Licence Professionnelle de l'Université de Provence
Systèmes Informatiques et Logiciels
Parcours *Nouvelles Technologies de l'Informatique*

Cours Réseau

Licence professionnelle SIL-NTI de l'Université de Provence
Catégorie : systèmes et réseaux (IP4)

Grégory Colpart <reg@evolix.fr>
Ingénieur en informatique Evolix

Table des matières

1 Introduction

1.1 Vocabulaire

1.1.1 Protocole/Norme/Standard/Spécifications

Protocole = ensemble de règles formelles décrivant un comportement. Les règles peuvent être écrites par des sociétés commerciales et rester secrètes ou bien être ouvertes.
`Exemples de protocole` : FTP, ICQ, MSN Messenger

Spécification = document technique décrivant de façon précise le fonctionnement de matériel, services, etc.
`Exemples de spécification` : Federal Standard 1037C [1]

Norme = document établi par un consensus et approuvé par un organisme de normalisation reconnu.
En voici quelques-uns : IEEE [2], IETF [3], W3C [4], ISO [5], et des organismes nationaux (ANSI [6], AFNOR [7])
`Exemple de normes` : ASCII, HTML

Standard = "norme de fait", c'est-à-dire répandue par une utilisation générale. Un standard n'est donc pas forcément détaillé dans un document officiel.
`Exemples de standard` : MPEG, PDF, HTML

1.1.2 Unités informatiques

L'unité élémentaire en informatique est le Bit. Un bit est un binaire (0 ou 1).
On a les équivalences suivantes : *1 octet = 1 o = 1 byte = 8 Bits*

Extensions informatiques :

K	kilo	10^3	1024
M	mega	10^6	1 048 576
G	giga	10^9	1 073 741 824 ~1 milliard (US : billion)
T	tera	10^{12}	1 099 511 627 776 ~1 billion (US : trillion)
P	peta	10^{15}	1 125 899 906 842 624 ~1 billiard (US : thousand billions)

`Exemple` : 1 To = 8.10^{12} huit billions de bits

Note : On devrait noter les extensions informatiques suivies d'un 'i' pour les différencier des extensions décimales : Kio, Mio, Gio, ...
`Exemple` : disque dur 120 Go = 120 000 000 octets soit environ 114.4 Gio

[1] http://www.its.bldrdoc.gov/fs-1037/fs-1037c.htm
[2] http://www.ieee.org/
[3] http://www.ietf.org/
[4] http://www.w3.org/
[5] http://www.iso.org/
[6] http://www.ansi.org/
[7] http://www.afnor.fr/

Unité de mesure :

```
1 baud = 1 Bps = 1 Bit/s
8 bauds = 8 Bits/s = 1 octet/s
512 Kbauds = 512 KBps = 64 Ko/s
1 MBps = 128 Ko/s
10 MBps = 1.25 Mo/s
100 MBPS = 12.5 Mo/s
```

1.1.3 Notation informatique

– BINAIRE :

$$a = \sum_{k=0}^{+\infty} a_k \cdot 2^k$$

où $a_k = 0$ ou 1

Conversion de décimal en binaire :

Tableau puissance de 2 :

2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	1	0	1	0	1	1	0	1

1. On prend la puissance de 2 inférieure au décimal et dans la colonne correspondante on note 1
2. On prend la puissance de 2 inférieure au (décimal - puissance de 2 précédente) et on note 1 dans la colonne relative
3. On itère et arrivé à 0, on complète le tableau par des 0

Conversion de binaire en décimal :

On utilise la formule

$$a = \sum_{k=0}^{+\infty} a_k \cdot 2^k$$

où $a_k = 0$ ou 1

Opérations élémentaires

Tables de vérité

opérateur ET LOGIQUE (AND) :

&	0	1
0	0	0
1	0	1

opérateur OU LOGIQUE (OR) :

|	0	1
0	0	1
1	1	1

opérateur OU EXCLUSIF (XOR) :

(+)	0	1
0	0	1
1	1	0

opérateur INVERSE :

inverse(0)=1
inverse(1)=0

– HEXADÉCIMALE

$$a = \sum_{k=0}^{oo} a_k \cdot 16^k$$

où $a_k = 0, 1, .., f$

Permet de coder 8 bits sur un octet en utilisant les caractères 0,1,2,..,e,f

Ainsi, un octet se note avec deux caractères hexadécimaux.
On précède souvent une écriture héxadécimale du préfixe '0x'

Exemples de notation hexadécimale pour 1 octet :

```
00 : 00000000
01 : 00000001
05 : 00000101
0a : 00001010
0f : 00001111
3a : 00111010
ff : 11111111
```

1.1.4 Fonctions de hashage

Une fonction de hashage est une fonction difficile à inverser.
En pratique, elle est utilisée pour générer des empreintes qui permettent de vérifier l'intégrité de données.
La probabilité de générer des données différentes ayant la même empreinte est très faible. (on parle alors de collision)

Soit $h : x \rightarrow h(x)$
Trouver $x' \neq x$ tel que $h(x) = h(x')$ est très difficile.

Remarque : Par très difficile, on entend *techniquement impossible* que ce soit au niveau algorithmique ou matériel

Principe :

Soit H fonction de hashage

A calcule $h = H$(message)
$A \rightarrow$ [message+h] $\rightarrow B$

B calcule $h' = H$(message reçu)
Si ($h = h'$) alors le message reçu est intègre

1.2 Historique des réseaux

De l'Arpanet à Internet, de l'Internet au Web [8]

1957 : Le ministère de la Défense américain crée l'Agence ARPA (Advanced Research Project Agency) dont l'objectif est de renforcer les développements susceptibles d'être utilisés à des fins militaires.

Contexte des années 60 : Guerre Froide, Spoutnik, lutte technologique et époque où les entreprises informatiques créent des logiciels non interopérables empéchant de partager des ressources
(but de ces dernières : créer un monopole et emprisonner le peu d'utilisateurs qui sont les centres de recherche et entreprises importantes.)

1968 : Le National Physical Laboratorie (laboratoire nationale de physique), en Grande-Bretagne, met en place le premier réseau à commutation de paquets (découpage par paquets de l'information numérisée). Le principe est le suivant : le message est divisé en blocs transmis aléatoirement à travers les connexions du réseau. A l'arrivée, le destinataire reconstitue le message indépendamment du chemin emprunté par ces derniers.

1969 : Sous l'égide de l'ARPA, quatre ordinateurs situés chacun dans des centres universitaires différents sont reliés : naissance de L'ARPANET. Ce premier réseau se développe rapidement et en 1973, il comporte 35 machines.

Technologiquement : Un premier protocole, le NCP XXX, est mis au point dès 1970. Les ingénieurs d'Arpanet proposent d'implanter dans les ordinateurs un langage qui leur serait commun une fois en réseau même si l'ordinateur possède son propre langage pour ses application propres.

[8] http://www.funoc.be/etic/doss001/art001.html

1974 : V. Cerf et B. Kahn publient le TCP/IP (Transfert Control Protocol/Internet Protocol) qui permet, cette fois, aux réseaux de communiquer entre eux même si chacun, pris individuellement, continue à parler son propre " langage".

1979 : Six universités s'associent pour réfléchir à la création d'un nouveau réseau ouvert à l'ensemble de la communauté scientifique, le CSNET (Computer Science Research Network)

1983 : Les militaires décident de diviser l'Arpanet en deux, le Milnet, un réseau uniquement dédié à la transmission de données militaires et Arpanet proprement dit, dédié à la communauté scientifique.

De son côté, le monde commercial crée lui aussi ses propres réseaux. La société IBM met au point le BITNET (Because It's Time NETwork, "parce c'est le temps des réseaux") qui permet de relier des ordinateurs IBM entre eux. Les laboratoires Bell définissent le protocole UUCP (UNIX to UNIX Copy Protocol, ("protocole de copie entre UNIX") qui assure le transfert de fichiers et la possibilité de commander à distance un ordinateur par un autre ordinateur. Ce protocole sera utilisé sur le réseau USENET (UNIX uSer Network, "réseau des utilisateurs UNIX"). Tous ces réseaux pourront communiquer entre eux grâce aux protocoles TCP/IP et l'ensemble prendra progressivement le nom d'Internet.

1989 : Création du World Wide Web par Tim Berners-Lee, physicien britannique (CERN). Idée née du besoin du CERN de gérer ses documents. WWW est formé par le protocole HTTP (HyperText Transfer Protocol), les URL (Uniform Ressource Locator) et le langage HTML (Hypertext Markup Language).

1990 : Naissance des premiers navigateurs (logiciels clients des serveurs HTTP capables d'interpréter le langage HTML)

1993 : navigateur Mosaic implémenté par le NCSA (National Center for Supercomputing Applications)

1994 : Jim Clarck rachète Mosaic pour créer Netscape

2 Modèles théoriques

2.1 L'art de la communication

2.1.1 Définition d'un réseau informatique

Question : Quels sont les sous-entendus qui font qu'un professeur échange de l'information avec des élèves ?

Quelques réponses possibles :

- support physique (salle adaptée remplie d'air)
- émetteurs et récepteurs adaptés au support physique
- parler un langage (grammaire, vocabulaire, etc.)
- comprendre le langage (niveau sonore suffisant, récepteurs attentifs)
- cohérence des échanges (logique dans l'échange)
- arbitrage des échanges (centralisation, système à jetons)

Sur un système informatique, la communication entre les processus, alias IPC (InterProcess Communication), peut fonctionner grâce à certains paramètres définis (PID - Process IDentifier, appels systèmes, etc.).
La capacité des systèmes informatiques à offrir différents services (fichiers, données, puissance de calcul, ...) repose sur la possibilité de faire communiquer des "processus distants".
L'idée générale a donc été de développer des mécanismes pour véhiculer l'information entre les systèmes. Mais les problématiques posées par la communication entre deux systèmes distants sont nombreuses (matériel, programmation) et imposent le choix de technologies communes à tous les systèmes.

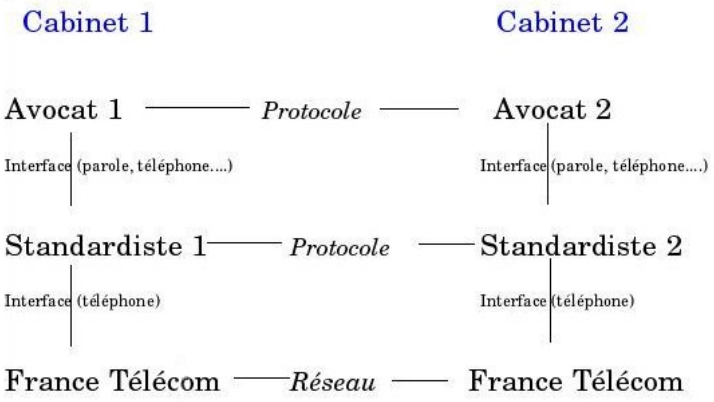
Un réseau informatique est donc constitué par :

1. les systèmes dont les processus de haut niveau interagissent
2. l'ensemble des équipements intermédiaires qui permettent à ces processus d'interagir

2.1.2 Les contraintes d'un réseau informatique

Question : Un avocat d'un cabinet souhaite entrer en contact avec un autre avocat d'un autre cabinet. Quelle est la structure d'un cabinet d'avocat afin d'organiser les échanges ?

Réponse possible :



D'une façon parallèle, les interactions au niveau d'un réseau informatique se situent entre les processus distants, les systèmes distants, les équipements intermédiaires, etc. On comprend bien la nécessité de mettre en place des règles de communication à plusieurs niveaux, ainsi que des choix en terme de rapidité, fiabilité, coût, etc.
Par exemple, la standardiste ralentit les communications mais offre des services à l'avocat : gestion de tous les problèmes de connexion (erreur de numéro, correspondant injoignable, etc.), filtrage des appels.
Cette structure nécessite des conventions : messages standards (sonnerie avec une fréquence particulière signifie que la ligne est occupée), format des échanges, etc.
Une fois la communication établie entre les avocats, elle se déroule de manière transparente aux services sous-jacents.

2.1.3 Les mécanismes d'un réseau informatique

Question : un employé d'une société en Provence envoie un message confidentiel à un employé d'une société en Allemagne.
Modéliser les échanges et les intervenants.

Solution possible :

Le message étant confidentiel, chaque niveau n'a besoin que de peu de renseignements :

– Le niveau 4 n'a besoin que de connaître les coordonnées des deux bureaux postaux

– Le niveau 3 n'a besoin que de connaître l'adresse de la société

– Le niveau 2 n'a besoin que de connaître le nom de l'employé

On observe donc ici des notions de couches, de services et de données encapsulées. Chaque niveau interagit de façon "verticale" pour encapsuler les données et communique de façon "horizontale" avec un niveau équivalent. Ainsi, la Poste n'a pas besoin de connaître le nom de l'employé B (l'enveloppe contenant son nom pourrait être mise dans une enveloppe plus grande ne contenant que l'adresse de sa société). Son action est de communiquer l'adresse de la société de l'employé B à la Poste distante (communication horizontale) et pour cela elle transmet l'adresse de la Poste distante au transporteur (communication verticale).

Le mécanisme qui consiste à insérer l'enveloppe reçue dans une enveloppe plus grande et contenant d'autres données s'apparente à l'encapsulation. En réception, les enveloppes sont ouvertes ce qui correspond à décapsuler les données. Comme pour la communication entre avocats, on voit également la nécessité de conventions telles que le format des échanges (noms, adresses,...), une suite logique des échanges et un service assuré par chaque entité.
Cet ensemble est appelé le protocole (à mettre en parallèle des protocoles officiels pour les chefs d'état destinés notamment à prendre en compte les coutumes différentes de chaque pays).

Dans le but d'organiser de manière logique les échanges des réseaux informatiques, des modèles ont été développés. Certains sont restés au stade théorique (modèle OSI) ou d'autres au stade pratique (TCP/IP).

2.1.4 Notions de base

Voici différents sigles sous la forme xAN (x Area Network) dont il est intéressant de connaître la signification (même si certains ne sont pas très utilisés) :

PAN : Personal Area Network (ou HAN : Home Area Network)

LAN : Local Area Network

VLAN : Virtual Area Network

WLAN : Wireless LAN (ou LAWN : Local Area Wireless Network)

ILAN : Industrial LAN

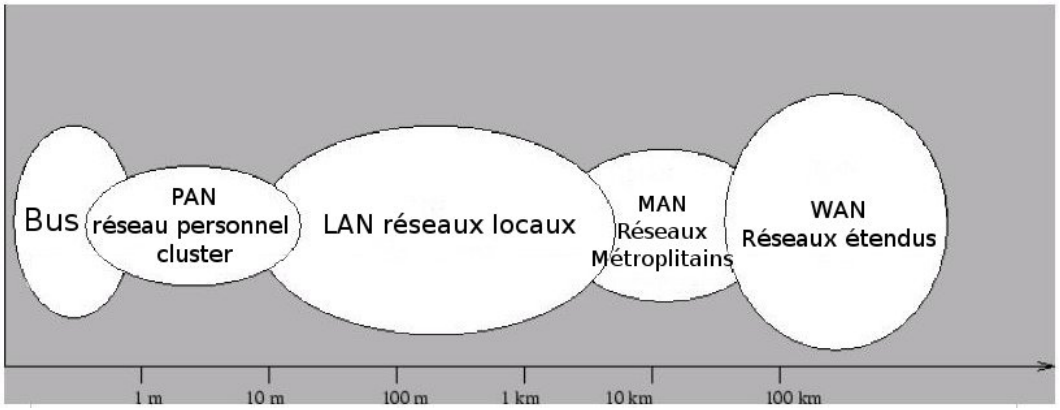
MAN : Metropolitan Area Network

WAN : Wide Area Network

EWAN : External WAN

WWAN : Wireless WAN

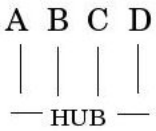
et connaître leur ordonnancement :



Topologie
Un réseau informatique est constitué d'ordinateurs reliés grâce à des cartes réseau, des câbles et des équipements réseau spécifiques. La topologie physique est la disposition physique de ces élements. On distingue :

– **la topologie en bus**
Toutes les machines sont reliées à une même ligne de transmission (souvent un câble coaxial)

– **la topologie en étoile**
Toutes les machines sont reliées à un équipement spécifique appelé concentrateur. Cette topologie est plus fiable (une machine peut se déconnecter sans déranger) mais plus coûteuse.



– **la topologie en anneau**
Les machines sont reliées en boucle et communiquent chacun leur tour.

En réalité, les machines sont reliées à **MAU** (Multistation Access Unit) qui est un répartiteur qui gérer le "temps de parole" de chaque machine.

Les technologies Token Ring et FDDI utilisent cette topologie.

La topologie logique représente le mode de circulation de données dans la topologie physique (Ethernet, Token Ring, FDDI, etc.)

Unicast, Broadcast, Multicast

Un émetteur peut envoyer des données à un système en particulier (Unicast) mais également à un ensemble de systèmes (Broadcast) ou à un groupe particulier de systèmes (Multicast).

Unicast : Il s'agit du mode de transmission le plus utilisé où l'émetteur envoie les données à un seul récepteur.
Broadcast : Le mode de transmission Broadcast est celui où un émetteur diffuse des informations à tous les hôtes d'un réseau à la fois. L'adresse Broadcast est une adresse de diffusion utilisée pour envoyer des données à tous les hôtes du réseau. Les paquets Broadcast ne peuvent pas traverser les routeurs. Il s'agit de l'adresse la plus "haute" dans le réseau.
Multicast : Le terme de Multicast définit une connexion réseau multipoint, en opposition à l'utilisation commune Unicast qui fonctionne d'un émetteur à un destinataire précis. Il s'agit d'un protocole standard normalisé dans la RFC 1112 qui permet d'envoyer un seul paquet qui sera reçu par plusieurs destinataires. Le protocole IGMP (voir plus bas) permet de gérer les informations sur les membres des

groupes de multidiffusion.
L'utilisation de Multicast sera précisée dans le 3.2.1.

Mode connecté ou non connecté

Il est important de comprendre les notions de "mode connecté" et "mode non connecté". Lors d'une connexion en "mode connecté", il y a notamment un échange préalable à la connexion pour s'assurer que les hôtes sont bien connectés. On pourrait comparer cela à un coup de téléphone où deux personnes commencent à se parler après avoir échangé un "allô". Lors d'une connexion en "mode non connecté", les données sont envoyées sans vérification préalable : l'émetteur espère que les données seront bien reçues sans garantie en terme de délai. On peut comparer cela à l'envoi d'une lettre par la Poste.

Transmission synchrone ou asynchrone

Une transmission synchrone signifie que l'émetteur et le récepteur se mettent d'accord sur une base de temps (un top d'horloge) qui se répète régulièrement durant tout l'échange. À chaque top d'horloge (ou k tops d'horloge k entier fixé définitivement) un bit est envoyé et le récepteur saura ainsi quand lui arrive les bits.
Une transmission asynchrone signifie qu'il n'y a pas de négociation préalable mais chaque caractère envoyé est précédé d'un bit de start et immédiatement suivi d'un bit de stop. Ces deux bits spéciaux servent à caler l'horloge du récepteur pour qu'il échantillonne le signal qu'il reçoit afin d'y décoder les bits qu'il transmet.

Architecture client/serveur

La communication entre des systèmes informatiques distants fonctionne souvent selon l'architecture client/serveur.

L'architecture client-serveur[9] se résume à la demande de services d'un programme client à un programme serveur. Il s'agit de l'extension logique du partitionnement des logiciels importants en modules donnant la possibilité de développement et de maintenance plus aisés. Les modules *demandeurs* sont qualifiés de *client* et les modules appelés sont nommés *service*. Ainsi les différents modules fonctionnent sur des plateformes différentes et appropriées à leur fonction. Par exemple, les systèmes de gestion de base de données tournent sur des plateformes logicielles et matérielles conçus pour optimiser les requêtes, ou les serveurs de fichiers tournent sur des plateformes adaptées pour la gestion de fichiers.

Le client est donc un programme qui envoie un message à un programme serveur, demandant à ce serveur un service. Les programmes clients sont en général constitués d'une interface permettant de valider les données entrées par l'utilisateur et d'un programme permettant de traiter et d'envoyer les requêtes aux programmes

[9] http://www.faqs.org/faqs/client-server-faq/

serveurs.

Le programme contient donc un certain nombre de facilités pour interagir avec l'utilisateur. Ainsi, il accède aux ressources locales (écran, clavier, processeur, périphériques, etc.).
Un des éléments souvent présent sur une machine de type poste de travail est une interface graphique : GUI (Graphical User Interface).
Normalement, c'est le Windows Manager qui détecte les actions de l'utilisateur, gère les différentes fenêtres et affiche les données.

Le serveur est un programme qui répond aux demandes du client en réalisant la tâche demandée. Les programmes serveurs recoivent en général des requêtes des programmes clients, exécutent des requêtes et mises-à-jour sur une base de données, contrôlent l'intégrité des données et répondent aux programmes clients. Le programme serveur devrait être sur une machine indépendante sur le réseau mais souvent plusieurs programmes serveurs sont sur la même machine et dans certains cas, la machine hébergeant le service est un poste de travail. Le programme serveur peut souvent accéder à des resources locales telles que les bases de données, imprimantes, interfaces et processeur(s).

2.2 Le Modèle OSI

Le modèle de référence est une architecture en sept couches développée par l'ISO (International Standards Organization).
Ce modèle appelé OSI (Open Systems Interconnection) fixe une base commune pour le domaine des télécommunications.
Voici comment il s'organise :

- **Couche application (Application Layer)**
  Cette couche est constituée des programmes informatiques connaissant différents paramètres (destinataire, qualité de service, etc.) et utilisant directement les possibilités en terme de réseau proposées par la couche de présentation.

- **Couche de présentation (Presentation Layer)**
  Cette couche effectue une traduction (changement de syntaxe) pour que les

données puissent être interprétées par la couche application (et vice-versa). On l'appelle parfois la couche de syntaxe (Syntax Layer).

- **Couche de session (Session Layer)**
  Cette couche gère l'établissement, la continuité et la fermeture des connexions entre les applications.

- **Couche de transport (Transport Layer)**
  Cette couche assure l'acheminement des données de manière transparente entre les systèmes en intégrant des contrôles plus ou moins avancés des erreurs durant le transfert.

- **Couche de réseau (Network Layer)**
  Cette couche réalise l'acheminement des données en gérant adressage, routage, etc.

- **Couche de liaison (Data Link Layer)**
  Cette couche contrôle la fiabilité des transmissions effectuées par la couche physique.

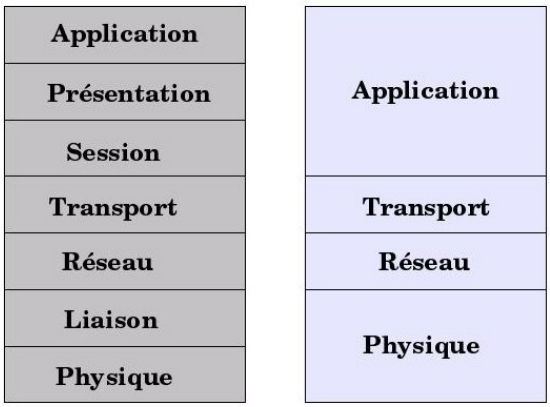
- **Couche physique (Physical Layer)**
  Cette couche contient les spécifications matérielles (électrique, physique) pour faire véhiculer le signal de transmission des données.

Le modèle est donc organisé en couches superposées les unes sur les autres. On parle de pile de protocole ("stack", en anglais). Ces couches définissent des fonctions assurées par un protocole. Les protocoles normalisés n'ont ainsi pas à se préoccuper des couches supérieures et inférieures et se "contentent" de remplir leur rôle. C'est ainsi que deux systèmes peuvent communiquer au niveau de la couche réseau alors que leurs couches physiques diffèrent.

En pratique, le modèle OSI n'est pas vraiment utilisé pour diverses raisons (matériel, temps de calcul, efficacité). Des modèles comportant moins de couches ont été élaborés et notamment le modèle DOD sur lequel s'appuie TCP/IP.

2.3 Le Modèle DOD

Le modèle de DOD (Department Of Defence) a été produit par le département de la défense des États-Unis qui a joué un rôle important dans le domaine de TCP/IP. Le modèle de DOD comporte quatre couches :



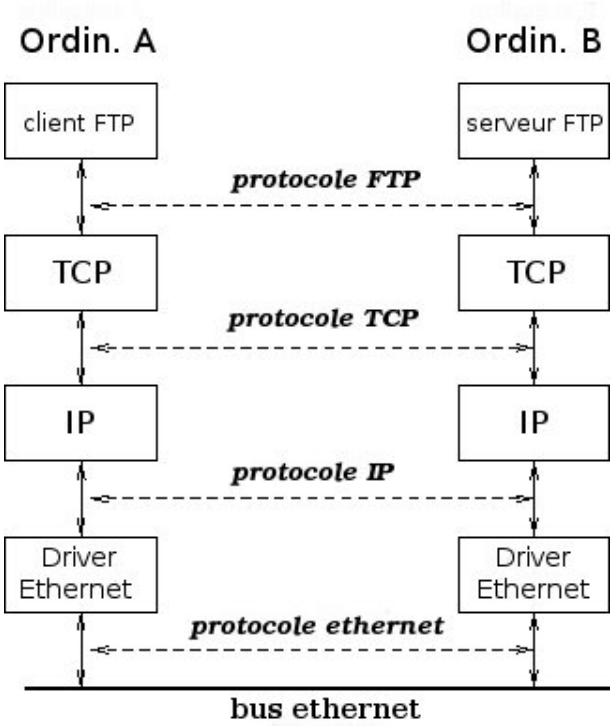
– **Couche application**
  Cette couche englobe les couches application, présentation et session du modèle OSI. Il existe de nombreux protocoles d'application offrant divers services à l'utilisateur (HTTP, FTP, DHCP, DNS, etc.).
  Voir le fichier `/etc/services` sous UNIX.

– **Couche transport**
  Cette couche est équivalente à la couche transport du modèle OSI. Les protocoles de la couche transport sont plus moins élaborés pour assurer la fiabilité des échanges (TCP, UDP, RIP, IGMP, etc.).
  Voir le fichier `/etc/protocols` sous UNIX.

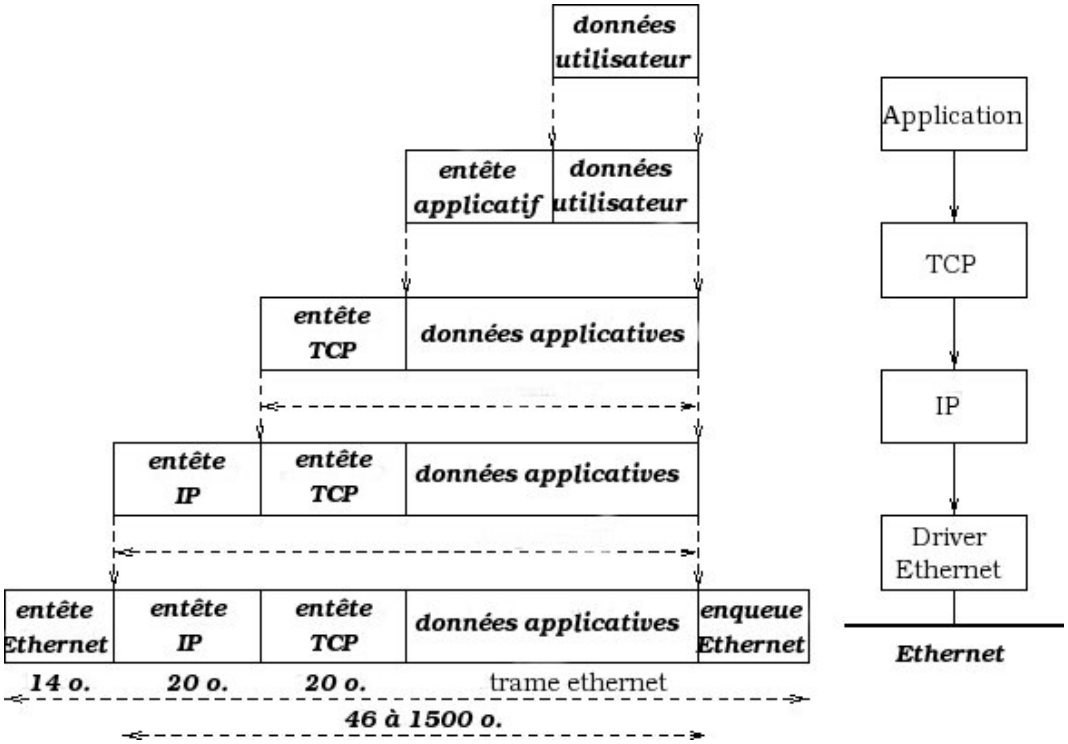
– **Couche réseau**
  Cette couche est équivalente à la couche réseau du modèle OSI. Le protocole de la couche réseau le plus répandu est IP (Internet Protocol) même si existent d'autres protocoles (AppleTalk, X.25, etc.)

– **Couche physique**
  Cette couche englobe les couches liaisons de données et physique du modèle OSI (bien que certains les laissent parfois séparées). Le protocole de la couche physique le plus répandu est Ethernet bien que de nombreux autres protocoles restent encore utilisés (PPP, FDDI, RNIS, etc.)

Exemple de pile TCP/IP :


Ordin. A
Ordin. B
client FTP
serveur FTP
protocole FTP
TCP
TCP
protocole TCP
IP
IP
protocole IP
Driver Ethernet
Driver Ethernet
protocole ethernet
bus ethernet

Encapsulation des données TCP/IP :



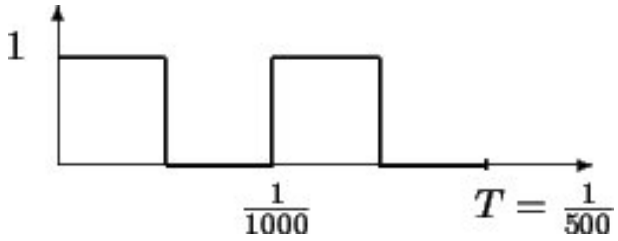
3 Couches du modèle DOD

3.1 Couche Physique

La couche physique est responsable du transport de signaux "physiques" (électronique, radio, laser), de l'émission et de la réception de bits et du découpage d'une série de bits en unité logique (couche 1 du modèle d'OSI) ainsi que des protocoles pour transporter ces signaux (couche 2 du modèle d'OSI).

Transmission :

– **Bande de base :** La transmission en bande de base consiste à envoyer directement les suites de bits sur le support à l'aide de signaux carrés constitués par un courant électrique pouvant prendre 2 valeurs (5 Volts ou 0 par exemple).
0101110 :



Il existe plusieurs types de codage pour une transmission en bande de base :

code "tout ou rien" : comme son nom l'indique, nul pour 0 et courant positif pour 1

code "NRZ" (non retour à zéro) : courant négatif pour 0 et positif pour 1

code bipolaire : idem à tout ou rien sauf que le 1 est alternativement positif ou négatif

code RZ (retour à zéro) : courant revenant systématiquement à zéro avant toute valeur

code Manchester (ou biphase) : un courant négatif puis positif pour 0 et l'inverse pour 1

code Miller : on diminue les translations en effectuant une transition (de haut en bas ou l'inverse) au milieu de l'intervalle pour coder un 1 et en n'effectuant pas de transition pour un 0 suivi d'un 1. Une transition est effectuée en fin d'intervalle pour un 0 suivi d'un autre 0

– **Modulée :**
Le principal problème d'une transmission en bande de base est la dégradation du signal en fonction de la distance. Ainsi en dehors d'un réseau local (inférieur à quelques kilomètres) l'utilisation de répéteurs et de ponts pour regénérer le signal s'avère trop coûteux.
Sur des longues distances on utilise donc un signal sinusoïdal (obtenu grâce à un modulateur-démodulateur alias modem).
Il existe trois types de modulation :

1. modulation d'amplitude
2. modulation de fréquence
3. modulation de phase

Passons maintenant à la couche de liaison.
Il existe diverses couches de liaison :

– Ethernet (IEEE 802.3) voir 3.1.2
– Token Bus (IEEE 802.4) voir 3.1.2
– Token Ring (IEEE 802.5) voir 3.1.3
– WI-FI (r) (IEEE 802.11) voir 3.1.4
– Bluetooth (IEEE 802.15) voir 3.1.5
– FDDI (norme ANSI)
– HDLC (norme OSI) voir 3.1.7
– SLIP/PPP voir 3.1.6
– RNIS voir 3.1.8
– ATM voir 3.1.9

3.1.1 IEEE 802

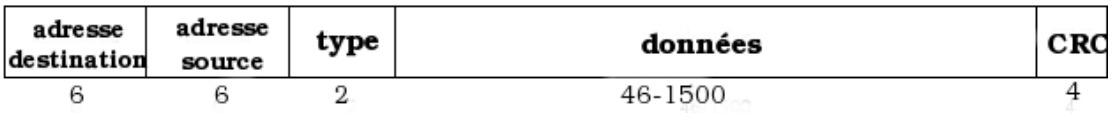
IEEE 802 est une commision d'experts chargée par l'organisme états-unien IEEE en février 1980 de statuer sur une norme commune pour les réseaux locaux. Cette commission a normalisé rapidement Ethernet (IEEE 802.3), Token Bus (IEEE 802.4) et Token Ring (IEEE 802.5).
Depuis d'autres normes ont été constituées[10].

3.1.2 Ethernet

Technologie inventée par Xerox au début des années 1970 et normalisée par l'IEEE en 1980 sous la norme IEEE 802.3
C'est la couche qui va nous intéresser car c'est la plus répandue sur les réseaux TCP/IP.

Format d'une trame Ethernet :

Trame Ethernet (RFC 894)



– **Préambule** :
Ce champ contient simplement des signaux permettant au récepteur de se synchroniser.

7 octets (10101010) + 1 octet de commencement effectif (10101011)

– **Adresse MAC (Media Access Control)**[11] :
Adresse physique d'une carte réseau attribuée directement par le constructeur. Elle est codée sur 48 bits et sensée être unique.

Exemple d'adresse MAC :
`00:C0:9F:23:C2:1B`
– **Type de données encapsulées :**

```
08:00 pour IP
08:06 pour IP-ARP
81:37 pour Novell
```
– **Données :**

Longueur minimum (présence éventuelle d'octets de "bourrage")

[10] http://ietf.org/rfc/rfc1042.txt
[11] http://www.coffer.com/mac_find/

– **FCS** (Frame Check Sequence) : il permet un contrôle d'erreur grâce à un calcul de type CRC (Cyclic Redundancy Check)

Pour information, les taux d'erreur sont de l'ordre de :

10^{-5} : ligne téléphonique
10^{-7} : cable coaxial
10^{-11} : fibre optique

En pratique...

On peut déduire de la commande suivante (et de sa sortie) :

```
# tcpdump -i eth0 -xx
```

```
18:43:24.172573 IP adsl-ns1.free.fr.domain > 192.168.7.181.33103:
31448 1/3/3 (192)
0x0000:  00c0 9f23 c21b 0002 4456 c7f2 0800 4500  ...\#....DV....E.
0x0010:  00dc 0000 4000 3e11 7ee8 d41b 20b0 c0a8  ....\@.>.~.......
0x0020:  07b5 0035 814f 00c8 4c33 7ad8 8180 0001  ...5.O..L3z.....
0x0030:  0001 0003 0003 0239 3003 3132 3103 3231  .......90.121.21
0x0040:  3202 3632 0769 6e2d 6164 6472 0461 7270  2.62.in-addr.arp
0x0050:  6100 000c 0001 c00c 000c 0001 0001 515b  a.............Q[
```

que :

```
client  : 192.168.7.181 (00:C0:9F:23:C2:1B)
routeur : 192.168.7.1   (00:02:44:56:C7:F2)
serveur : 212.27.32.176 (adsl-ns1.free.fr)
```

Il faut bien comprendre que toutes les trames captées par un système sont décapsulées afin de savoir à qui elles sont destinées.

Question : pourquoi l'adresse MAC de destination se trouve en premier dans l'entête Ethernet ?

Réponse : chaque trame ethernet étant décapsulée par chaque système pouvant la "voir" passer (c'est-à-dire sur le même segment de réseau), placer l'adresse MAC de destination en premier permet au système de se rendre compte le plus tôt possible si la trame lui est destinée ou non.
Les trames destinées à une adresse MAC acceptable pour le système (l'adresse MAC de la carte ethernet, une adresse de broadcast) sont transmises à la couche réseau.

Token Bus

La technologie Token Bus a été mise au point par la société General Motors. Cette

technologie a été normalisée par l'IEEE sous la norme IEEE 802.4

Token Bus signifie littéralement "bus à jeton". Elle a été principalement développée afin de combler une lacune d'Ethernet : l'accès aléatoire au support de transmission. Token Bus spécifie un accès déterministe au réseau en utilisant un jeton virtuel qui circule et permettant uniquement à l'hôte "saisissant" ce jeton d'émettre.

3.1.3 Token Ring

La technologie Token Ring [12] a été développée par IBM dans les années 1970 et normalisée par l'IEEE sous la norme IEEE 802.5

Token Ring signifie littéralement "anneau à jeton". Organisée avec une topologie de concentrateurs en anneau, chaque machine doit "saisir" un jeton virtuel circulant en permance pour pouvoir émettre des données. Alors que cette technologie était dans les années 1980 en réelle concurrence avec Ethernet, Token Ring est désormais totalement surpassé.

3.1.4 WI-FI

Le WI-FI (WIreless FIdelity[13]) correspond au départ à une certification délivrée à certains types de technologies sans fil. La norme est IEEE 802.11[14]

Les données sont transmises par modulation des ondes radio-électriques. La méthode d'accès à la couche physique est proche de celle d'Ethernet. Il existe plusieurs normes qui évoluent en permanence :

IEEE 802.11a émet sur une fréquence de 5 Ghz. Cette bande de fréquence ne subit que peu d'interférences. Le débit théorique est de 54 Mbit/s. Mais les distances de connexion sont moins grandes, le matériel coûteux et la legislation française ne permet pas l'utilisation de cette fréquence en extérieur.

IEEE 802.11b émet sur une fréquence de 2.4 GHz (qui peut subir des perturbations). Le débit théorique n'est que de 11 Mbit/s mais avec des coûts très abordables et des distances de connexion intéressantes, cette norme a été adoptée à grande échelle.

IEEE 802.11g émet également sur une fréquence de 2.4 GHz mais avec un débit théorique de 54 Mbit/s. Cette norme est de plus en plus utilisée car elle peut être compatible avec IEEE 802.11b

[12] http://www.ieee802.org/5/www8025org/
[13] http://wi-fi.com/
[14] http://grouper.ieee.org/groups/802/11/

3.1.5 Bluetooth

Le Bluetooth [15] correspond à la norme 802.15 adaptée pour les PAN. L'objectif principal du Bluetooth est de remplacer les ports séries, parallèles et USB. Les données sont transmises par ondes radio-électriques sur une fréquence 2.4 GHz (similaire à ceraines normes de WI-FI). En comparaison avec le WI-FI, la technologie Bluetooth est de plus faible portée (de l'ordre de quelques mètres), de plus faible débit (de l'ordre d'un Mbps), moins répandue mais consomme beaucoup moins (environ 10 fois moins).

3.1.6 SLIP/PPP

Il est possible de se connecter par une ligne téléphonique au travers d'un modem (modulateur/démodulateur), un appareil qui convertit les données numériques en signaux analogiques. Par le réseau téléphonique, seules deux machines peuvent communiquer, comme le RTC (Réseau Téléphonique Commuté).
On parle alors de liaison "point à point". Étant données ces contraintes et et le débit plutôt faible par rapport à une liaison Ethernet, des protocoles "modems" ont été mis au point.

SLIP (Serial Line Internet Protocol) est un protocole très simple dont le principe est de transmettre une trame composée de données et du code ASCII 192 (qui signifie END).

PPP (Point to Point Protocol) est un protocole plus élaboré qui comprend :
- encapsulation de paquets (principalement des datagrammes)
- contrôle de la liaison (test et configuration) par LCP (Link Control Protocol)
- contrôle de réseau NCP (Network Control Protocol) : IPCP (IP Control Protocol)

Une connexion PPP se déroule par étapes, notamment :
- Un paquet LCP envoyé lors de la connexion
- En cas de demande d'authentification, un paquet correspondant à une authentification sera envoyé : PAP (Password Authentification Protocol), CHAP (Challenge Handshake Authentification Protocol), Kerberos, etc.
- Si le besoin s'en fait ressentir, une négociation grâce à IPCP est entreprise pour obtenir une adresse IP
- La connexion est maintenant établie : des données sont échangées ainsi que des informations de contrôle grâce à NCP. Les datagrammes IP sont encapsulés et traduits grâce à un modem pour transiter sur le support physique (souvent en signaux analogiques). La connexion est "point-à-point", les paquets sont donc échangés exclusivement avec un hôte distant qui se retraduit souvent les données analogiques en données numériques puis route les paquets.
- Pour la déconnexion, un paquet LCP est envoyé

[15] http://www.bluetooth.com/

Tout ceci est transcrit et normé dans la *RFC 1661*

3.1.7 HDLC

HDLC (High Level Data Link Control) est un protocole synchrone développé par l'ISO (nternational Organization for Standarization)

| fanion (8) | adresse (8) | contrôle (8) | données | fanion (8) |

fanion : délimiteur de trame pour la synchronisation = 01111110
Si les données contiennent des bits identiques au fanion, il existe un procédé pour les différencier du fanion (un 0 est automatiquement ajouté après cinq 1 successifs)

adresse : il s'agit de l'adresse du destinataire, mais elle n'est pas utilisée en liaison point à point.

contrôle : FCS (Frame Check Sequence) permet la détection des erreurs

3.1.8 RNIS

RNIS [16] signifie Réseau Numérique à Intégration de Services - en anglais ISDN (Integrated Services Digital Network).

RNIS utilise une connexion numérique d'une extrémité à l'autre en utilisant des canaux de type B (transport de données à 64 Kbit/s) et de type D (signalisation des communications). Cela permet donc une séparation fonctionnelle entre la signalisation et le transport, des débits garantis pour le transfert de données (fax, réseau) et une intégration de multiples services (signal d'appel, rappel automatique, renvoi d'appel, etc.) d'où RNIS tire son nom.

En France, le réseau RNIS de France Télécom s'appelle Numéris. En pratique pour se connecter, il faut un adaptateur TNA (Terminal Numérique d'Abonné) qui permet d'obtenir un débit garanti de 64 Kbit/s (ou 128 en utilisant 2 canaux de type B).

3.1.9 ATM

ATM (Asynchronous Transfer Mode) est un protocole asynchrone permettant d'envoyer simultanément voix et données. Les paquets des réseaux ATM circulent sous forme de "cellules" de 53 octets (dont 5 octets d'entête) comprenant notamment des informations sur la qualité de service.
L'ATM peut théoriquement couvrir jusqu'aux niveaux 1 à 5 du modèle OSI mais il est parfois utilisé comme simple protocole de niveau 2 notamment pour l'ADSL

[16] http://www.wellx.com/fr/prest/rnis.htm

avec PPPoA (PPP over ATM).

3.1.10 Matériel

Carte réseau :
Les signaux captés par une carte réseau sont interprétés par le système d'exploitation au moyen d'un pilote adapté au modèle de la carte. Le pilote est donc un programme (souvent écrit en langage C) qui implémente des fonctions traduisant les signaux reçus sous une forme standard pour le système d'exploitation et vice-versa.
Les cartes réseaux les plus répandues dans le grand public sont :

- la carte ethernet qui possède une prise de type RJ-45 et est connectée sur un contrôleur de la carte mère d'un système (PCI, ISA, USB, PCMCIA, etc.)
- la carte sans fil (Wi-Fi ou Bluetooth) qui possède une antenne pour émettre et recevoir des ondes radios.

Répéteur :
Un répéteur [17] est un équipement qui permet de regénérer un signal. Il agit simplement sur la couche 1 du modèle d'OSI c'est-à-dire au niveau physique. En effet, les signaux se propageant sur une ligne de transmission subissent des affaiblissements notamment par rapport à la distance. Ainsi au-delà de quelques centaines de mètres pour l'ethernet, le signal ne peut plus atteindre l'hôte distant sans un équipement répéteur. Le rôle du répéteur est donc de renforcer les signaux électriques pour leur permettre d'aller plus loin. On ne peut généralement pas brancher plus de 4 répéteurs en cascade.

Convertisseur :
Un convertisseur est un répéteur permettant d'avoir deux types de câblage de part et d'autre (BNC, RJ-45, fibre, etc.)

Concentrateur :
Un concentrateur est un équipement qui concentre le trafic de plusieurs hôtes et régénère les signaux (on l'apelle aussi répéteur multiports). Comme un répéteur, il n'agit que sur la couche 1 du modèle OSI. Étant donné que le contrateur permet d'avoir une topologie en étoile, un concentrateur est souvent appelé hub (moyeu de roue, en anglais).

Pont :
Un pont est un équipement qui permet de relier deux segments de réseaux utilisant le même protocole réseau. Il agit sur les couches 1 et 2 du modèle OSI. Il permet de relier des réseaux avec des supports physiques différents (convertisseur) et des topologies différentes (un pont est souvent utilisé pour relier des sous-anneaux à un anneau central). Travaillant au niveau 2, un pont ne transfère sur un segment que les paquets destinés aux noeuds situés

[17] http://www.commentcamarche.net/lan/repeteurs.php3

sur ce segment (il transfère toujours les paquets Broadcast et Multicast) en se contruisant une table d'appartenance à partir des adresses sources (ainsi des hôtes n'émettant jamais ne seront jamais dans la table).
Concrètement, cela permet de désengorger le trafic entre deux segments de réseaux ou de prolonger la distance entre deux points (on contourne la limite des 4 répéteurs en cascade).
Un pont a donc des capacités plus évoluées qu'un concentrateur, car il filtre le trafic physique de manière transparente.

Switch de niveau 2 :
Un switch de niveau 2 [18] est un équipement de type concentrateur agissant également sur la couche 2 du modèle d'OSI. D'un point de vue utilisateur, on peut le considérer comme un "hub intelligent" car comme le hub, il est constitué de plusieurs ports mais il ajoute en plus des fonctions de filtres car il ne transfère vers un segment (ou un hôte) que les paquets destinés aux hôtes sur ce segment (comme le pont).
Il est en réalité constitué d'un ensemble de hubs et de ponts. Un switch divise donc le segment Ethernet en plusieurs brins (il leur permet donc au d'émettre en même temps) et construit une table de correspondance entre ses ports et des adresses MAC. Il construit cette table à partir des adresses MAC sources des paquets qu'il voit passer. Les paquets contenant une adresse MAC de destination inconnue ou de Broadcast sont propagés sur tous les brins.

Note : En général, le trafic broadcast ne devrait pas excéder 20% du trafic total.

3.2 Couche réseau

La couche réseau gère l'acheminement des données sur le réseau global en s'occupant notamment du routage. Le protocole qui s'est incontestablement imposé notamment sur Internet est IP (Internet Protocol).

Il existe diverses couches réseau :

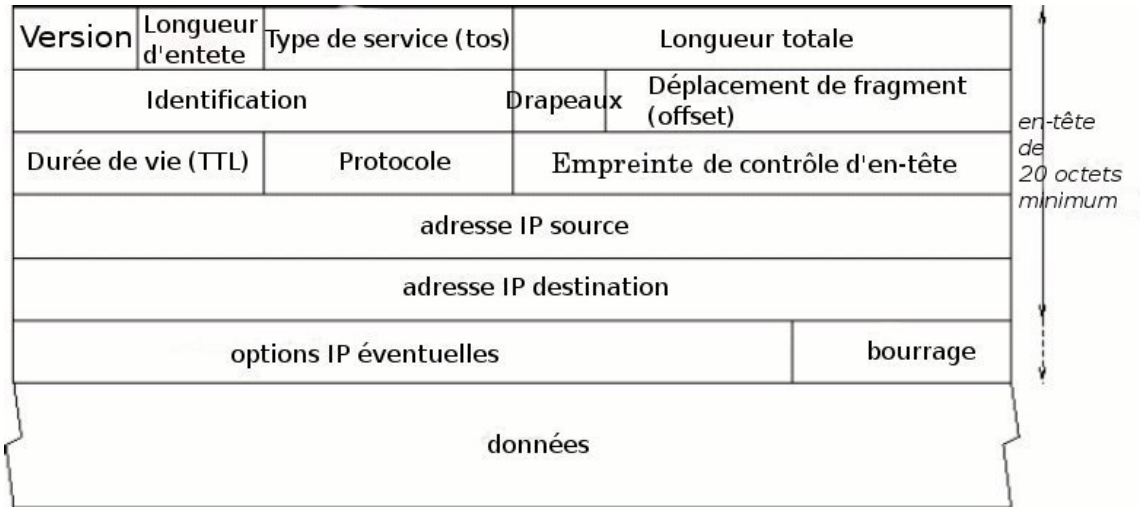
- *IP* voir 3.2.1
- *IPX* voir 3.2.4
- *X.25* voir 3.2.3
- *AppleTalk* voir 3.2.2
- *CLNP* voir 3.2.5
- *ICMP* voir 3.2.7
- *IGMP* voir 3.2.8
- *ARP* voir 3.2.6
- *OSPF* voir 4.0.5
- *RIP* voir 4.0.5

[18] http://sawww.epfl.ch/SIC/SA/publications/FI98/fi-6-98/6-98-page4.html

3.2.1 Le protocole IP

IP (Internet Protocol, RFC 791) assure un service non connnecté (les datagrammes sont indépendants entre eux) et non fiable (aucune garantie de livraison : best effort delivrery) de datagrammes IP.

Détaillons l'entête d'un datagramme IP :



Version (4) : la version courante est 4 (IPv4). Les logiciels vérifient ce paramètre, cela permet de tester des nouveaux protocoles sans interférer avec l'existant (IPv6).

Longueur d'entête (IHL : Internet Header Length) (4) : la longueur d'entête en multiple de 4 octets. La longueur courante est de 20 octets mais une longueur supérieure peut être observé en cas d'options supplémentaires.

Type de service (TOS : Type Of Service) (8) :
| DSCP (Differenciated Services CodePoint) (6) | ECN (2) |

DSCP : | Priorité (3) | délai (1) | débit (1) | fiabilité (1) |

– *Priorité* :
111 : Network Control
110 : Internetwork Control
101 : CRITIC/ECP
100 : Flash Override
011 : Flash
010 : Immediate
001 : Priority
000 : Routine

– *Délai* :
  0 : normal
  1 : faible

– *Débit* :
  0 : normal
  1 : faible

– *fiabilité* :
  0 : normale
  1 : elevée

  Ces 3 bits ne sont pas strictement nécessaires mais améliorent la qualité du routage.

Exemples :
```
telnet/SSH : 100
FTP : 100 (contrôle) et 010 (données)
```

ECN (Explicit Congestion Notification) :
Le 8e bit du TOS mais aussi le dernier bit (DSCP) sont utilisés pour ECN.
Ces bits particuliers sont destinés à transmettre des informations sur l'état du trafic (à utiliser avec précaution !)

– **ECT** (ECN-Capable Tansport) : indique la capacité à utiliser ECN (0 : non, 1 : oui)

– **ECN-CE** : la valeur par défaut est 0. Un routeur peut changer cette valeur de 0 à 1 pour indiquer aux extrémités un trafic encombré. En aucun cas, un routeur ne doit changer ce bit si il est à 1.

  longueur totale (Total Lenght) (16) : Taille en octets donc longueur totale limitée à 2^{16} -1 = 65536 octets.

  identification (16) : Numéro attribué à chaque fragment afin de permettre leur réassemblage.

  drapeau (3) :
  | Reserved Bit (1) | DF (Don't Fragment) (1) | MF (More Fragment) (1) |
  Le premier bit ne sert pas.
  Le bit DF indique si le datagramme peut être fragmenté. Si le bit n'est pas nul, le datagramme ne peut pas être fragmenté. Dans ce cas, si un routeur ne peut l'acheminer sans le fragmenter, il sera rejeté avec un message d'erreur.
  Le bit MF indique s'il s'agit d'un fragment de données. Si le bit est nul, il s'agit du dernier fragment ou bien le datagramme n'est pas fragmenté.

décalage (13) : Ce champ indique à quel datagramme appartient ce fragment. Les fragments sont comptés en unités de 8 octets. Ce format permet 8 192 fragments de 8 octets pour un total 65 536 octets.

durée de vie (TTL : Time To Live) (8) : Numéro indiquant le nombre maximal de routeurs que peut traverser le datagramme. Initialisé par la station émettrice, cette valeur est décrémentée par chaque routeur. Si un routeur reçoit un datagramme avec une valeur TTL nulle, il le détruit et envoie un message ICMP code 11 à l'émetteur.

protocole (protocol) (8) : spécifie quel protocole [19] de plus haut niveau a créé ce datagramme. ICMP = 0x01, TCP = Ox06, etc.

controle entête (header checksum) (16) : contrôle d'intégrité calculé à partir des champs invariants de l'entête.

adresse IP source (32) : adresse du système source

adresse IP de destination (32) : adresse du sytème de destination

options + bits de bourrage (32) : options très peu utilisées spécifiant des paramètres de routage, sécurité, etc.

données : on peut déduire où commencent les données grâce à la longueur d'entête.

Fragmentation, MTU (Maximum Transfert Unit) La longueur d'un datagramme est indiqué dans son entête par un champ de 16 bits. Sa longueur maximale théorique est donc de $2^{16} - 1 = 65535$ octets. Mais en pratique, un réseau possède une taille maximale pour les trames qui circulent appelée MTU (Maximum Transfert Unit).

```
MTU Ethernet : 1500
MTU PPP : 1492
MTU FDDI : 4470
MTU Token Ring 4 Mbit : 4464
MTU Token Ring 16 Mbit : 17914
```

Si le MTU [20] traversé est assez élevé pour encapsuler un datagramme, il n'y aura pas de fragmentation. Mais si le datagramme est trop long pour être contenu dans une seule trame, il va être fragmenté. La taille du fragment choisie est la plus grande possible (elle doit être un multiple de 8 octets). Ces fragments transitent ensuite de manière indépendante et ne sont réassemblés qu'à l'arrivée : le destinataire recevant un premier fragment décremente son TTL en attente des autres fragments.

[19] http://www.iana.org/assignments/protocol-numbers
[20] http://christian.caleca.free.fr/pppoe/mtu_mss_etc.htm

Exemple : A envoie à B un datagramme de 1300 octets.

```
A - MTU = 1500 - R1 - MTU = 620 - R2 - MTU = 1500 - B
```

Le datagramme initial est alors découpé en trois fragments (600 octets, 600 octets, 100 octets). Les champs de l'entête IP renseignant sur la fragmentation éventuelle d'un datagramme sont le drapeau et le décalage.

Note : PPPoE (PPP Over Ethernet) limite les trames IP à 1492 octets ce qui peut poser des problèmes lorsque le paquet ne doit pas être fragmenté (champ DF de l'entête IP). Certains routeurs choisissant de ne pas transmettre de trafic ICMP, le paquet ICMP signalant cette erreur peut ne pas être reçu ! Des méthodes pour contourner ce problème existent.

– **Adressage IPv4**
Chaque ordinateur du réseau Internet dispose d'une adresse IP.
Les adresses IPv4 sont codées sur 32 bits, souvent notés en décimal pointé :

8 bits . 8 bits . 8 bits . 8 bits

Exemple : `216.239.57.104`

Au départ, Internet a été divisé en plusieurs classes.

Seules les trois premières classes ont une signification
IP = HostID + NetID :

```
A     0.0.0.0 à 127.255.255.255 0...
B   128.0.0.0 à 191.255.255.255 10...
C   192.0.0.0 à 223.255.255.255 110...
D   224.0.0.0 à 239.255.255.255 1110... -> multicast
E   240.0.0.0 à 247.255.255.255 11110... -> RFU (Reserved For Use)
```

Adresses particulières :
Boucle locale : 127.x.y.z
Classes privées :
10.0.0.0 → 10.255.255.255
172.16.0.0. → 172.31.255.255
192.168.0.0 → 192.168.255.255
Auto-configuration :→ 169.254.x.y
Le développement d'Internet dépassant les prévisions, il a fallu trouver des astuces pour gérer plus efficacement l'adressage. C'est de ce constat que sont nées les notions de sous-réseau et de masque.

– **Le masque :**

NetID = IP & Masque HostID = IP - NetID

Quelques exemples de masques :
masque /27 : 5 bits disponibles → 2^5 = 32 adresses possibles
masque /24 : 8 bits disponibles → 256 adresses possibles
masque /20 : 12 bits disponible → 4 096 adresses possibles
masque /16 : 16 bits possibles → 65 536 adresses disponibles
masque /13 : 19 bits possibles → 524 288 adresses disponibles

Parmi les adresses disponibles, il faut retirer les adresses HostID et Broadcast

Une classe A doit avoir un masque avec un minimum de 8 bits non nuls
Une classe B doit avoir un masque avec un minimum de 16 bits non nuls
Une classe C doit avoir un masque avec un minimum de 24 bits non nuls

– **Le Broadcast**
Il s'agit de l'adresse la plus haute dans le réseau :

Wildcard = inverse(Masque)
Broadcast = NetID | Wildcard

Exemples :
L'adresse Broadcast du réseau 192.168.63.0/24 est 192.168.63.255
L'adresse Broadcast du réseau 192.168.63.96/27 est 192.168.63.127
L'adresse Broadcast générale est 255.255.255.255

Possibilités de configuration :

1. **automatique** Si un serveur DHCP est présent sur le réseau, la configuration est automatique.
2. **manuelle** On préférera néanmoins une configuration statique pour des raisons de sécurité. La configuration réseau peut se définir dans un fichier (par exemple dans `/etc/network/interfaces` sous Debian GNU/Linux). Voici un exemple :

```
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
address 192.168.12.67
netmask 255.255.255.0
network 192.168.12.0
broadcast 192.168.12.255
```

```
gateway 192.168.12.254
```

3. **auto-configuration** Certains systèmes, notamment Microsoft, utilisent une méthode d'auto-configuration en attribuant des adresses particulières (169.254.x.y) aux interfaces non configurées.

– **Le Multicast :** En pratique, l'émetteur communique avec un groupe d'ordinateurs identifiés par une adresse IP de classe D. Le groupe d'ordinateurs est dynamique (chaque hôte peut s'abonner ou se désabonner à tout moment), illimité (pas de restrictions d'adresses à priori) et peut recevoir des données d'hôtes non membres. On peut comparer le Multicast à l'abonnement à une liste de diffusion ouverte.

Classe D : adresses de 224.0.0.0 à 239.255.255.255

Quelques adresses Multicast particulières :

– *LNCB (Local Network Control Block)* sont les adresses 224.0.0.0/24 qui sont réservées aux protocoles de routage ou de découverte de topologies.

```
224.0.0.1 : groupe all-hosts
224.0.0.1 : groupe all-routers
224.0.0.4 : groupes routeurs DVMRP
224.0.0.5 : groupes routeurs OSPF
224.0.0.13 : groupe routeurs PIM
```

Voici quelques mots de vocabulaire touchant au Multicast :

– *MOSPF (Multicast Open Shortest Path First protocol)*
Extension du protocole OSPF (décrite dans la RFC 1584), il permet la création d'arbres de distribution Multicast. Il est réputé plus adapté dans le cas d'hôtes peu nombreux et distants.

– *PIM (Protocol Independent Multicast)*
Issu du groupe IDMR (Inter-Domain Multicast Routing), PIM se veut plus efficace que DVMRP ou MOSPF pour un large déploiement.

– *MBone (Multicast backBone)*
C'est un réseau expérimental de diffusion multipoint. Ce réseau est principalement constitué de machines exécutant le logiciel *mrouted* reliées par des tunnels. Le *FMBone* est la partie française du *MBone*. Les applications sont principalement la diffusion de contenus multimédias (images de satellites en temps réel, retransmissions d'évènements de la NASA, etc.) et le travail collaboratif (visioconférences, documents partagés).

3.2.2 AppleTalk

Appletalk est un protocole de communication d'Apple. Il fut employé par les ordinateurs Macintosh jusqu'à la fin des années 1990. Ces derniers utilisent désormais le protocole Ethernet.

3.2.3 X.25

X.25 est un protocole standard définit par l'ITU-T (International Telecommunication Union-Telecommunication Standardization Sector). Il définit la façon dont les communications de type WAN (Wide Area Network) sont établies. X.25 est notamment utilisé dans les "packet switching network" (fournisseur d'accès X.25) ou dans des entreprises de téléphonie. X.25 définit en réalité 3 couches du modèle OSI : la couche physique, la couche données et la couche réseau.

3.2.4 IPX

IPX (Internetwork Packet Exchange) est un protocole réseau dérivé des protocoles XNS (Xerox Network System) de Xerox. Il est surtout utilisé par le système d'exploitation de réseau Netware de Novell. Il utilise le mode non connecté. Utilisé avec le protocole transport SPX (Sequenced Packet Exchange), il forme une pile IPX/SPX comparable à TCP/IP permettant aux serveurs Novell de communiquer à travers un LAN ou un WAN.

3.2.5 CLNP

CLNP (ConnectionLess Network layer Protocol) est un protocole réseau avec des mécanismes proches de ceux d'IP. Il est défini par la norme ISO 8473
La couche de transport liée à CLNP est CNLS (ConnectionLess Network Service).
CLNP est encore utilisé, même pour du matériel récent, pour les transmissions des réseaux de télécommunications.

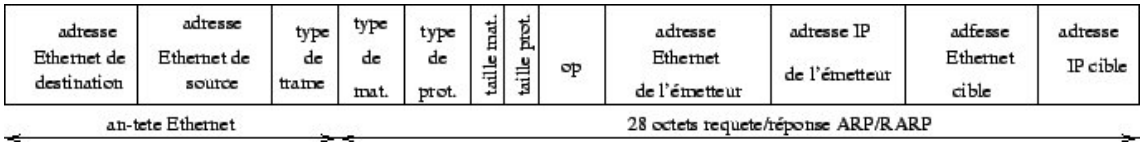
3.2.6 ARP

Le protocole ARP (Address Resolution Protocol) permet de traduire les adresses IP en adresses MAC.

Lors d'un échange de données sur un réseau IP sur de l'Ethernet, la couche réseau de l'émetteur est "sensée" connaître l'adresse IP du destinataire, mais pas son adresse MAC. Pour connaître son adresse MAC, le protocole ARP sera utilisé.

Le principe est que l'émetteur va émettre une trame Ethernet en broadcast physique pour demander "à qui appartient telle adresse IP ?" et que la machine ayant l'adresse IP lui répondra "je suis là, j'ai telle adresse MAC". L'émetteur connait maintenant son adresse MAC pourra lui envoyer sa trame Ethernet.
Afin d'éviter de trop nombreuses requêtes en broadcast physique, ARP utilise une

mémoire-cache. Ainsi, si l'association adresse IP-adresse MAC est dans sa mémoire-cache, il l'utilise et ne fera pas de requête. Au bout d'un certain temps (souvent quelques secondes), les valeurs non utilisées de sa mémoire-cache deviennent "à confirmer" (le prochain échange confirmera la valeur). Au bout d'un temps plus long (souvent quelques minutes) les valeurs toujours non utilisées seront effacées de la mémoire-cache.



Exemple :

```
# tcpdump -i eth0 arp -xx -n

17:54:54.322469 arp who-has 192.168.4.66 tell 192.168.4.121
        0x0000:  ffff ffff ffff 00c0 9f23 c21b 0806 0001
        0x0010:  0800 0604 0001 00c0 9f23 c21b c0a8 0479
        0x0020:  0000 0000 0000 c0a8 0442
17:54:54.322677 arp reply 192.168.4.66 is-at 00:0b:cd:85:6b:5b
        0x0000:  00c0 9f23 c21b 000b cd85 6b5b 0806 0001
        0x0010:  0800 0604 0002 000b cd85 6b5b c0a8 0442
        0x0020:  00c0 9f23 c21b c0a8 0479 ffff ffff ffff
        0x0030:  ffff ffff ffff ffff ffff ffff
```

Exemple de manipulation du cache ARP :

```
# ip neigh show
192.168.1.4 dev eth1 lladdr 00:e0:7d:d2:00:11 nud delay
192.168.0.21 dev eth2 lladdr 00:e0:4c:03:36:4e nud reachable
192.168.0.121 dev eth2 lladdr 00:09:5b:4b:3f:3b nud stale

# ip neigh delete 192.168.1.4 dev eth1
# ip neigh show
192.168.1.4 dev eth1  nud failed
192.168.0.21 dev eth2 lladdr 00:e0:4c:03:36:4e nud reachable
192.168.0.121 dev eth2 lladdr 00:09:5b:4b:3f:3b nud reachable

# ip neigh flush dev eth2
# ip neigh show
192.168.1.4 dev eth1  nud failed
192.168.0.21 dev eth2  nud failed
192.168.0.121 dev eth2  nud failed
```

États différents :

- `nud permanent` : valide définitivement (seul l'admin. peut la modifier)
- `nud noarp` : valide jusqu'à ce qu'expire le durée de vie

– `nud reachable` : valide jusqu'à ce qu'expire le *reachability timeout*

– `nude stale` : valide mais incertain : à revalider à la prochaine connexion

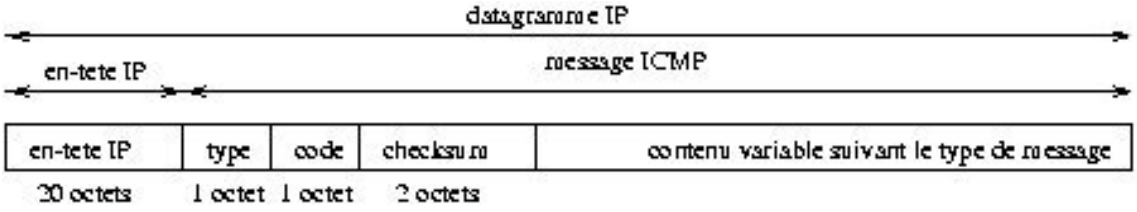
– `nud failed` : invalide

RARP : Le protocole RARP (Reverse Address resolution protocol) joue le rôle inverse du protocole ARP : il permet de déterminer l'adresse IP en fonction d'une adresse MAC. Ce protocole n'est utilisé que dans certains cas précis (démarrage de terminaux X, etc.) et seuls les serveurs RARP vont répondre à ce type de requête.

3.2.7 ICMP :

ICMP (Internet Control Message Protocol) est considéré comme un protocole de niveau 3 (couche réseau) bien qu'il se trouve au-dessus de protocole IP. ICMP permet de contrôler les erreurs de transmissions entre les machines.

Note : un message ICMP étant bien sûr également susceptible de souffrir d'erreurs de transmission, la règle est qu'aucun message ICMP ne doit être délivré pour signaler une erreur relative à un message ICMP. Car sinon, on risquerait d'avoir une "avalanche" de messages ICMP !



Le champ type peut prendre 15 valeurs différentes spécifiant de quelle nature est le message envoyé. Pour certains types, le champ code sert à préciser encore plus le contexte d'émission du message. Le checksum est une somme de contrôle de tout le message ICMP calculée comme dans le cas de l'en-tête d'un datagramme IP 3.2.1. Le détail des différentes catégories de messages est donné dans la liste ci-dessous où chaque alinéa commence par le couple (type,code) de la catégorie décrite.

– type/code :
  (0,0) ou (8,0) Demande (type 8) ou réponse (type 0) d'écho dans le cadre de la commande ping

  (3,0-13) Compte-rendu de la destination inaccessible fournie quand un routeur ne peut délivrer un datagramme. Le routeur génère et envoie ce message ICMP à l'expéditeur de ce datagramme. Il obtient l'adresse de cet expéditeur en l'extrayant de l'en-tête du datagramme, il insère dans les données du message ICMP toute l'en-tête ainsi que les 8 premiers octets du

datagramme en cause. Une liste non exhaustive des différents codes d'erreurs possibles est :
0 = réseau inaccessible
1 = hôte inaccessible
2 = protocole non disponible
3 = port non accessible
4 = fragmentation nécessaire mais bit de non fragmentation positionné à 1
5 = Échec de routage de source.
6 = Réseau de destination inconnu.
7 = Machine destinataire inconnue.
8 = Machine source isolée (obsolète)
9 = Communication avec le réseau de destination administrativement interdite.
10 = Communication avec la machine de destination administrativement interdite.
11 = Réseau inaccessible pour ce type de service.
12 = Machine inaccessible pour ce type de service.
13 = Communication administrativement interdite par filtrage.

(4,0) Demande de limitation de production pour éviter la congestion du routeur qui envoie ce message.
(5,0-3) Demande de modification de route expédiée lorsqu'un routeur détecte qu'un ordinateur utilise une route non optimale, ce qui peut arriver lorsqu'un ordinateur est ajouté au réseau avec une table de routage minimale. Le message ICMP généré contient l'adresse IP du routeur à rajouter dans la table de routage de l'ordinateur. Les différents codes possibles ci-après expliquent le type de redirection à opérer par l'ordinateur.
0 = Redirection de datagramme sur la base du réseau
1 = Redirection de datagramme sur la base de l'adresse d'hôte
2 = Redirection de datagramme sur la base du réseau et du Type de Service
3 = Redirection de datagramme sur la base de l'hôte et du Type de Service

(9,0) Avertissement de routeur expédié par un routeur.
(10,0) Sollicitation de routeur diffusée par une machine pour initialiser sa table de routage.
(11,0) TTL détecté à 0 pendant le transit du datagramme IP, lorsqu'il y a une route circulaire ou lors de l'utilisation de la commande traceroute
(11,1) TTL détecté à 0 pendant le réassemblage d'un datagramme.
(12,0) Mauvaise en-tête IP.
(12,1) Option requise manquante.
(13-14,0) Requête (13) ou réponse (14) timestamp, d'estampillage horaire.
(15,0) et (16,0) devenues obsolètes.
(17-18,0) Requête (17) ou réponse (18) de masque de sous-réseau.

– Contrôle : empreinte permettant de vérifier l'intégrité du paquet

– identificateur : permet d'associer la demande et sa réponse (l'envoi de 5 paquets ECHO_REQUEST avec la commande `ping -c 5` aura par exemple le même identificateur)

– numéro : c'est le numéro de séquence associant la demande et sa réponse (pour un ping avec le même identificateur, le numéro s'incrémentera avec chaque paquet ECHO_REQUEST)

– données :
Données de remplissage contenant souvent des octets tels que :
```
00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18
19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31
32 33 34 35 36 37
```
ICMP est notamment exploité avec la commande `ping`.

Quelques options de la commande `ping` sous Unix :

`-c n` : envoie n paquets ECHO_REQUEST
`-i n` : envoie les paquets avec un intervalle de n secondes
`-f` : envoie les paquets le plus rapidement possible (flood)

Note : la commande `ping` demande des droits CAP_NET_RAWIO pour être exécutée. Ainsi, sous Linux, elle possède souvent les droits set-uid root.

3.2.8 IGMP

IGMP (Internet Group Management Protocol [21]) est un protocole lié au Multicast. IGMP transporte les messages de gestion Multicast (souscription à un groupe Multicast, etc.) Il existe plusieurs versions d'IGMP : la première intégrée aux spécifications du Multicast (RFC 1112), la deuxième qui met en place un certain nombre de fonctionnalités supplémentaires (RFC 2236) et la troisième représente un véritable tournant par rapport aux versions précédentes en apportant de nombreuses évolutions (RFC 3376).

3.2.9 Protocoles de routage

Il existe de nombreux protocoles de routage : RIP, OSPF, BGP, EGP, etc.
Voir 4.0.5

3.2.10 Matériel

Le matériel qui décapsule la couche réseau afin d'interpréter les entêtes est appelé un routeur. Il propose donc des fonctionnalités beaucoup plus avancées que le matériel de niveau 2 de la couche d'OSI. Un routeur participe activement à l'acheminement des paquets sur le réseau en utilisant sa table de routage.

[21] http://www.routage.org/igmp1.html

Certains commutateurs peuvent également regarder la couche réseau afin d'offrir des fonctionnalités avancées à l'administrateur. On appelle généralement un commutateur de ce type : "switch de niveau 3".

Pour récapituler le besoin de matériel de ce type, nous pouvons faire un parallèle avec les chemins de fer :
Imaginons, qu'au début une seule voie reliait toutes les gares : un train ne pouvait démarrer qu'à la condition qu'aucun train ne cicurle déjà (parrallèle avec un bus Ethernet).
Afin d'améliorer le trafic, une grue a été construite coupant la ligne en deux afin de faire transiter la marchandise entre les deux portions de voies : ainsi deux trains pouvait circuler en même temps (parallèle avec le routeur). Mais sur le même segment de voie, seul un train peut circuler : on choisit une gare centrale et on relie chacune des gares secondaires à cette gare d'aiguillage (réseau switché). L'aiguillage se fait en fonction de la pancarte de destination du train. Le trafic est désormais fluide sur tout le réseau. Le seul point de "ralentissement" potentiel est la grue. Pour résoudre ce dernier problème, la grue est également remplacée par un système d'aiguillage mais les décision d'aiguillage sont plus complexes : l'aiguillage ne se fait plus en fonction des pancartes de destination du train car elles indiquent toutes "vers la grue", mais l'aiguillage se fait alors en fonction des étiquettes de destination des marchandises (routing switching).

3.3 Couche transport

La couche de transport est chargée de contrôler de façon plus ou moins poussée l'acheminement des données. Les protocoles de transport les plus utilisés pour contrôler IPv4 sont les protocoles TCP et UDP.

Au niveau de la couche transport, il existe la notion de ports. L'organisme IANA (Internet Assigned Numbers authority) contrôle les numéros de port :
- les numéros de ports inférieurs à 1024 sont réservés à des services connus : ils sont considérés comme des "ports privilégiés" et ne doivent pas être utilisés.
- les numéros de port compris entre 1024 et 49151 sont des numéros de "ports enregistrés"
- les ports entre 49151 et 65535 sont des numéros de "ports privés" utilisable pour tout usage.

Sur un système UNIX, on peut retrouver les numéros de port dans le fichier `/etc/services`

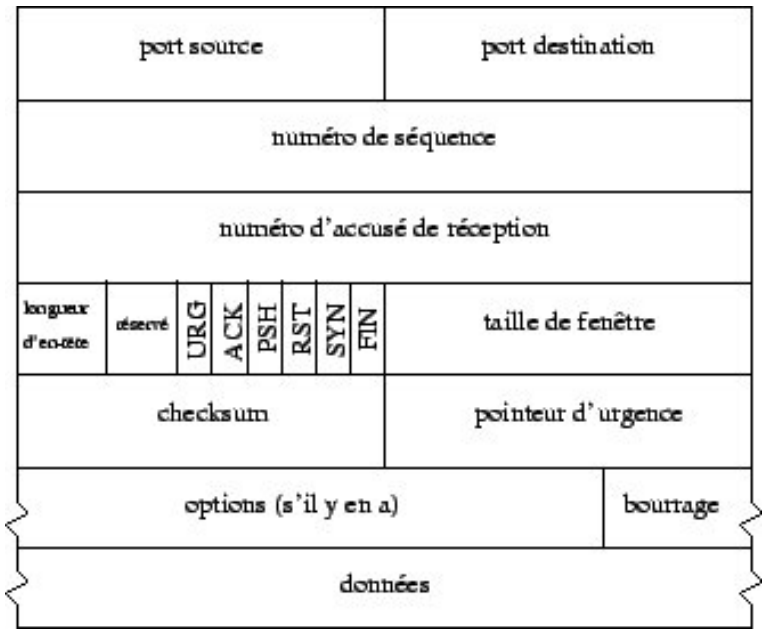
Note : le terme de "socket" est utilisé pour l'association d'une adresse IP, d'un protocole de transport et d'un numéro de port

3.3.1 TCP (Transmission Control Protocol)

TCP est un protocole qui assure un mode connecté, c'est-à-dire un mode qui impose un échange particulier au début et à la fin de toute connexion et où les

segments TCP sont classés avec un numéro de séquence et un numéro de confirmation.

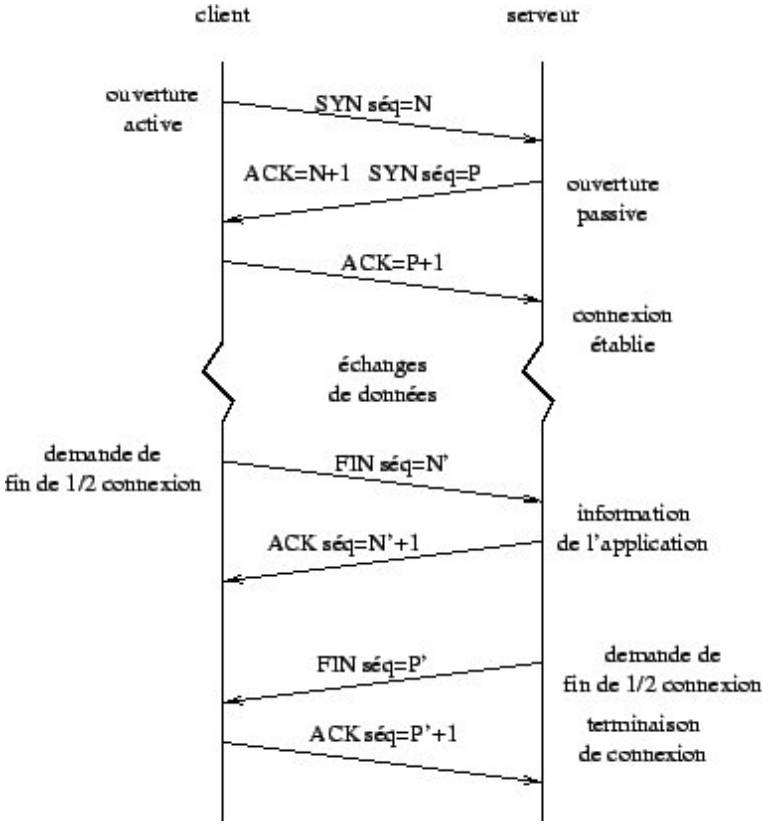
– *Entête TCP :*



– **port source (16)** : numéro du port de la machine d'où est émis le paquet
– **port de destination (16)** : numéro du port de la machine destinataire du paquet
– **numéro de séquence (sequence number) (32)** : numéro utile pour échange SYN/ACK car il donne la position du segment dans le flux de données envoyé
– **numéro de confirmation (acknowledgement number) (32)** : numéro utile pour échange SYN/ACK
– **longueur d'entête (header lenght) (4)** : la longueur d'entête en multiple de 4 octets. La longueur courante est de 20 octets mais une longueur supérieure peut être observée (notamment lors de l'initialisation d'une session)
– **réservé (6)** : il s'agit de bits destinés à un usage futur, notamment CWR (Congestion Window Reduced) ou ECN-Echo. Ces bits doivent être nuls pour le moment.
– **drapeau (flag) (6)** :
  le drapeau permet de spécifier des spécificités du paquet :
  – *Urgent :* données "urgentes"
  – *Ack :* confirmation
  – *Push :* envoi des données à la couche supérieure
  – *Reset :* réinitialisation de session
  – *Syn :* début de session
  – *Fin :* fin de session
– **taille de fenêtre (window size) (16)** : ce champ sert au contrôle du flux. L'émetteur ajuste le nombre d'octets qu'il envoie selon la valeur "windows size" reçue dans les accusés de réception du destinataire.

– **checksum (16)** : somme de contrôle calculée par l'émetteur et vérifiée par le récepteur.
– **pointeur d'urgence (8)** : associé avec le bit Urgent du drapeau, il permet d'indiquer le dernier octet de données "urgentes" (peu utilisé)
– **option ()** : il y a plusieurs options facultatives. Signalons une option souvent utilisée qui est de spécifier la taille maximale du segment TCP qu'une extrémité de la connexion souhaite recevoir. En réglant cette taille en fonction du MTU, cela permet d'éviter de fragmenter des paquets. L'exemple typique est pour PPPoE (voir ci-dessus).
– **bits de bourrage ()**
– **données applicatives**

Établissement de la session complète (échange SYN/ACK) :



On part de l'hypothèse qu'aucun échange n'a encore eu lieu.

A envoie une demande de session à B :

1. `A -> B SYN=X, pas de ACK` → "Salut, mon numéro de séquence est X"

   Si B reçoit une demande valide d'ouverture de session, B envoie un accusé de réception :

2. `A <- B SEQ=Y et ACK=X+1` → "Mon numéro de confirmation est ton numéro de séquence incrémenté. Mon numéro de séquence est Y"

   Si A reçoit un accusé de réception conforme à sa demande, A confirme la

réception de l'accusé (et considère la session comme établie) :

3. `A -> B SEQ=X+1 et ACK=Y+1` → "Mon numéro de confirmation est ton numéro de séquence incrémenté. Mon numéro de séquence vaut ta confirmation."
   Si B reçoit une confirmation conforme à son accusé de réception, B considère la session comme établie.

   La session est donc établie.

   Envoi de données :
   La session établie, l'application de A envoie des données à B :

4. `A -> B SEQ=X+1 et ACK=Y+1` → "J'utilise les numéros de séquence de confirmation de ma dernière trame. Je t'envoie des données (applicatives)."
   Si B reçoit des données conformes, il envoie un accusé de réception :
5. `A <- B SEQ=Y+1 et ACK=X+1+s` → "Mon numéro de confirmation suit ton numéro de séquence. Mon numéro de séquence vaut ta confirmation."
   Si A reçoit un accusé de réception valide, il sait donc que B a bien reçu ses données. Des échanges de données peuvent donc avoir lieu entre les hôtes A et B.
   Prenons le cas où A est un client et B un serveur. Dans les étapes 5 et 6, L'application cliente de A a envoyé une requête à l'application serveur de B. L'application serveur de B va donc répondre : envoyer des données à A :
6. `A <- B SEQ=Y+1 et ACK=X+1+s` → "J'utilise les numéros de séquence et de confirmation de ma dernière trame. Je t'envoie des données (applicatives)"
   Si A reçoit des données conformes, il envoie un accusé de réception :
7. `A -> B SEQ=X+1+s et ACK=Y+1+s` → "Mon numéro de confirmation suit ton numéro de séquence. Mon numéro de séquence vaut ta confirmation."
   Si B reçoit un accusé de réception valide, il sait donc que A a bien reçu ses données. Prenons le cas où B va maintenant clore la session :
8. `A <- B SEQ=Y+1 et ACK=X+1+s` → "J'utilise les numéros de séquence et de confirmation de ma dernière trame."
   Si A reçoit une demande de B reçoit une demande valide de fermeture de session, A envoie un accusé de réception :
9. `A -> B SEQ=X+1+s et ACK=Y+1+1` → "Mon numéro de confirmation est ton numéro de séquence incrémenté. Mon numéro de séquence vaut ta confirmation."
   Si B reçoit un accusé de réception conforme à sa demande, B confirme la réception de l'accusé (et considère la session comme fermée) :
10. `B -> A SEQ=Y+1+1 ACK=X+1+s+1` → "Mon numéro de confirmation est ton numéro de séquence incrémenté. Mon numéro de séquence vaut ta confirmation."

Si A reçoit une confirmation conforme à son accusé de réception, A considère la session comme fermée.
La session est donc fermée.

- **Principe de l'acquittement retardé**
  Il y a souvent deux types de paquets TCP : les paquets de transfert de masse et les paquets de transfert interactif.
  Les transferts de masse (FTP, HTTP, SMTP, etc.) envoient des paquets de longueur proche du maximum alors que les transferts interactifs (Telnet, SSH, etc.) envoient des paquets de faible taille. En effet, lors d'un shell à distance par exemple, chaque touche tapée doit être interprétée par le système puis l'affichage exportée ce qui génère en théorie un trafic que 4 paquets pour une lettre de tapée. Ainsi, en tapant la commande "uptime" on a plus de 28 (petits) paquets qui transitent. L'acquittement retardé consiste à transporter l'acquittement avec l'affichage par exemple. Ainsi lorsque l'on tape une touche, seul 3 paquets sont échangés.

- **Algorithme de Nagle**
  Malgré le principe de l'acquittement retardé, 21 paquets seront transportés en tapant la commande "uptime". Lorsque le trafic réseau n'est pas chargé, cela ne pose pas de problème (c'est d'ailleurs ce qui se passe dans la pratique) mais lorsque le temps de transfert est non négligeable, une solution est proposée par John Nagle : détecter la "qualité" du réseau à partir de l'option "timstamp" de TCP et lorsque la qualité est mauvaise, accumuler plusieurs informations en un seul paquet. Cet algorithme est transparent car il s'autorégule en fonction de la charge du réseau. Au niveau utilisateur, on observe ce phénomène lorsque sur un shell distant, on voit arriver les lettres tapées par "paquet" et non une par une.

- **Fenêtre glissante**
  La technique de la fenêtre glissante [22] est utilisée par TCP pour contrôler les transferts de masse. Elle permet notamment aux destinataires d'éviter de saturer son buffer (cas d'un client recevant des données d'un gros serveur). L'émetteur utilise donc le champ "window size" émis par le destinataire : cela fixe la taille de la fenêtre et limite l'envoi de paquets non acquittés. Une fois les premiers paquets acquittés, la fenêtre se "glisse".

- **Départ lent**
  En complément de la technique de la fenêtre glissante, TCP utilise un algorithme appelé "slow start" [23]. En effet, tous les paquets autorisés par la fenêtre ne sont pas envoyés en une fois au début. Il envoie tout d'abord un paquet, puis, l'acquittement reçu, il en envoie deux, puis quatre, etc. Durant cette augmentation, si des paquets sont perdus, la fenêtre est réduite afin d'optimiser la retransmission.

[22] http ://www.info.univ-angers.fr/pub/pn/poly/node45.html
[23] http://www.laissus.fr/cours/node11.html

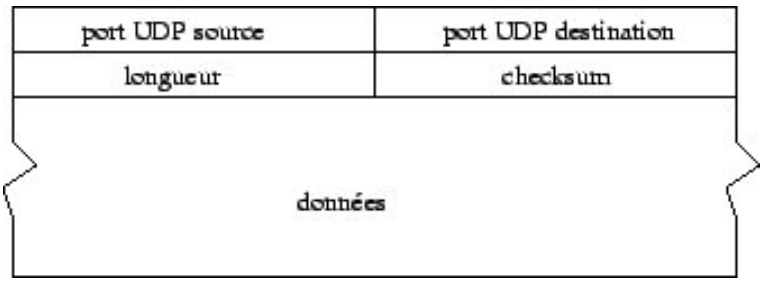
– **Faille TCP (avril 2004) :**
Il est possible de réinitialiser une connexion TCP établie en envoyant des paquets RST et SYN appropriés. Un attaquant peut exploiter cette possibilité mais jusqu'ici on pensait [24] que la probabilité de trouver le numéro de séquence correct était de l'ordre de $1/2^{32}$. Or, dans la pratique, une numéro de séquence à peu près bon (compris dans la "window size") convient.

[24] http://www.k-otik.com/bugtraq/bulletins/142

3.3.2 UDP : (User Datagram Protocol)

UDP est un protocole beaucoup plus léger que TCP. Il s'utilise en mode non connecté

– *Mode non connecté*



- **port source (8) :** numéro du port de la machine d'où est émis le paquet
- **port de destination (8) :** numéro du port de la machine destinataire du paquet
- **taille (16) :** taille totale du paquet (entête et donnée) en octets
- **checksum (16) :** somme de contrôle vérifiée par le récepteur.
- **données applicatives**

Question :
Quel protocole transport va-t-on utiliser pour :
- applications de consultation de données ?
- applications d'envoi/réception de son et/ou image ?
- applications de transfert de fichiers de taille importante ?
- application de discussion texte en temps réel ?
- application de type FTP ?
- application de type DNS ?
- application de type client DHCP ?

3.4 Couche application

La couche application est la couche dite de *haut-niveau* car elle se contente de définir des protocoles applicatifs afin d'utiliser les couches inférieures. Il existe donc de nombreux protocoles : HTTP, FTP, SMTP, DNS, SSH, POP, IMAP, etc. Ces protocoles, au nom souvent familier pour l'utilisateur de base, se basent généralement sur une architecture client/serveur et décrivent des actions à la fois pour les clients et pour les serveurs.

3.4.1 HTTP

Nous pouvons réaliser les actions faites par un navigateur Internet en lançant une connexion TCP vers le port 80 d'un serveur web à l'aide de la commande "tel-

net" :

```
$ telnet google.fr 80
Trying 216.239.39.104...
Connected to google.fr.
Escape character is '~]'.
GET / HTTP/1.1
Host: www.google.fr

HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/html
...
```

3.4.2 SMTP

On peut aisément faire de même avec une connexion SMTP et se passer ainsi de client mail !

```
$ telnet mail.e-cml.org 25
Trying 194.167.168.2...
Connected to servecml.e-cml.org.
Escape character is '^]'.
220 servecml.e-cml.org ESMTP
HELO evolix.fr
250 servecml.e-cml.org
MAIL FROM: <reg@evolix.fr>
250 ok
RCPT TO: <login@e-cml.org>
250 ok
DATA
From: "Gregory Colpart" <reg@evolix.fr>
To: "Laurent Ogin" <login@e-cml.org>
Date: Mon, 30 Aug 2004 11:07:31 +0200
Subject: Re: commande M. COLPART
Cc: "Renaud Oot" <root@e-cml.org>
Salut,
Comment vas-tu ?
--
Grégory C.
.
250 ok 1106589139 qp 47947
QUIT
221 servecml.e-cml.org
Connection closed by foreign host.
```

En lisant les RFC correspondant aux protocoles FTP, POP, etc. on pourrait faire de même.
Ainsi, on comprend bien la fonction des serveurs qui est de répondre à certaines requêtes bien précises et la fonction des clients qui est de savoir formuler les requêtes appropriées et d'interpréter correctement les réponses.

En analysant les données contenues dans les segments TCP, on voit bien sûr en clair cet échange d'information. On se rend compte aisément que si le protocole de la couche application ne spécifie pas de chiffrement, toutes les informations circulent en clair.

3.4.3 Services DNS

Le service DNS (Domain Name System) permet la conversion de noms de domaine en adresses IP (et inversement). On parlera de "nom pleinement qualifié" pour designer un nom de domaine complet (typiquement sousdomaine.domaine.tld). Il est plus facile pour l'être humain de retenir des noms que des numéros : il a donc été décidé d'associer des noms de domaine aux adresses IP. Au départ, un simple fichier HOSTS.TXT rassemblait les correspondances entre les adresses IP et les noms. Mais rapidement, ce système est devenu ingérable et il fut décidé d'opter pour une base de données distribuées en arbre (similaire au système de fichier UNIX) : la racine est gérée par l'ICANN[25] et les sous-domaines de la racine (TLD) sont délégués à des organismes : par exemple, chaque pays se voit confier un TLD (fr pour la France, us pour les États-Unis, uk pour la Grande-Bretagne, etc.). Ces organismes permettent ensuite à certaines sociétés, sous certaines conditions, de revendre des sous-domaines. À chaque sous-domaine acheté est associé un ou plusieurs serveurs DNS maître. Il est conseillé d'utiliser un serveur maître primaire et un serveur maître secondaire (ou serveur esclave) qui se synchronise sur le serveur primaire.

Serveur DNS

Un serveur DNS est (souvent) maître pour un certain nombre de noms de domaines, c'est-à-dire qu'il a autorité sur plusieurs zones. Il peut ainsi résoudre un certains nombres de demandes (il contient les informations grâce à des fichiers) ou déléguer des sous-zones à d'autres serveurs de domaine. Le serveur répond donc aux requêtes où il a autorité (en répondant directement ou en redirigeant vers le serveur ayant autorité) ou bien se redirige vers un serveur "racine" capable lui-même de rediriger vers les serveurs appropriés.
Au niveau réseau, un serveur DNS écoute sur le port 53 en TCP et UDP.

Resolvers DNS

Un résolver, ou client DNS, peut formuler deux types de demandes à un serveur DNS :

- les demandes en mode récursif qui demandent une résolution complète au serveur DNS, l'obligeant à faire lui-même les requêtes pour apporter la réponse
- les demandes en mode itératif qui demandent une réponse sans générer de requête de la part du serveur (sera donc sovent redirigé vers un serveur racine ou délégué)

Exemple de résolution "classique"

Le résolveur adresse une requête en mode récursif : le serveur local envoie une requête en mode itératif à un serveur de nom faisant autorité sur la demande. Il essaiera le serveur le plus à même de répondre. Dans le pire des cas il contactera un serveur "racine" et suivra les références jusqu'à obtenir une réponse complète (généralement par le serveur de nom s'occupant du réseau où se trouve la machine).
Une fois la réponse trouvée, le serveur local envoie la réponse au résolveur.
Attention, il faut noter que les applications ou les systèmes ont souvent leur propre cache DNS.

[25] http://www.icann.org/

Types d'enregistrement
Bien que le principal but soit de convertir un nom de domaine en adresse IP, il existe différents types d'enregistrement au niveau d'un serveur DNS. L'enregistrement principal est l'enregistrement A qui associe directement un nom de domaine à une adresse IP mais il enexiste d'autres : l'enregistrement CNAME (qui lie un nom de domaine vers un autre nom de domaine), MX (qui permet de spécifier plusieurs serveurs SMTP avec différentes priorités), NS (qui spécifie les serveurs de noms), TXT (qui permet d'avoir un champ texte), PTR (pour les enregistrements invers), etc. Il existe également des enregistrements pour IPv6 où il faut à priori utiliser les enregistrements AAAA (il peut exister également des enregistrements A6).

Résolution inverse
La résolution d'une adresse en nom n'est pas triviale car le processus de résolution devrait connaître à peu près toutes les branches de l'arbre pour pouvoir répondre, (sans compter que plusieurs noms pointent souvent vers la même adresse). L'idée a donc été de développée un sous-domaine pour les adresses. Ce sous-domaine est in-addr.arpa et peut contenir toutes les adresse IP existantes (elles sont inversées afin de partir des feuilles jusqu'à la racine). Ainsi, pour qu'une adresse IP fasse l'objet d'une résolution inverse, elle doit avoir un enregistrement dans ce sous-domaine.

Mémoire cache et durée de vie
En plus des différents mécanismes de résolution, l'utilisation d'une mémoire-cache accélère efficacement ces résolutions. En effet, lorsqu'un serveur de nom effectue une requête récursive il obtient de nombreuses informations (résolutions complètes, serveurs faisant autorité, etc.). Il va donc stocker ces informations afin de pouvoir répondre plus rapidement à de prochaines requêtes utilisant ces informations, et ainsi éviter d'interroger de nouveau les serveurs concernés. Il conserve en fait ces données pendant un temps fixé par le serveur DNS où il a été cherché ses données. Les serveurs DNS ayant des données qui varient fréquemment adapteront donc cette durée appelée TTL (Time To Live).

Les outils
L'outil "historique" pour provoquer une résolution DNS est *nsloookup*. Sous Unix, on trouvera les outils *host* et *dig*.

Exemple d'une requête avec dig :

```
$ dig @192.168.1.4 cmi.univ-mrs.fr +norecurse +notcp \
+noquestion +noauthority +noadditional MX

; <<>> DiG 9.3.1 <<>> @192.168.1.4 cmi.univ-mrs.fr
+norecurse +notcp +noquestion +noauthority +noadditional MX
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4914
```

```
;; flags: qr ra; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 2

;; ANSWER SECTION:
cmi.univ-mrs.fr.        86236   IN      MX      10 gyptis.univ-mrs.fr.

;; Query time: 6 msec
;; SERVER: 192.168.1.4#53(192.168.1.4)
;; WHEN: Sun Nov  6 19:56:15 2005
;; MSG SIZE  rcvd: 157
```

4 Routage

Le routage est le processus consistant à acheminer des datagrammes d'après les informations contenues dans les entêtes des datagrammes et les tables de routage. Le routage est forcément l'un des élements essentiels d'un réseau mais surtout pour gérer les interconnexions entre réseaux. La mise en oeuvre du routage sur Internet est ainsi un défi majeur.
Tous les systèmes connectés à un réseau ont donc une table de routage. Cette table de routage peut être statique (ajout de routes à la main) ou dynamique (routes gérées par un protocole de routage).

4.0.4 Décision de routage :

Un système A (IPA, MasqueA) s'adresse à un système B (IPB).
Le système A prend une décision de routage d'après les élèments suivants :

- Si IPB appartient au même sous-réseau que IPA, A lui envoie directement ses données.
- Sinon A envoie ses données à un routeur de son réseau spécifié dans sa table de routage.

L'appartenance au même sous-réseau est testée en vérifiant l'égalité des deux ensemble suivants : IPA & MasqueA et IPB & MasqueA

4.0.5 Tables de routage

Il s'agit de données stockées par un système destinées aux traitements des datagrammes entrants et sortants. Un poste de travail classique situé dans un petit réseau local (LAN) possède une table de routage limitée : il s'y trouve des informations pour contacter les autres sytèmes du réseau local et éventuellement des informations pour accéder à Internet.

Exemple :

```
# route -n

Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic Metric Ref    Use Iface
192.168.4.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
0.0.0.0         192.168.4.1     0.0.0.0         UG    0      0        0 eth0
```

On observe donc les informations suivantes :
Destination : Valeur à laquelle l'IP est comparée
Passerelle : Routeur à utiliser pour atteindre la destination
Genmask : Masque à utiliser pour comparer la correspondance entre l'IP et la Destination

Indic : Caractéritiques de cette route. Voici quelques valeurs intéressantes pour l'indicateur : U : indique que la route est opérationnelle H : indique qu'il s'agit d'une route vers un hôte unique G : indique que la route passe par un routeur
Metric : "Coût" de la route utilisable par les protocoles de routage dynamique.
Ref : Nombre de références faites à cette route
Use : Nombre de consultations de cette route
Iface : Nom de l'interface utilisée par cette route

– **Routage statique** À l'aide de la table de routage, le système va déterminer vers qui transmettre les paquets : le système choisit la route avec le masque comportant le plus de bits non nuls possible.

  Ainsi, pour des systèmes comme des postes de travail, les tables de routage sont statiques. Elles sont entrées manuellement par l'administrateur du poste ou bien elles peuvent être issues d'un serveur DHCP.

  Des systèmes plus complexes, comme des petits routeurs, ont souvent besoin d'une table de routage un peu plus complexe. Néanmoins si le nombre de routes est limité et si les routes changent relativement peu souvent, la table de routage sera construite manuellement.

  Remarque : pour un poste de travail, la configuration des paramètres IP (IP, masque, routeur) du poste suffisent souvent à construire la table de routage.

– **Routage dynamique**
  Un protocole de routage est chargé de construire une table de routage de façon dynamique. Il existe deux types de protocoles de routage : les protocoles intérieurs et extérieurs.

Les protocoles de routage intérieurs
  Un protocole de routage intérieur s'utilise sur un réseau autonome : il s'agit de déterminer les meilleures routes et de répandre ces informations entre les routeurs du réseau. Il existe les protocoles RIP (Routing Information Protocol), OSPF (Opn Shortest Path First) [26], Hello, IS-IS (Intermediate System to Intermediate System), etc. Les différences entre ces protocoles viennent des techniques pour déterminer le plus court chemin. RIP priviliégie la route possédant le plus petit nombre de sauts de routeurs, Hello privilégie la durée d'un aller-retour de la source à la destination, OSPF utilise une technique d'états de liaison, etc.

Les protocoles de routage extérieurs
  Un protocole de routage extérieur est utilisé pour échanger des informations de routage entre les réseaux autonomes. EGP (Exterior Gateway Protocol) est un protocole basique qui consiste à découvrir les voisins et à échanger de routage. BGP (Border Gateway Protocol) est plus souvent utilisé : il peut

[26] http://en.wikipedia.org/wiki/OSPF

notamment tenir compte de la politique de routage.

4.1 L'explosion des réseaux privés

Après avoir utilisé les masques de sous-réseaux pour faire face à l'explosion de la demande d'adresse IP, l'utilisation des classes privées s'est développée grâce à de nouvelles technologies : le NAT puis le NPAT. Il s'agit de technologies de routage pour pouvoir limiter la distribution d'adresses IP car permettant aux réseaux locaux d'accéder à l'extérieur tout en possédant une adresse non routable sur Internet.

Connexions de 2 hôtes au site Internet possédant l'adresse IPOM :

Voici notamment les informations contenues dans les deux datagrammes (avant puis après le routeur) :
```
IPA IPB 1053 80 -> IPR IPB 1053 80
IPB IPB 1784 80 -> IPR IPB 1784 80
```
Au niveau du routeur, une table contient donc les informations suivantes :
```
A : IPA, 1053, tcp, IPOM, 80
B : IPB, 1053, tcp, IPOM, 80
```

NAT (Network Adress Translation) Le routeur substitue l'adresse IP source par son adresse publique et retient dans une table que le port utilisé est associé à l'adresse privée.

Remarque : problème lorsque deux machines utilisent le même port ! !

En effet, les ports "clients" sont choisis aléatoirement par le système, on pourrait donc se retrouver avec ces informations dans les deux datagrammes :
```
IPA IPOM 1053 80 -> IPR IPOM 1053 80
IPB IPOM 1053 80 -> IPR IPOM 1053 80
```
Comment résoudre ce problème ?

La réponse donnée est que le routeur substitue également le port client des hôtes afin de pouvoir les différencier.

Voici donc par exemple, dans ce cas, les informations contenues dans les deux datagrammes (avant puis après le routeur) :
```
IPA IPOM 1053 80 -> IPR IPOM 1427 80
IPB IPOM 1053 80 -> IPR IPOM 1861 80
```
Sur le routeur, on a ainsi une table contenant un peu plus d'informations :
```
A : IPA, 1053, tcp, IPOM, 80, 1427
B : IPB, 1053, tcp, IPOM, 80, 1861
```

NPAT (Network Adress Port Translation) Le routeur substitue l'adresse IP source et le port source par son adresse publique et un des ses ports disponibles. Il retient que ce port est désormais occupé et l'associe à l'adresse

privée ET son port source.

5 Sécurité et cryptographie

La notion de sécurité informatique n'englobe pas seulement la sécurité réseau mais également tout ce qui touche à la sécurité des données, l'utilisation de technologies RAID, de processus éprouvés de sauvegarde locale et distante.
Même l'écriture de documentations relève de la sécurité au cas où l'administrateur aurait un accident ! On peut ainsi penser à enfermer les mots de passe importants dans un coffre accessible par un supérieur hiérarchique.
On voit donc que la sécurité est un sujet bien vaste !
Nous nous contenterons d'aborder ici la sécurité réseau...

5.1 Firewall

L'un des élements essentiels en terme de sécurité réseau est le firewall (littéralement "mur de feu").
Il s'agit d'un système capable de réaliser un filtrage avancé sur les différentes couches du modèle de DOD sur plusieurs interfaces réseaux.
Un firewall est souvent placé en tête de réseau afin de filtrer toutes les connexions extérieures. La configuration d'un firewall consiste à décider d'un ensemble de règles et de les traduire dans un langage compréhensible pour le système.
Pour mettre en place des règles "classiques" il faut bien comprendre les différentes notions :
- paquet entrant : paquet destiné au système
- paquet sortant : paquet émis par le système
- paquet traversant : paquets non destinés au système transitant par le système

Lorsque que le firewall est capable de faire du PNAT (on se contente souvent de dire NAT), la possibilité de faire de la translation d'adresse source ou bien de la translation d'adresse de destination existe.

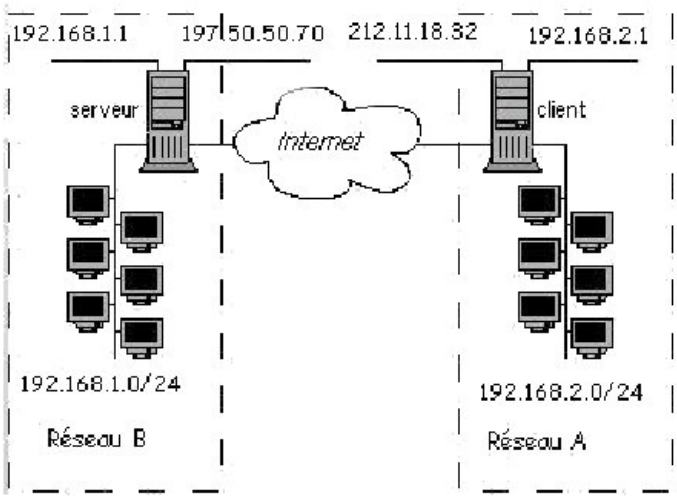
Un firewall peut également faire du marquage de paquet c'est-à-dire marquer les paquets selon certains critères et ainsi permettre, par exemple, à une couche logicielle de traiter les paquets marqués. Un firewall peut également générer des journaux pour certaines règles. Des règles spécifiques peuvent aussi être implémentées, etc.

5.2 VPN

5.2.1 Définition d'un VPN

L'acronyme VPN signifie Virtual Private Network, ie Réseau Virtuel Privé.
Lorsqu'on aborde pour la première fois le sujet des réseaux privés virtuels, on s'aperçoit que le nombre de définitions d'un VPN est trés élevé.
Une première définition simple et approximative serait :
Un VPN est un réseau privé construit sur l'infrastructure d'un réseau public, comme l'est Internet.
Un réseau est une collection de dispositifs qui peuvent communiquer entre eux et

donc s'échanger des données.
Les réseaux VPN sont privés, car aucune connexion externe, et donc aucune communication de réseau avec l'extérieur n'est présente.
La communication virtuelle entre deux ou plusieurs machines se justifie par la fait que les machines qui ne sont pas concernées par la communication n'ont pas la possibilité de percevoir des données.
Un VPN peut étre créé entre deux systèmes, entre deux organisations, entre plusieurs systèmes et une organisation ou entre plusieurs organisations répandues dans Internet, soit encore entre des applications individuelles ou une combinaison des ces possibilités.



Un VPN est un environnement de communication dans lequel l'accès est contrôlé, afin de permettre des connections entre une communauté d'intérêts seulement.
Son objectif est de fournir aux utilisateurs les conditions d'exploitation, d'utilisation et de sécurité à travers un réseau public identiques à celles disponibles sur un réseau privé.

5.2.2 Exemples d'utilisation pratique

Télé-travail : Raccordement de télé travailleurs ou travailleurs mobiles. Ceux-ci se raccordent aux ressources de l'entreprise par modem ou tout autre moyen de connexion.

Connexion de sites distants : Interconnexion de succursales. Des sites distants d'une même entreprise qui partagent les mêmes ressources sans avoir recours à des lignes spécialisées.

Transport de la voix

Externalisation

WAN clefs en main

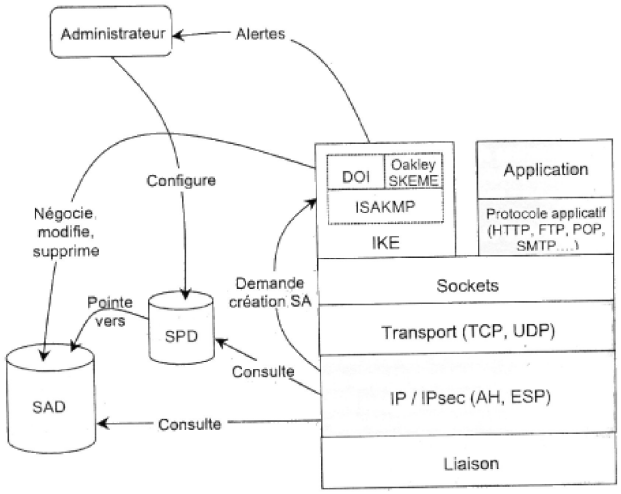
5.2.3 Principe VPN

Il est basé sur la technique du tunnelling. Cela consiste à construire un chemin virtuel après avoir identifié l'émetteur et le destinataire. Ensuite la source chiffre les données et les achemine en empruntant ce chemin virtuel.
Les données à transmettre peuvent appartenir à un protocole différent d'IP. Dans ce cas le protocole de tunnelling encapsule les données en rajoutant une entête permettant le routage des trames dans le tunnel. Le tunneling est l'ensemble des processus d'encapsulation, de transmission et de désencapsulation.

5.2.4 IPSEC

À l'heure actuelle, plusieurs solutions VPN existent et sont utilisées sur Internet : IPSec est celle que l'on retrouve le plus souvent, et de nombreux facteurs indiquent que cette situation tend à se généraliser (supporté nativement par de nombreux systèmes d'exploitation, protocole ouvert, etc.).
Revenons aux principes de base d'IPSec.

IPSec se présente sous la forme d'un ensemble de mécanismes permettant d'initier, au niveau réseau, des connexions entre systèmes distants.
Le schéma suivant rappelle les principes de son fonctionnement :



On note à la vue de ce schéma qu'IPSec repose sur le protocole IKE, qui permet une connexion sécurisée entre les entités désirant communiquer, et les protocoles AH et ESP, qui traitent les donnéees utiles de la couche IP afin de les protéger selon la politique choisie.

Avant que les paquets ne puissent être sécurisés par IPSec, une SA (Associations de Sécurité) doit exister. Elle peut être crée manuellement ou dynamiquement. Le protocole IKE est utilisé pour la création dynamique de cette SA ; il s'agit d' un protocole hybride basé sur les protocoles ISAKMP, Oakley et SKEME : il utilise les

bases de ISAKMP, les modes de Oakley et les techniques de partage des clés de SKEME.

5.3 Cryptographie

5.3.1 Définition

La cryptographie ou chiffrement est le processus de transcription d'une information intelligible en une information inintelligible par l'application de conventions dont l'effet est réversible à condition de connaître un certain secret, la clé.

Deux grands types de chiffrement existent du fait que la clé est secrète ou publique. On parle respectivement de cryptographie symétrique et de cryptographie asymétrique.

- **la cryptographie symétrique** : la même clé est utilisée pour chiffrer et pour déchiffrer les messages
- **La cryptographie asymétrique** : ce n'est pas la même clé qui chiffre et qui déchiffre. L'utilisateur possède une clé privée qu'il garde secrète et d'une clé publique, qu'il distribue à tout le monde.

5.3.2 Cryptographie à clefs secrètes

Petit historique

- **Algorithmes de chiffrement par substitution mono-alphabétique :** Algorithme de chiffrement de César : substitution mono-alphabétique définie par une rotation circulaire de lettres.

- **Algorithmes de chiffrement par substitution poly-alphabétique :**
  - Le carré de Vigenère (Vigenère, alchimiste, écrivain, historien,diplomate au service des Ducs de Nevers et des Rois de France, auteur, en 1586, du *Traité des Chiffres ou secrètes manières d'écrire*)
  - Algorithme de chiffrement de Hill (1929, Cryptography in an algebraic alphabet) : substitution par groupe de m lettres, basé sur une transformation linéaire dans Z/nZ.

- **Chiffrement de Vernam** : clé aussi longue que le message et utilisable une unique fois (téléphone rouge, valise diplomatique)

- **Data Encryption Standard (DES)**(accepté par la National Security Agency en 1970 sur proposition de IBM et devenu le standard de chiffrement américain pendant 20 ans)
  D'une manière générale, on peut dire que DES fonctionne en trois étapes :

1. permutation initiale et fixe d'un bloc
2. le résultat est soumis à 16 itérations d'une transformation, qui dépendent à chaque ronde d'une autre clé partielle de 48 bits, calculée à partir de la clé initiale de l'utilisateur. Lors de ces rondes, le bloc de 64 bits est découpé en deux blocs de 32 bits, et ces blocs sont échangés l'un avec l'autre. Le bloc de 32 bits ayant le poids le plus fort (celui qui s'étend du bit 32 au bit 63) subira une transformation.
3. le dernier résultat de la dernière ronde est transformé par la fonction inverse de la permutation initiale.

– **Triple DES (aussi appelé 3DES)** 3 applications successives de l'algorithme DES sur le même bloc de données de 64 bits, avec 2 ou 3 clés DES différentes.

– **AES Advanced Encryption Standard**, en français *standard de chiffrement avancé* remplacant du DES en 2001.

Fonctionnement : L'algorithme prend en entrée un bloc de 128 bits (16 bytes), la clé fait 128, 192 ou 256 bits.
Les 16 bytes en entrée sont permutés selon une table définie au préalable. Ces bytes sont ensuite placés dans une matrice de 4x4 éléments et ses lignes subissent une rotation vers la droite. L'incrément pour la rotation varie selon le numéro de la ligne.
Une transformation linéaire est ensuite appliquée sur la matrice, elle consiste en la multiplication binaire de chaque élément de la matrice avec des polynômes issues d'une matrice auxiliaire, cette multiplication est soumise à des règles spéciales.
Finalement, un XOR entre la matrice et une autre matrice permet d'obtenir une matrice intermédiaire.
Ces différentes opérations sont répétées plusieurs fois et définissent un *tour*. Pour une clé de 122, 192 ou 256, AES nécessite respectivement 10, 12 ou 14 tours.

Résumé et Applications pour la cryptographie symétrique
Caractéristiques : Opérations de chiffrement et déchiffrement identiques (clés identiques)
Avantages :
Rapidité, et facilité de mise en oeuvre sur des circuits « bon marché »
Inconvénient :
Problème de la distribution des clefs

5.3.3 Cryptographie à clés publiques

Préambule

Le concept de cryptographie à clef publique a été inventé par Whitfield Diffie et Martin Hellman en 1976, dans le but de résoudre le problème de distribution des

clefs posé par la cryptographie à clef secrète.
Depuis, de nombreux algorithmes permettant de réaliser un cryptosystème à clef publique ont été proposés.
La sécurité de ces algorithmes est conditionnée par la difficulté mathématiques qui y est lié (ex : décomposer un grand nombre en produit de deux nombre premiers).

Contrairement au cas précédent, on se contentera ici de décrire rapidement deux cryptosystèmes qui reflètent bien la cryptologie asymétrique : Diffie Hellman et RSA.

– **Diffie Hellman** C'est la dénomination d'un protocole cryptographique qui permet à deux tiers de générer un secret partagé sans informations préalables l'un sur l'autre.(ce protocole du nom de ses concepteurs comme indiqué plus haut est à l'orgine de la cryptologie à clef publique)
Sa sécurité dépend de la difficulté de calculer des logarithmes discrets sur un corps fini.
Le secret généré à l'aide de cet algorithme peut ensuite être utilisé pour dériver une ou plusieurs clefs (clef secrète, clef de chiffrement de clefs...).

Voici le déroulement de l'algorithme :

1. Alice et Bob se mettent d'accord sur un grand entier n tel que $(n-1)/2$ soit premier et sur un entier g primitif par rapport à n. Ces deux entiers sont publics.

2. Alice choisit de manière aléatoire un grand nombre entier a, qu'elle garde secret, et calcule sa valeur publique, A = ga mod n. Bob fait de même et génère b et B = gb mod n.

3. Alice envoie A à Bob ; Bob envoie B à Alice.

4. Alice calcule $K_AB = B_a$ mod n
Bob calcule $K_BA = A_b$ mod n.
$K_AB = K_BA = gab$ mod n est le secret partagé par Alice et Bob.

Une personne qui écoute la communication connaît g, n, $A = ga$ mod n et $B = gb$ mod n, ce qui ne lui permet pas de calculer gab mod n : il lui faudrait pour cela calculer le logarithme de A ou B pour retrouver a ou b.

– **RSA Rivest Shamir Adleman** [27] En 1978, l'algorithme à clé publique de Rivest, Shamir, et Adelman (d'où son nom RSA) apparaît. Cet algorithme sert encore en 2002 à protéger les codes nucléaires de l'armée américaine et russe.

Schéma :

1. On choisit p, q deux « grands » nombres premiers

2. On calcule n=p.q

[27] http://fr.wikipedia.org/wiki/RSA

3. Un entier e est choisi de façon à être premier avec (p-1) et (q-1)

4. L'entier d tel que ed = 1 [(p-1)(q-1)] est calculé, avec l'algorithme d'Euclide
   Le couple d'entier (n,e) représente la clef publique.
   L'entier d représente la clef

Résumé et Applications pour la cryptographie asymétrique

Problèmes : Algorithmes asymétriques très lents

Applications de la cryptographie asymétrique : les signatures numériques (authentification et intégrité) et le chiffrement (des données).

- **Confidentialité** : tout le monde peut écrire en utilisant la clé publique et seul son possesseur peut déchiffrer les messages qui lui sont destinés à l'aide de sa clé secrète.
- **Authentification** : l'utilisateur signe ses messages avec sa clé secrète et les destinataires peuvent vérifier son identité à l'aide de sa clé publique.

5.3.4 En pratique coté réseaux

Comme précédemment évoqué il existe deux concepts principaux qui sur un réseau sont primordiaux : (problèmes auxquels s'attaque la cryptographie), l'**authentification** de l'origine des données et leur **intégrité**
Comment résoudre ces problèmes ?

- Avec un canal sûr en parallèle du canal de communication normal, on peut communiquer l'empreinte des messages (fonction de hachage) pour assurer l'**intégrité** des données transférées.

- Sans canal sûr : si l'on transfère l'empreinte sur un tel canal un intercepteur peut modifier les données puis recalculer l'empreinte (man in the middle).
  Il faut trouver une méthode pour s'assurer que seul l'expéditeur est capable de calculer l'empreinte. Par exemple, une fonction de hachage à sens unique qui fonctionne de plus avec une clef secrète ou privée peut faire l'affaire.
  On notera qu'alors on fournit également l'**authentification** de l'origine des données.

Inversement, si l'on désire fournir l'**authentification** de l'origine des données et que l'on utilise pour cela un moyen qui ne garantit pas l'**intégrité** des données authentifiées, un intrus peut modifier le message et donc faire accepter comme authentifiées des données qu'il a choisies. C'est pourquoi intégrité et authentification de l'origine des données sont généralement fournies conjointement par un même mécanisme. On utilise le terme d'**authenticité** pour désigner l'**intégrité** jointe à l'**authentification** des données.

Théorie VS Pratique

Une signature numérique (qui ne peut être générée par essence que par l'expéditeur) fournit les services d'**authentification** de l'origine des données, d'**intégrité** des données et de **non-répudiation**. Ce dernier point la différencie des codes d'authentification de message, et a pour conséquence que la plupart des algorithmes de signature utilisent la cryptographie à clef publique.
La théorie voudrait qu'une signature consiste à signer un message avec sa clé privé. Problème pratique de lenteur fait qu'en fait on ne chiffre qu'une empreinte du message.

Certains algorithmes asymétriques ne sont adaptés qu'au chiffrement, tandis que d'autres ne permettent que la signature. Seuls trois algorithmes sont utilisables à la fois pour le chiffrement et pour la signature : RSA, ElGamal et Rabin.

Pour des applications où il faut échanger de nombreuses données on utilise donc des **systèmes hybrides** :

- *échange d'une clé sur le canal par un algorithme de chiffrement asymétrique,*
- *cette clé servant ensuite à la communication des données à l'aide d'un algorithme de chiffrement symétrique ;*
- *Exemple : PGP (Pretty Good Privacy), utilisé notamment pour le courrier électronique*

5.4 SSL

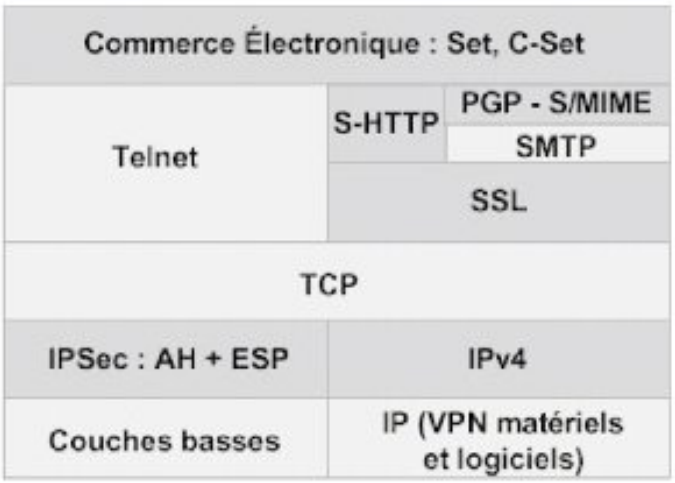
5.4.1 Définition :

SSL signifie Secure Sockets Layer, en français couche de sockets sécurisée.
Le protocole SSL a été développé par Netscape Communications de façon à améliorer la confidentialité des échanges de tout protocole basé sur TCP/IP.
En 1994, Netscape Communications diffuse le protocole SSL 2.0[28]
En 1996, Netscape Communications diffuse une amélioration : le protocole SSL 3.0[29]
En 2001, le brevet de SSL a été racheté à Netscape Communications par l'IETF (Internet Engineering Task Force) et a été renommé TLS (Transport Layer Security), qui peut-être vu comme la version 3.1 de SSL
Le protocole TLS 1.0 est désormais standard. Il est décrit dans le RFC 2246[30]

[28] http://wp.netscape.com/eng/security/SSL_2.html
[29] http://wp.netscape.com/eng/ssl3/
[30] ftp://ftp.rfc-editor.org/in-notes/rfc2246.txt

5.4.2 Caractéristiques du protocole SSL 2.0

L'un des points forts du protocole SSL est qu'il s'insère entre le protocole applicatif et le protocole TCP. Il est donc transparent pour le protocole applicatif.



SSL assure l'authentification des extrémités du circuit de communication ainsi que la confidentialité et l'intégrité des informations transmises :

- le serveur s'identifie auprès du client par un certificat vérifiable par le client. Cette authentification est obligatoire. Le client peut optionnellement s'identifier auprès du serveur par les mêmes mécanismes.

- le flux d'informations transmises entre les deux extrémités du circuit est segmenté en enregistrements (cf. SSL Record Protocol Specification). Chaque enregistrement est chiffré par un algorithme négocié au moment de l'ouverture du circuit.

- En outre, chaque enregistrement est accompagné d'une signature MAC (Message Authentication Code) garantissant l'intégrité de l'enregistrement.

5.4.3 Caractéristiques du protocole SSL 3.0

Le protocole SSL 3.0 corrige un certain nombre de faiblesses de la version 2 et apporte de nouvelles fonctionnalités. En plus du protocole d'ouverture de session (SSL Handshake Protocol), SSL 3.0 prévoit deux protocoles supplémentaires : le protocole d'échange de spécifications de chiffrement (SSL Change Cipher Spec) et le protocole d'alerte (SSL Alert Protocol).
Tous deux, à l'instar du SSL Handshake Protocol, s'appuient sur le protocole de transfert d'enregistrement (SSL Record Protocol).

Les messages échangés par ces protocoles sont donc chiffrés dès que possible.
Le protocole d'alerte est utilisé pour la transmission de codes d'erreurs entre le client et le serveur.
Les codes d'erreurs sont accompagnés d'un indice de sévérité de l'erreur (avertissement, erreur fatale).
Le protocole SSL 3.0 spécifie que si une erreur fatale est détectée lors d'une session, alors la session doit être immédiatement interrompue et doit être effacée du cache des sessions, à la fois sur le client et sur le serveur.
Ainsi, une erreur dans le protocole SSL force, lors de la prochaine ouverture de session, la renégociation des spécifications de chiffrement et la génération d'une nouvelle clé-maître. Le protocole d'alerte est également utilisé pour indiquer la fin d'une session entre le client et le serveur.
Cette notion n'existe pas dans SSL 2.0 , où l'une des extrémités indique la fin d'une session SSL en fermant la connexion TCP/IP sous-jacente.
Ce mécanisme rend SSL 2.0 vulnérable aux attaques par "troncature" : en effet, on peut très bien imaginer un intrus (*man in the middle*) provoquant la fin prématurée d'une session SSL en envoyant les paquets ad-hoc au client et au serveur.
Dans le protocole d'ouverture de session, le serveur et le client ont la possibilité de négocier un algorithme de compression des données avant leur chiffrement.
En outre, lorsqu'ils transmettent leur certificat, ils transmettent également la chaîne d'autorités de certification qui valide le certificat en question.
De même, lorsque le serveur demande le certificat du client, il lui transmet également la liste des autorités de certification qu'il accepte.
Le client est donc en mesure de sélectionner le certificat le plus approprié.

5.4.4 Négociation sous SSL

Une session SSL commence par une phase de négociation (handshake).
Le protocole SSL combine simultanément l'utilisation de clés publiques et de clés symétriques. Les clés publiques privées ou clés asymétriques procurent en effet une très bonne méthode pour l'authentification mais son utilisation est coûteuse en terme de bande passante. A l'opposé, le mécanisme de clé symétrique (identique pour chiffrer et déchiffrer) est extrêmement rapide mais pas réellement adapté à l'authentification d'un tiers. Ainsi SSL va utiliser son protocole de négociation qui va être apte à partir des clés publiques et privées du client et du serveur d'établir une communication entre les deux entités avec une clé secrète (symétrique) de taille nettement inférieure à celle rencontrée pour des clés publiques (souvent 128 bits contre 1024 ou plus).

Mécanisme de la négociation

1. Le client envoie au serveur sa version du protocole SSL, ses paramètres de chiffrement, des données générées aléatoirement et d'autres informations dont le serveur a besoin.
2. Le serveur renvoie sa version de SSL, ses paramètres de chiffrement, des données générées aléatoirement et d'autres informations dont le client a besoin.

Le point essentiel est l'envoi par le serveur de son propre certificat avec des informations concernant ce certificat. Si le client demande une information nécessitant un certificat, il demande également un certificat client.

3. Le client utilise les informations envoyées par le serveur pour l'authentifier. Si le serveur ne peut pas être authentifié, la connexion n'a pas lieu.
4. Avec les données préalablement échangées, le client est en mesure d'envoyer au serveur une clé secrète, qu'il chiffre avec la clé publique du serveur. Si le serveur a requis une authentification du client, ce dernier renvoie également au serveur un bloc de données signé ainsi que son certificat.
5. Si le serveur a requis une authentification, il authentifie le client. Le serveur utilise alors sa clé privée de façon à pouvoir déchiffrer la pré clé secrète. Le serveur effectue alors une suite d'actions (également effectuées par le client) pour obtenir une clé secrète à partir de la pré clé secrète.
6. Le client et le serveur utilisent la clé secrète pour générer des clés de session qui seront les clés symétriques utilisées pour le chiffrement, déchiffrement des données et l'intégrité.
7. Le client envoie alors un avertissement au serveur le prévenant que les prochains messages seront chiffrés avec la clé de session. Puis il envoie un message chiffré indiquant que la phase de négociation est terminée.
8. Le serveur envoie alors un avertissement au client comme quoi les prochains messages seront chiffrés avec la clé de session. Puis il envoie un message (chiffré cette fois) indiquant que la phase de négociation est terminée.
9. La phase de négociation est alors terminée.

Dans le cas du SSL il est pour le moment assez rare de rencontrer une authentification du client. En effet, la plupart des applications utilisées sur Internet ne requièrent pas un tel niveau de sécurité.

Toute la sécurité de SSL repose sur l'authentification du serveur par le client et donc la vérification de la validité du certificat par le serveur.
Par exemple, lorsque l'on se connecte à un serveur web sécurisé, un navigateur web présente le certificat en faisant apparaître une fenêtre contenant des informations sur le certificat. L'idéal serait de vérifier de façon sûre l'empreinte du certificat du serveur. Mais en pratique, cette méthode est irréalisable. En revanche, pour limiter les risques, le client devra vérifier (souvent visuellement) certaines informations sur le certificat du serveur :

1. **Vérification de la date de validité du certificat du serveur.**
2. **Est-ce que l'autorité de certification est une autorité de confiance ?** Pour vérifier cela, les navigateurs intègrent une liste de certificats d'autorités. Mais les certificats ne sont pas toujours validés par des autorités de confiance.
3. **Vérification de la clé publique à partir de la signature**. Le client vérifie la validité de la signature (chiffrée avec la clé privée) fournie dans le certificat grâce à la clé publique qui a été fournie elle aussi dans le certificat.

4. **Vérification du nom de domaine du serveur.** Cette étape permet de vérifier que le nom de domaine du serveur défini dans le certificat correspond bien à la même adresse Internet. Cette étape n'est pas obligatoire et dépend de l'implémentation du client SSL. Elle permet cependant d'éviter qu'un usurpateur ne vienne jouer les intermédiaires entre le client et le serveur et se fasse passer pour l'un et l'autre auprès des deux entités. Vérifier la validité du nom de domaine est le seul moyen d'éviter ce genre de faille qui permet, dans ce cas, à la personne malveillante d'intercepter les informations transitant pendant la négociation et donc ultérieurement de prendre la place du client une fois cette phase passée. Pour usurper l'identité du serveur, le pirate devra également se procurer la clé privée du serveur.
5. Le serveur est maintenant authentifié, la phase de négociation se poursuit.

5.4.5 Utilisation

On a pu voir que SSL permet de chiffrer une communication client/serveur.

HTTPS Le protocole HTTPS est l'implémentation du protocole HTTP (Hyper Text Transfer Protocol) au dessus de SSL. Du fait que le démarrage d'une session SSL se fait à l'initiative du client, l'utilisation de HTTPS pour le transfert de pages WWW doit être spécifié dans l'URL (Uniform Resource Locator). Le protocole HTTPS est communément utilisé dans les applications de commerce électronique, au moins dans la phase de transfert des coordonnées bancaires (numéros de cartes bancaires, notamment).
On peut repérer concrètement cette utilisation par l'apparition d'un cadenas en bas à droite du navigateur, ou d'une notification lors de l'arrivée ou la sortie d'une page sécurisée. Il faut garder à l'esprit que seul le transfert de données est réellement sécurisé. L'utilisation de HTTPS n'augmente en rien la sécurité du serveur lui-même, ni de l'application basée sur HTTPS .

SSL et la messagerie Les technologies de Webmail, utilisées pour les accès à distance aux boîtes aux lettres, sont souvent sécurisées par HTTPS et permettent de visualiser les courriels avec un protocole chiffré.
Il existe des protocoles spécifiques pour consulter des courriels tels POP ou IMAP. Ces derniers sont encapsulables dans des sessions SSL. L'idée est donc la même que pour HTTPS : le client et le serveur établissent en premier lieu une connexion SSL et déroulent ensuite le protocole classique au dessus de cette connexion.
Les protocoles qui en résultent ont reçu des numéros de ports officiels par l'IANA : il s'agit de 993 pour IMAP sécurisé et 995 pour POP sécurisé.

Le format S/MIME Le format S/MIME est une extension du format MIME [31](format spécifique d'extensions multimédias pour les courriels) pour les messages chiffrés ou signés (Macintosh utilise plutôt BinHex.).
Il définit un certain nombre de types MIME pour décrire un message chiffré

[31] http://www.linux-france.org/article/memo/smtp/index18.htm

ou un message signé. S/MIME s'appuie sur le standard PKCS7 qui décrit une syntaxe générale pour des données auxquelles on a appliqué des techniques de cryptographie, c'est-à-dire essentiellement, la signature digitale et le chiffrement. Les clients de messagerie sont en mesure de générer et interpréter correctement des messages au format S/MIME.
La signature d'un message S/MIME définit plusieurs formats pour des messages signés.

– Le plus couramment utilisé correspond au type MIME multipart/signed. Dans ce format, le message transmis est la réunion de deux parties MIME. La première correspond au message à signer. Il a un type MIME et un encodage classique. La deuxième partie est l'empreinte correspondante chiffrée avec la clé privée du signataire. Elle contient également le certificat du signataire et les informations sur les algorithmes utilisés pour calculer l'empreinte du message et chiffrer cette empreinte. (Plus précisément, cette partie est un objet PKCS7 de type signedData avec un champ contentInfo vide. On lui attribue le type MIME application/pkcs7-signature.)

– Le type MIME application/pkcs7-mime est utilisé dans le cas d'un message ou d'une partie de message chiffré.
La partie chiffrée du message est un objet PKCS7 de type envelopedData. Il correspond au résultat du chiffrement du message par la clé publique du destinataire.

– Dans le cas d'un message à la fois signé et chiffré, il y a deux possibilités.

1. soit on chiffre le message signé,
2. soit on signe le message chiffré.

Les deux solutions ont chacune leur intérêt.
Dans le premier cas, la liste des signataires est masquée par le chiffrement. Dans le deuxième cas, on peut vérifier les signataires sans être en mesure de déchiffrer le message. C'est donc l'application de messagerie qui décide quel mécanisme employer.

Pour pouvoir utiliser S/MIME, il est nécessaire d'être titulaire d'un certificat d'authentification. L'autorité signataire de ce certificat doit être connue de tous les correspondants. Dans le cas contraire, le destinataire d'un message ne pourra pas être en mesure de vérifier la signature de l'émetteur du message. En principe, un certificat d'utilisateur ne contient que la clé publique. La manière habituelle de transmettre à l'utilisateur son certificat et sa clé privée est de chiffrer l'ensemble par un algorithme symétrique standard (triple-DES, par exemple) et de communiquer la clé de chiffrement par un moyen sûr (communication téléphonique chiffrée, courrier en main propre, etc.)
Le format dans lequel est chiffré le certificat et la clé privée s'appelle PKCS12 . Importer un certificat au format PKCS12 depuis un navigateur standard est trivial.

OpenSSL OpenSSL [32] est une suite libre d'outils implémentant à la fois les protocoles SSL 2.0 et SSL 3.0 et le protocole TLS 1.0 ainsi que des librairies cryptographiques.
OpenSSL est basé sur l'excellente librairie SSLeay [33] développée par Eric A. Young et Tim J. Hudson.
Le projet est géré par une communauté de développeurs répartis partout dans le monde.
Les outils OpenSSL sont sous une license de style Apache qui en permet une utilisation commerciale ou non.
Toutes les fonctions principales nécessaires pour mettre au point une politique de sécurisation sont accessibles par la ligne de commande : algorithmes de hashage, algorithmes de chiffrement clé privé ou public, diffie-hellman, signature ...

5.5 Attaques réseau

Dans cette partie, nous nous parlerons uniquement des attaques orientées réseau. Nous ne parlerons pas des attaques orientées système (virus, chevaux de Troie, failles logiciels, dépassement de mémoire) ou des attaques de "social-engineering" (pishing, crackage, etc.)
Il existe de nombreux types d'attaques réseau qui peuvent avoir des conséquences diverses comme la mise hors-service d'un système ou l'interception de données. De nouvelles possibilités ou failles sont découvertes en permanence. Nous n'allons pas dresser une liste exhaustive mais parler de certaines attaques, quasi-historiques, qui permettent de bien comprendre certains problèmes posés par un réseau. Les systèmes récents ont souvent des parades pour prévenir ces types d'attaques.

5.5.1 Le flooding

Le principe du flooding est d'inonder un destinataire avec des requêtes réseau trop nombreuse afin de saturer ses ressources.

- **TCP SYN :** Il s'agit de générer de nombreux paquets SYN (c'est-à-dire des paquets d'initialisation de connexions TCP) en simulant des émetteurs factices (spoofing).
  Le serveur attendra la suite de la connexion pendant un certain temps et occupera donc de la place dans une mémoire. Si les paquets sont envoyés assez rapidement, cela va saturer la mémoire du destinataire (IP SPOOFING).

- **Ping :**
  Il s'agit d'envoyer des paquets ICMP le plus rapidement possible afin de ralentir voire faire planter le destinataire.
  Il s'agira de message de type "ECHO REQUEST" provenant de d'hôtes possédant des caractéristiques supérieures (bande passante, CPU, mémoire) au

[32] http://www.openssl.org
[33] http://www2.psy.uq.edu.au/~ftp/Crypto/

destinataire, ou bien d'envoyer des "ECHO REQUEST" à des nombreux hôtes en prenant l'adresse de la victime (spoofing) afin de le saturer sous les paquets "ECHO REPLY".

– **UDP :**
Il s'agit d'envoyer des paquets avec une adresse source égale à l'adresse de la victime (spoofing) pour délivrer un flux UDP.
Cela permet de générer un trafic UDP en direction de la victime.

5.5.2 Le spoofing :

Le principe du spoofing est d'usurper l'identité d'un système afin de pouvoir manipuler les équipements réseau.

– **adresse MAC :** `ifconfig eth0 hw ether` $11:22:33:44:55:66$ `up`
Un attaquant prenant l'adresse MAC d'un hôte du réseau peut piéger les équipements de type "switch de niveau 2". En effet, ces équipements contiennent des tables associant port et adresses MAC. En voyant passer deux adresses MAC identiques issues de deux ports différents, le switch peut avoir un comportement imprévisible (transformation en hub ou même reboot !) mais logiquement il devrait mettre à jour sa table en fonction du dernier paquet passé. Ainsi si l'attaquant émet un paquet, le switch , lui, redirigera tout le trafic à destination de l'hôte (tant que l'hôte n'émettra pas de paquet). Pour éviter que l'hôte ne ré-emette des paquets, cette attaque peut être couplée avec une attaque par *deny of service*.

– **adresse IP :** `ifconfig eth0 11.22.33.44`
Un attaquant prenant l'adresse IP d'un hôte du réseau peut piéger les tables ARP des routeurs. En effet, ces équipements contiennent des tables assoçiant adresse MAC et adresses IP.
Ainsi lorsqu'il demande l'adresse MAC associée à l'adresse IP d'un hôte du réseau, l'attaquant répondra également à cette requête. Le routeur recevant peut avoir un comportement imprévisible. Afin que l'hôte ne réponde plus, une attaque par déni de service pourra également être effectuée.
Ce type d'attaque suppose néanmoins que le cache ARP du routeur ne contienne pas l'association entre l'hôte et son adresse MAC, ce qui n'est pas forcément évident. Des techniques consistent à tenter de corrompre le cache ARP d'un hôte existant en envoyant des requêtes astucieuses (arp-sk[34], etc.)

5.5.3 ICMP :

Outre le ping flooding, les messages ICMP peuvent s'avérer dangereux pour certains systèmes.

– **Ping Of Death :**

[34] http://www.arp-sk.org/

Il s'agit d'envoyer un paquet ICMP d'une taille non prévue (supérieure à 65535 octets). Le paquet est donc fragmenté et certains systèmes ne gèrent pas cette fragmentation et peuvent se bloquer voire planter.

– **Redirect bombs**
Il s'agit d'envoyer des messages ICMP de type 5 (redirect) afin de détourner le trafic vers un autre routeur.

5.5.4 Fragmentation

Il s'agit de construire des paquets fragmentés de taille très faible (un octet) où une partie des entêtes de la couche transport serait seulement dans le deuxième fragment. Souvent les firewalls laissent passer ce type de paquets et le système du destinataire peut planter.

6 Introduction à IPv6

IPv6 est le protocole de la couche réseau (couche 3 sur le modèle d'OSI) qui devrait remplacer IPv4 (la version actuelle de la couche réseau sur Internet) dans les prochaines années. L'un des intérêts d'IPv6 est d'avoir un nombre très important d'adresses routables. Les adresses IPv6 sont codées sur 128 bits (16 octets) donnant un nombre théorique d'adresses IPv6 supérieur à 3.10^{38}, soit 7.10^{29} fois plus qu'IPv4 ! Mais IPv6 apporte également d'autres avantages en terme de mécanismes (configuration automatique, découverte des voisins, ICMPv6, mobilité), ou de sécurité (authentification, confidentialité, etc.). La transition d'IPv4 vers IPv6 se fait très lentement pour de nombreuses raisons (matériels ou applications souvent incompatibles avec IPv6, véritable phase de transition) mais est irrémédiable à cause des limitation d'IPv4 et surtout la pénurie d'adresses (forte demande de pays émergeants, besoin d'adresses pour les matériels téléphoniques ou électroménagers, etc.).
Nous sommes actuellement dans une phase où la connectivité IPv6 commence à se développer en parallèle de la connectivité IPv4 : une longue phase où IPv4 et IPv6 seront disponibles sera bien sûr nécessaire avant... une disparition complète d'IPv4 !

Pour en savoir plus sur IPv6, on se réfèrera à la nombreuse documentation sur Internet, notamment http://mboucey.free.fr/Linux+IPv6-HOWTO-fr/,
ou bien au livre "IPv6 Essentiels" (Collection O'Reilly).

Attention, cependant, le protocole IPv6 n'est pas officiellement terminé et des changements interviennent encore régulièrement.

7 À propos de ce document

Je remercie Rolland Agopian et Pascal Nicolas.

Ce support de formation s'inspire (forcément) de documentations officielles, de pages Internet, de livres ou de magazines soumis à des droits d'auteurs. Dans la mesure du possible, les liens vers les sources ont été cités.

7.1 Liens / bibliographie

http://www.info.univ-angers.fr/pub/pn/poly/
http://www.univ-st-etienne.fr/monnet/formation/cours/informatique/ethernet/
http://www.eisti.fr/~bp/doc/reseaux/
http://netfilter.org/documentation/
http://www.frameip.com/
http://www.routage.org/
http://ietf.org/
http://www.rfc-editor.org/