Network).

En 1990 le terme de ARPANET fut abandonné et céda la place à **Internet** qui représente de nos jours l'ensemble des réseaux internationaux reliés par le protocole TCP/IP. Le succès de ce réseau est tel que le nombre de machines connectées connaît actuellement une **croissance exponentielle**. Ainsi en 1981, seulement 213 machines étaient enregistrées sur Internet, en 1989 on en dénombrait 80 000. En octobre 1990 le chiffre de 313 000 était atteint et trois mois plus tard, en janvier 1991, le nombre de machines alors connectées dépassait les 376 000. Un an plus tard, au mois de janvier 1992, ce nombre avait presque doublé pour atteindre les 727 000 machines. En fait, au moment ou vous lirez ces lignes, plus de 1,5 millions de sites, dans plus de 45 pays, seront connectés entre eux sur un seul réseau : **INTERNET**.

I-1.2 Spécifités d'utilisation

Un tel succès auprès de l'ensemble des constructeurs et des utilisateurs ne peut pas n'être qu'un phénomène de mode dû au progrès technologique de ces dernières années. En fait, si le protocole de communication de données TCP/IP a émergé comme un standard pour plus de 90% des réseaux actuels c'est qu'il possède des atouts non négligeables.

- TCP/IP a été distribué gratuitement dés le départ, déjà en 1980 il était intégré à la version 4.1 de Unix BSD.
- TCP/IP est indépendant du réseau physique. Ainsi on peut très bien l'utiliser comme protocole de communication sur un réseau **ethernet**, **token ring** ou **X25** etc...
- TCP/IP utilise un système d'adressage simple qui permet de sélectionner un site parmi un réseau international.
- TCP/IP est distribué avec un ensemble d'applications standardisées qui donnent à l'utilisateur l'ensemble des fonctions de base nécessaires à l 'échange de données.

ADMINISTRATION RESEAU

Rapport de fin d'année :
spéciale informatique 1992-93
François Borderies,
Olivier Chatel,
Jean-christophe Denis,
Didier Reis.

Ce document est une introduction à l'administration réseau, destinée aussi bien au simple utilisateur qui désire en savoir plus sur le fonctionnement du réseau qu'à de futurs administrateurs réseaux.

Remerciements :

Nous tenons à remercier Monsieur Serge ROUVEYROL, pour avoir permis de travailler sur une présentation de l'administration réseau TCP/IP.

Nous remercions aussi l'ensemble des personnes ayant participées directement ou, indirectement à l'élaboration de cet ouvrage.