Introduction aux réseaux TCP/IP.

David TILLOY . Ingénieur Systèmes & Réseaux
Neuronnexion

Table des matières.

I. Rappels sur le modèle en couche.

1. Introduction.

Tous les applicatifs réseaux doivent pouvoir communiquer entre eux, quelque soit l'architecture ou la plate-forme utilisée. Pour cela, les opérations sur les réseaux ont été divisées en plusieurs phases de base, de manière à simplifier le portage des applicatifs sur toutes les plates-formes. C'est ce que l'on appelle le modèle en couche. Un standard a alors été créé, normalisé par l'Open Systems Interconnection Reference Model (modèle de référence d'interconnexion des systèmes ouverts) sous la référence OSI-RM, utilisant 7 couches distinctes.

L'architecture TCP/IP est similaire à ce modèle en couche, mais ne dispose que de 4 couches dans la plupart des cas.

7	Application
6	Présentation
5	Session
4	Transport
3	Réseau
2	Liaison de données
1	Physique

Fig. 1 - *Modèle de référence OSI*

Application
Transport
Internet
Réseau Internet physique

Fig. 2 - *Modèle TCP/IP (Internet)*

Les couches 5 à 7 du modèle OSI sont des couches dites d'application. Elles sont orientées application, et fournissent une interface entre une application et le réseau.

Les couches 1 à 4 sont des couches dites de liaison. Ce sont elles qui se chargeront du routage, afin de correctement acheminer les paquets d'un point à un autre.

Le modèle TCP/IP ne suis pas tout à fait l'architecture en couche du modèle OSI. Après expérimentation, on s'est aperçu qu'une carte réseau devait regrouper les couches 1 et 2 pour obtenir des performances correctes.

Toutefois, il existe quelques cas où les couches 1 et 2 sont différenciées dans le modèle TCP/IP. C'est le cas par exemple d'une connexion par modem, qui comporte donc une couche de liaison de données (PPP : Point to Point Protocol).

On peut aussi trouver parfois une couche de niveau présentation (6), c'est par exemple le cas du SSL (Secure Socket Layer).

Remarque : dans le modèle TCP/IP, la couche de transport utilise soit T.C.P (Transmission Control Protocol), soit U.D.P (User Datagram Protocol). Par contre, il n'existe qu'un seul protocole de niveau Réseau : I.P (Internet Protocol).

2. Protocoles et Services.

On appelle **protocole**, un dialogue connue par les deux parties, entre deux couches de même niveau. Une couche de niveau (n) ne sera capable de dialoguer qu'avec une autre couche de même niveau qu'elle.

On appelle **service** l'ensemble des fonctions que doit absolument remplir une couche, fournissant l'interface pour transmettre des données de la couche (n) à la couche (n+1).

3. Les standards de l'Internet.

En 1980, le D.A.R.P.A créé un groupe pour donner un ensemble de standards pour l'Internet : le I.C.C.B (Internet Configuration Control Board) et devient en 1983 l'I.A.B (Internet Activities Board), chargé de concevoir, mettre en œuvre et gérer l'Internet.

1986, l'I.A.B se décharge de la normalisation sur l'Internet Engineering Task Force (I.E.T.F), et la recherche à long terme est confiée à l'Internet Research Task Force (I.R.T.F). l'I.A.B donne son aval sur les projets de ces deux organismes.

1992, formation de l'Internet Society, l'I.A.B devient l'Internet Architecture Board.

Pour tous renseignements complémentaires, vous vous référeraient aux R.F.C (Request For Comment) : ftp://ftp.ibp.fr/pub/rfc/rfc/

4. Attribution des adresses IP.

Les adresses Internet (32 bits en Ipv4) identifient de manière **unique** une machine dans la toile du réseau. Elles sont délivrées par des organismes chargés de gérer le bon déploiement de ces adresses (pour l'Europe, il s'agit R.I.P.E (Réseau I.P Européen)).

Une adresse IP est composée de deux champs : l'adresse réseau et l'adresse machine. L'adresse réseau est placée sur les bits de poids forts, alors que l'adresse de machine est calculée sur les bits de poids faible.

Toutefois, dans les communications entre machines, un autre type d'adresse est parfois utilisé, il s'agit de l'adresse M.A.C (Media Access Control), en accord avec le protocole A.R.P (Address Resolution Protocol).

Il existe plusieurs classes d'adresses. On parle des classes A, B, C, D et E. Elles sont différenciées par les bits de poids forts qui les compose.

A	0000	Identifiant du réseau	Identifiant de la machine
B	1000	Identifiant du réseau	Identifiant de la machine
C	1100	Identifiant du réseau	Identifiant de la machine
D	1110	Identifiant du réseau	Identifiant de la machine
E	1111	Non utilisé	Non utilisé

Une adresse IP est toujours de la forme a.b.c.d.

Dans le cas d'une classe A, on peut librement fixez les valeurs b, c et d. On pourra donc adresser théoriquement 16 777 214 machines.

Une classe B fixe librement les valeurs de c et d. On pourra alors adresser 65 534 machines.

Une classe C fixe uniquement la valeur de d. On pourra donc adresser 254 machines.

La classe D est une classe quelque peu différente, puisqu'elle est réservée à une utilisation particulière : le multicast.

La classe E est quant à elle une classe non usitée à ce jour.

On dispose donc en théorie des plages d'adresses suivantes :

Classe	Plage	
A	0.0.0.0	127.255.255.255
B	128.0.0.0	191.255.255.255
C	192.0.0.0	223.255.255.255
D	224.0.0.0	239.255.255.255
E	240.0.0.0	247.255.255.255

Il existe quelques adresses dites non routables. Ces adresses sont réservées à un usage interne, ou dans le cas de réseaux privés. Elles ne sont en théorie jamais routées sur l'Internet. Il existe 3 classes d'adresses IP :

Classe A : 10.0.0.0
Classe B : 172.16.0.0 à 172.31.0.0
Classe C : 192.168.0.0 à 192.168.255.0

127.0.0.0 est aussi une classe A particulière, puisqu'elle ne sera jamais non plus routée sur un réseau. Elle est réservée pour un usage interne d'adresses IP. On l'appelle aussi interface loopback (interface de bouclage).

5. Quelques « services » présents sur l'Internet.

Il existe de multiples services utiles sur l'Internet, comme Telnet, F.T.P (File Tranfert Protocol), S.M.T.P (Simple Mail Transport Protocol), D.N.S (Domain Name Service), R.P.C (Remote Process Control)...

Tous ces services sont sur la couche haute du modèle OSI. Ils sont soient au-dessus d'une couche de transport TCP ou UDP, mais tous au dessus de la couche réseau IP.

Telnet : il sert à se connecter sur une machine à distance, et de pourvoir y travailler de la même manière que devant la machine même.

F.T.P : Il permet de transférer des données d'une machine à une autre. Ce service sert aussi à diffuser des données sans authentification (FTP anonyme).

S.M.T.P : Il permet d'échanger du courrier entre deux serveurs de courrier. Ce protocole est totalement transparent pour l'utilisateur, les serveurs se chargeant entre eux de correctement transférer les données.

D.N.S : C'est le service de nom de domaine. Il permet de convertir un nom de machine en adresse réseau (ou I.P), et vice versa.

II. Protocoles de haut niveau.

1. Le protocole Telnet.

Telnet (Telecommunications Network) permet à une machine client se connecter sur un serveur, et ceux, quelles que soient leurs localisations dans le monde, du moment que ces deux machines sont raccordées à l'Internet.

Une machine disposant d'un serveur telnet permettra donc à n'importe quelle machine de part le réseau de s'y connecter, au moyen d'un client telnet. Les clients telnet existent sur la quasi-totalité des plates-formes (Windows, Unix, MacOS, BeOS…).

Lorsqu'une machine A désire se connecter à une machine B, A doit disposer d'un client telnet, et B d'un serveur telnet. Dans le monde Unix, le serveur telnet est in.telnetd et le client est telnet. Toutes les architectures ne sont pas dotées en standard d'un serveur telnet (c'est le cas de Windows NT, de MacOS…). D'une manière plus générale, un serveur est souvent le nom du service proposé suivi de la lettre 'd'.

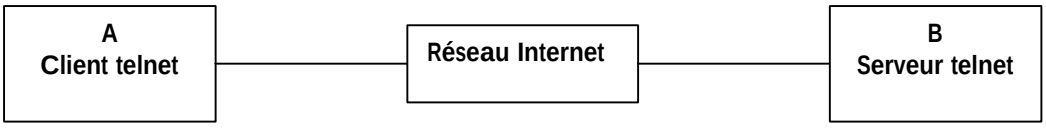

Fig. 3 – *Connexion telnet de A vers B***.**

Lorsque A envoie une requête à la machine B, celle-ci est à l'écoute permanente des requêtes présentées sur le port TCP (23 par défaut). B répond alors par une demande d'authentification, à laquelle A doit répondre (login + password). Lorsque cette phase est réussie, l'entrée standard est redirigée sur le clavier de A, et la sortie standard est redirigée sur l'écran de A. Tout se passe comme ci l'utilisateur de A était devant la machine B, alors que des milliers de kilomètres peuvent les séparer.

Le protocole Telnet est basé sur le protocole de plus bas niveau TCP. C'est donc la couche TCP qui se charge d'assurer l'intégrité de l'échange des données. Telnet est un protocole de haut niveau, qui ignore donc tout de l'architecture utilisée, soit sur la machine, soit sur la topologie du réseau. Il ne fait que renvoyer un écho des données envoyées ou reçues.

```
[dav@dav ~/dav/trav/cours/iut (5)]$ telnet localhost
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Red Hat Linux release 4.2
Kernel 2.0.30 on an i586
login: dav
Password:
Last login: Sun Sep 28 20:27:49 on tty1
[dav@dav ~/ (1)]$ whoami
dav
[dav@dav ~/ (2)]$ logout
Connection closed by foreign host.
[dav@dav ~/dav/trav/cours/iut (6)]$
```

Fig. 4 – *Exemple de connexion telnet.*

2. Le protocole F.T.P.

FTP est utile dès qu'il s'agit de transférer des données entre deux machines A et B. Comme en telnet, la machine A doit être équipée d'un client ftp, alors que la machine B est elle équipée d'un serveur FTP.

FTP utilise un langage de commande définit par des mots clefs de 4 caractères. Ce sont les commandes FTP internes. Mais il dispose aussi de commandes utilisateur. Les commandes internes servent à établir et à maintenir la connexion FTP, alors que les commandes utilisateur permettent d'effectuer des opérations à l'aide de cette connexion.

Lorsque A envoie une demande de connexion à B, le serveur FTP renvoie alors le message de login définit par l'administrateur de B. L'utilisateur de A envoie alors la commande **USER login** attendue par B, où login est le nom de l'utilisateur. B attend alors la commande **PASS password**, où password est le mot de passe correspondant à l'utilisateur **login**.

De nombreux serveurs autorisent les connexions dites anonymes, c'est à dire que n'importe quel utilisateur peut s'y connecter pour prendre des fichiers, et sur certains serveurs déposer des fichiers. Dans une phase de connexion anonyme, on envoie généralement **ftp** ou **anonymous** comme nom de login, et son adresse email en mot de passe. Une fois l'identification effectuée, le client envoie la commande **SYST**, de manière à connaître le système distant.

A cette étape, vous avez ouvert un canal de commande sur le port TCP 21 (par défaut), mais vous n'êtes pas prêt à transférer des données. Les données ne sont pas présentées sur le même port TCP que les commandes. Vous devez donc ouvrir un autre port TCP de manière à transférer vos données.

Le protocole TCP utilise par convention le port TCP/21 pour les commandes, et le port TCP/20 pour les données. Le port TCP/21 est appelé l'interpréteur de protocole (Protocol Interpreter ou PI), alors que le port TCP/20 est appelé processus de transfert de données (data transfert process ou DTP).

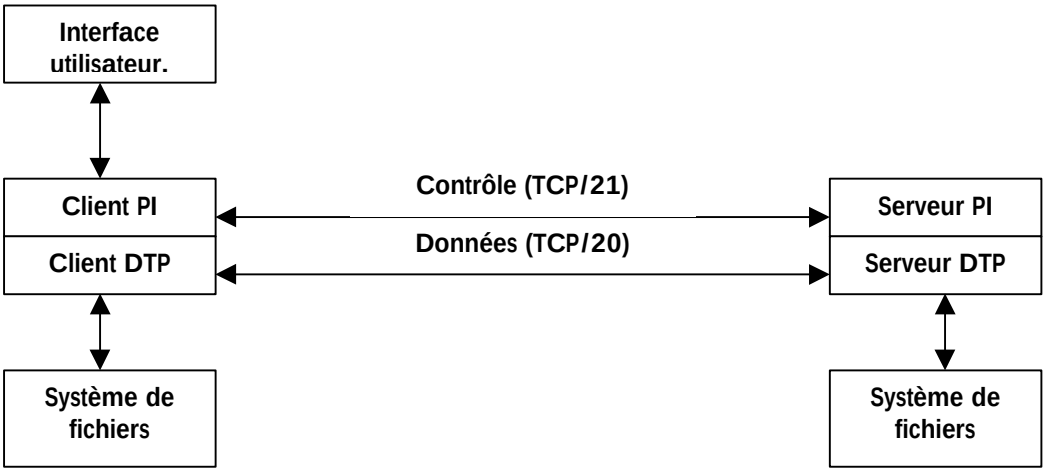

Fig. 5 – *Connexion FTP de A vers B*.

Une fois le canal de données ouvert, vous pourrez alors transférer des données entre les machines A et B.

Remarques :

(i) A l'heure actuelle, le protocole FTP ne conserve pas les droits d'accès sur les fichiers transférés. Il vous appartient de maintenir ces droits si besoin.

(ii) FTP utilise deux modes de transfert, ascii ou binaire. Pensez à vous placer dans le bon mode avant d'entamer un transfert.

(iii) FTP effectue tous ses transferts en avant-plan, donc vous devez attendre que les transferts soient achevés avant d'entamer un autre transfert.

(iv) FTP, comme telnet, repose sur le protocole TCP, c'est donc la couche réseau qui sera chargée de gérer l'intégrité des données.

```
[dav@dav ~/dav/trav/cours/iut (7)]$ ftp localhost
Connected to localhost.
220 dav.neuronnexion.fr FTP server (Version wu-2.4.2-academ[BETA-14](1) Thu
Sep 11 00:49:43 MET DST 1997) ready.
Name (localhost:dav): ftp
331 Guest login ok, send your complete e-mail address as password.
Password:

230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 20068
drwxr-xr-x   7 root     root         1024 Sep 25 19:40 .
drwxr-xr-x   7 root     root         1024 Sep 25 19:40 ..
d--x--x--x   2 root     root         1024 Aug 15 21:12 bin
drwxr-xr-x   2 root     root         1024 Sep 25 19:40 bsd
-rwxr-xr-x   1 root     root     20007844 Sep 25 19:41 bsd.tgz
d--x--x--x   2 root     root         1024 Aug 15 21:12 etc
drwxr-xr-x   2 root     root         1024 Aug 15 21:12 lib
dr-xr-sr-x   2 root     ftp          1024 Dec 24  1996 pub
-rw-r--r--   1 root     root       451430 Sep 10 21:47 vmlinuz
226 Transfer complete.
ftp> bye
221 Goodbye.
[dav@dav ~/dav/trav/cours/iut (8)]$
```

Fig. 6 – *Exemple de connexion FTP.*

```
[dav@dav ~/dav/trav/cours/iut (9)]$ ftp -d localhost
Connected to localhost.
220 dav.neuronnexion.fr FTP server (Version wu-2.4.2-academ[BETA-14](1) Thu
Sep 11 00:49:43 MET DST 1997) ready.
Name (localhost:dav): ftp
---> USER ftp
331 Guest login ok, send your complete e-mail address as password.
Password:

---> PASS XXXX
230 Guest login ok, access restrictions apply.
---> SYST
215 UNIX Type: L8
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
ftp: setsockopt (ignored): Operation not permitted
---> PORT 127,0,0,1,4,106
200 PORT command successful.
---> LIST
150 Opening ASCII mode data connection for /bin/ls.
total 20068
drwxr-xr-x   7 root     root         1024 Sep 25 19:40 .
drwxr-xr-x   7 root     root         1024 Sep 25 19:40 ..
d--x--x--x   2 root     root         1024 Aug 15 21:12 bin
drwxr-xr-x   2 root     root         1024 Sep 25 19:40 bsd
-rwxr-xr-x   1 root     root     20007844 Sep 25 19:41 bsd.tgz
d--x--x--x   2 root     root         1024 Aug 15 21:12 etc
drwxr-xr-x   2 root     root         1024 Aug 15 21:12 lib
```

```
dr-xr-sr-x   2 root     ftp          1024 Dec 24  1996 pub
-rw-r--r--   1 root     root       451430 Sep 10 21:47 vmlinuz
226 Transfer complete.
ftp> bye
---> QUIT
221 Goodbye.
[dav@dav ~/dav/trav/cours/iut (10)]$
```

Fig. 7 – *Exemple de connexion FTP en mode debug.*

3. Le protocole S.M.T.P.

Le protocole SMTP (Simple Mail Transfert Protocol) est certainement un des protocoles le plus utilisé sur l'Internet. Il est totalement transparent à l'utilisateur, ce qui le rend convivial, et dispose de clients et de serveurs sur la majorité des architectures.

Son but est de permettre le transfert des courriers électroniques. Il est similaire au protocole FTP, de part son langage de commande. Il est généralement implémenté sur le port TCP/25. Sur système Unix, sendmail est généralement utilisé, et se comporte comme client et comme serveur.

SMTP utilise des files d'attente pour gérer les transferts de courrier. Lorsqu'un message est envoyé au serveur SMTP, celui-ci le place dans une file d'attente, puis tente de le livrer à la machine de destination. Si cette machine n'est pas accessible, le serveur SMTP tentera selon la configuration de le transmettre ultérieurement. Le serveur SMTP, contrairement au serveur telnet ou FTP, est généralement présent sur le système en tant que démon, et n'est donc pas lancé à la demande.

Tous les messages sont transférés dans un format ascii, donc avec un codage sur 7 bits. La fin d'un message est indiquée par un '.' sur une ligne vierge.

Dans la phase d'échange de courrier entre deux serveurs SMTP, la première phase est l'authentification de la machine émettrice.

La machine qui demande la connexion envoie la commande HELO suivi de son nom de domaine. La machine réceptrice renvoie alors un message de bienvenue, et présente les commandes disponibles. La machine émettrice va maintenant donner le nom de l'expéditeur, par la commande MAIL FROM: login. Ensuite, l'émetteur indique à qui s'adresse ce courrier, par la commande RCPT TO: login. A ce niveau, les machines sont prêtes à échanger les messages. La machine émettrice envoie alors la commande DATA, puis termine cette phase de transfert du message en envoyant un point sur une ligne vierge. La connexion reste alors établie, et les deux machines peuvent continuer à transférer des courriers, ou retourner leur mode de connexion (celle qui émettait devient réceptrice, et celle qui recevait devient émettrice).

Si plusieurs destinataires sont spécifiées dans le champ RCPT, le message est alors envoyé à tous les destinataires, mais il n'est transféré qu'une fois entre les deux serveurs.

```
[dav@dav ~/dav/trav/cours/iut (11)]$ telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 dav.neuronnexion.fr ESMTP Sendmail 8.8.4/8.8.4; Sun, 28 Sep 1997
20:41:35 +0200
helo neuronnexion.fr
250-dav.neuronnexion.fr Hello localhost [127.0.0.1], pleased to meet you
250-EXPN
250-VERB
250-8BITMIME
250-SIZE
250-DSN
250-ONEX
250-ETRN
250-XUSR
250 HELP
```

```
mail from: dav@dav.neuronnexion.fr
250 dav@dav.neuronnexion.fr... Sender ok
rcpt to: dav@dav.neuronnexion.fr
250 dav@dav.neuronnexion.fr... Recipient ok
data
354 Enter mail, end with "." on a line by itself
Un exemple de message sans client, directement, sur le port SMTP 25/TCP
.
250 UAA00614 Message accepted for delivery
quit
221 dav.neuronnexion.fr closing connection
Connection closed by foreign host.
[dav@dav ~/dav/trav/cours/iut (12)]$
```

Fig. 8 – *Exemple de dialogue SMTP.*

Lorsque le message a été correctement acheminée sur la machine de destination, celui-ci est ensuite placé en attente dans la boîte aux lettres de l'utilisateur. Sur la plupart des configurations Unix, les messages sont placés dans le répertoire /var/spool/mail/. Chaque utilisateur y a donc un fichier sous son nom de login, qui contient les messages. Chaque message possède une entête qui renseigne la provenance du message, l'heure à laquelle il est arrivé, etc.

```
From dav@dav.neuronnexion.fr  Sun Sep 28 20:42:29 1997
Return-Path: <dav@dav.neuronnexion.fr>
Received: from neuronnexion.fr (localhost [127.0.0.1])
          by dav.neuronnexion.fr (8.8.4/8.8.4) with ESMTP
        id UAA00614 for dav@dav.neuronnexion.fr; Sun, 28 Sep 1997 20:41:58
+0200
Date: Sun, 28 Sep 1997 20:41:58 +0200
From: David TILLOY <dav@dav.neuronnexion.fr>
Message-Id: <199709281841.UAA00614@dav.neuronnexion.fr>

Un exemple de message sans client, directement sur le port SMTP 25/TCP
```

Fig. 9 – *Exemple d'un message dans une boîte aux lettres.*

4. Le D.N.S

Lorsque vous vous adressez à une machine de l'Internet, il est souvent plus pratique de mémoriser un nom symbolique plutôt que son adresse IP. Toutefois, vous pouvez utiliser indifféremment l'une ou l'autre des deux formes d'adresse. Ceci est possible grâce au DNS (Domain Name Service), qui est chargé de convertir si besoin les adresses IP en noms symboliques ou les noms symboliques en adresses IP.

A l'heure actuelle, nous utilisons le protocole IP version 4, qui permet d'adresser 2^{32} machines (codage des adresses sur 4 octets), mais vu la forte croissance que connaît le réseau Internet, il est maintenant nécessaire d'étendre cet espace d'adressage. C'est le but de la prochaine version de l'IP, IP version 6, qui permettra un codage des adresses IP sur 128 bits (16 octets), et donnera ainsi la possibilité d'adresser 2^{128} machines. A l'heure actuelle, l'IPv6 est encore en phase de test, et son déploiement sur les réseaux n'est prévu que pour 2003.

Lorsque vous recherchez l'adresse IP ou le nom associé à une adresse IP d'une machine du réseau, vous émettez une requête à votre serveur DNS, dont l'adresse vous a été fournit par votre administrateur réseau. C'est ensuite lui qui s'occupera de demander à qui de droit de résoudre l'adresse ou le nom fournit, puis vous retournera l'information. Chaque serveur D.N.S gère une plage d'adresses IP. C'est ce que l'on appelle une zone, et le D.N.S qui contrôle cette zone est appelée primaire de la zone.

Prenons par exemple le domaine nommé « internet.fr », et donnons lui la classe C 127.0.0.0 et le netmask de 255.255.255.0. Cet organisme dispose donc de 256 adresses IP, dont 2 réservées (0 et 255). C'est lui qui va décider de l'organisation de cette plage d'adresse. Dès lors, une machine est donc installée et désignée comme serveur DNS. C'est sur cette machine que toutes les informations adresses / noms symboliques seront entrées.

Donc, dans notre exemple, nous décidons d'installer trois machines, « a », « b » et « c », plus un serveur dns que nous appellerons « ns1 ». L'administrateur de ce site a choisi de répartir ainsi ses adresses :

ns1 127.0.0.1
a 127.0.0.10
b 127.0.0.11
c 127.0.0.200

Remarque : On appelle « reverse » l'opération qui consiste à obtenir un nom symbolique en fonction de l'adresse IP.

Il va donc rentrer ces informations dans son serveur DNS. Il va aussi les adresses des « ROOT SERVERS ». Les ROOT SERVERS sont quelques machines réparties dans le monde qui maintiennent et s'échange quotidiennement des bases de données référençant chaque couple (plage d'adresses / serveur DNS).

Imaginons maintenant que vous faites une requête à ce serveur. Si la requête concerne votre plage d'adresses (par exemple, la machine « a » demande l'adresse de la machine « b ») alors votre serveur DNS répond de manière autonome. Par contre, si jamais votre requête est en dehors de votre domaine, le serveur DNS va demander à un ROOT SERVER (le premier dans la liste qui lui a été donnée) à quelle adresse il doit demander cette information. Si le premier ROOT SERVER ne répond pas, il demande alors au suivant, et ce jusqu'à ce qu'un serveur réponde ou que la liste des ROOT SERVER soit épuisée. Le ROOT SERVER va donc retourner une adresse de serveur DNS ayant autorité sur la zone demandée (on appelle la zone indifféremment la plage d'adresses ou le nom de domaine). Lorsque le DNS sait à qui demander l'information, il va alors contacter ce serveur pour lui poser la question. Le serveur DNS alors contacter va alors renvoyer sa réponse.

Les noms de domaines sont normalisés, on observe donc une certaine hiérarchie dans l'attribution des noms de domaines.

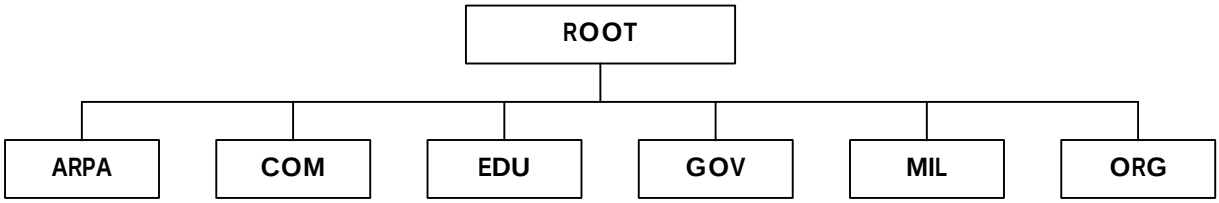

Fig. 10 – *Hiérarchie des noms de domaine T.L.D (Top Level Domain)*

Il faut rajouter à cet arbre les noms de domaines propres à chaque pays (.fr pour la France, .be pour la Belgique, .ar pour l'Argentine, etc.).

Chaque domaine devra donc être préalablement enregistré auprès de l'organisme de gestion officiel (l'INTERNIC pour les domaines ARPA, COM, EDU, GOV, MIL et ORG, le NIC France pour les .fr, etc.).

Chacun de ses domaines de premier niveau a une signification :

.ARPA organismes spécifiques à l'Internet (obsolète),
.COM Entreprises à but commercial,
.EDU Organismes d'enseignements,
.GOV Organismes gouvernementaux,
.MIL Entités militaires,
.ORG Organisations à but non lucratif.

Remarque : la France a elle aussi adoptée une sous hiérarchisation de ses noms de domaines. On dispose maintenant des domaines de premier niveau suivants : .gouv.fr, .asso.fr, .barreau.fr, .cci.fr, .ac.fr, tm.fr…

Aux U.S.A, un nouveau type de domaines de premier niveau sera prochainement lancé (.stor, .firm, .chop, .web…), qui ne seront plus gérés par l'Internic, mais par des organismes indépendants.

Chaque zone dispose donc d'un serveur de nom, mais la plupart dispose d'un second serveur de nom (serveur de nom secondaire), qui contient exactement les mêmes informations que le serveur de nom primaire. Cela assure des réponses aux requêtes, même si le serveur de nom primaire est indisponible. Deux méthodes sont possibles : soit le second serveur dispose exactement de la même configuration que le serveur primaire (cas le plus sûr, mais aussi le plus lourd à gérer dans le cas de changements fréquents), soit on opère un transfert de zone, c'est à dire que le serveur secondaire reçoit à des intervalles de temps configurés une copie complète de la zone contrôlée par le serveur primaire. Lorsque ce laps de temps est achevé, si le serveur primaire ne répond pas aux demandes de transferts du serveur secondaire, celui-ci maintient les données qu'il possède et n'opère pas de mise à jour de la base contenue. Dès que le serveur primaire sera à nouveau disponible, il détruira alors la copie de la base conservée jusqu'ici, et obtiendra une nouvelle copie plus récente du serveur primaire. Ainsi, quel que soit l'état du serveur primaire, les requêtes sont correctement achevées par le serveur secondaire.

Remarques :

(i) **Lorsque vous émettez votre requête, celle-ci est envoyée aux serveurs par le protocole UDP, et les réponses vous sont retournées elles aussi par protocole UDP. Lors de transfert de zone (dans le cas de serveurs DNS primaire et secondaire), le protocole utilisé est TCP, afin d'assurer l'intégrité des données qui est importante dans ce cas. Les requêtes comportant généralement peu de données, UDP convient bien à la rapidité de réponse demandée, et donne une probabilité d'erreur relativement faible.**

(ii) **Lorsque vous demandez une résolution à un ROOT SERVER qui est justement l'adresse du serveur de nom, celui ne vous répondra jamais directement, mais vous communiquera l'adresse du serveur ayant autorité sur la zone. Cette technique a été préférée pour des raisons de sécurité informatique (Cf. IP Spoofing).**

(iii) **Lorsque votre requête a été correctement remplie par votre resolveur, celui place la demande dans un cache, ce qui permettra la fois suivante de ne plus refaire la même requête, mais de répondre directement. Ce cache est vidé au bout d'un temps fixé (TTL : Time To Live).**

270	15.375	alpha	fgw	DNS	0x3403:Std Qry for 5.23.23.10.in-addr.arpa. of type Dom. name ptr on class	10.23.23.36	10.23.23.23
271	15.380	fgw	alpha	DNS	0x3403:Std Qry Resp. for 5.23.23.10.in-addr.arpa. of type Dom. name ptr on	10.23.23.23	10.23.23.36
291	15.655	alpha	fgw	DNS	0x7F61:Std Qry for zapan.nnn.fr. of type Host Addr on class INET addr.	10.23.23.36	10.23.23.23
292	15.659	fgw	alpha	DNS	0x7F61:Std Qry Resp. for zapan.nnn.fr. of type Host Addr on class INET add	10.23.23.23	10.23.23.36
496	27.292	fgw	alpha	ARP RARP	ARP: Request, Target IP: 10.23.23.36		

Fig. 11 – *Trames d'un réseau local avec requêtes DNS.*

Dans l'exemple de la Fig. 11, on a « sniffé » les trames d'un réseau ethernet, et on a isolé les requêtes DNS.

Paquet 270 : la machine alpha a demandé la résolution de l'adresse 10.23.23.5 (demandé en mode reverse). Cette demande a été faite au resolver (la machine fgw).

Paquet 271 : la machine fgw a résolu l'adresse demandée, et renvoie alors la réponse.

Paquet 291 : la machine alpha demande une nouvelle résolution, et demande quelle est l'IP associée à la machine zapan.

Paquet 292 : la machine fgw répond alors en renvoyant l'adresse IP de la machine zapan.

III. Adressage entre machines.

Dans un réseau TCP/IP, chaque machine est configurée avec une adresse IP. Une configuration TCP/IP comprend aussi une adresse de sous-réseau, un masque de sous-réseau, ainsi qu'une adresse de passerelle.

L'adresse IP est codée sur 2^{32} bits, soit 4 octets. Chaque adresse est unique dans le réseau Internet, ce qui est une première étape nécessaire pour le bon adressage des machines connectées au réseau. Le réseau est en fait composé d'une multitude de sous-réseaux. Même si chaque machine d'un sous-réseau est connectée au réseau mondial, seule une machine dispose de la liaison physique la reliant au sous-réseau concerné ainsi qu'au reste de l'Internet.

Remarque : Les organismes de normalisation ont défini l'écriture d'une adresse IP sous trois formes :

192.168.1.16 (base 10),
0300.0250.01.020 (base 8, préfixe '0'),
0xc0a80110 (base 16, préfixe '0x').

1. Informations nécessaires pour une configuration TCP/IP.

Pour configurer une machine en TCP/IP, on fournit plusieurs informations :
Son adresse IP, son masque de réseau et son adresse de passerelle. Ces informations vont servir à correctement orienter les dialogues avec les autres machines. Certaines adresses IP sont réservées à un usage particulier :

- **La première adresse d'un sous-réseau (ayant donc tous les bits à 0, sauf ceux identifiant le sous-réseau) est appelée adresse de sous-réseau, et est réservée. Elle ne pourra en aucun cas être utilisée.**
- **La dernière adresse d'un sous-réseau (ayant donc tous les bits à 1, sauf ceux identifiant le sous-réseau) est appelée adresses de broadcast, ou adresse de diffusion. Elle ne pourra en aucun cas être utilisée.**

1.2 Détermination du netmask.

Imaginons un sous-réseau de classe C 192.168.16.x (0 < x < 255). La première chose à faire est de déterminer le masque de sous-réseau (netmask). Le netmask permet de définir le réseau dans lequel vous vous trouvez. Dans le cas du réseau 192.168.16.x, nous avons donc un réseau de 255 machines.
On écrit donc n=255 (nombre de machines),
Netmask = NON (n)
Donc, ici, le netmask est de NON(0.0.0.255), soit (255.255.255.0).

Le netmask est donc utile pour connaître le nombre de machine présente dans le même sous-réseau que le nôtre. Cette option est intéressante, car elle va permettre de diviser une classe C en plusieurs sous-réseaux, par exemple :

Adresse de sous-réseau	Netmask	Nb d'adresses dans ce sous-réseau.
192.168.16.0	255.255.255.128	128
192.168.16.128	255.255.255.192	64
192.168.16.192	255.255.255.224	32
192.168.16.224	255.255.255.224	32

1.3 Calcul de l'adresse de diffusion.

L'adresse de diffusion, ou adresse de broadcast, est calculée à partir du netmask et de l'adresse du sous-réseau. Si R est l'adresse du sous-réseau, et N le netmask associé, on connaît donc B, l'adresse de broadcast par la formule suivante :

B = NON (N) ET (R)

Exemple :
Netmask N = 255.255.255.128
Sous-réseau R = 192.168.16.0

B = NON (255.255.255.128) ET 192.168.16.0
= (0.0.0.127) OU 192.168.16.0
B = 192.168.16.127

L'adresse de broadcast permet d'adresser toutes les machines du même réseau que le vôtre d'une seule opération. Imaginons que vous souhaitiez envoyer la commande ping à toutes les machines appartenant au même réseau que vous, vous pourrez alors entrer : ping 192.168.16.127 et toutes les machines de ce réseau vous répondront.

Il est maintenant clair qu'une adresse de sous-réseau ou de broadcast ne peut pas être attribuée à une machine, du fait de son usage quelque peu particulier.

1.4 L'adresse de passerelle.

L'adresse de passerelle indique si nécessaire à quelle machine doit-on s'adresser lorsqu'une requête n'est pas destinée à une machine de notre réseau. La passerelle est chargée de correctement transmettre les paquets de notre réseau aux autres passerelles des autres réseaux, mais elle doit aussi nous transmettre les paquets des autres réseaux à destination de notre réseau.

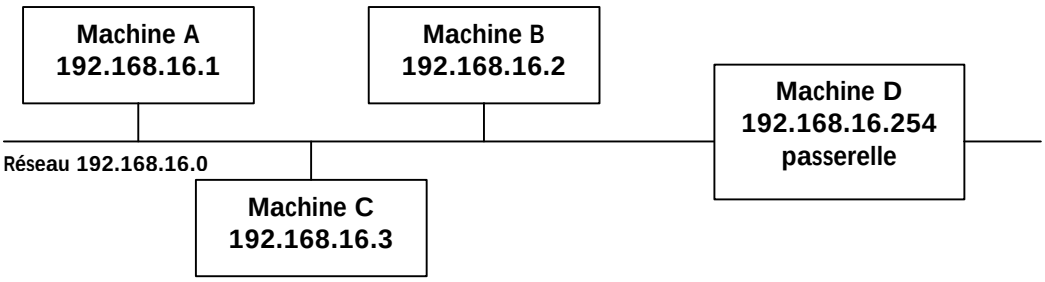

Fig. 12 – *Réseau local interconnecté au réseau mondial.*

Dans la Fig. 12, on a trois machines A, B et C, respectivement d'adresse IP 192.168.16.1, 192.168.16.2 et 192.168.16.3. La quatrième machine D est généralement un routeur, d'adresse IP 192.168.16.254. Chacune de ces machines appartient à la classe C 192.16.168.0, qui permet donc à l'administrateur de ce site d'utiliser selon sa volonté les adresses de 0 à 255. Le netmask de ce réseau est donc 255.255.255.0.

Un routeur est un ordinateur ne disposant pas d'écran ni de clavier, mais de deux interfaces réseau. Par exemple, on peut envisager le routeur comme un PC muni du logiciel adéquat et de deux cartes réseau, configurées chacune sur des sous-réseaux différents.

Lorsqu'une requête est émise, si l'adresse de destination ne concerne pas le réseau dans lequel l'émetteur se trouve, alors le paquet sera dirigé vers le routeur pour être routé sur un autre tronçon du réseau. Si par contre la machine de destination est dans le même réseau que la machine émettrice, la passerelle n'est pas utilisée, car elle serait inutile.

Pour déterminer si la machine émettrice se trouve dans le même réseau que la machine qu'elle souhaite contacter, l'opération suivante est effectuée : V = IP_{dest} ET Netmask. Si le résultat V de cette opération donne la même adresse de sous-réseau que la machine émettrice, alors la machine de destination se trouve dans le même réseau que la machine émettrice, et la passerelle n'est pas utilisée. Elle l'est dans tous les cas contraires.

Remarque : Certains réseaux non connectés (réseaux locaux) ne disposent pas de passerelle. Lors de la configuration, on omet alors l'adresse de passerelle.

2. Résolution d'adresses logique / physique : A.R.P.

Tous les dialogues entre machines sont faits par l'intermédiaire d'interfaces réseau. Dans la plupart des cas, il s'agit de cartes réseau branchées sur des PC.

Lorsqu'un paquet transite sur le réseau, la carte doit alors déterminer si ce paquet s'adresse à elle ou non. Or, l'adresse IP se trouve codée dans la couche de niveau 3 du modèle OSI. Si chaque interface devait remonter jusqu'au niveau 3 pour déterminer si ce paquet s'adresse à elle ou non, alors les débits seraient extrêmement faibles, et demanderait beaucoup de ressources aux machines. Pour palier à ce problème, on utilise l'adresse MAC (Media Access Control), qui est connue des couches de niveau 1, puisque ce sont des adresses physiques. Elles sont codées sur 6 octets (dans le cas de l'Ethernet), de la forme xx:xx:xx:yy:yy:yy, et sont généralement exprimées en hexadécimal. Les adresses M.A.C peuvent être localement administrées (l'adresse est arbitraire et est fixé par l'administrateur du site) ou plus généralement universellement administrées (les adresses sont fixées à la sortie d'usine des cartes). Dans les cas d'adresses M.A.C universellement administrées, les trois premiers octets identifient le constructeur de la carte (Ex : **00:0a:24**:yy:yy:yy indique une interface réseau de marque 3COM).

La première étape dans le dialogue entre deux machines est donc la création du couple (Adresse IP, Adresse MAC). Pour cela, on utilise le protocole A.R.P (Address Resolution Protocol).

ARP commence par émettre un paquet sur toutes machines en utilisant l'adresse de broadcast. Ce paquet contient l'adresse IP de la machine émettrice, son adresse IP, ainsi que l'adresse IP de la machine qu'il cherche à contacter. Lorsque les interfaces réseaux identifient une requête ARP, elles regardent toutes si l'IP recherchée est la leur ou non. Seule la machine concernée par la requête ARP saura donc répondre, et renverra à la machine demandeuse une réponse ARP complète, puisqu'elle vient d'y ajouter son adresse MAC.

Les deux machines (émettrice et réceptrice) placent dans un buffer le couple (Adresse IP, Adresse MAC) qu'elles viennent de découvrir. Ces données sont conservées un certain temps, ce qui permet de ne plus réémettre la même requête ARP si jamais un nouveau dialogue est initié entre ces deux même machines.

Lorsque la machine demandée ne fait pas partie du même réseau que la machine demandeuse, la requête ARP est émise avec l'adresse IP de la passerelle, qui se chargera elle de router correctement les paquets jusqu'à leurs destinations.

3. Routage.

Le routage est primordial pour l'interconnexion des réseaux. Le réseau Internet est en fait composé d'une multitude de petits réseaux connectés entre eux. Chaque réseau envoie et reçoit ses informations par le biais de passerelles (voir points précédents).

Chaque réseau connecté comprend au minimum une passerelle. Chaque passerelle est obligatoirement connectée à une autre passerelle, appartenant à un autre réseau. Les passerelles sont généralement des routeurs, appareils dédiés au routage de paquets.

Un routeur est donc nécessaire pour relier deux réseaux entre eux, car il n'est pas concevable de relier tous les réseaux par liaison ethernet classique. En effet, si on prend le cas de l'IUT d'Amiens, il est totalement inconcevable de relier l'IUT situé au campus à la faculté de Sciences située en centre ville par une liaison ethernet. Déjà parce que les distances limites de transmission physique par liaison ethernet seraient largement dépassées, mais aussi à cause du coût que cela engendrerait.

On fait donc appel aux lignes louées, qui sont fournies par les opérateurs télécom. Chaque ligne dispose donc de son propre protocole et de son propre débit (E1 : 2Mb/s, E3 : 34Mb/s).

Les routeurs ne remontent jamais au-dessus de la couche 3 du modèle OSI. Par contre, comme les routeurs retranscrivent les trames d'un protocole dans un autre, il faut que le logiciel intégré dans le routeur soit capable de router ce protocole.

Si on utilise un protocole non routable, le routeur ne fait que transporter le paquet d'un point à un autre, et on parle alors de pontage. Ainsi, un pont ne fait que bêtement transcrire des trames entre deux réseaux reliés par une interface autre que celle du réseau local, alors qu'un routeur sera capable d'orienter les paquets selon leur destination. Un pont n'est pas capable de comprendre un modèle en couche au-delà du niveau 2.

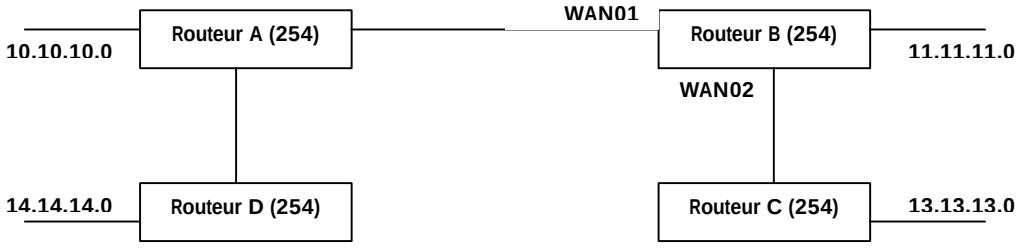

Fig. 13 – *Réseaux locaux interconnectés par routeurs.*

Dans l'exemple ci-dessus, on a 4 réseaux (10.10.10.x, 11.11.11.x, 12.12.12.x et 13.13.13.x). Chaque réseau dispose d'un routeur (respectivement A, B, C et D) qui leur permette de dialoguer entre eux. On a donc configuré les routeurs de manière à permettre les échanges entre chaque réseau.

On a donc ce que l'on appelle des tables de routage pour chacun des routeurs.

Exemple pour B :

Destination	Masque	Interface
10.10.10.0	24	WAN01
13.13.13.0	24	WAN02
11.11.11.0	24	LAN01
0.0.0.0	0	WAN02

Chaque paquet à destination du réseau 10.10.10.0 sera donc orienté vers l'interface WAN01, et ainsi de suite pour tous les paquets. Par défaut, la destination 0.0.0.0/0 est utilisée, et donc tous les paquets ne trouvant pas leur destination dans une table de routage seront orientés vers cette interface.

Remarque : un routeur ne diffuse pas les messages de broadcast, et « isole » ainsi un réseau local des autres réseaux. De la même manière, il ne diffuse pas les requêtes ARP, mais les émet lui-même à l'autre routeur auquel il est relié.

IV. Sécurité des réseaux.

La sécurité informatique est importante dans les réseaux globaux, car vos données sont accessibles à tout le reste du monde. La mise en place d'une politique de sécurité informatique appartient à l'administrateur réseau, qui devra prendre en compte les besoins des utilisateurs, ainsi que les risques encourus.

Lorsqu'un service présente une défaillance dans le dispositif de sécurité, on parle de trou de sécurité. Il peut alors donner des accès à d'autres personnes non autorisées, ou même donner les pleins pouvoirs sur le système à une personne extérieure.

IV.1 Qu'est-ce que la sécurité informatique.

On appelle sécurité informatique les moyens matériels et logiciels permettant à vos systèmes de fonctionner normalement. Cela implique une **confidentialité** dans vos données, qui ne devront être accessibles que par les personnes habilitées.

Les données ne doivent être modifiables que par les personnes autorisées à le faire, donc **l'intégrité** des données est elle aussi importante.

Une politique de sécurité correctement établie doit assurer aux utilisateurs une **disponibilité** constante dans les outils informatique. Chacun doit pouvoir lancer les commandes sur lesquels il a reçu une autorisation.

Il existe deux politiques de sécurité selon les sites à protéger :

- tout ce qui n'est pas explicitement permis est interdit,
- tout ce qui n'est pas explicitement interdit est autorisé.

La première méthode est la plus sécurisée, mais elle présente de grosses contraintes pour vos utilisateurs, et n'est donc pas toujours applicable. Il n'existe pas de règle d'or quant à savoir quelle méthode appliquer, il faut donc étudier les besoins au cas par cas.

IV.2 Que faire en cas d'intrusion.

Dans le cas où votre système aurait été pénétré, ou si vous soupçonner son intégrité, vous devez tout de suite isoler la machine concernée du réseau, et mettre en place des agents de surveillance sur les autres machines sensible. Vous devrez ensuite en informer votre direction informatique, de manière à ce qu'elle prenne les mesures nécessaires, puisqu'elle est la seule à pouvoir décider d'une action en justice. Relevez et archivez ensuite toutes les traces possibles dans vos fichiers de logs. Ceux-ci pourront éventuellement vous aider à définir la source de l'attaque, bien que généralement les attaques proviennent de comptes déjà préalablement piratés, et donc sans aucune valeur.

Il est ensuite important de vérifier chacune des commandes de votre système, notamment les commandes qui s'exécutent en tant qu'administrateur (ex : /bin/login, /bin/su, etc.) car elles sont généralement les cibles favorites des pirates.

IV.3 Règles importantes.

Dans tous les cas, il existe des règles à ne pas contourner. Votre système doit correctement gérer les groupes d'utilisateurs, afin de définir précisément chaque outil ou donnée pour chaque groupe d'utilisateur.

Certains fichiers de configuration, ainsi que certaines commandes n'ont pas à être exécutées par une autre personne que l'administrateur lui-même, donc inutile de donner les droits aux utilisateurs sans pouvoir. Par exemple, la commande ifconfig, qui permet de configurer une interface réseau, n'a aucun intérêt à être exécuté par un utilisateur sans pouvoir. On désactivera donc les droits d'exécution pour les utilisateurs.

Les fichiers de configuration du système (généralement placées dans /etc) ne doivent absolument pas être modifiables par une autre personne que l'administrateur.

Refuser les accès par telnet depuis la machine que vous sécurisé. Une des méthodes des pirates pour ne pas se faire repérer consiste à obtenir un accès sur un serveur en tant qu'utilisateur standard, puis de faire 'telnet localhost'. Ainsi, vos traces dans le système ne laisseront apparaître que des connexions depuis cette machine, ce qui présente à priori peu de danger d'intrusion.

Evitez les services inutiles, et notamment les services ne demandant pas d'authentification, tels rsh, rlogin, rpc, NFS, etc.

Ne donner pas accès à trop d'informations sur vos serveurs. Une bonne méthode pour se prémunir contre les tentatives d'accès est de se protéger par une chrootage des répertoires. On entend par chrootage l'impossibilité pour un utilisateur de remonter au-delà de l'arborescence pour laquelle il est autorisé. Un bon exemple de mise en place est le protocole FTP. Les personnes qui se connectent en FTP n'ont aucune raison de prendre votre fichier /etc/passwd, donc limiter les accès FTP à la racine des répertoires de chaque compte.

IV.4 Méthodes d'intrusion les plus courantes.

IV.4.1 Buffer overflow (dépassement de capacité).

Une des méthodes d'intrusion les plus courantes consiste à exécuter une commande avec le set-uid bit de positionner, puis de stopper cette commande avec une violation d'accès, en opérant par exemple un buffer overflow (dépassement de capacité). Par exemple, un trou connu fut celui de sendmail, le gestionnaire de courrier. En quelques lignes de C, on pouvait alors obtenir un shell root sur n'importe quelle station. La technique était simple :

1. Ouvrir le fichier /bin/sh (fopen par exemple)
2. Exécuter une commande set uid (dans notre cas, la commande /usr/sbin/sendmail),
3. Empiler volontairement et sans arrêt des variables dans la pile mémoire du système.

Lors de l'exécution, le programme est alors appelé, puis le fichier /bin/sh ouvert. Ensuite, la commande sendmail est exécutée, mais comme nous l'avons vu dans le protocole SMTP, celle-ci attend un certain nombre de commandes. Durant ce temps, nous empilons des variables dans la pile de données du système, ce qui va finir par la faire déborder. Lorsque qu'elle débordera, vous aurez alors une violation de segment, mais le système vous aura laissé les droits de l'administrateur jusqu'à la fin de votre shell d'exécution. Vous aurez donc une commande sh ayant le set-uid bit de positionné. Exécutez cette commande, et vous aurez alors tous les privilèges de l'administrateur.

Pour palier à ce problème, vous pouvez enlever le droit de lecture sur la majeure partie des commandes utilisateurs. Par exemple, /bin/sh n'a pas besoin du bit de lecture, ce qui permettra aux utilisateurs d'exécuter cette commande, mais pas de la lire, et donc le système refusera d'ouvrir le fichier.

IV.4.2 Sniffeurs de trames.

Les sniffeurs de trames permettent depuis n'importe quelle machine reliée au LAN de voir ce qui transite dans les paquets du réseau. On a vu précédemment que chaque interface réseau était capable de déterminer si le paquet lui était ou non adressé, mais chacune des interfaces voit en fait tous les paquets. Il existe donc des outils sous la majeure partie des systèmes d'exploitation qui permettent de voir tous les paquets. Le plus célèbre est sans doute tcpdump sous Unix, qui donne une foule d'informations, ainsi que le contenu des paquets.

Lorsqu'une personne tente de se connecter sur une machine distante, elle doit s'identifier sur cette machine, et entre donc son nom de login et son mot de passe. Ces deux informations passent en clair dans les trames du réseau. En sniffant les trames, vous retrouverez alors le nom de login et le mot de passe de tous les utilisateurs qui se connecteront à distance.

Il n'existe pas vraiment de solutions à ce problème, à cause du gouvernement français. Une bonne solution serait l'usage de protocole chiffré, mais le gouvernement s'oppose à son usage sur le territoire français. Il existe un outil de connexion chiffré : ssh. Lorsque vous sniffer les trames, vous

verrez alors apparaître le nom de connexion et le mot de passe, mais codé, donc peu ou pas du tout utilisable.

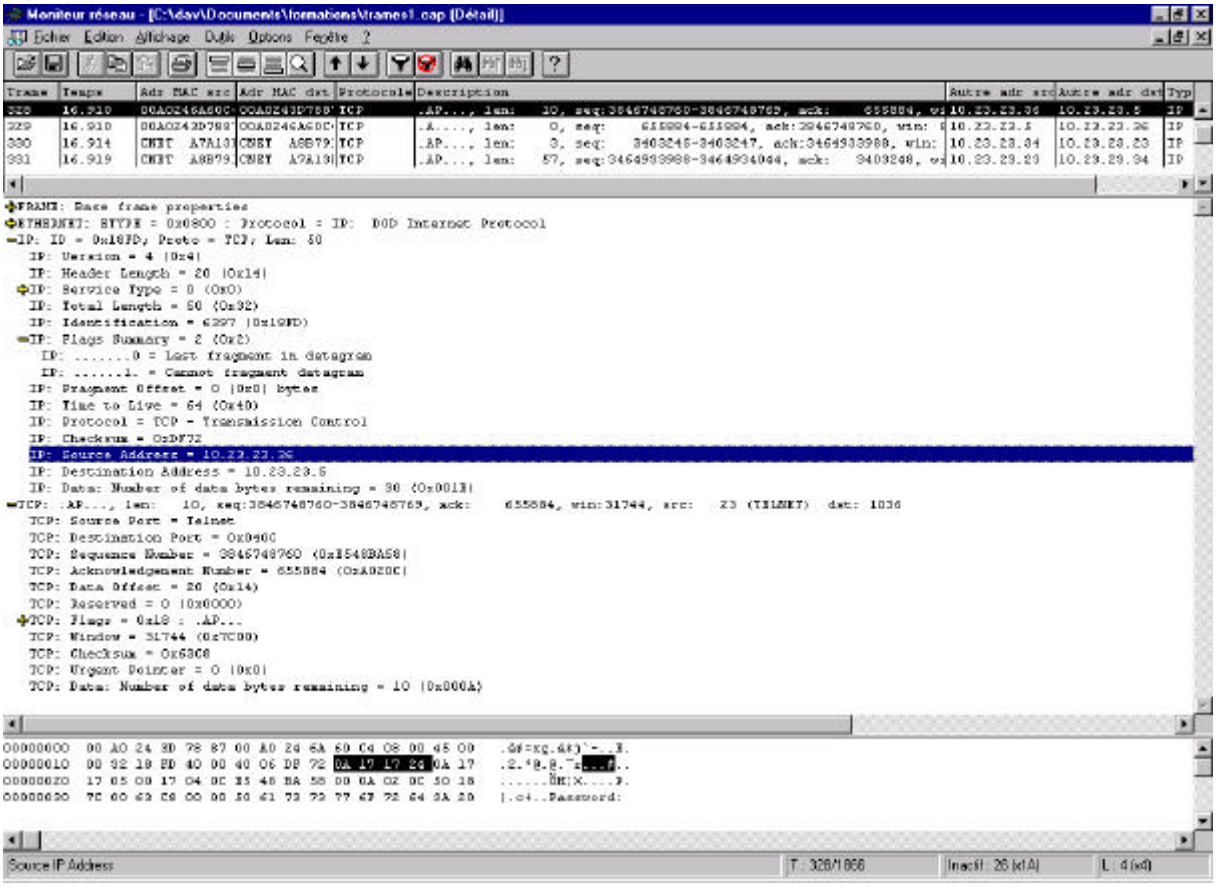

Fig. 13 – *Exemple d'une trame « sniffé » sur un réseau local (Microsoft S.M.S).*

IV.4.3 Exploitation de bugs.

Il arrive que certains systèmes présentent des bugs importants dans le fonctionnement du système. Ce fut le cas l'an passé sur la couche réseau. La majorité des systèmes Unix étaient affectés.

Une trame ICMP n'est pas prévue pour excéder une longueur de 16bits, soit 65535. Or, les développeurs de la couche réseau n'avaient pas testé cette limite. Un simple ping avec une longueur de trame excédant 65535 suffisait donc à crasher les serveurs. Ce bug a été corrigé au bout de 4 heures après sa diffusion sur l'Internet pour le système Linux, et plusieurs semaines pour les systèmes Windows (NT et 95) de Microsoft.

IV.4.4 Déni de service (Denial of Service).

Les attaques par denial of service ne donne aucun moyen à l'agresseur de pénétrer votre système, mais permet à quiconque de stopper le fonctionnement de vos machines. Le bug énoncé pour ping dans le point précédent est un exemple d'attaque par déni de service.

Sous Windows, on notera la présence du port 139 (port RPC : Remote Process Control) qui présente un trou de sécurité. Vous pouvez très bien vous connecter par telnet sur ce port, puis entrer quelques caractères et vous déconnecter. Les services réseaux de la machine cible ne sont alors plus disponibles jusqu'au prochaine redémarrage.

IV.5 Protection par « mur coupe-feu » (FireWall).

Le firewall est sans doute l'outil de sécurité le plus performant à l'heure actuelle. Il permet de se protéger du réseau global en isolant le réseau local. Il n'exclut toutefois pas les risques de piratage interne, mais solutionne le problème d'intrusion depuis l'extérieur.

Un firewall est en fait une passerelle avec un minimum de services installés. Elle dispose des logiciels nécessaires pour acheminer à votre place vos requêtes. Il se place entre le réseau à protéger et le réseau global.

Fig. 14 – *Protection d'un réseau local avec un Firewall.*

Aucun paquet ne doit passer entre le réseau local et le réseau global. Donc, tous les services de routage doivent être désactivés sur le firewall. Chaque application qui sera autorisée nécessitera donc un proxy. Le proxy reçoit la requête du client, regarde si celui-ci est autorisé à utiliser ce service, et si oui, effectue la requête à sa place avant de lui renvoyer la réponse. Ainsi, les machines à l'intérieur du réseau sont invisibles pour l'extérieur du réseau.

Toutes les machines se trouvant sur le réseau local devront avoir des adresses IP non routable (voir les classes d'adresses IP non routables).

On configure alors sur le firewall quels services seront disponibles par les proxies appropriés, et quelles machines seront autorisées à les utiliser. C'est ainsi que la protection est la mieux assurée.

Remarque :

(i) **Attention, un réseau équipé d'un firewall mal configuré est souvent pire qu'un réseau sans protection, car si quelqu'un trouve une faille dans votre firewall, et parvient à le pénétrer, toutes vos machines seront alors en péril.**

(ii) **Il est grandement conseillé de ne jamais installer un serveur Windows NT sans utiliser de firewall pour la protection.**