Ecole Nationale Supérieure des Mines
SAINT-ETIENNE

Support de Cours.

Inititation aux réseaux informatiques

Enseignant :
Maxime MORGE
courriel : morge@emse.fr
page web : http ://www.emse.fr/~mmorge

Table des matières

Déroulement

- **public** : MIAS 1 ;
- **CM** : 3 séances de 2 heures ;
- TD : 3 séances de 2 heures (2 groupes) ;
- TP : 1 séance de 2 heures (2 groupes).

Chapitre 1

Principe des réseaux

Nous aborderons les grands principes régissant les équipements matériels et logiciels permettant d'échanger des données et qui forment les réseaux informatiques.

1.1 Introduction

L'usage de l'ordinateur par l'homme évolue :

1. Une machine - des hommes : de gros serveurs, des cartes perforées, un opérateurs...
2. Une machine - un homme : l'ère du *Personnal Computer*, qui l'est de moins en moins. Un ordinateur sans connexion est un ordinateur mort...
3. Des machines - un homme : Un réseau personnel (*Personnal Area Network*) interconnecte (souvent par des liaisons sans fil) des équipements personnels comme un ordinateur portable, un agenda électronique...

On peut faire une première classification des réseaux à l'aide de leur taille :

- réseau local (*Local Area Network*) : réseau d'une entreprise. Il peut se développer sur plusieurs bâtiments et permet de satisfaire tous les besoins internes de cette entreprise (1m à quelques kms).
- réseau métropolitain (*Metropolitan Area Network*) : relie différents sites d'une université ou d'une administration, chacun possédant son propre réseau local (quelques kms à quelques dizaine de kms).
- réseau étendu (*Wide Area Network*) permet de communiquer à l'échelle d'un pays, ou de la planète entière, les infrastructures physiques pouvant être terrestres ou spatiales à l'aide de satellites de télécommunications (plusieurs centaines de km).

On peut différencier les réseaux selon leur topologie comme illustré dans la figure 1.1.1 :

- mode point à point : les équipements partagent le même support (sensible aux pannes de support) ;
- mode diffusion : un lien spécifique entre chaque noeud (sensible aux pannes d'équipements).

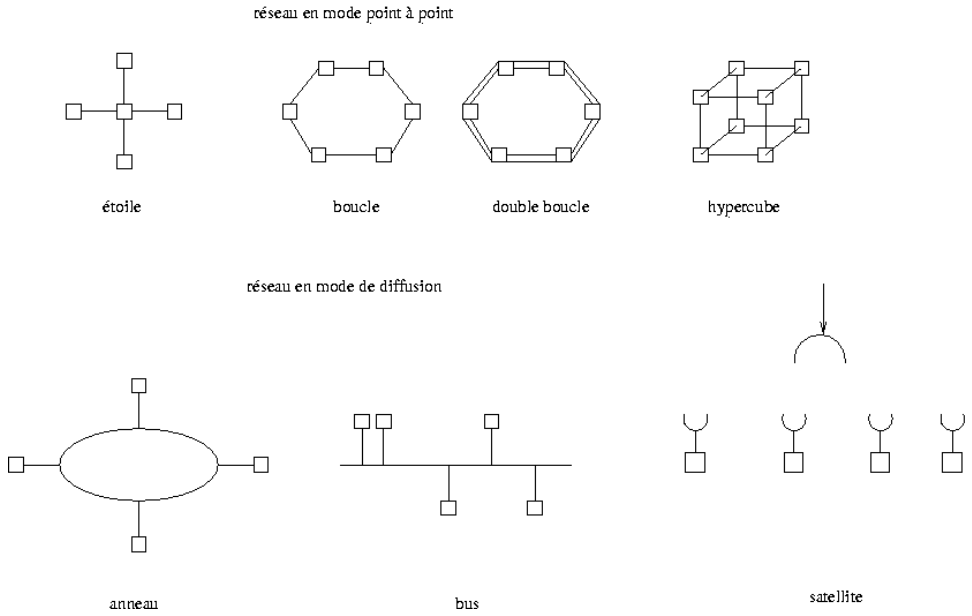

Fig. 1.1.1: Topologie des réseaux

De cette topologie, dépendent :
- le coup du câblage ;
- la fiabilité du réseau ;
- le routage.

Un réseau peut fonctionner selon deux modes :
- *avec* connexion (sécurisation, Q.S.) ;
- *sans* connexion (sans lourdeur,Multi-point).

il existe plusieurs types de commutation :
- la commutation de circuits : création de circuit entre un émetteur et un récepteur (ex : téléphone). La réservation de circuit a un coût ;
- la commutation de messages : les messages sont reçus et ré-expédiés. Il faut pouvoir contrôler le flux des messages pour éviter la saturation du réseau ;
- la commutation de paquets : les messages sont découpés, transmis par des routes différentes et reconstitués dans le bon ordre.

1.2 Le modèle OSI

Dans les années 70, l'organisme ISO (*International Standard Organization*) a développé un modèle dans le but de répondre à l'ensemble de ces questions indépendamment les unes des autres, par conséquence, d'interconnecter des réseaux selon une norme OSI (*Open System Interconnexion*).

Chaque COUCHE ou INTERFACE (de niveau n) fournit un certain nombre de FONCTIONNALITÉS mises à la disposition de la couche immédiatement supérieure (de niveau $n+1$) et permet de communiquer avec la couche de même niveau (n) d'un autre dispositif selon un PROTOCOLE qui spécifie la séquence des actions possibles. Ce modèle, par la définition de différents niveaux d'ABSTRACTION, permet à une couche de S'AFFRANCHIR des problèmes adressés par les couches de niveau inférieur. Cette structuration a un coût. C'est un modèle théorique qui se distingue des implémentations qui ont pu en être réalisées.

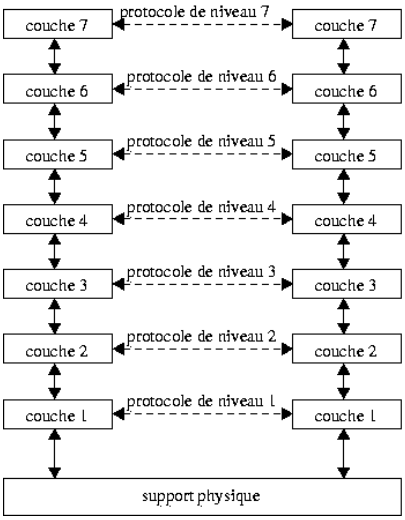

Fig. 1.2.1: Le modèle OSI

Nous allons détailler les caractéristiques de chacune de ces couches en précisant les services et fonctions attendues.

1.3 La couche physique

Définition 1 *La couche physique fournit les moyens* MÉCANIQUES, ÉLECTRIQUES, FONCTIONNELS *et* PROCÉDURAUX *nécessaires à l'activation, au maintien et à la désactivation des connexions physiques destinées à la* TRANSMISSION DE BITS *entre deux entités de liaison de données.*

Une liaison physique peut être :
- **simplex** : unidirectionnelle, on parle alors d'émetteur et de récepteur (ex : monitoring) ;
- **half-duplex** : bidirectionnelle à l'alternat, les rôles d'émetteur et de récepteur peuvent être interverti (ex : tallkie-walkies) ;
- **full-duplex** : bidirectionnelle simultanée, les entités peuvent émettre et recevoir dans le même temps (ex : téléphone).

la transmission des bits peut s'effectuer :
- **en parallèle/série** : les bits sont-ils tous envoyer en même temps ?
- de façon **synchrone/asynchrone** : les horloges sont-elles synchronisées ?

Quel que soit le mode de transmission retenu, l'émission est toujours cadencée par une horloge. Le *baud* représente le nombre de top d'horloge en une seconde Le signal peut prendre 2^n valeurs distinctes on dit alors que sa **valence** est de n, ainsi à chaque top d'horloge n bits peuvent être transmis simultanément et si le débit de la ligne est de $x$bauds il est en fait de $n.x\text{bit}/s$.

1.3.1 Transmission en 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). Voici quelques exemples de codage de l'information (cf figure 1.3.1) pour une transmission en bande de base :
- **le code tout ou rien** : c 'est le plus simple, un courant nul code le 0 et un courant positif indique le 1 ;
- **le code bipolaire** : c'est aussi un code tout ou rien dans lequel le 0 est représenté par un courant nul, mais ici le 1 est représenté par un courant alternativement positif ou négatif ;
- **le code NRZ** (non retour à zéro) : pour éviter la difficulté à obtenir un courant nul, on code le 1 par un courant positif et le 0 par un courant négatif ;
- **le code RZ** (retour à zéro) : le 0 est codé par un courant nul et le 1 par un courant positif qui est annulé au milieu de l'intervalle de temps prévu pour la transmission d'un bit ;
- **le code Manchester** (codage biphase) : ici aussi le signal change au milieu de l'intervalle de temps associé à chaque bit. Pour coder un 0 le courant sera négatif sur la première moitié de l'intervalle et positif sur la deuxième moitié, pour coder un 1, c'est l'inverse. Autrement dit, au milieu de l'intervalle il y a une transition de bas en haut pour un 0 et de haut en bas pour un 1 ;
- **le code Miller** : on diminue le nombre de transitions 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.

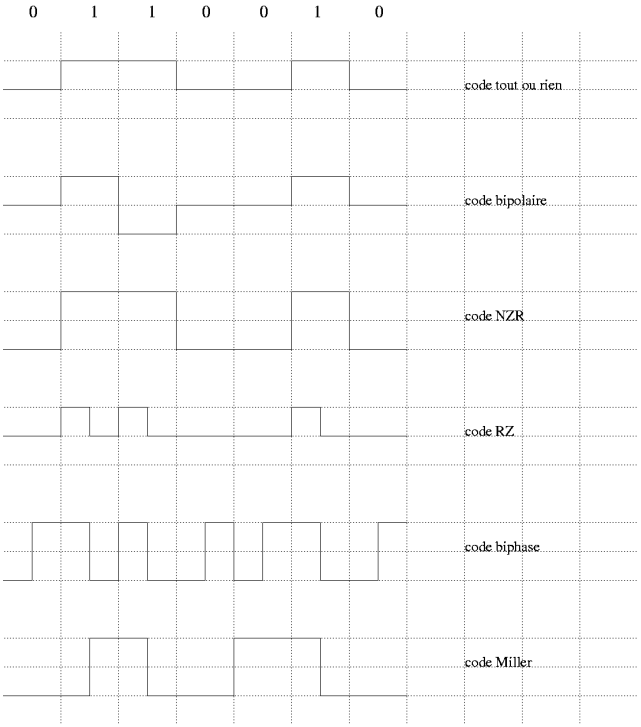

Fig. 1.3.1: Différents codage en bande de base

1.3.2 Les supports de transmission

L'objectif de la couche 1 du modèle OSI est aussi de fixer les caractéristiques des matériels utilisés pour relier physiquement les équipements d'un réseau. Nous décrivons succinctement quelques uns des supports de transmission les plus usités :

- **le câble coaxial** : il est constitué d'un cœur qui est un fil de cuivre. Ce cœur est dans une gaine isolante elle-même entourée par une tresse de cuivre, le tout est recouvert d'une gaine isolante. On le rencontre dans sa version 10 Base2 (ou Ethernet fin 10 Mbit/s sur 200 m maximum) pour la réalisation de réseaux locaux à topologie en bus, relié sur le poste à l'aide connecteur BNC ;
- **la paire torsadée** : c'est un câble téléphonique constitué à l'origine de deux fils de cuivre isolés et enroulés l'un sur l'autre. Actuellement on utilise plutôt des câbles constitués de 2 ou 4 paires torsadées. chaque extrémité d'un tel câble étant muni d'une prise RJ45. Son intérêt principal est que cette même paire torsadée peut servir au réseau téléphonique, au réseau informatique et vidéo d'une même entreprise et de plus elle pourra être utilisée ultérieurement pour évoluer vers des réseaux 100 Base T et même Gigabits ;

- **la fibre optique** : c'est un support d'apparition plus récente (cf figure 1.3.2). Elle permet des débits de plusieurs Gbit/s sur de très longues distances. En plus de ses capacités de transmission, ses grands avantages sont son immunité aux interférences électromagnétiques et sa plus grande difficulté d'écoute ;
- **les liaisons sans fil** : ce sont des liaisons infrarouges, laser ou hertziennes sur de courtes distances et grâce aux faisceaux hertziens pour les liaisons satellitaires. Les débits sont très élevés mais les transmissions sont sensibles aux perturbations et les possibilités d'écoute sont nombreuses.

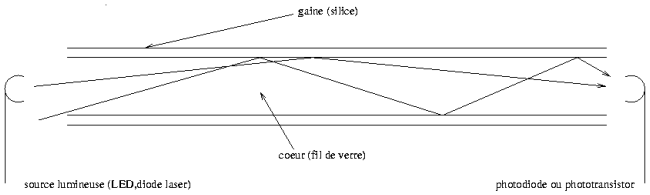

Fig. 1.3.2: Schéma d'une fibre optique

1.4 La couche liaison

La couche liaison de données fournit les moyens **fonctionnels** et **procéduraux** nécessaires à l'*établissement*, au *maintien* et à la *libération* des connexions de liaison de données entre entités **adjacentes** du réseau. Par adjacentes, nous entendons que les machines sont physiquement connectés par un canal de transmission. La couche liaison de données *détecte et corrige*, si possible, les erreurs dues au support physique et signale à la couche réseau les erreurs irrécupérables. Elle supervise le fonctionnement de la transmission et définit la *structure syntaxique des trames*, *la manière d'enchaîner les échanges* selon un protocole normalisé ou non.

La couche liaison de données peut fournir trois types de services :

1. le service sans connexion et sans acquittement (trafic en temps réel) ;
2. le service sans connexion et avec acquittement (plus fiable) ;
3. le service orienté connexion (fournit l'équivalent d'un canal fiable) ;

La couche *Liaison de Données* découpe un train de bits en trame et calcul un ensemble de bits de contrôle pour chaque trame. Pour réaliser ceci, on peut utiliser l'une des méthodes suivantes :

- **compter les caractères d'une trame** : On utilise un caractère "nombre de caractère". Le faits que ces caractères peuvent être altérés est une limite de cette méthode ;

- **utiliser des caractères de début et de fin de bits** : DLE STX (*Data Link Escape Start of TeXt*), DLE ETX (*Data Link Escape End of TeXt*). On doit adjoindre un caractère DLE devant tout DLE qui apparaît dans les données. La nécessité d'utiliser un codage ASCII est une limite à cette méthode ;
- **utiliser des fanions de début et de fin de bit** : idem que précédemment avec un fanion 011111110. Un utilise un bit de transparence 0 pour casser une séquence de 6 bits 1 consécutifs.

Nous allons examiner les différentes techniques de détection et correction d'erreur (changement de 1 par 0 ou vice-versa).

1.4.1 Détection et correction d'erreurs

Le taux d'erreurs de transmission est de l'ordre de :
- 10^{-5} sur une ligne téléphonique ;
- 10^{-7} à 10^{-8} sur un coaxial ;
- 10^{-10} à 10^{-12} sur une fibre optique.

Les messages acheminés par le support physique sont des blocs de n bits. avec un taux d'erreur e :
- quelle est la probabilité qu'un bloc soit indemne ? $(1-e)^n$;
- quelle est la probabilité qu'un blocs soit éronné ? $1-(1-e)^n$.

Les techniques de codes correcteurs ou codes détecteurs d'erreurs transforment la suite de bits à envoyer en lui ajoutant de l'information à base de **bits de redondance** ou **bits de contrôle**. Le récepteur se sert de cette information ajoutée pour **déterminer** si une erreur s'est produite et pour la **corriger** si la technique employée le permet.

La parité est une technique qui consiste à ajouter à chaque bloc de i bits émis un bit de parité de telle sorte que parmi les $i+1$ bits émis le nombre de bits à 1 soit toujours pair.

bloc	bit de parité	bloc émis
1000001	0	10000010
0110100	1	01101001
1110110	1	11101101

Exemple 1

Cette technique ne permet pas de détecter 2n erreurs dans le même bloc de bits transmis, car dans ce cas la parité ne sera pas changée. Si à la réception, le nombre

de bit à 1 est dans le cas d'une parité paire est impaire, il y a eu une erreur de transmission. Il doit donc avoir ré-émission.

Le Code de Hamming. Soit x, y deux mots sur un alphabet , N la longueur du codage de ces mots. x_i (respectivement y_i) désigne le $i^{\text{ème}}$ bit de x (respectivement y). La distance d'édition $d(x,y) = \sum_{i=1}^{N}(x_i - y_i)_{[2]}$ permet de compter le nombre de bits qui diffèrent entre x et y. On définit alors la distance de Hamming d'un alphabetA par $d_H = \inf_{(x,y)\in\mathcal{A}^2, x\neq y} d(x,y)$

alphabet	k	l	m	n
A	0	01	10	11
B	00000	01111	10110	11001
	00001	01110	10111	11000
caractères	00010	01101	10100	11011
erronés	00100	01011	10010	11101
possibles	01000	00111	11110	10001
	10000	11111	00110	01001

Tab. 1.4.1: alphabet A et B

Exemple 2 *Par exemple, on peut coder les 4 mots de la manière illustrée dans la table 2. Ainsi si un bit (parmi les 5 émis) est erroné on sait quand même déterminer quel caractère a été émis, car comme on peut le voir dans la table 2 la modification d'un bit ne peut pas faire passer d'un caractère initial à l'autre. On a des ensembles d'"erreurs possibles" totalement disjoints. Par contre la modification de 2 bits dans cet exemple peut amener à des confusions et à l'impossibilité de corriger les erreurs.*

1Calculer les distances entre les mots dans les alphabets A et B.
Calculer la distance de Hamming de l'alphabet A et de B.

2 Chaque erreur sur un bit d'un caractère x donne un caractère x' tel que $d(x,x') = 1$, donc pour pouvoir détecter une seule erreur il faut que $d_H \geq 2$ et pour la corriger il faut que $d_H \geq 3$. Pour corriger 2 erreurs il faut que $d_H \geq 5$. D'une manière générale on détecte et corrige n erreurs quand la distance de Hamming est $2n+1$.

1.5 La couche réseau

La couche réseau assure toutes les fonctionnalités de relai et d'amélioration de services entre entité de réseau, à savoir :
• l'adressage et le routage ;

- le contrôle de flux ;
- la détection et correction d'erreurs non réglées par la couche 2.

1.5.1 le contrôle de flux

Le contrôle de flux consiste à gérer les paquets pour qu'ils transitent le plus rapidement possible entre l'émetteur et le récepteur. Il cherche à éviter les problèmes de congestion du réseau qui surviennent lorsque trop de messages y circulent. On peut citer les quelques méthodes suivantes :

- **Le contrôle par crédit** (mode sans connexion) : on distribue N jetons pour s'assurer de limiter le trafic à N paquets. Le problème réside dans leur distribution. On peut dédier les jetons aux nœuds. Dans cette situation, les jetons sont renvoyés à la réception à l'expéditeur du paquets ;
- **Le contrôle par fenêtre** (mode avec connexion)) : les paquets sont numérotés *modulo 8* et contiennent deux compteurs :
  - ⋆ $P(S)$ un compteur de paquets émis ;
  - ⋆ $P(R)$ un compteur de paquets reçus.

  L'émetteur n'est autorisé à émettre que les paquets inclus dans la fenêtre, c'est-à-dire les paquets dont le compteur de paquet émis est tel que :

$$P(R) \leq P(S) \leq P(R) + W$$

  Comme le montre la figure, si $W = 4$ et $P(R) = 1$, l'émetteur peut envoyer les paquets 1, 2, 3 et 4.

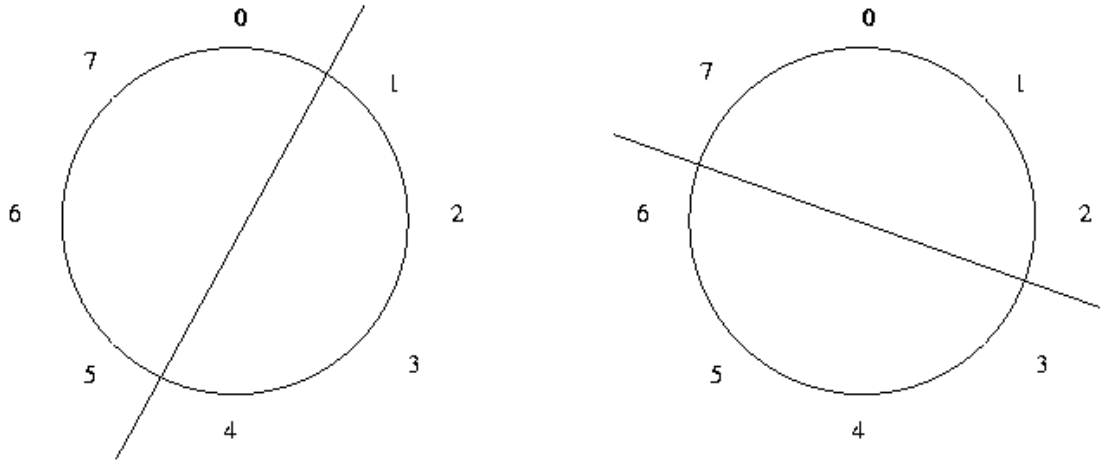

Fig. 1.5.1: Position initiale et finale de la fenêtre

1.5.2 le routage

le routage c'est à dire le chemin d'un transmission (une connexion ou plusieurs *paquets*) est établi à partir de table de routage qui dépendent :

- du coût des liaisons ;
- du coût de passage dans un nœud ;
- du débit demandé ;
- du nombre de nœuds à traverser ;
- de la sécurité de transport de certains paquets ;
- de l'occupation des mémoires des nœuds de commutation,
- ...

Le routage peut être :

- **centralisé** : géré par un nœud particulier du réseau qui reçoit des informations de chacun des nœuds du réseau et leur envoie leur table de routage ;
- **décentralisé** : le réseau ne possède pas de centre de contrôle et les règles de passage d'un paquet (paquet d'appel pour établissement d'un circuit virtuel) sont :
  - ⋆ *l'inondation* : à la réception d'un paquet celui-ci est renvoyé sur toutes les lignes de sortie. Cette technique simpliste et rapide est efficace dans les réseaux à trafic faible et où le temps réel est nécessaire mais elle est pénalisante en flux de données, inadaptée aux réseaux complexe et au circuit virtuel.
  - ⋆ *la technique "hot patatoes"* : un paquet reçu est renvoyé le plus tôt possible par la première ligne de sortie vide. On améliore ce principe en affectant des coefficients à chaque ligne de sortie en fonction de la destination voulue. On tient compte de l'état des nœuds voisins, sans utiliser de paquet de contrôle, mais simplement en comptabilisant le nombre de paquets reçus de chacun d'eux. Ils peuvent également envoyer de manière synchrone ou asynchrone un compte-rendu de leur état, permettant ainsi de choisir la "meilleure" ligne à un instant donné. Mais ceci reste local et une panne du réseau localisée au-delà du premier nœud voisin ne pourra pas être prise en compte.
  - ⋆ *le routage adaptatif* à la fois dans l'espace et dans le temps demande, de la part de chaque nœud, une connaissance complète du réseau. Les différents nœuds s'échangent donc des messages, mais si chacun envoie des messages à tous les autres le trafic va augmenter de manière beaucoup trop grande. C'est pourquoi un nœud ne transmet un compte-rendu qu'à ses voisins immédiats qui doivent en tenir compte dans leur propre compte-rendu.

Dans le cas du routage centralisé, la mise à jour des tables de routage peut se faire de manière :

- **fixe** : en fait il n'y a pas de mise à jour, la table de routage est fixée une fois pour toute en fonction de la topologie du réseau ;
- synchrone : toutes les tables sont mises à jour au même moment par le centre

de contrôle qui reçoit des informations de la part de tous les nœuds à intervalles réguliers (toutes les 10 sec par exemple) ;
- **asynchrone** : les tables sont mises à jour indépendamment les unes des autres dans certaines parties du réseau, chaque nœud envoyant un compte-rendu de son état au centre de contrôle lorsqu'il observe des changements significatifs ;

Pour éviter les congestions suite à des erreurs de routage, ou d'adressage les paquets sont munis de compteur qui s'incrémente à chaque changement de nœud et qui est détruit au delà d'une certaine valeur ce qui limite sa durée de vie dans le réseau.

1.6 La couche transport

La couche transport assure un transfert de données transparents entre entités en les déchargeant des détails d'exécution. Elle a pour rôle d'**optimiser** l'utilisation des services réseaux disponibles afin d'assurer au **moindre coût** les performances requises par la couche session.

C'est une couche intermédiaire qui se définit par la notion de **Qualité de Service** (Q & S). La qualité est évaluée sur certains paramètres avec trois types de valeurs possibles : *préféré, acceptable et inacceptable* qui sont choisis lors de l'établissement d'une connexion. La couche transport surveille alors ces paramètres pour déterminer si la couche réseau sous-jacente assure la qualité de service demandée. Les différents paramètres de la qualité de service sont :

- **le temps d'établissement de la connexion transport** : c'est la durée qui s'écoule entre une demande de connexion et sa confirmation. Plus ce délai est court meilleure est la qualité de service ;
- **la probabilité d'échec d'établissement** : elle mesure la chance qu'une connexion ne puisse s'établir dans un délai maximum défini en considérant plutôt les problèmes d'engorgement de réseau ;
- **le débit de la liaison** : il mesure le nombre d'octets utiles qui peuvent être transférés en une seconde, ce débit est évalué séparément dans les deux sens ;
- **le temps de transit** : il mesure le temps écoulé entre le moment où l'utilisateur du service de transport envoie un message et celui où l'entité de transport réceptrice le reçoit, ce temps est évalué séparément dans les deux sens ;
- **le taux d'erreur résiduel** : est le rapport entre le nombre de messages perdus ou mal transmis et le nombre total de messages émis au cours d'une période considérée. Ce nombre, en théorie nul, a une valeur faible ;
- **la probabilité d'incident de transfert** : elle mesure le bon fonctionnement du service transport. Lorsqu'une connexion est établie, un débit, un temps de transit, un taux résiduel d'erreurs sont négociés. La probabilité d'incident mesure

la fraction de temps durant laquelle les valeurs fixées précédemment n'ont pas été respectées ;

- **le temps de déconnexion** : il mesure la durée s'écoulant entre une demande de déconnexion émise et la déconnexion effective du système distant ;
- **la probabilité d'erreur de déconnexion** : c'est le taux de demandes de déconnexion non exécutées pendant le temps maximum défini ;
- **la protection** : elle est définie comme la possibilité de se prémunir contre les intrusions passives (interférences sur une même ligne) et actives (écoute et modification des données transmises) ;
- **la priorité** : elle permet à l'utilisateur de privilégier certaines transmissions par rapport à d'autres ;
- **la résiliation** : c'est la liberté laissée à la couche transport de décider elle-même de la déconnexion suite à un problème.

Lorsqu'une connexion est demandée, les valeurs désirées et minimales sont spécifiées. Une procédure, la **négociation des options**, a lieu.

1.7 Les couches hautes : session, présentation, application

Les couches hautes du modèle OSI offrent des services orientés vers les utilisateurs, alors que les couches dîtes basses sont concernées par des communications fiables de bout en bout.

1.7.1 La couche session

La couche session fournit aux entités de la couche présentation les moyens d'**organiser** et **synchroniser** les dialogues et les échanges de données. C'est une couche **mince** offrant relativement peu de services. Malgré cela nous ne détaillerons pas tous ces services.

La principale fonction de la couche session est de fournir aux utilisateurs (entité de la couche de présentation ou processus de la couche application) les moyens d'établir des connexions appelées **sessions** et d'y transférer des données en bon ordre.

On s'intéresse particulièrement au fonctionnement d'une session en mode connecté. Bien que très similaires, une session et une connexion établie par la couche transport ne sont pas nécessairement identiques. On distingue les trois cas de figure suivants :

1. il y a correspondance exacte entre une session et une connexion de transport ;

2. plusieurs sessions successives sont établies sur une seule et même connexion de transport (ex : agence de voyage) ;
3. plusieurs connexions de transport successives sont nécessaires pour une seule et même session (ex : connexion de la couche transport tombe en panne) ;

Contrairement à la couche réseaux qui peut multiplexer plusieurs sessions de la couche transport sur une même connexion, la couche transport ne peut multiplexer plusieurs sessions de la couche session sur une même connexion de transport. Le multiplexage sert à réduire le coût où à améliorer les performances, fonctions qui sont du ressort de la couche transport.

1.7.2 La couche présentation

Chaque ordinateur ayant un mode de représentation propre, il est nécessaire de prévoir **des mécanismes de conversion** afin de s'assurer que des machines différentes puissent se comprendre. Le terme "présentation" est donc mal choisi. On devrait plutôt parlé de couche de "représentation". C'est à ce niveau que peuvent être implantées des techniques de compression et de chiffrement de données.

1.7.2.1 Compression de données

La compression des données consiste à réduire la taille de la représentation des données que cela soit pour minimiser l'espace disque qu'il occupe ou le temps de transfert. On distingue deux types d'approches :

1. **les méthodes réversibles** : sans perte ;
2. **les méthodes irréversibles** : avec pertes.

On mesure l'efficacité de la compression par le taux de compression :

$$\sigma = \frac{\text{nb bits de la représentation compressée}}{\text{nb de bits de la représentation originale}} \tag{1.7.1}$$

Il est plus difficile de mesurer la qualité de la compression. Soient les données $n_1, ..., n_N$ et $n'_1, ..., n'_N$ ces même données obtenues après compression puis décompression. Il y a perte de données si les données obtenues ne sont pas identiques. On définit l'erreur quadratique moyenne (EQM) de la manière suivante :

$$\text{EQM} = \sum_{i=1}^{N} (n'_i - n_i)^2 \tag{1.7.2}$$

Nous nous intéresserons ici uniquement aux méthodes statistiques sans pertes. On a pour habitude de coder tous les caractères avec le même nombre de bits. C'est le cas

de la table ASCII. Ce n'est certainement pas le codage le plus efficace possible. Il est en effet plus efficace d'utiliser des codes plus courts pour des valeurs fréquentes et de réserver des codes plus longs pour les valeurs moins fréquentes. On parle de codage à longueur variable (*VLC Variable Lenght Code*).

Comment coder `abracadabra` ?

lettre	occurrence	fréquence	codage ASCII	codage naïf	codage de Huffman	codage de Shanon-Fano
a	5	43 %	1100001	0	0	1
b	2	19 %	1100010	00	10	01
r	2	19 %	1110010	11	110	001
c	1	8,5 %	1100011	01	1110	0000
d	1	8,5 %	1100100	10	1111	0001

Tab. 1.7.1: codage

On s'aperçoit que lors du décodage du code dit naïf, il y a ambiguïté (00 signifie B ou AA ?). Afin de résoudre ce problème, il suffit de préfixer le codage. Le `codage Huffman` est l'une des solutions possibles (cf figure 1.7.1). Le codage de Shanon-Fano en est une autre (cf figure 1.7.2 et table 1.7.2).

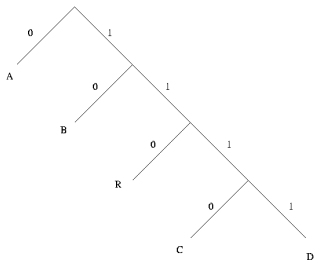

Fig. 1.7.1: Arbre binaire du codage de Huffman

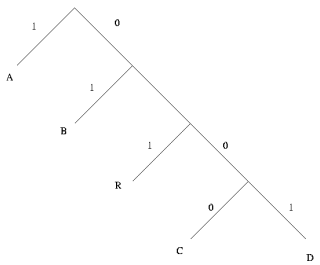

Fig. 1.7.2: Arbre binaire du codage de Shanon-Fano

lettre	occurrence	fréquence	bit 1 de poids fort	bit 2	bit 3	bit 4
a	5	43 %	1	∅	∅	∅
b	2	19 %	0	1	∅	∅
r	2	19 %	0	0	1	∅
c	1	8,5 %	0	0	0	0
d	1	8,5 %	0	0	0	1

Tab. 1.7.2: codage de Shanon-Fano

1.7.2.2 Cryptage des données

La cryptographie désigne l'ensemble des techniques permettant de chiffrer des messages, c'est-à-dire permettant de les rendre inintelligibles sans une action spécifique, basée le plus souvent sur des opérations arithmétiques.

Le cryptage se fait généralement à l'aide d'une clef de chiffrement, le décryptage nécessite quant à lui une clef de déchiffrement. On distingue généralement deux types de clefs :

- **les clés symétriques** : il s'agit de la même clé utilisée pour le chiffrement ainsi que pour le déchiffrement ;
- **les clés asymétriques** : il s'agit de clés différentes utilisées pour le chiffrement et pour le déchiffrement.

La cryptographie a pour but de garantir :

- la **confidentialité** de l'information : consiste à rendre l'information inintelligible à d'autres personnes que les acteurs de la transaction ;
- l'**intégrité** de l'information : les données n'ont pas été altérées durant la communication (de manière fortuite ou intentionnelle) ;
- l'**authentification** : elle consiste à assurer l'identité de l'émetteur du message ;
- La **non-répudiation** de l'information : elle garantie qu'aucun des correspondants ne pourra nier la transaction.

Ce chiffrement à clé publique (cf figure 1.7.3) est basé sur une fonction facile à calculer dans un sens, appelée **fonction à trappe à sens unique** (*one-way trapdoor function*), mais qui est mathématiquement très difficile à inverser sans la clé privée. Ce chiffrement garantie donc la confidentialité de l'information.

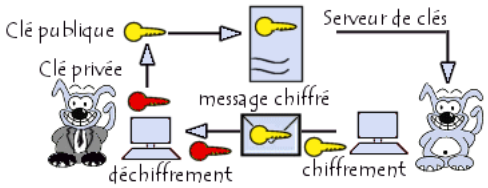

Fig. 1.7.3: Principe du chiffrement à clé publique

La notion de signature électronique permet de garantir l'intégrité, l'authentification et la non-répudiation de l'information. Une **fonction de hachage**, parfois appelée fonction de condensation, est une fonction permettant d'obtenir un **condensé**, appelé aussi haché, d'un texte, c'est-à-dire une suite assez courte de caractères représentant le texte qu'il condense. La fonction de hachage doit être telle qu'elle associe un et un seul haché à un texte en clair (cela signifie que la moindre modification du document entraîne la modification de son haché). D'autre part, il doit s'agir d'une fonction à sens unique (*one-way function*) afin qu'il soit impossible de retrouver le

message original à partir du condensé. Actuellement, l'algorithme de hachage le plus utilisés est MD5. En expédiant un message accompagné de son haché, il est possible de garantir l'intégrité d'un message (cf figure 1.7.4).

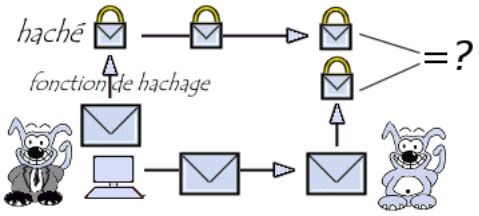

Fig. 1.7.4: Principe du hachage

Pour garantir l'authentification du message, il suffit à l'expéditeur de chiffrer, on dit généralement signer, le condensé à l'aide de sa clé privée, le haché signé est appelé **sceau** et d'envoyer le sceau au destinataire (cf 1.7.5).

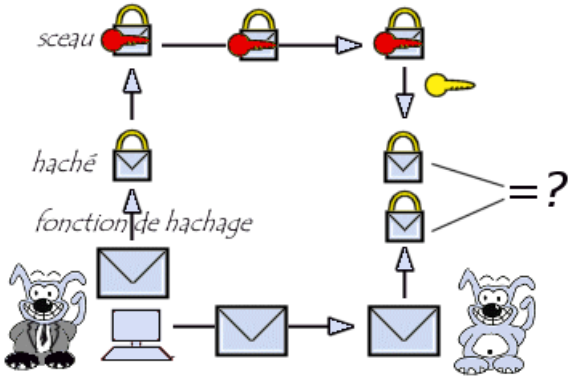

Fig. 1.7.5: Principe du scellement

La publication de l'ensemble des clés publiques est assurée à travers un annuaire électronique ou un site web. Toutefois ce mode de partage a une grande lacune : rien ne garantit que la clé est bien celle de l'utilisateur a qui elle est associée. En effet un pirate peut corrompre la clé publique présente dans l'annuaire en la remplaçant par sa clé publique. Ainsi, le pirate sera en mesure de déchiffrer tous les messages ayant été chiffrés avec la clé présente dans l'annuaire. Les clés publiques s'accompagnent d'un certificat délivré par une autorité. Ces informations sont signées par l'autorité de certification, cela signifie qu'une fonction de hachage crée une empreinte de ces informations, puis ce condensé est chiffré à l'aide de la clé publique de l'autorité de certification, clé publique ayant été préalablement largement diffusée ou elle même signée par une autorité de niveau supérieur. Ce système permet de garantir la non-répudiation de l'information.

1.7.3 La couche application

La couche application est constituée par l'ensemble des programmes courants à disposition de l'utilisateur nécessitant des ressources réseaux et définis selon certaines normes pour leur inter-operabilité.

- transfert de fichier ;
- partage d'espace disque ;
- partage de mémoire ;
- messagerie électronique ;
- terminaux virtuel ;
- appel de procédure à distance ;
- ...

L'ensemble de ces applications ainsi que leur fonctionnalités (synchronisation, cohérence, intégrité) seront décrites dans le chapitre suivant à l'aide d'exemples.

Chapitre 2

Le réseau Internet

On désigne par **internet**, l'interconnexion des différents réseaux (*internetworking*) dans un unique environnement de communication homogène dont le point commun est de fonctionner en suivant les protocoles TCP/IP (*Transmission Control Protocol/Internet Protocol*) dont nous allons étudier le fonctionnement.

Un peu d'histoire :
- au milieu des années 70, l'ARPA (*Advanced Research Project Agency*) développe un réseau pour relier ses centres de recherches militaires ;
- début des années 80, l'ARPA adopte les nouveaux protocoles TCP/IP et subventionne l'université de Berkeley pour qu'elle intègre TCP/IP à son système d'exploitation Unix (BSD) ;
- au milieu des années 80, sous l'impulsion de la NSF (National Science Foundation), la quasi totalité des départements d'informatique des universités américaines commencèrent à se doter de réseaux locaux ;
- en 1993, des sociétés commerciales se connecte, c'est le début du **web**...

Les protocoles TCP/IP sont structurés en **quatre couches** qui s'appuient sur une couche matérielle comme illustré dans la figure 2.0.1 :
- la **couche de liens** (couche OSI 1 et 2) : c'est l'interface avec le réseau et est constituée d'une carte réseau (le plus souvent une carte Ethernet) et d'un "driver" pour cette carte, c'est à dire un certain nombre de fonctionnalités intégrées au système d'exploitation qui permet de gérer le fonctionnement de cette carte ;
- la **couche IP** (couche OSI 3) : elle gère la circulation des paquets à travers le réseau en assurant leur routage. Elle comprend aussi le protocoles ICMP (*Internet Control Message Protocol*) ;
- la **couche transport** (couche OSI 4) : elle assure tout d'abord une communication de bout en bout en faisant abstraction des machines intermédiaires entre l'émetteur et le destinataire. Elle s'occupe de réguler le flux de données. Les pro-

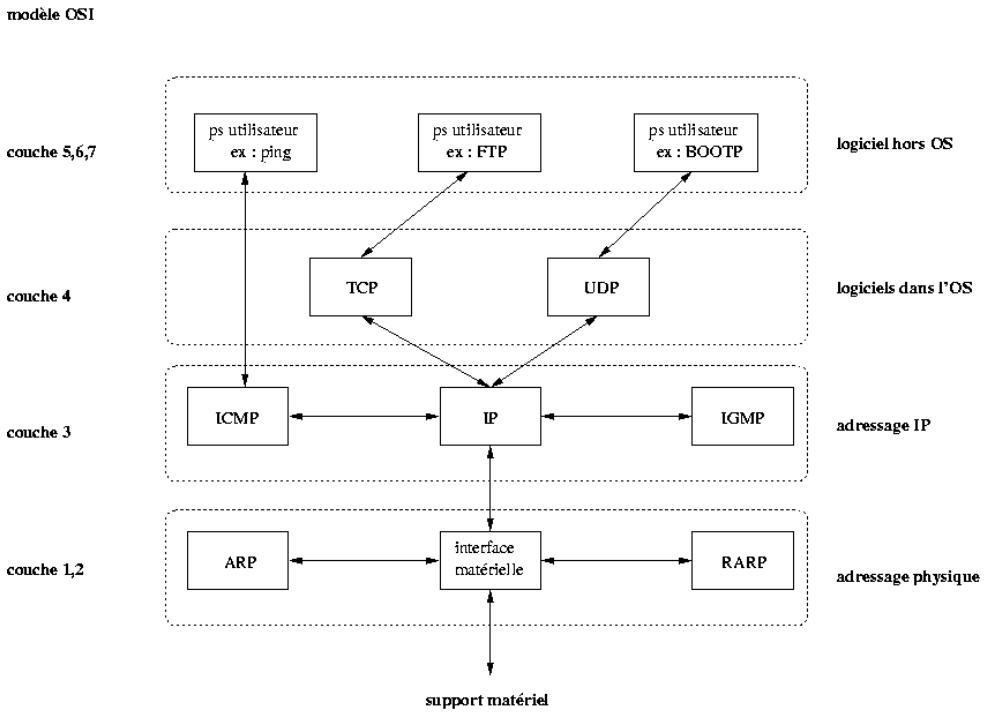

Fig. 2.0.1: Architecture d'une pile TCP/IP

tocoles TCP et UDP (*User Datagram Protocol*) seront détaillés par la suite (cf section 2.5).

- la **couche application** (couche OSI 5,6 et 7) : c'est celle des programmes utilisateurs comme telnet (connexion à un ordinateur distant), FTP (*File Transfert Protocol*), SMTP (*Simple Mail Transfert Protocol*), etc...

Cette architecture permet surtout de réaliser l'interconnexion de réseaux éventuellement **hétérogènes**.

On désigne par **encapsulation**, le fait que chaque couche concatène des informations sous la forme d'en-têtes ou de remorques (cf figure 2.2.1).

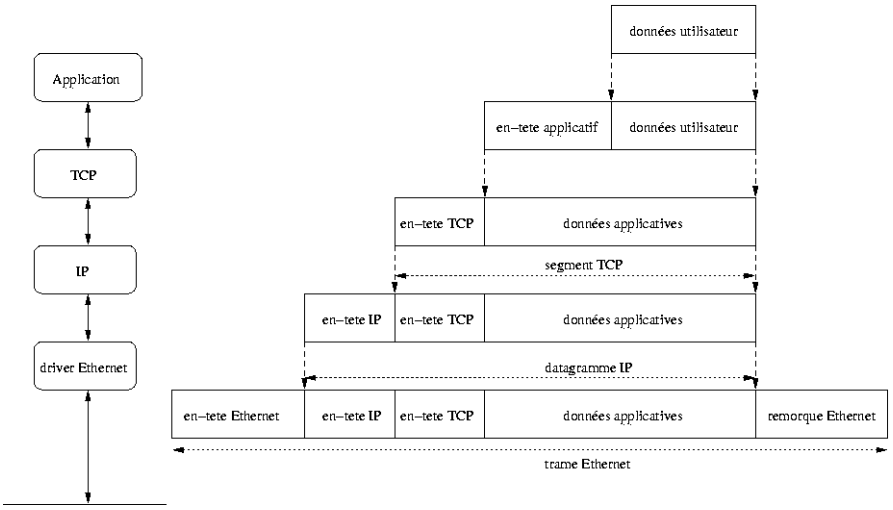

Fig. 2.0.2: Encapsulation des données par la pile des protocoles TCP/IP

2.1 Adressage

L'**adresse IP** est un nombre binaire de 32 bits qui identifie, de manière unique, un noeud (ordinateur, imprimante, routeur, etc.) d'un réseau TCP/IP. Les adresses IP sont généralement exprimées dans un format décimal pointé, fait de quatre nombres en base 10, compris entre 0 et 255, séparés par des points.

Exemple 3 *adresse IP au format décimal pointé : 195.83.83.36*
adresse IP au format binaire : 11000011 01010011 01010011 00100100

Au même titre que l'adresse physique d'un personne est constituée de la ville et de l'adresse de l'individu dans cette ville, un adresse IP se décompose en un adresse de réseau (ID réseau) et en une adresse d'hôte (ID hôte). Il existe trois classes de réseaux : A,B,C (cf tableau 2.1.1).

classe	adresse IP en notation décimale pointé	ID réseau	ID hôte
classe A	0wwwwwww.xxxxxxxxx.yyyyyyyyy.zzzzzzzzz.	w.	x.y.z.
classe B	10wwwwww.xxxxxxxxx.yyyyyyyyy.zzzzzzzzz.	w.x.	y.z.
classe C	110wwwww.xxxxxxxxx.yyyyyyyyy.zzzzzzzzz.	w.x.y.	z.

Tab. 2.1.1: Les différentes classes de réseaux

La classe définit aussi le nombre maximal de réseaux et le nombre maximal d'hôtes par réseau.

classe	plage ID réseau (1er octet)	nombre de réseaux	nombre d'hôtes ds le réseau
classe A	1-126	126	16 777 214 ($2^{24} - 2$)
classe B	128-192	16 384 (2^{14})	65 534 ($2^{16} - 2$)
classe C	192-223	2 097 152 (2^{21})	254 ($2^8 - 2$)

Tab. 2.1.2: Les différentes classes de réseaux

Un **masque de sous-réseau** est une adresse sur 32 bits qui permet de "masquer" une partie de l'adresse IP pour différencier l'ID de réseau de l'ID d'hôte. C'est le masque de sous-réseau qui permet à TCP/ IP de savoir si une certaine adresse IP se trouve sur le réseau local ou sur un autre réseau. Le masque par défaut dépend de la classe d'adresses. Un réseau peut être découpé en sous-réseau chacun disposant de son propre ID et de son propre sous-masque (cf tableau 2.1.3).

ID hôte	193.83.83.036
masque sous-réseau	255.255.255.128
adresse de diffusion	195.83.83.127
ID réseau	195.83.83.0 (mask$\wedge$host)

Tab. 2.1.3: Mon sous-réseau

2.2 Nommage

Bien que la numérotation IP à l'aide d'adresses numériques soit suffisante techniquement, il est préférable pour un humain de désigner une machine par un nom explicite. Pour faire façe à l'explosion du nombre de machine relié à l'internet, un système de base de données distribuée a été mise en place : **système de noms de domaine** (*Domain Name System*). Il fournit les correspondances entre les noms d'hôtes et les numéro IP. La responsabilité du nommage est subdivisée par niveau. Les espaces de noms de domaines sont hiérarchisés.

Les serveurs de noms peuvent fonctionner en **mode récursif** ou non, mais ils doivent toujours implanter le mode non récursif. Dans tous les cas, lorsqu'un serveur de noms reçoit une demande, il vérifie si le nom appartient à l'un des sous-domaines qu'il gère. Si c'est le cas il traduit le nom en une adresse en fonction de sa base de données et renvoie la réponse au demandeur. Sinon :

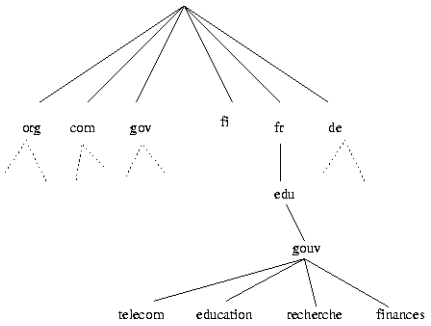

Fig. 2.2.1: Système de noms de domaine

- en mode **non-récursif** : le serveur indique au client un autre serveur de noms qui saura lui répondre ou à son tour transmettre la requête à un autre serveur ;
- en mode **récursif** : c'est le serveur qui se charge de l'interrogation successive des serveurs de noms et qui retourne finalement la réponse au client.

On peut améliorer la résolution des noms en stockant les noms fréquemment utilisés dans une mémoire **cache** afin de réduire le nombre de requêtes au serveur du niveau supérieure. Afin de palier à d'éventuelle pannes matérielles, les domaines sont munis de deux serveurs DNS : un **primaire** et un **secondaire**.

2.3 la couche de liens

2.3.1 le réseau Ethernet

Ethernet est le nom donné à une des technologies les plus utilisées pour les réseaux locaux. Elle a été inventée par Xerox au début des années 70 et normalisée par l'IEEE (Institute for Electrical and Electronics Engineers) vers 1980 sous la norme IEEE 802. il existe plusieurs technologies physiques pour établir un réseau Ethernet :

- **10 base 5** : l'ordinateur vampirise un câble coaxial ;
- **10 base 2** : l'ordinateur est relié au câble co-axial à l'aide d'une prise BNC en T ;
- **10 base T** : les ordinateurs sont reliés par l'intermédiaire d'un câble du type *pair torsadée* au travers de prise *RJ45* un même équipement *électrique* appelé **hub**. Le fonctionnement en bus est simulé alors que la typologie physique est une étoile ;

Chaque carte réseau capte toutes les trames qui sont émises sur le câble et les rejette les trames qui ne lui sont pas destinées, c'est-à-dire celles dont l'adresse de destination est égale à celle de la carte réseau. Comme il n'y a pas d'autorité centrale qui gère l'accès au câble, il est possible que plusieurs stations veuillent émettre simultanément sur le câble. On parle de **collisions**.

Chaque carte réseau est munie d'une unique adresse Ethernet de 6 octets (par exemple `00 :D0 :59 :90 :1C :60`).

une trame Ethernet est constituée :

- d'une adresse destination ;
- d'une adresse source ;
- d'un type ;
- d'un CRC.

Un **répéteur** est un équipement qui se contente de transmettre et amplifier les signaux. Un **pont** est équipement qui relie deux segments disjoint d'un même réseau Ethernet. Un **commutateur** est un pont multi-port.

2.3.2 la liaison PPP

Le protocole **PPP** (*Protocol Point to Point*) permet d'envoyer des paquets IP entre deux équipements reliés par des liaisons séries (un modem branché sur le port RS232 fera l'affaire). C'est le dispositif classiquement mis en œuvre pour relier les abonnées d'un fournisseur d'accès à internet.

2.3.3 le protocole ARP et RARP

Il est nécessaire d'établir les **correspondances bi-univoques** entre adresses IP et adresses matérielles des ordinateurs d'un réseau (Ethernet/ou autre). Ceci est l'objet des protocoles **ARP** (*Address Resolution Protocol*) et **RARP** (*Reverse Address Resolution Protocol*). ARP fournit une correspondance dynamique entre une adresse IP connue et l'adresse matérielle lui correspondant, RARP faisant l'inverse.

une requête/réponse ARP/RARP est constituée des champs suivants :
- un type de matériel ;
- un type d protocole ;
- une taille d'adresse matérielle ;
- une taille d'adresse de protocole ;
- un code opérateur pour distinguer une requête d'une réponse ;
- l'adresse Ethernet de l'émetteur ;
- l'adresse IP de l'émetteur ;
- l'adresse Ethernet destinataire ;
- l'adresse IP destinataire ;

2.4 La couche IP

le protocole IP (*Internet Protocol* est au cœur du fonctionnement de l'internet. Il assure *sans connexion* un service *non fiable* de délivrance de datagrammes IP.

Le rôle du protocole IP est centré autour des trois fonctionnalités suivantes :
- définir le format du datagramme IP qui est l'unité de base des données circulant sur Internet ;
- définir le routage dans Internet ;
- définir la gestion de la remise non fiable des datagrammes.

Un datagramme IP est constitué des champs suivants :
- numéro de version : (IPv4) ;
- longueur d'en-tête : parfois supérieure à 20 octets à cause des options ;
- un type de service : pour préciser la fiabilité, la priorité et la sécurité ;

- la longueur totale ;
- identification, drapeaux et déplacement de fragment : utilisés pour la fragmentation ;
- la durée de vie : indique le nombre maximal de routeurs que peut traverser le datagramme ;
- le type de protocole : 1 pour ICMP, 2 pour IGMP, 6 pour TCP ou 17 pour TCP ;
- bits de contrôle d'en-tête ;
- adresse IP source ;
- adresse IP destination ;
- le champ options : pour les militaires ;

Pour optimiser le débit il est préférable qu'un datagramme IP soit **encapsulé** dans une seule trame de niveau inférieur (Ethernet par exemple). Mais, comme un datagramme IP peut transiter à travers Internet sur un ensemble de réseaux aux technologies différentes il est impossible de définir une taille maximale des datagrammes IP qui permette de les encapsuler dans une seule trame quel que soit le type de réseau. On appelle la **taille maximale d'une trame** d'un réseau le **MTU** (*Maximum Transfert Unit*) et elle va servir à *fragmenter* les datagrammes trop grands pour le réseau qu'ils traversent.

Le **routage IP** consiste à choisir la manière de transmettre un datagramme IP à travers les divers réseaux d'un internet. On appellera **ordinateur** un équipement relié à un seul réseau et **routeur** un équipement relié à au moins deux réseaux (un ordinateur ou non). D'une manière générale on distingue la **remise directe**, qui correspond au transfert d'un datagramme entre deux ordinateurs du même réseau, et la **remise indirecte** qui est mise en œuvre dans tous les autres cas, c'est-à-dire quand au moins un routeur sépare l'expéditeur initial et le destinataire final. La remise indirecte nécessite de déterminer vers quel routeur envoyer un datagramme IP en fonction de sa destination finale. Ceci est rendu possible par l'utilisation d'une **table de routage**. L'essentiel du contenu d'une table de routage est constitué de quadruplets **(destination, passerelle, masque, interface)** où :

- **destination** : c'est l'adresse IP d'une machine ou d'un réseau de destination ;
- **passerelle** (*gateway*) : c'est l'adresse IP du prochain routeur vers lequel envoyer le datagramme pour atteindre cette destination ;
- **masque** : c'est le masque associé au réseau de destination ;
- **interface** : elle désigne l'interface physique par laquelle le datagramme doit réellement être expédié ;

Une table de routage contient notamment une **route par défaut** qui spécifie un routeur par défaut vers lequel sont envoyés tous les datagrammes pour lesquels il n'existe pas de route dans la table.

L'un des protocoles de routage les plus populaires est **RIP** (*Routing Information*

Protocol) qui est un protocole de type vecteur de distance ;

Le protocole **ICMP** (*Internet Control Message Protocol*) organise un échange d'information permettant aux routeurs d'envoyer des messages d'erreurs à d'autres ordinateurs ou routeurs. Le but d'ICMP n'est pas de fiabiliser le protocole IP, mais de fournir le **compte-rendu d'une erreur détectée** dans un routeur.

2.5 La couche transport

Les deux principaux protocoles de la couche transport d'Internet sont les protocoles **TCP** (*Transmission Control Protocol*) et UDP (*User Datagram Protocol*). Tous les deux utilisent IP comme couche réseau, mais TCP procure une couche de transport fiable (alors même que IP ne l'est pas), tandis que UDP ne fait que transporter de manière non fiable des datagrammes.

2.5.1 Le protocole UDP

Le protocole UDP utilise IP pour acheminer en **mode non fiable** des datagrammes qui lui sont transmis par une application (cf figure 2.0.1).

Ce protocole n'assure ni **accusé de reception** ni de **reséquencement** ni de **contrôle de flux** mais un contrôle **l'intégrité** du message.

UDP permet de distinguer plusieurs applications destinatrices sur la même machine par l'intermédiaire des ports. Un **port** est une destination abstraite sur une machine identifié par un numéro qui sert d'interface à l'application.

```
ftp-data        20/tcp
ftp-data        20/udp
# 21 is registered to ftp, but also used by fsp
ftp             21/tcp
ftp             21/udp          fsp fspd
ssh             22/tcp                          # SSH Remote Login Protocol
ssh             22/udp                          # SSH Remote Login Protocol
telnet          23/tcp
telnet          23/udp
# 24 - private mail system
smtp            25/tcp          mail
smtp            25/udp          mail
http            80/tcp          www www-http    # WorldWideWeb HTTP
```

```
http            80/udp          www www-http    # HyperText Transfer Protocol
pop3            110/tcp         pop-3           # POP version 3
pop3            110/udp         pop-3
printer         515/tcp         spooler         # line printer spooler
printer         515/udp         spooler         # line printer spooler
nfs             2049/tcp        nfsd
nfs             2049/udp        nfsd
```

Le format d'un datagramme UDP est le suivant :
- port UDP source ;
- port UDP destination ;
- longueur ;
- bits de contrôle (*checksum*) ;
- données.

2.5.2 Le protocole TCP

TCP est un protocole qui procure un service de flux d'octets orienté connexion et fiable. L'**accusé de réception** ainsi que la **numérotation des messages** évite la PERTE DES MESSAGES et leur éventuelle DUPLICATION. L'acquittement de données est **cumulatif**. Cette connexion est bidirectionnelle simultanée (*full duplex*) et composée de **deux flots** de données indépendants de sens contraire (enfin presque).

L'en-tête, sans option, d'un segment TCP a une taille totale de 20 octets et se compose des champs suivants :
- port source ;
- port destination ;
- La position du segment dans le flux de données envoyées par l'émetteur ;
- Le numéro d'accusé de réception : si l'émetteur a reçu correctement k octets, il renvoie $k + 1$;
- La longueur d'en-tête : elle peut dépasser les 20 octets dans le cas d'option ;
- un champ réservé à un usage ultérieur ;
- plusieurs champs qui permettent de spécifier le rôle et le contenu du segment TCP pour pouvoir interpréter correctement certains champs de l'en-tête :
  - ⋆ `URG`, le pointeur de données urgentes est valide ;
  - ⋆ `ACK`, le champ d'accusé de réception est valide ;
  - ⋆ `PSH`, ce segment requiert un push ;
  - ⋆ `RST`, réinitialiser la connexion ;
  - ⋆ `SYN`, synchroniser les numéros de séquence pour initialiser une connexion ;
  - ⋆ `FIN`, l'émetteur a atteint la fin de son flot de données.

- La taille de fenêtre : elle permet de réguler le flux de données ;
- Le checksum ;
- Le pointeur d'urgence : ajouté au numéro de séquence du segment, indique le numéro du dernier octet de donnée urgente ;
- option : le plus souvent la taille maximale du segment TCP qu'une extrémité de la connexion souhaite recevoir.

L'établissement et la terminaison d'une connexion suit le diagramme d'échanges de la figure 2.5.1.

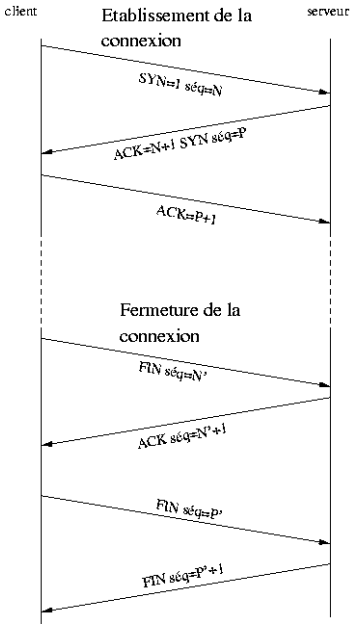

Fig. 2.5.1: Établissement et fermeture d'une connexion TCP/IP

La terminaison d'une connexion peut être demandée par n'importe quelle extrémité et se compose de deux demi-fermetures puisque des flots de données peuvent s'écouler simultanément dans les deux sens.

2.6 La couche application

On propose ici une liste non-exhaustives des applications sur Internet. Elles sont bâties sur le modèle **client-serveur** à savoir qu'une des deux extrémités de la connexion (TCP UDP)/IP rend des services à l'autre extrémité.

- **Telnet** est une application qui permet à un utilisateur de se connecter à distance sur un ordinateur, pourvu que cet utilisateur y dispose d'un accès autorisé (i.e. pour rlogin,ssh,...) ;

- **NFS** (*Network File System*) est un système qui permet de rendre transparente l'utilisation de fichiers répartis sur différentes machines ;
- **FTP** (*File Transfert Protocol*) permet de transférer des fichiers d'une machine à une autre ;
- **SMTP** (*Simple Mail Transfer Protocol*) permet d'envoyer des messages électroniques ;
- **POP3, IMAP** : permet de recevoir des messages électroniques ;
- **NNTP** (*Network News Transfert Protocol*) est le protocole d'échange des news ou forums de discussions à travers Usenet (nom donné au réseau logique constitué des serveurs de news disséminés sur la planète).
- **HTTP** (*HyperText Transfer Protocol*) est le protocole de communication du web permettant d'échanger des documents hypertextes contenant des données sous la forme de texte, d'images fixes ou animées et de sons.

Table des figures

Liste des tableaux