Réseaux

MASTER Informatique

Université Bordeaux 1 – Année 1 – Semestre 1 – 2005/2006

Présentation générale

Introduction

- 18ème siècle : révolution industrielle
- 20ème siècle : révolution numérique dans les domaines de l'informatique, de la téléphonie et des médias. Séparés à l'origine, ces trois domaines sont maintenant fusionnés.

Utilisations

- Divertissements (jeux, films, Pay Per View)
- Accès à des informations distantes (Bases de Données)
- Communication (mail, chat, SMS, forums,...)
- Nomadisme (wifi, GPRS/UMTS, TMC)

Ces utilisations reposent sur des modèles techniques et économiques variés, ce qui implique un certain impact sur la société (protection, sécurité de l'information).

Evolutions

- Modèle centralisé : un « gros » ordinateur + périphériques dans une salle (imprimantes)
- Périphériques en dehors de la même salle : MAC (1963) : 30 périphériques via réseaux téléphoniques.
- Modèles répartis :au départ par un petit ensemble de mini-ordinateur et maintenant par un grand ensemble de micro-ordinateurs.

Topologie

On distingue 2 grandes catégories de topologies des réseaux :

- les réseaux centralisés
- les réseaux répartis

Réseaux centralisés :

- étoilés : réseaux en forme d'étoile
- hiérarchique : en étoile, avec plusieurs niveaux (une machine fait alors partie de 2 hiérarchies et effectue le lien entre les deux groupes hiérarchique)
- multiplexage : sur une même ligne de communication, on fait circuler, à des fréquences différentes et en simultané plusieurs messages en utilisant des slots de fréquence distincts pour les différents messages.
- Réseau bouclé : forme de réseau en boucle avec un ordinateur particulier ayant un rôle « central » permettant de faire le lien entre les autres ordinateurs/périphériques.

Cette topologie de réseau n'est pas optimisée car elle est limité en général à une communication par canal.

Réseaux répartis :

Un modèle réparti peut être représenté par un graphe sur lequel les sommets sont les ordinateurs/périphériques et les arêtes, les liens de communication. On utilise des algorithmes de routage (ou de choix de chemins).

On distingue alors deux types de routages :

- fixe : les chemins sont calculés lors de la conception du réseau et fixe ensuite (tables de routages)

- adaptable (ou dynamique) : de manière centralisée (les calculs se font sur un poste fixe) ou répartis (les calculs se font sur tous les postes en fonction du besoin).

Les réseaux répartis permettent un meilleur partage des ressources matérielles (imprimante par exemple), logicielles, humaines et des données. Ils sont plus fiables en ce qui concerne la destruction ou la surcharge de données. L'optimisation des ressources permet une optimisation des coûts d'utilisations. Une évolution en ce termes d'utilisation des structures répartis : commutation.

Commutation

On distingue 3 types de commutations :

- Par circuit (téléphonie) : ces circuits étaient des cables aux débuts de la téléphonie, ils sont maintenant remplacés par des circuits électroniques. Cette commutation « physique » implique une même vitesse et aucune reprise sur erreur.
- Par message : on décompose la communication en plusieurs segments indépendants. Chaque segment peut alors être envoyé à des vitesses différentes. On utilise alors (message par message) un stockage local (sur les serveurs par exemple) permettant la reprise sur erreur et la ré-émission si l'information a été perdue.
- Par paquets : le principe reste le même que pour la commutation par message, mais on décompose ici les informations en paquets de taille fixe. Cela permet d'optimiser le stockage, la détection d'erreurs et le multiplexage.

Techniques de communication

On distingue 2 techniques de communication :

- Diffusion : un seul canal de communication est partagé par n utilisateurs. Ainsi, un émetteur peut envoyer un message à n récepteurs, ou à un seul (par sélection/rejet). Exemple de réseaux par diffusion : le réseau local.
- Point à point (paire de machines connectées) : décomposition d'une seule communication en plusieurs parties => plusieurs chemins possibles.

Modèles d'architectures

Il existe deux types de modèles d'architectures des réseaux :

- Modèle client/serveur : un ordinateur serveur auquel se connectent plusieurs ordinateurs clients.
- Modèle P2P (peer to peer : « poste à poste ») : tous le monde est à la fois client et serveur.

Classification des réseaux

On peut classer les réseaux en fonction de leur importance et étendue :

- Réseau personnel (PAN) : ordinateur + souris, claviers, imprimante et autres périphériques
- Réseau local (LAN : Local Area Network) : salle, batiment, entreprise/campus,...
- Réseau métropolitain : grâce à la mise au point des accès ADSL
- Réseau à longue distance (WAN : Wide Area Network) : à l'échelle d'un pays
- Interconnexion de réseaux : Internet (à l'échelle de la planète).

Interprocessor distance	Processors located in same	Example
1 m	Square meter	Personal area network
10 m	Room	Local area network
100 m	Building	
1 km	Campus	
10 km	City	Metropolitan area network
100 km	Country	Wide area network
1000 km	Continent	
10,000 km	Planet	The Internet

Dans ce cours, nous ne traiterons que des réseaux locaux et longues distances. Le schéma ci-contre résume la classification des réseaux selon leur importance et étendue.

Réseau local

Propriétés :

La taille est restreinte (inférieure à quelques kilomètres). Cela permet d'avoir des délais (temps) de transmission bornés et donc d'avoir du (très) haut débit (de 10 Mb/s à 10Gb/s).

Allocation :

L'allocation d'un réseau local peut se faire de manière :

- statique : partage figé de l'accès (temps, fréquence)
- dynamique : accès à la demande

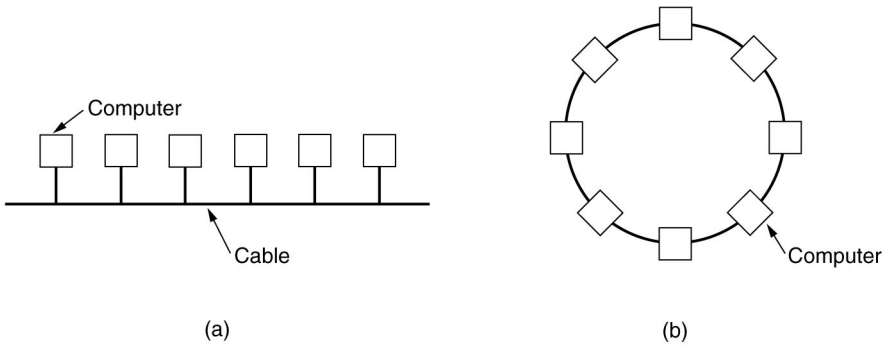
Topologie :

Deux topologies classiques du réseau local :

- bus (Ethernet, norme 802.3) : tous les postes sont reliés à un « bus » (cable). Les ordinateurs communiquent entre eux en passant en en-tête des messages l'adresse de l'ordinateur récepteur.
- Anneau (Token Ring, norme 802.5) : Les postes sont liés en forme d'anneau sur lequel l'information ne circule que dans un sens. La sélection du récepteur d'un message se fait par sélection/rejet : si un message arrive sur une machine et que celui ci ne lui est pas destiné, elle transfère le message au poste suivant.

La topologie de l'anneau est peu performante s'il y a beaucoup d'ordinateurs sur le réseau (notamment en raison de la diffusion : une seule machine à le droit d'émettre à la fois). La topologie de bus permet des hiérarchies dans le réseau.

Le schéma ci-dessous présente les deux topologies :



Réseau longue distance

Un réseau longue distance se compose d'hotes (applications utilisateurs) et de sous réseaux (permettant l'acheminement des messages). Un sous réseau se compose de :

- lignes de transmissions (pas nécessairement physiques)
- équipements de commutation (ordinateurs) connectant les lignes de transmission
- commutateurs, routeurs (routage = sélection)

Certaines liaisons (lignes de transmissions) sont maintenant assurées via satellite géostationnaire (36000km) qui servent donc à la fois de station d'emission et de réception.

Logiciels de réseaux

L'évolution des structures (support physique, architecture) et du fonctionnement (services, débits) des réseaux ont imposé la mise au point de logiciels de réseaux. Ces logiciels (devenus complexes avec l'évolution) sont maintenant décomposés. Ainsi, on trouve des logiciels de conception, de maintenance, d'échange,...

Pour faire fonctionner correctement ces logiciels, on a mis en place la notion de couche, protocole et service.

Définition : couche réseau

Les modules logiciels qui sont regroupées fonctionnellement forment une couche de niveau N. Leur rôle est d'assurer des services à la couche N+1 en lui masquant les détails d'implémentation. On peut faire ici l'analogie avec les machines virtuelles et l'encapsulation.

Il existe alors un dialogue (communication virtuelle) entre couches N (entités paires) de machines distantes. Ce dialogue est possible grâce à un ensemble de conventions (règles, formats) : c'est ce que l'on appelle un protocole.

Mécanismes des couches :

- Adressage
- contrôle d'erreurs
- séquencement (remettre en ordre les paquets)
- contrôle de flux

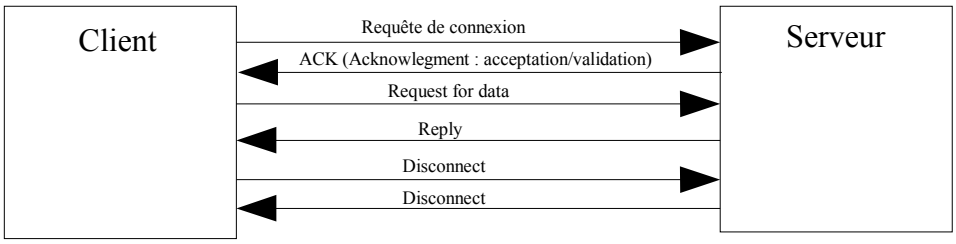
Primitives :

Exemple de primitives (transfert avec établissement de connexion) :

LISTEN	On attend une connexion libre
CONNECT	émet un appel d'établissement de connexion
RECEIVE	attente d'un message (une fois connecté)
SEND	envoi d'un message (une fois connecté)
DISCONNECT	déconnexion, libère les ressources

Exemple :

Un exemple de dialogue entre un client et un serveur :



Remarque, problème :

Au début des réseaux, il existait uniquement des réseaux propriétaires (IBM, Bull, DEC) et des interconnexions de systèmes hétérogènes (machine, OS, liaisons) qui rendaient très difficile la connexion entre couches.

La solution, venue il y a 20 ans a consisté à :

- standardiser les produits du marchés les plus répendus : IBM, PC, Windows
- normaliser : instances internationnales (ISO, AFNOR, ANSI). Exemple de normes : RS 232, IEEE 802.3

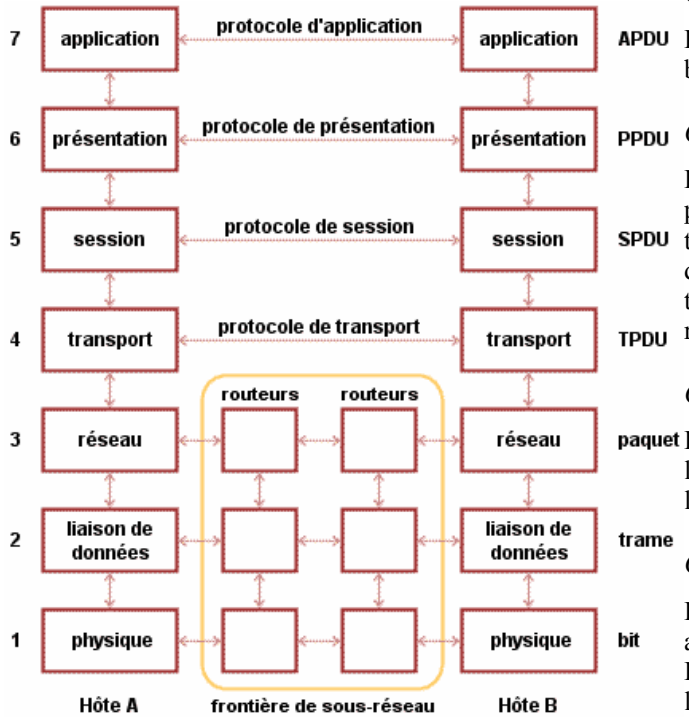
Modèles de références

Modèle OSI

Le modèle OSI (Open System Interconnexion) c'est présenté comme une solution d'interconnexion des réseaux à l'époque des réseaux propriétaires. En effet, les réseaux propriétaires tels que ceux d'IBM ou DEC ont des architectures différentes. Afin de facilité leur interconnexion, les organismes internationnaux de la normalisation (ISO) ont développés un modèle de référence : le modèle OSI.

Ce modèle décrit les concepts utilisés et la démarche suivie pour normaliser l'interconnexion de systèmes ouverts (un réseau est composé de systèmes ouverts lorsque la modification, l'adjonction ou la suppression d'un de ces systèmes ne modifie pas le comportement global du réseau).

Le modèle OSI décrit le fonctionnement de 7 couches (mais pas des protocoles, ni des services), allant du support physique jusqu'à l'application.



Couche physique :

La couche « physique » s'occupe de la transmission des bits de façon brute sur un canal de communication.

Couche liaison de données :

La couche « liaison de données » transforme la couche physique en une liaison à priori exempte d'erreurs de transmission pour la couche réseau. Elle fractionne les données d'entrée de l'émetteur en trames, transmet ces trames en séquence et gère les trames d'acquittement renvoyées par le récepteur.

Couche réseau :

La couche « réseau » permet de gérer le sous-réseau, i.e. le routage des paquets sur ce sous-réseau et l'interconnexion des différents sous-réseaux entre eux.

Couche transport :

La couche « transport » est responsable du bon acheminement des messages complets au destinataire. Le rôle principal de la couche transport est de prendre les messages de la couche session, de les découper s'il le faut en unités plus petites et de les passer à la couche réseau, tout en s'assurant que les morceaux arrivent correctement de l'autre côté. Cette couche effectue donc aussi le réassemblage du message à la réception des morceaux. Cette couche est également responsable de l'optimisation des ressources du réseau, ainsi que du type de service à fournir à la couche session, et finalement aux utilisateurs du réseau : service en mode connecté ou non, avec ou sans garantie d'ordre de délivrance, diffusion du message à plusieurs destinataires à la fois... C'est l'une des couches les plus importantes, car c'est elle qui fournit le service de base à l'utilisateur, et c'est par ailleurs elle qui gère l'ensemble du processus de connexion, avec toutes les contraintes qui y sont liées.

Couche session :

Cette couche organise et synchronise les échanges entre tâches distantes. Elle réalise le lien entre les adresses logiques et les adresses physiques des tâches réparties. Elle établit également une liaison entre deux programmes d'application devant coopérer et commande leur dialogue.

Couche présentation :

Cette couche s'intéresse à la syntaxe et à la sémantique des données transmises : c'est elle qui traite l'information de manière à la rendre compatible entre tâches communicantes. Elle va assurer l'indépendance entre l'utilisateur et le transport de l'information.

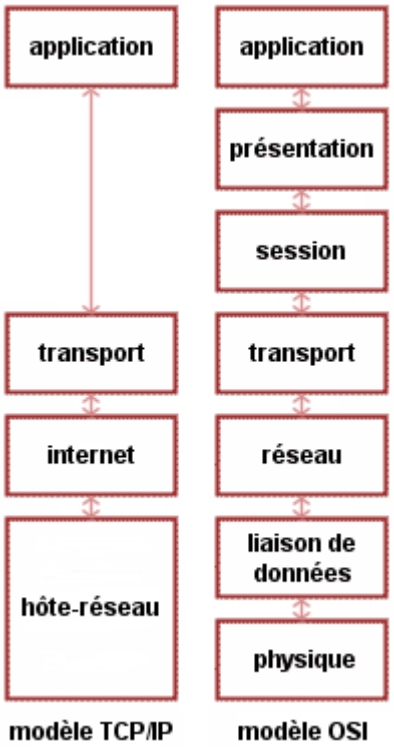
Couche application :

Cette couche est le point de contact entre l'utilisateur et le réseau.

Modèle TCP/IP

TCP/IP désigne communément une architecture réseau, mais cet acronyme désigne en fait 2 protocoles étroitement liés : un protocole de transport, TCP Transmission Control Protocol) qu'on utilise "par-dessus" un protocole réseau, IP (Internet Protocol). Ce qu'on entend par "modèle TCP/IP", c'est en fait une architecture réseau en 4 couches dans laquelle les protocoles TCP et IP jouent un rôle prédominant, car ils en constituent l'implémentation la plus courante. Par abus de langage, TCP/IP peut donc désigner deux choses : le modèle TCP/IP et la suite de deux protocoles TCP et IP.

Le modèle OSI, est très peu utilisé par rapport au modèle TCP/IP. OSI a le défaut d'avoir des couches peu équilibrées. Les couches 2 et 3 du modèle OSI sont très importantes et fournies. En revanche, les couches 5 et 6 sont peu développées, voire même maintenant, apparues inutiles : on ne les trouve donc pas dans le modèle TCP/IP.



La couche hôte-réseaux

Cette couche, qui n'a pas été vraiment spécifiée, regroupe la couche physique et la couche liaison de données du modèle OSI (le modèle OSI a été mis à coté sur le schéma ci-contre pour faciliter la comparaison avec le modèle TCP/IP). La seule contrainte de cette couche est de permettre à un hôte d'envoyer des paquets IP sur le réseau. L'implémentation de cette couche est laissée libre. De manière plus concrète, cette implémentation est typique de la technologie utilisée sur le réseau local. Par exemple, beaucoup de réseaux locaux utilisent Ethernet ; Ethernet est une implémentation de la couche hôte-réseau.

La couche internet

La couche internet réalise l'interconnexion des réseaux (hétérogènes) distants sans connexion. Son rôle est de permettre l'injection de paquets dans n'importe quel réseau et l'acheminement des ces paquets indépendamment les uns des autres jusqu'à destination.

La couche transport

Son rôle est identique à celui de la couche transport du modèle OSI. Cette couche n'a officiellement que deux implémentations : le protocole TCP (Transmission Control Protocol) et le protocole UDP (User Datagram Protocol).

La couche application

Directement au dessus de la couche transport (les couches session et présentation sont apparues inutiles). Cette couche contient tous les protocoles de haut niveau, comme par exemple Telnet, FTP (File Transfer Protocol), SMTP (Simple Mail Transfer Protocol), HTTP (HyperText Transfer Protocol).

Schéma des protocoles du modèle TCP/IP

Telnet FTP SMTP DNS	Application
TCP UDP	Transport
IP	Network
NetWork, ARPANET, SATNET, Packets Radio, LAN	

} protocoles

networks

Exemple : Internet

Pendant la guerre froide : le réseau téléphonique est vulnérable car centralisé. Vers les 60's : l'idée d'une structure maillée (P. Baron) est rejetée. Mais en 1967, ARPA conçoit un réseau de mini-ordinateurs appelés IMP (Interface Message Processor : ancètre des commutateurs/routeurs) reliés par des lignes 56kbits/s. Chaque IMP est relié à au moins 2 autres IMP. Un noeud, constitué d'un IMP et d'un ordinateur hote reliés par un cable court, permet l'accès au réseau.

ARPA va rapidement se développer : en 1969, 4 universités sont financées pour se connecter à ARPAnet. En 1972, plus de 40 universités dans le monde sont connectées.

Les extensions d'ARPAnet vont conduire au développement d'outils efficaces et beaucoup diffusés (TCP/IP par exemple) et au déploiement d'outils (sockets, utiliaires, Unix BSP, adressage DNS).

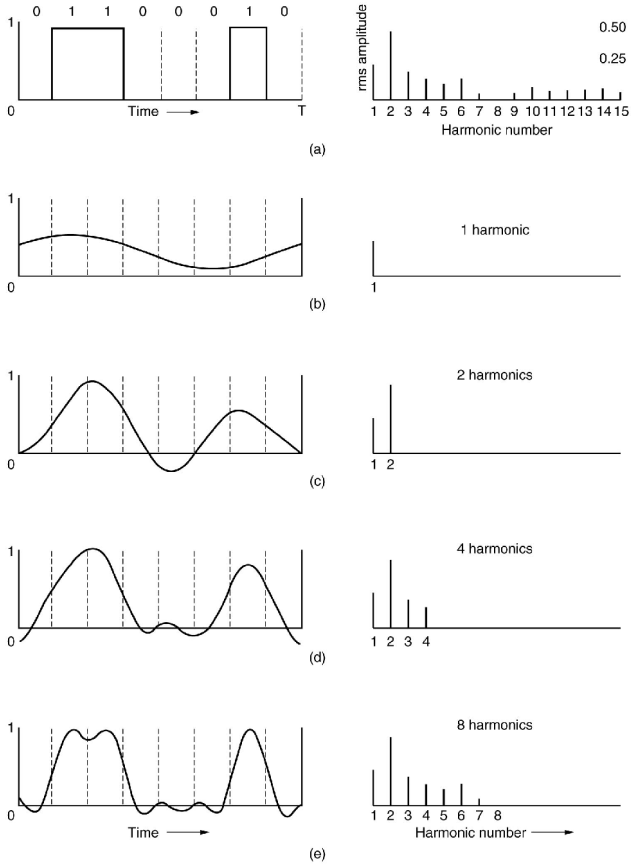
Reprenant l'idée d'ARPAnet, un nouveau réseau NSFnet financé dans les 70's et reliant 6 centres de calculs va ensuite intégrer de nombreux site, puis être raccordé à ARPAnet. Son évolution va directement donner naissance à Internet dans les années 90's, notamment avec le développement des email, newsgroup, telnet, Mosaïc (le premier navigateur avant Internet Explorer, Mozilla,...) l'utilisation dans les universités et les entreprises, la mise en place des fournisseurs d'accès pour le grand public.

Autre exemple de réseau basé sur TCP/IP : Ethernet

Développement dans les années 70's de ALOHAnet : réseau local, émission avec collision. Ethernet est développé par Xerox : débit initial : 2.94Mbits/s. En 1978, la norme IEEE 802.3 à 10Mbits/s apparaît. On connait maintenant les évolutions à 100 Mbits/s et 1Gbits/s aujourd'hui.

Couche physique

Bases de la théorie de transmission



Signal sinusoïdal

Pour transmettre les données, on utilise un signal sinusoïdal. Les informations sont transmises sur un support en faisant varier les caractéristiques (tension, courant) de f(t). Pour cela, on utilise la transformée de Fourier afin de décomposer un signal en une série de fonctions sinusoïdales (harmoniques) :

$y(t) = a.\sin(2\pi ft + \varphi)$

avec :

- a : amplitude
- f : fréquence (= 1/T)
- φ : phase

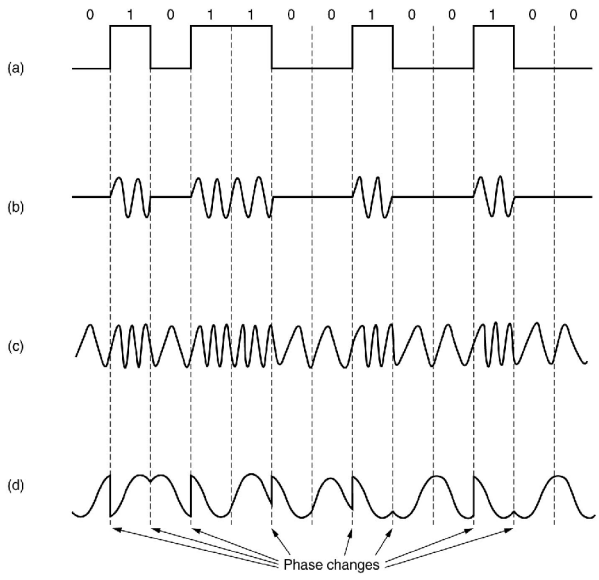
On représente alors y(t) par son spectre d'énergie (fréquences dont il est composé). Le schéma ci-contre résume le fonctionnement du signal sinusoïdal, transformé en spectre, transmis, et reconstruit en signal sinusoïdal. On remarquera qu'il faut transmettre un certain nombre d'harmoniques afin de reconstruire un signal proche du signal de départ.

Modes de transmission

On distingue deux modes de transmission :

- Numérique (bande de base) : les données binaires sont envoyées sous forme de signaux numériques (potentiels électriques). Problème : le signal s'attenue sur de longues distances.
- Analogique (modulation par ondes porteuses) : le signal de base (onde) est modifié en fonction des informations à transmettre.

Techniques de modulation



Les techniques de modulation vont permettre, sur une transmission analogique de moduler/démoduler le signal. Pour cela, on utilise des modems (abréviation de « modulateur/démodulateur ») ETCD (Equipement Terminal de Circuit de Données).

Sur l'onde porteuse, on peut distinguer 3 types de tensions :

- tension continue : modulation de l'amplitude
- tension carrée : modulation à largeur d'implusion
- tension alternative : modulation d'amplitude, de fréquence et de phase.

La rapidité de modulation correspond à l'inverse du temps le plus court existant entre deux niveaux de modulation (unité : baud ; différent du bits/s). Le schéma ci-contre présente le principe de fonctionnement de modulation d'un signal (a) binaire par modulation d'amplitude (b), de fréquence (c) et de phase (d).

Codage

Avant sa transmission, une information doit être numérisée. Par exemple, du texte en suite de caractères code ASCII, les images/sons en suites de bits. En général 1 bit (respectivement 1 octet) est associé à 1 (8) bascule(s) 0.5V.

On distingue alors de 2 types de transmissions :

- parallèle : transmission simultanée des 8 bits sur 8 fils (par exemple : bus microProcesseur) Ce mode de transmission, très efficace sur de courtes distances devient très couteux sur de longues distances (8 fils au lieu d'un seul!)
- série : transmission bit après bit sur un fil : il est alors nécessaire de voir les circuits de convertion Série/Parallèle (registres à décalages) entre un émetteur E et un récepteur R. Il va ya avoir des problèmes de synchronisation entre E et R.

Synchronisation

Pour synchroniser E et R, il faut prendre en compte une horloge H. Le problème consiste à ce que R obtienne le même H que E. Une première solution consiste à transmettre en parallèle H (ce qui est plus couteux car il faut un deuxième fil). Une seconde solution : intégrer H aux données émises. Il faut alors faire en sorte que H_E et H_R soient sensiblement équivalentes, puis synchronisées (réajustées) avec des bits du message.

On distingue 2 types de synchronisation :

- Asynchrone : le message est groupé en trame délimitées par un début et une fin. Au début, on fournit la synchronisation de H_R. Dans le mode asynchrone, on est limité par la taille des données (pour éviter une erreur de synchronisation) et par la vitesse (car ajout des informations de services). SLIP et PPP sont deux exemples de protocole de transmission asynchrone.
- Synchrone : H_R est recalée en permanence sur transition du signal.On envoi un message de synchronisation au préalable. On a pas de limitation de taille ou de vitesse. Problème : suite « continue » de 0 et de 1. Une solution consiste en un limiteur d'état permanent. A l'émission, on inverse tous les bits au dela du $n^{ième}$ constant. A la réception, on effectue le processus inverse. Exemples de protocoles : BSC, SDLC, HDLC, PPP (qui peut permuter entre le mode synchrone et le mode asynchrone).

Supports physiques de transmission

On peut distinguer deux types de supports physiques de transmission :

- supports guidés : paires torsadées, cable coaxial, fibre optique
- supports libres : faisceaux hertziens, liaisons satellites

Supports guidés

Paires torsadées (Unshielded Twisted Pair : UTP) :

Il s'agit de paires de fils. On les trouve pour :

- Les dessertes téléphoniques (1 paire)
- LAN (4 paires)
- réseau téléphonique (quelques dizaines de paires)

Elles sont torsadées pour diminuer les radiations parasites. Sur de courtes distances, on atteint un débit de quelques Mbits/s. Un avantage majeur de ce support : simplicité et coût. En revanche, le signal est rapidement affaibli après quelques kms, d'ou la nécessité de « régénateurs » pour ré-amplifier le signal.

Les transmissions par paires torsadées sont classées par catégories de fils, en fonction de la bande passante et du débit qu'elles proposent. Par exemple :

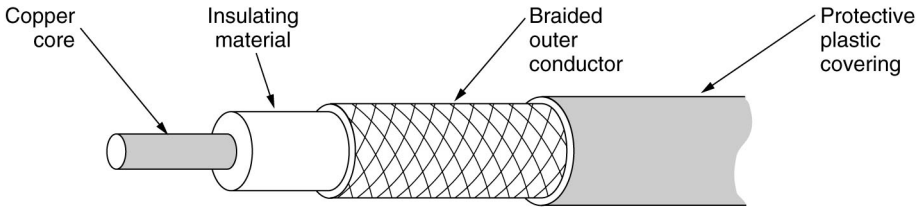
- UTP3 (16Mhz, voix numérique)
- UTP6 (250Mhz, Gigabit, ...)

Cable coaxial :

Le cable coaxial consiste en deux conducteurs cylindriques coaxiaux et isolés, d'un diamètre de quelques millimètre. Offrant une meilleur protection face aux radiations parasites, on obtient des débits plus élevés par rapport aux paires torsadées (bande passante de 1 Ghz, débit de quelques 100Mbits/s). La cable coaxial présente de nombreux avantages :

technique robuste, faible coût et débits élevés.

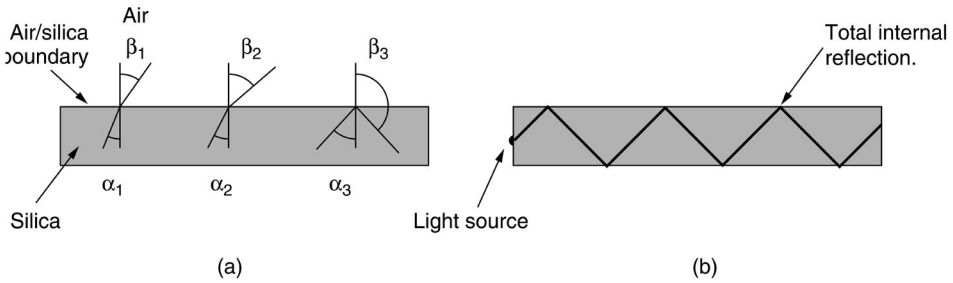
Ci-dessous, le schéma d'un cable coaxial :



Fibre optique :

Dans les années 60's : apparition du laser (faisceau lumineux très directif et stable en amplitude et en fréquence). D'ou la nécessité de systèmes pour guider la lumière : en 1972, la fibre optique devient « le piège à lumière » toujours d'actualité.

Le schéma ci-dessous explique comment se déplace le rayon lumineux piégé dans la fibre optique (On fera appel aux souvenirs des cours d'optique du lycée!) :

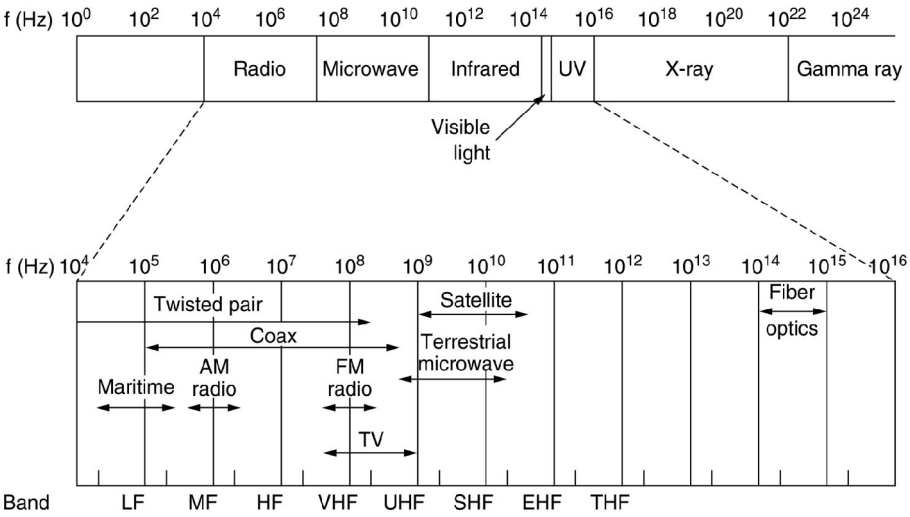


Le système de fibre optique consiste en un émetteur (équipé d'une diode), de la fibre optique et du récepteur (une photodiode). Ce système présente de nombreux avantages :

- performances : bande passante de plusieurs Ghz, des débits de 10 à 100 Gbits/s (en théorie, on peut aller jusqu'à 50 kGbits/s), très faible atténuation du signal
- Insensibilité aux parasites magnétiques
- Compacité : diamètre de quelques millimètres et poids léger : quelques grammes par km

Cependant, on ne peut négliger des inconvénients importants tels que les coûts de fabrications, la difficulté de la connectique et de la pose.

Supports libres



Il s'agit de transmission sans fils (wireless). Ces technologies sont surtout développées dans un but d'accès permanent (nomadisme) et des difficultées d'accès dues aux infrastructures. Les supports libres fonctionnent grâce à la progression d'ondes électromagnétiques : d'où nécessité de réguler l'accès à des gammes d'ondes (en France : ART, en Europe : ECC).

Le schéma ci-contre présente les différentes bandes de fréquences et leur utilités.

Téléphonie mobile (cellulaire) :

On distingue trois générations de téléphonie mobile :

- voix analogique
- voix numérique
- voix et données numériques

La téléphonie mobile (et ses évolutions) ont été inventée aux USA mais développées en Europe. D'ou un problème de norme unique s'opposant à plusieurs système (les longueurs d'ondes utilisées aux USA sont différentes que celles utilisées en Europe par exemple).

Historique : En 1946, le premier système de radiotéléphone est mis au point à Saint Louis. Il s'agit d'un canal unique, « push to talk » (CB). En 1960 : IMTS : deux fréquences (émission, réception) mais des délais importants. En 1982 : AMPS (Bell Labs) est l'ancètre de la première génération de portable (G1).

L'AMPS consiste en un découpage de l'espace (territoire) en cellules (de 10 à 20 kms). Chaque cellule utilise une bande de fréquences différente des cellules adjacentes. Chaque cellule est équipée d'une antenne. La diminution de la taille de la cellule permet :

- une augmentation du nombre d'appels simultanés
- une diminution de la puissance des antennes
- une diminution de la puissance (et donc du poids et de la consommation) des téléphones.

Lors d'évènements importants, on va même jusqu'à utiliser des microcellules (une cellule est équipée de plusieurs dizaines d'antennes qui ne sont activées que pour les évènements – matches par exemple – nécessitant un plus important nombre de connexion simultanées sur la même zone.

Fonctionnement : Un téléphone est dans une cellule, relié à la station de base. Quand il sort de la cellule, la station de base constate un affaiblissement du signal. Elle communique avec les stations de bases adjacentes et passe le relais à celle qui reçoit un signal fort. Le téléphone est alors prévenu et change de canal.

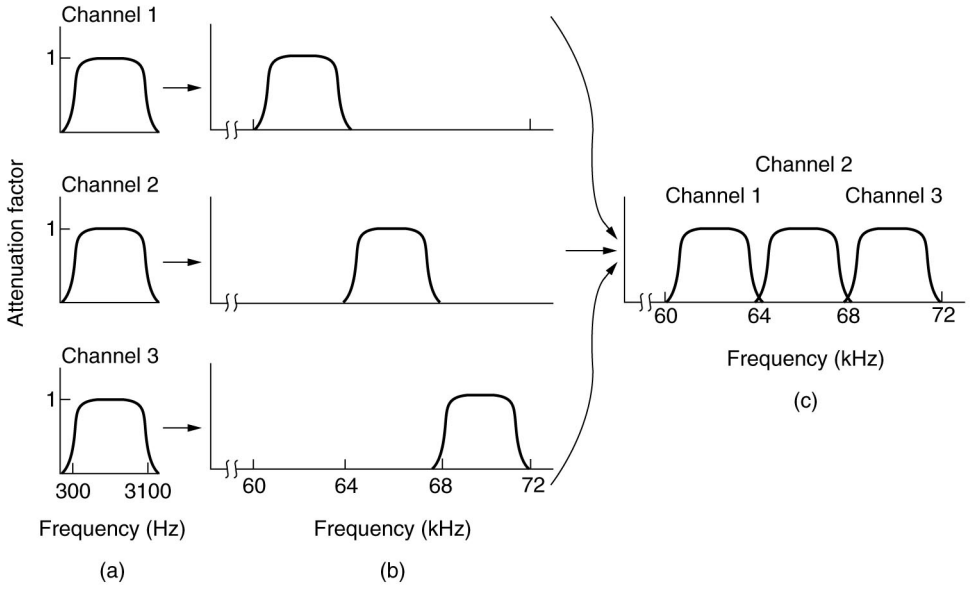
Le système deuxième génération (G2 : voix numérique, évolution vers le WAP et UMTS ou G3) émet sur une fréquence et reçoit sur une autre. Il y a donc multiplexage fréquentiel et temporel constitué de 124 paires de canaux simplex (les informations circulent toujours dans le même sens. La bande passante est découpée :

- bande passante montante : 890 – 915 Mhz
- bande passante descendante : 935 – 960 Mhz

Un canal simplex fonctionne sur 200kHz et propose 8 connexions. Ainsi, une cellule comporte environ 992 canaux.

Multiplexage fréquentiel :

Le schéma ci-dessous explique le fonctionnement du multiplexage fréquentiel :



Couche liaison

Introduction

La couche liaison permet à deux machines connectées de communiquer de façon fiable. Il faut alors prendre en compte que les liaisons peuvent être imparfaites (pertes, corruptions, délais).

La couche liaison sert donc à :

- traiter les erreurs de transmission
- réguler, adpater les flux d'entrées/sorties.

Pour cela, elle utilise les trames.

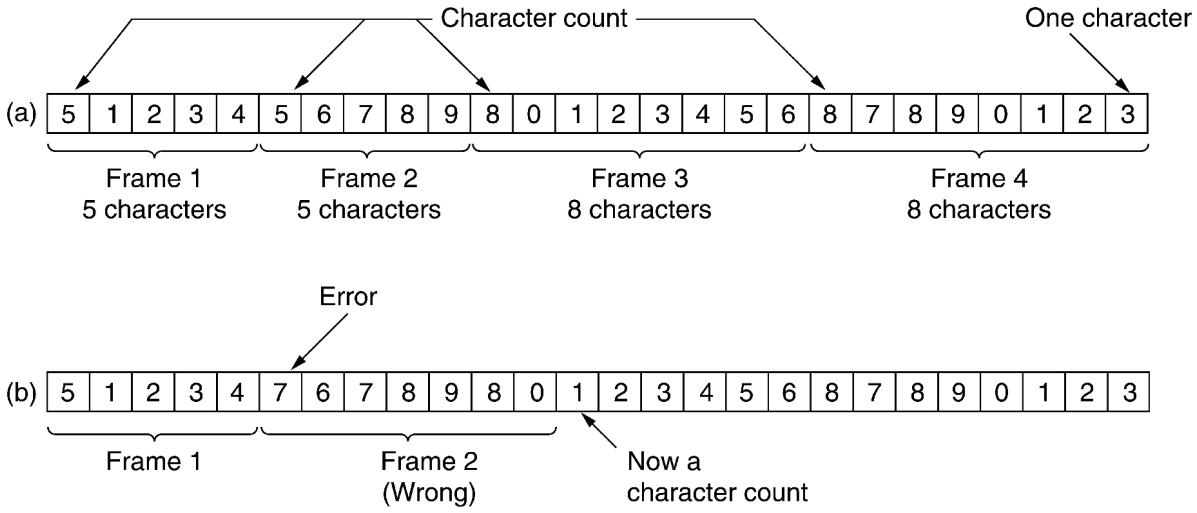
Trames

Sur un canal de transmission imparfait, le nombre de bits envoyés peut être inférieur, égal ou supérieur au nombre de bits envoyés. Dans le cas ou le nombre de bits est différent, il est évident qu'il y a eu erreur lors du transfert. Mais des erreurs peuvent exister aussi lorsque le nombre de bits est égal : en effet, les valeurs peuvent être modifiées sur des liaisons imparfaites.

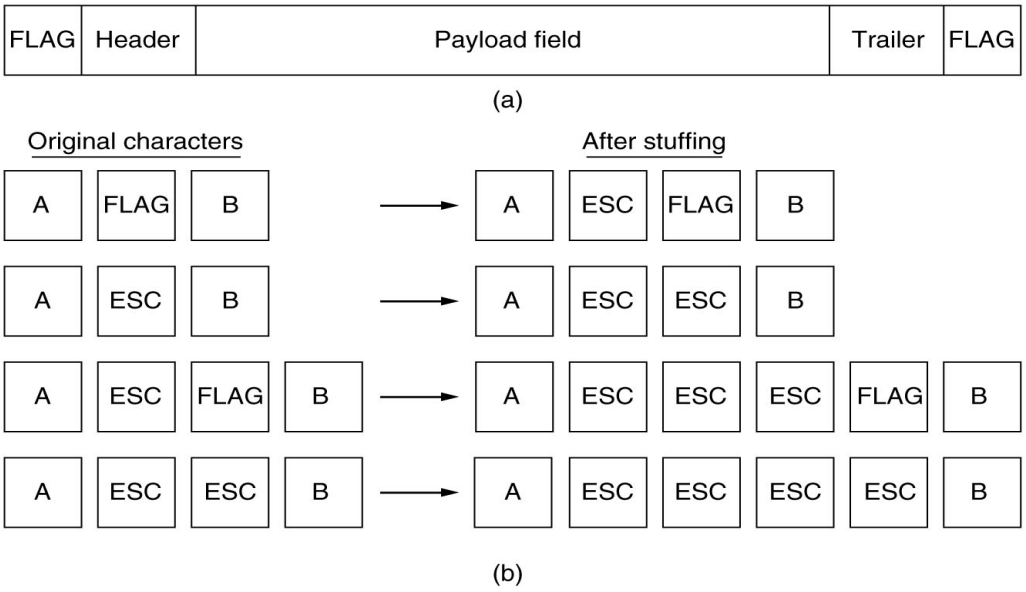
Pour controler ces erreurs, on utilise des trames. Ces trames sont controlées par un « checksum » calculé par l'émetteur. Inséré dans la trame envoyé, il peut alors être vérifié par le récepteur.

Il existe 3 sortes de trames :

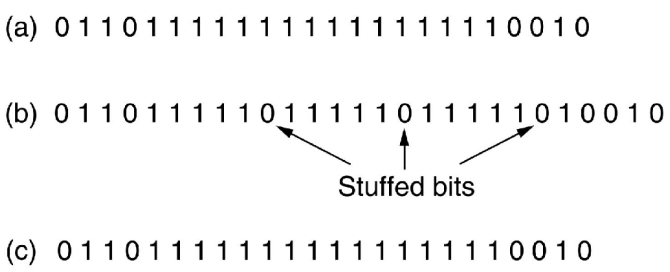
- trames comptées : le champ dans l'en-tête indique le nombre de caractères envoyés (le schéma ci-dessous montre le fonctionnement de telles trames (a) dans le cas ou il n'y a pas d'erreurs. On remarquera que le caractère de contrôle est compté dans le nombre de caractères envoyés. Le cas (b) montre la détection d'une erreur).



- Trames délimitées par des octets spéciaux (fanions, flags). Le problème, c'est qu'on peut avoir le fanion dans un transfert binaire. La solution dans ce cas là consiste à ajouter un caractère spécial (ESC). C'est ce qu'on appelle le « remplissage de caractère » (d'octet). Le schéma ci-dessous (page suivante) présente différents cas figures. Le schéma (a) présente le détail d'une trame. (b) : dans le premier cas on veut envoyé un flag, on effectue un remplissage avant pour signifier que le caractère suivant est un caractère spécial (ici un flag). Si on veut envoyé ESC comme caractère, on le précède de ESC pour le traiter comme caractère spécial (cas b2). Et il en est de même à chaque caractère, comme le montre les cas b3 et b4. (cette méthode est utilisée dans le protocole PPP, mais elle est limitée à des caractères de 8 bits... ce qui commence maintenant à poser problème)



- Trames de taille quelconque délimitées par des motifs binaires particuliers. Par exemple, le motif binaire peut être 01111110. Quand l'émetteur détecte 5 bits à 1, il insère un bit à 0 (pour être sur de ne pas retrouver le motif à l'intérieur du message et ne pas être ambigü). Inversement, lorsque le récepteur reçoit le message, il l'analyse et supprime le bit qui suit après 11111 dans le message. Le schéma ci-contre résume le fonctionnement de ce type de trame. En (a), le message à transmettre. En (b) le même message après ajout des bits à 0 suivant une chaine de 11111. En (c), le message analysé par le récepteur. Ce type de trame est utilisé dans le protocole HDLC.



Services

On distingue trois types de services :

- Sans connexion et sans accusé de réception : rapide mais peu fiable, il est utilisé uniquement pour les liaisons sures (fibre optique, temps réel – voix).
- Sans connexion et avec accusé de réception : plus fiable, utilisé par exemple pour les liaisons sans fils.
- Avec connexion et avec accusé de réception : divisé en trois phases (ouverture, transmission, fermeture) ce service est très fiable (trames numérotées), et garanti (pas de perte de trames, pas de duplication, ordre des trames respecté).

Traitement des erreurs

Sur un canal imparfait, (Shanon) l'ajout limité d'informations redondantes rend arbitrairement petit le taux d'erreurs résiduel (car les informations ajoutées sont calculées à partir des informations à transmettre). Il faut alors trouver un compromis entre la taille des informations, la vitesse de transmission, le taux d'erreur et le coût du matériel.

Le principe est toutefois assez simple :

- l'émetteur calcule une séquence de contrôle à partir des informations
- l'émetteur transmet les informations et la séquence calculée SC_E
- le récepteur reçoit le message et la séquence calculée. Il calcule une séquence SC_R à partir des informations reçues
- le récepteur compare SC_E avec SC_R

On distingue plusieurs stratégies :

- Données très redondantes : restitution des données émises à partir des données reçues
- Données peu redondantes : demande de réémission

Il existe donc deux solutions :

- Code correcteur d'erreur (ECC) : canal peu fiable (exemple réseau sans fil)
- Code détecteur d'erreur (EDC) : utilisé sur des canaux fiables (fibre)

Code correcteur : contrôle de parité

On contrôle la parité du nombre de bits. Par exemple, le nombre de bits à 1 sur 7 bits décide de la parité du 8ème bit. Par exemple, le code ASCII du caractère 'A' est 100001. Le nombre de bits à 1 est 2 (pair). Le bit ajouté est donc 0 (pair). Le code trnasmis est alors 10000010.

On remarquera toutefois qu'il est possible de ne pas détecter les erreurs si le nombre d'erreurs sur un transfert est pair. Pour remédier à cela, on peut faire un contrôle longitudinal. Ainsi, on ne contrôle pas trame par trame, mais une série de trames.

On veut par exemple transmettre 4 caractères codés sur 7 bits : 1000001, 0101011, 1000011 et 1001010. On ajoute le bit de parité sur ces 4 trames. On obtient les trames :

- 1000001 0
- 0101011 0
- 1000011 1
- 1001010 1

Et on crée un nouveau mot pour controler ces 4 trames. Le mot créé correspond à la parité sur les colonnes des 4 trames précédente. Ici, le mot trouvé est : 11000110.

On envoie alors la suite de trames :

1000001	0
0101011	0
1000011	1
1001010	1
1100011	0

Supposons maintenant qu'il y ait une erreur durant le transfert, le bit indiqué en rouge si-dessous a été modifié :

1000001	0
0111011	0
1000011	1
1001010	1
1100011	0

La parité indiquée par le bit de parité (0) est donc fausse. On peut donc détecter qu'il y a une erreur dans la deuxième trame envoyée. Mais la parité en colonne, indiquée par le 5ème mot créé, indique qu'il y a une erreur sur la 3ème colonne. On peut donc détecter ou est l'erreur : 2ème ligne et 3ème colonne : le bit rouge est donc faux.

Attention toutefois, s'il y a plusieurs erreurs, il est possible qu'on ne puisse plus les détecter toutes sans ambiguité.

Cependant, il ne faut pas oublier de plus, que si l'on a :

- d bits de données
- c bits de contrôles
- t = d + c : longueur du mot transmis

On a alors :

- 2^d combinaisons de bits de données
- 2^t combinaisons de mots

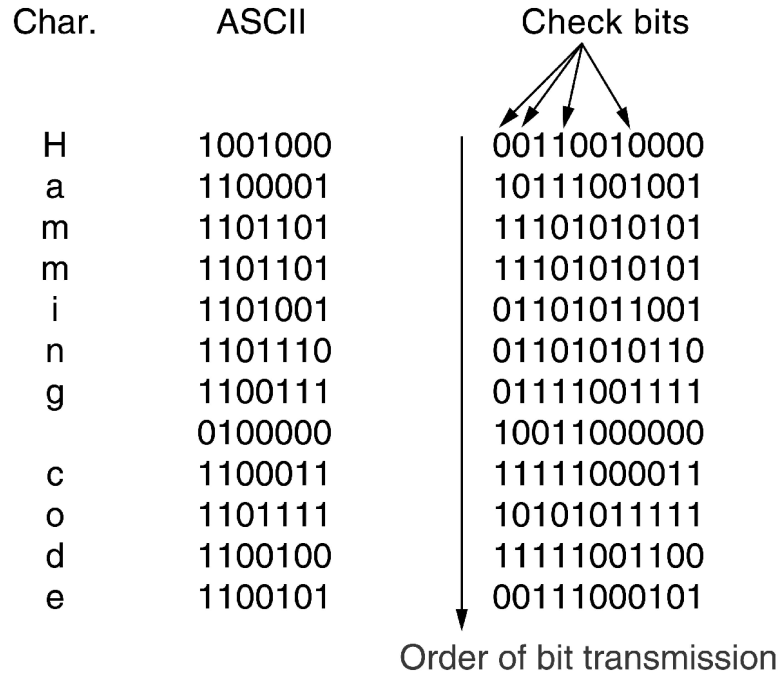
Mais parmis 2^t mots, seule une partie est valide (c'est à dire appartient à un langage). Cela peut être un moyen de vérification (correction) supplémentaire.

Code de Hamming

Le code de Hamming fait parti des codes correcteurs les plus connu. Il consiste à :

- inserer des bits de contrôle dans les bits de données aux position 2^n.
- Le bit de données k est contrôlé par les bits dont les positions sont les coefficients de la décomposition de k en puissances de 2.

Par exemple, si k = 13 = 1 + 4 + 8. Le bit 13 est donc contrôlé par les bits 1, 4 et 8. La valeur d'un bit de contrôle est choisie pour assurer une parité. Le schéma ci-dessous montre comment passer du code ASCII à la représentation du message calculé par le code de Hamming. La contrôle de parité se fait sur les bits de contrôles reçus. S'il y a, par exemple, une erreur sur les bits 1, 2, 8, alors le bit 11 est faux.



Code polynomial (CRC)

Un code est cyclique s'il est binaire et si toute permutation linéaire d'un mot du code est encore un mot du code. Par exemple C = {000, 101, 110, 011}. Les bits d'informations à transmettre sont les coefficients (0, 1) d'un polynome. Les mots du code peuvent donc être représenté sous forme polynomiale. Le code C précédent peut donc être écrit : C={0, $1+x^2$, $1+x$, $x+x^2$}.

On traite donc des mots à l'aide de l'arithmétique polynomiale modulo 2 ($ou_{EXCLUSIF}$). Par exemple 01100+11010=10110.

Pour tout code cyclique, il existe un unique polynome générateur G(X). Soit l'information à transmettre représentée sous forme polynomiale U(X). On multiplie U(X) par G(X) et on transmet le résultat. A la réception, on effectue la division euclidienne du message reçu par G(X) et le reste de la division doit être nul (sinon, c'est qu'il y a eu une erreur durant le transfert).

Ce type de calcul est effectué hardware (utilisation de circuits : registres à décalage, additionneurs, multiplicateurs,...)

Exemple de protocole : HDLC

HDLC (High Level Data Link Control) est un protocole basé sur l'insertion de bit (indépendant du réseau). Normalisé par l'ISO en 1976 :

- ISO 3309-76 : normalisation de la structure de la trame
- ISO 4335-77 : éléments de procédure

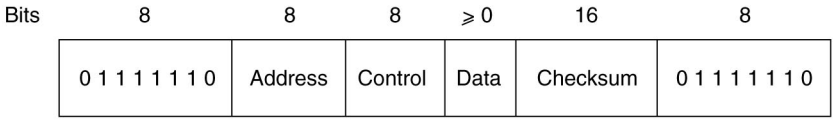
Trames

Les trames sont délimitées par 2 fanions, et il n'y a pas plus de 5 bits à 1. Elles sont constituées des champs suivants :

- adresse (du destinataire)
- commande (type, numéro)

- informations utilisateurs
- contrôle (CRC : $1 + x^5 + x^{12} + x^{16}$, adresse + commande + informations).

Le schéma suivant présente la structure de la trame HDLC : 8 bits (01111110) du fanion, 8 bits d'adresse, 8 bits de contrôle, les données utilisateurs (un nombre quelconque positif ou nul de bits), le contrôle (checksum) sur 16 bits et le fanion de fin (01111110) :



Le transfert s'effectue en 3 phases : ouverture, transfert, fermeture.

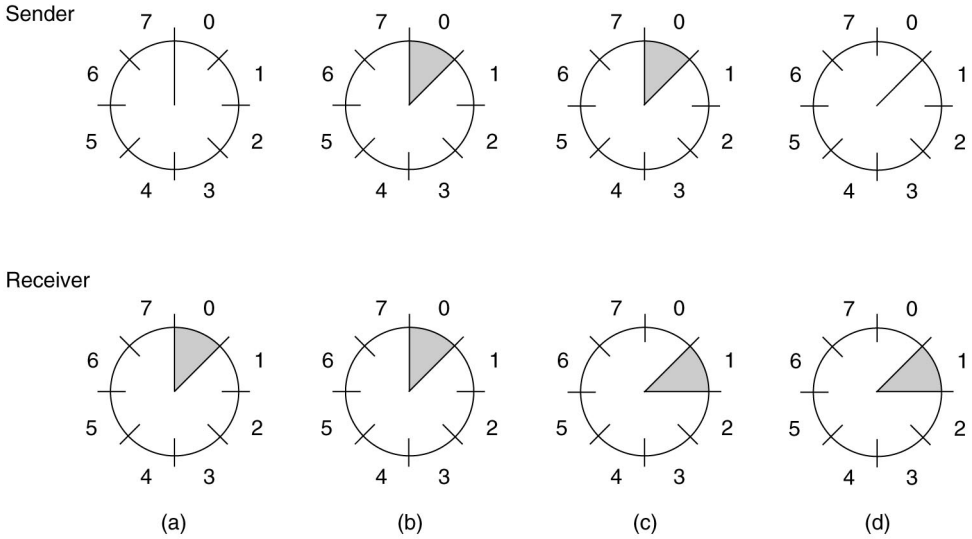
Transfert

Les trames sont numérotées. L'acquittement contient le numéro de la dernière trame correctement reçue. On utilise une fenêtre coulissante de taille 7 (modulo 8). On distingue deux types de demande de ré-émission en cas de problème :

- trame REJ (>n) : demande de réémission de toutes les trames à partir de la trame n
- trame SREJ (=n) : demande de réémission de la trame n uniquement.

Le contrôle de flux peut être demandé par blocage de l'émetteur (trames RNR ou RR).

Le schéma ci-dessous explique le principe de fonctionnement des fenêtres coulissantes :

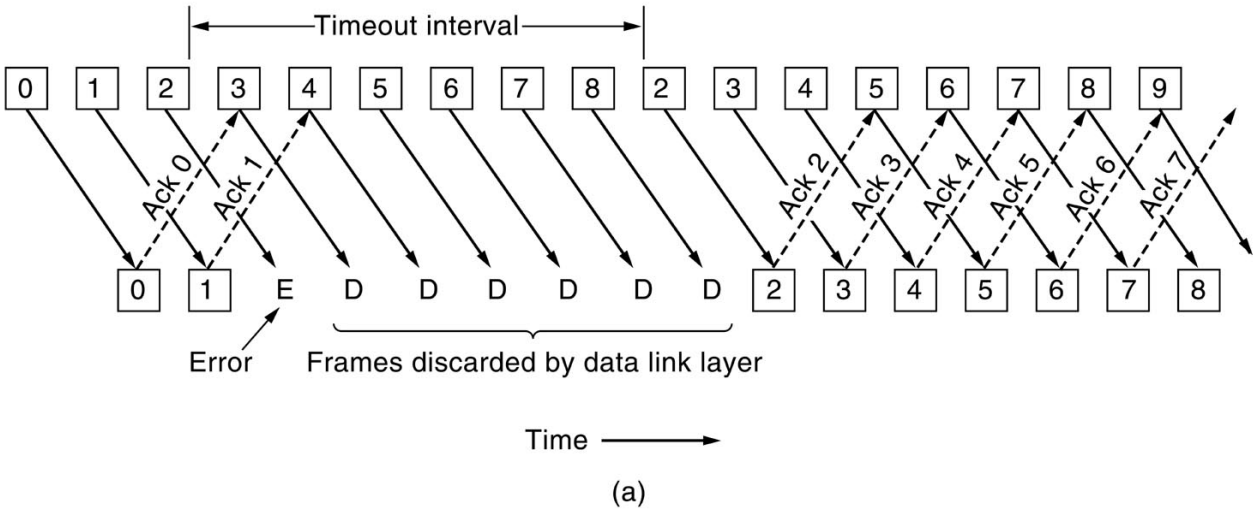


- (a) : le récepteur attend la trame 0, l'émetteur se positionne sur 0.
- (b) : l'émetteur envoie la trame 0. Le récepteur attend la trame 0. S'il y a éventuellement une erreur, il reste en position 0.
- (c) : l'émetteur est toujours en position 0. Le récepteur passe en position 1 pour attendre la trame 1. Le récepteur envoie un acquittement pour préciser que la trame 0 a été correctement reçue.
- (d) : l'émetteur à reçu l'acquittement, il va passer en position 1 pour envoyer la trame 1. Le récepteur est déjà positionné dans la fenêtre 1 et attend la prochaine trame.

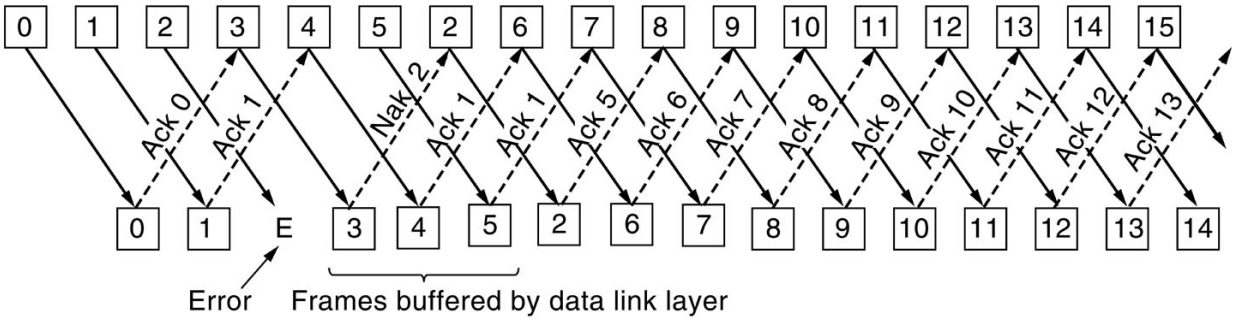
Le deuxième schéma explique deux modes de transfert des trames :

- (a) : les trames 0 et 1 sont correctement reçue par le récepteur. L'émetteur n'attend pas les acquittements pour continuer à envoyer les trames. Les acquittements ne sont pas interprétés par l'émetteur. L'acquittement des trames 0 et 1 est envoyé. L'émetteur envoie les trames 3 à 8. Cependant, au bout d'un certain temps de contrôle (timeout interval), l'émetteur n'a toujours pas reçu l'acquittement de la trame 2. Il réenvoie la trame 2 et toutes les trames depuis la trame 2. Les frames 3 à 8 reçues par le récepteur avant la réémission de la trame 2 ne sont pas concervées (elles sont ignorées).
- (b) : les trames 0 et 1 sont correctement envoyé. Il y a une erreur de transfert sur la trame 2. Le récepteur reçoit la trame 3. Il interprète donc cela comme une erreur de transfert et envoie un acquittement négatif (Nak2) pour

préciser à l'émetteur qu'il y a eu une erreur. (remarque : le numéro d'acquittement envoyé par le récepteur correspond au numéro de la dernière trame correctement reçue). Cependant, les trames 3, 4 et 5 sont correctement reçues par le récepteur qui les stocke dans un buffer. L'émetteur reçoit l'acquittement négatif de la trame 2. Il garde en mémoire le numéro de trame auquel il en était (trame 6), envoie la trame 2 et reprend directement à la trame 6. Le récepteur reçoit correctement la trame 2. Il récupère les trames 3 4 et 5 dans le buffer et donne l'acquittement positif (trame 5 reçue). Le transfert se poursuit alors normalement.



(a)



(b)

Sites HDLC

On distingue 3 sortes de sites (stations) HDLC :

- primaire (maître) : contrôle les opérations, envoie les commandes
- secondaire (esclave) : sous contrôle, envoie des réponses
- mixte : à la fois primaire et secondaire

Types de transfert HDLC

On distingue 3 types de transferts HDLC :

- NRM (Normal Response Mode) : liaison asymétrique, transfert à l'initiative du primaire uniquement.
- ABM (Asynchronous Balanced Mode) : liaison symétrique, les transferts peuvent être fait sans invitation préalable. On trouve en général ce type de transfert entre deux sites mixtes.
- ARM (Asynchronous Response Mode) : la liaison est asymétrique, les transfert peuvent être fait sans invitation préalable mais seul le primaire concerve la responsabilité de la ligne.

Procédure HDLC

A l'ouverture, le primaire anvoie une demande de connexion (SARM, SNRM ou SABM en fonction du mode de transfert demandé). Le secondaire acquitte par un UA (Unnumbered Acknowledgment).

A la fermeture, le primaire envoie une trame DISC pour demander la déconnexion. Le secondaire répond par un UA pour préciser au primaire qu'il a bien reçu la demande de déconnexion. Cependant, il peut continuer à effectuer les éventuelles émissions en cours ou en attente. Lorsque le secondaire a terminé toutes ses émissions, il envoie une trame DM (Disconnect Mode). Le primaire envoie alors une trame RD (Request to Deconnexion) pour confirmer la déconnexion au secondaire.

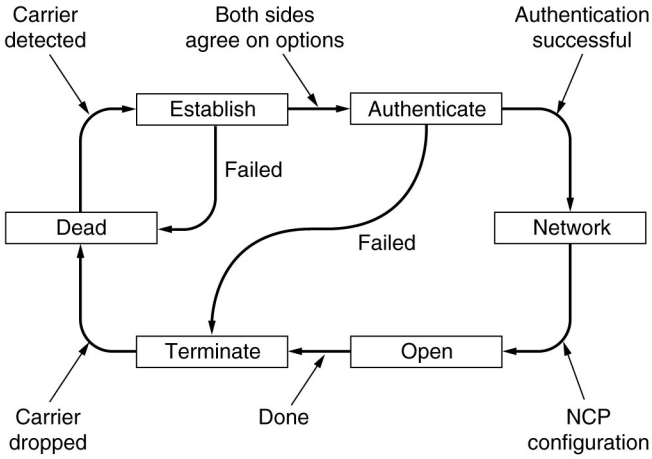
S'il y a une erreur de réception (trame invalide par exemple), une trame FRMR (Frame Reject) signifie qu'on a reçu une trame, mais qu'elle est incorrecte.

Bilan

HDLC est un protocole assez simple, mais la description des éléments de procédure peut parfois être longue et ambiguë. Il est alors nécessaire de mettre en place des méthodes formelles de description, spécification et vérification. On utilise pour cela des machines à états finis (FSM) tels que les automates ou les réseaux de Petri.

Automates et réseaux de Petri

Un automate A est un ensemble d'états E (initial, final, courant) et de transitions (changement d'état en fonction d'un événement). L'exemple si dessous montre un automate décrivant une demande de connexion (état initial : « Dead ») :

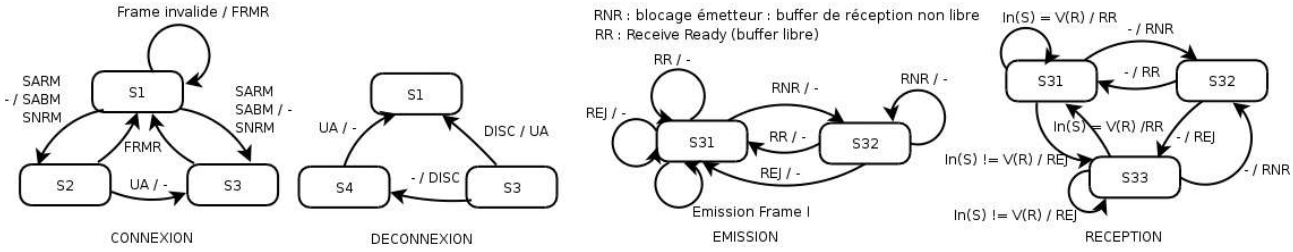


Il existe plusieurs types d'automates. Dans le cadre de ce cours, on utilisera les automates E/S (à entrées/sorties) :

- un événement entrant correspond à la réception d'un message
- un événement sortant correspond à l'envoie d'un message
- notation des transitions : me/ms = « si on reçoit le message entrant 'me', on envoie le message sortant 'ms' » et on change d'état.

Exemple des automates HDLC

En utilisant les automates tels que définis précédemment, nous allons voir quels sont les 4 automates (partiels) correspondant à l'ouverture, la fermeture, l'émission et la réception dans le cas de l'utilisation du protocole HDLC :

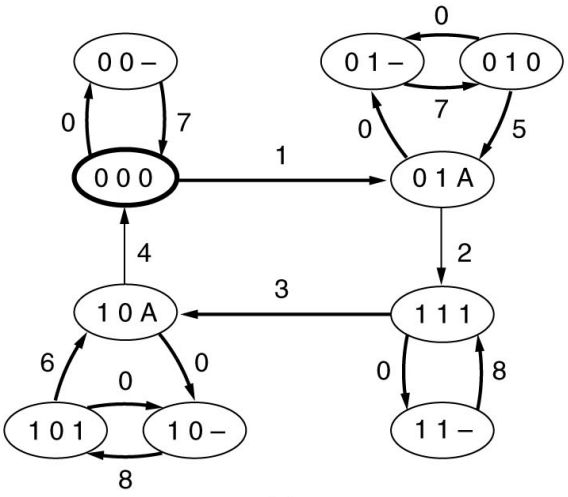


Système global

L'automate de système global permet de représenter sous forme d'automate l'ensemble des états du système global, à savoir : l'émetteur, le récepteur et l'état du canal. Pour simplifier Etats(SG) = {Etat(E), Etat(R), Etat(C)} ou :

- Etat(E) = état courant de l'émetteur (par exemple S32, si on garde les notations de l'automate HDLC précédent)
- Etat(R) = état courant du récepteur
- Etat(C) = état du canal, c'est à dire le contenu du canal et/ou des files d'attentes.

Le schéma ci-dessous est un exemple d'automate du système global :

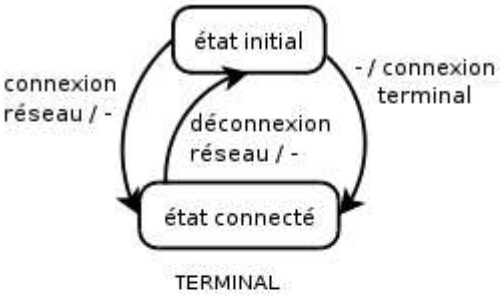


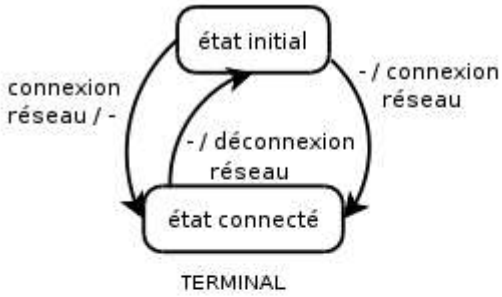
Transition	Who runs?	Frame accepted	Frame emitted	To network layer
0	–	(frame lost)		–
1	R	0	A	Yes
2	S	A	1	–
3	R	1	A	Yes
4	S	A	0	–
5	R	0	A	No
6	R	1	A	No
7	S	(timeout)	0	–
8	S	(timeout)	1	–

(b)

L'étude du graphe des états du système global permet la détection de deadlocks (c'est à dire les états inacessibles, les composantes connexes). Il s'agit alors de vérifier qu'il n'y a pas d'états puits (pas de sortant : si on arrive dans cet état on est bloqué : deadlock) et d'états non accessible (pas d'entrant).

L'exemple ci-dessous explique la connexion/déconnexion du point de vue du réseau et d'un terminal.

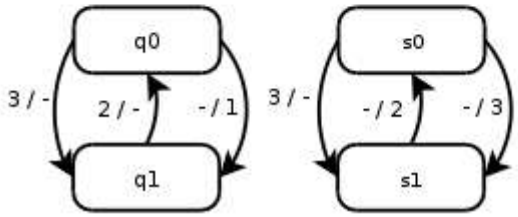




Notons :
- q0 = "état initial" du TERMINAL
- q1 = "état connecté" du TERMINAL
- s0 = "état initial" du RESEAU
- s1 = "état connecté" du RESEAU
- 1 = "connexion terminal"
- 2 = "déconnexion réseau"
- 3 = "connexion réseau"

CONNEXION - DECONNEXION
TERMINAL - RESEAU

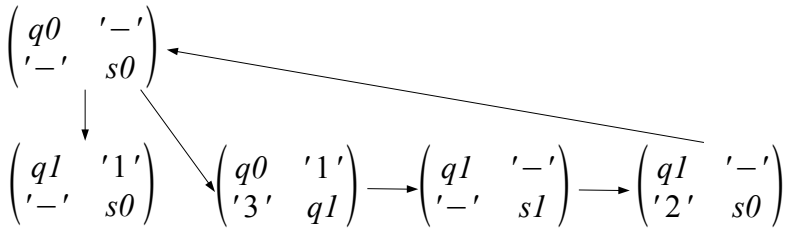
Avec les nouvelles notations ci contre on obtient les deux diagrammes ci dessous (équivalents aux deux précédents) :



On peut écrire, en utilisant les notations spécifiées dans le schéma ci-dessus, l'état courant du système global sous forme de matrice comme suit :

$$\begin{pmatrix} Etat\ Emetteur & Transmission\ E\ vers\ R \\ Transmission\ R\ vers\ E & Etat\ Récepteur \end{pmatrix}$$

On peut alors définir un graphe des changements d'état du système global par représentation de ces matrices :



Le graphe ci dessus est incomplet, mais suffisant pour remarquer qu'il y a déjà des erreurs : en effet, arrivé à l'état dus système global $\begin{pmatrix} q1 & '1' \\ '-' & s0 \end{pmatrix}$, on reste bloqué (deadlock) dans cet état.

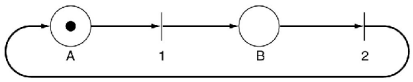
Ce type de vérification permet donc de détecter :

- les deadlocks
- les transitions non utilisées
- les réceptions non spécifiées
- les composantes connexes
- les boucles infinies

Réseaux de Petri

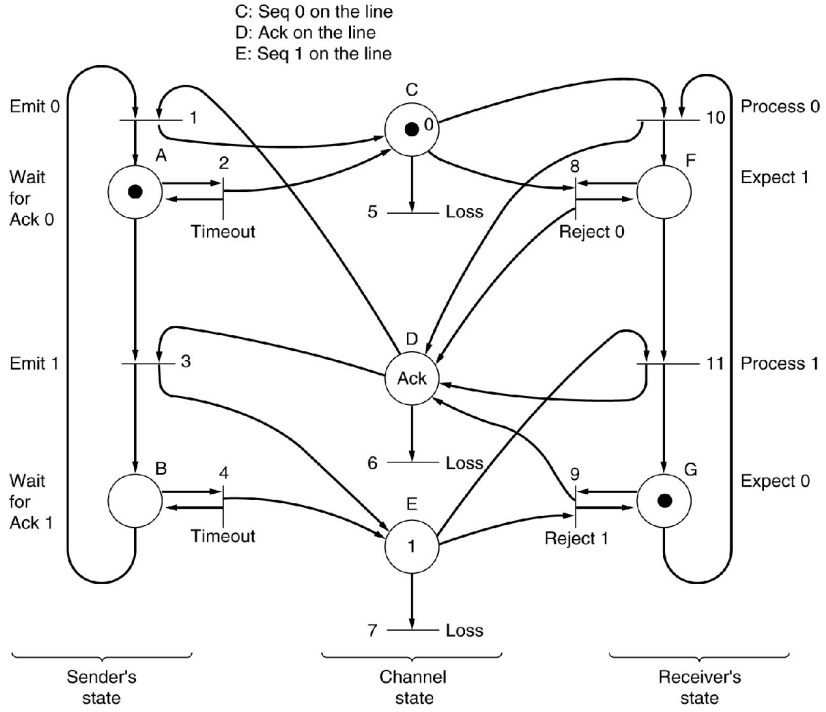
Un réseau de Pétri est un autre moyen d'interpréter le fonctionnement d'un réseau. Un réseau de Pétri est :

- un ensemble de places
- un ensemble de transitions
- un/des jetons qui peuvent être transmis d'une place à une autre. La quantité de jetons transmissibles est indiquée par la « valeur » d'une transition.



Le shéma ci-contre explique brièvement le fonctionnement d'un réseau de Pétri. Le jeton en A peut être transmis en B car la liaison entre A et B permet le passage d'un jeton. Au tour suivant le jeton en B pourra de nouveau être transmis en A par la liaison de valeur « 2 ».

Le schéma ci-dessous représente sous forme de réseau de Pétri le fonctionnement d'un transfert sur un réseau :



Couche MAC

La couche MAC s'occupe du contrôle d'accès au canal. Elle est utilisée par exemple pour le CSMA ou Ethernet. Elle fonctionne au point à point (WAN) ou par diffusion (LAN) et le canal est partagé (à accès multiple ou aléatoire). Il y a donc nécessité d'établir une stratégie de contrôle d'accès au canal. C'est la couhe MAC, une sous couche de la couche liaison, qui s'occupe de ce contrôle.

Remarque : en allocation statique, le multiplexage fréquentiel (FDM) est complètement inefficace s'il y a trop peu ou trop d'utilisateurs.

Exemple : CSMA (Carrier Sense Multiple Access)

Le CSMA est protocole à détection de porteuse, c'est à dire qu'il a la capacité « d'écoute » du canal.

CSMA persistant

Si une station veut émettre, elle écoute : si le canal est libre, elle émet, sinon elle attend qu'il se libère. En cas de collision, elle fait une pause (durée aléatoire) puis elle recommence au début.

CSMA non persistant

L'algorithme CSMA non persistant est identique au précédent. Si le canal est occupé quand la source veut émettre, elle n'attend pas qu'il se libère mais fait une pause (d'une durée aléatoire) et recommence au début. Le fait que la pause soit aléatoire permet une meilleure répartition car toutes les tentatives d'émission ne tentent pas de recommencer en même temps.

CSMA – CD (Collision Détectée)

Cet algorithme se présente de la même manière que le CSMA non persistant, si ce n'est qu'on écoute la ligne pendant l'émission (et non plus « avant » d'émettre). En cas de collision, on arrête immédiatement le transfert, fait une pause de durée aléatoire, puis tente une nouvelle fois d'émettre. Cette méthode permet une économie de temps et de bande passante. De plus, il est facile à maintenir. Le CSMA – CD est utilisé dans Ethernet.

Exemple de réseau : Ethernet

Réseau Ethernet : norme IEEE 802.3.

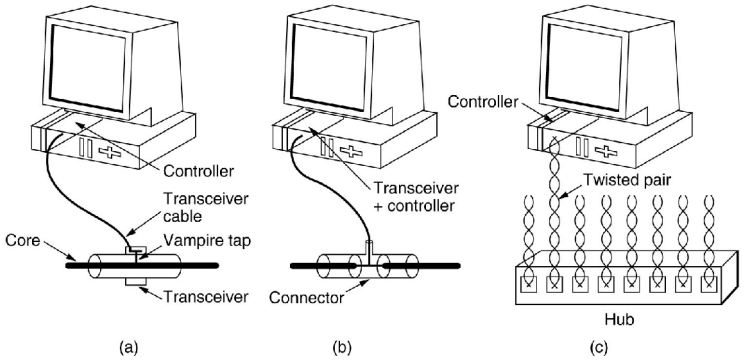
Support physique

Le tableau ci-dessous résume les différents supports physiques (et caractéristiques) utilisés pour un réseau Ethernet.

Name	Cable	Max. seg.	Nodes/seg.	Advantages
10Base5	Thick coax	500 m	100	Original cable; now obsolete
10Base2	Thin coax	185 m	30	No hub needed
10Base-T	Twisted pair	100 m	1024	Cheapest system
10Base-F	Fiber optics	2000 m	1024	Best between buildings

La carte Ethernet est alors un « controller » gérant l'émission et la réception, la transmission des trames et le calcul des contrôles. Le schéma ci-dessous montre les différents mode de connection de la carte réseau sur le réseau

- (a) et (b) : cas de connexion sur un réseau dont le support physique est un cable coaxial

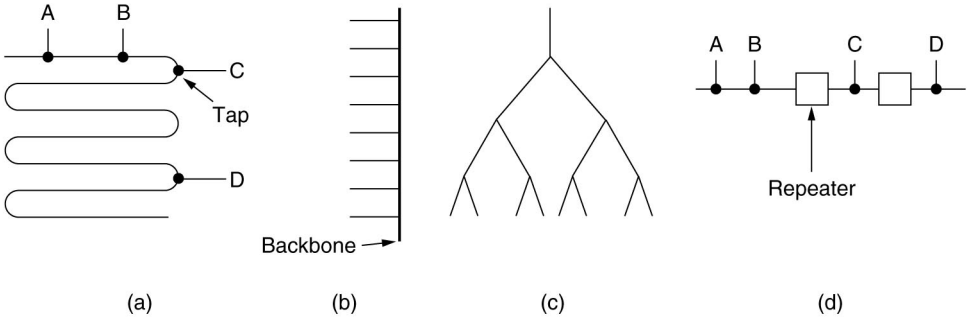


- (c) : pour les réseaux utilisant les paires torsadées (cable RJ 45 reliés à un HUB)

Topologie

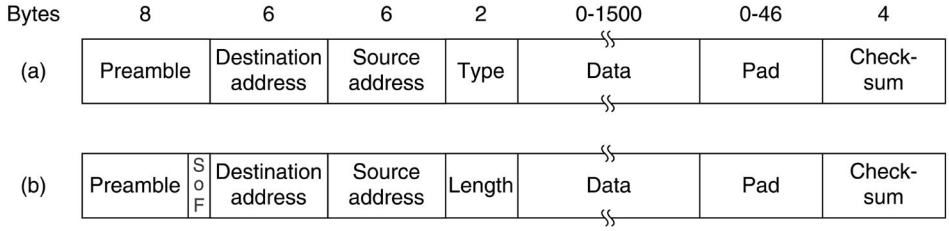
Il existe 4 grands types de topologie des réseaux Ethernet (voir le schéma ci-dessous) :

- (a) : avec des ponts, acheminants uniquement certains paquets entre plusieurs sous réseaux.
- (b) : topologie en BUS
- (c) : topologie en étoile (avec un hub central et des hubs intermédiaires)
- (d) : avec des répéteurs (permet d'étendre le réseau plus loin que les limitations dues aux cables)



Trame DIX (Dec, Intel, Xerox)

Le schéma ci-dessous montre le format d'une trame utilisée sur un réseau Ethernet.

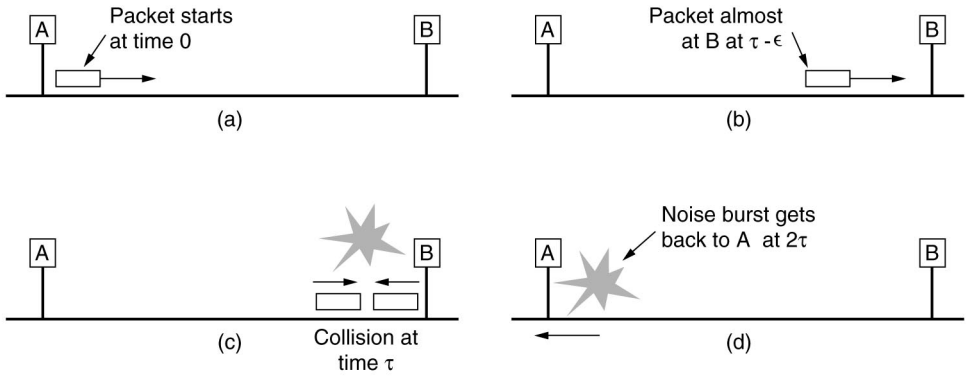


Si le bit de poids fort dans l'adresse destinataire est égal à 1, il s'agit d'un adressage de groupe, de diffusion restreinte, de multidestinataires ou multicast. Si tous les bits sont à 1, il s'agit d'une diffusion générale (broadcast).

Le type correspond à l'identifiant du protocole réseau souhaité.

La taille minimale d'une trame est de 64 octets. Si nécessaire, on complète avec des octets de remplissages.

Détection de collision



On note τ le temps de propagation jusqu'à l'extrémité B. Le temps de transmission est supérieur à 2τ. De cette valeur dépend la taille de la trame. En effet, la taille d'une trame assure que la collision sera détectée et donc traitée (sinon on envoie toute la trame avant la collision).

Si l'on a un débit de 10Mbits/s et une longueur maximale de 2500m, avec un temps de propagation 2τ = 50us et un

temps de transmission (1 bit) = 100us, une trame doit alors avoir une longueur minimale de 500 bits, soit 64 octets (éventuellement remplissage).

Si l'on a un débit de 1Gbits alors la taille minimale d'une trame est 6400 octets (pour 2500 m), 640 octets (250 m).

Extentions d'Ethernet

Ethernet commuté

Utilisation d'un commutateur (switch) et décomposition en plusieurs sous réseaux. On a alors un domaine de collision par sous réseaux (on réduit l'espace de collision, car seules les branches utiles sont utilisées).

Evolutions

- Fast Ethernet (802.3a) : 100Mbits/s, support (UTP5, fibres), mêmes trames
- Gigabit (802.3z) : 1Gbits/s, nouvelles trames, topologie multipoints

Couche réseau

La couche réseau propose des services utilisés par la couche transport :

- indépendances des technologies de routeur et de la topologie du sous-réseau
- système d'adressage homogène (WAN, LAN)

On distingue deux approches :

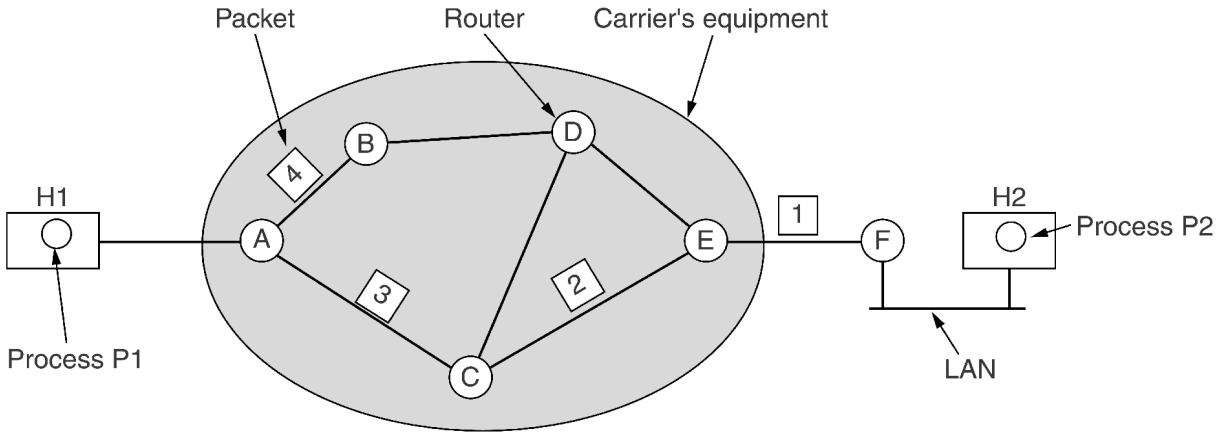
- Internet : transport des paquets uniqument (send, receive), pas de contrôle de flux ni de classement.
- Télécom : service fiable en mode connecté, qualité de service, trafic, temps réel (voix, vidéo).

Mode sans connexion

Le mode sans connexion est un mode datagramme avec transfert individuel des paquets.

Par exemple (voir le schéma ci-dessous) P1 veut transmettre un message m à P2. La couche réseau découpe ce message m en 4 paquets. Les paquets sont envoyés au routeur. Une table de routage précise l'adresse du destinaire et le port de sortie, afin d'envoyer correctement le message. Le port de sortie est déterminé par un algorithme de routage en fonction des données de la table de routage et du paquet reçu.

Les différents routeurs du réseau sont indépendants et distribue le trajet des paquets du message de manière indépendante.



A's table

Dest. Line	initially	later
A	–	–
B	B	B
C	C	C
D	B	B
E	C	B
F	C	B

C's table

A	A
B	A
C	–
D	D
E	E
F	E

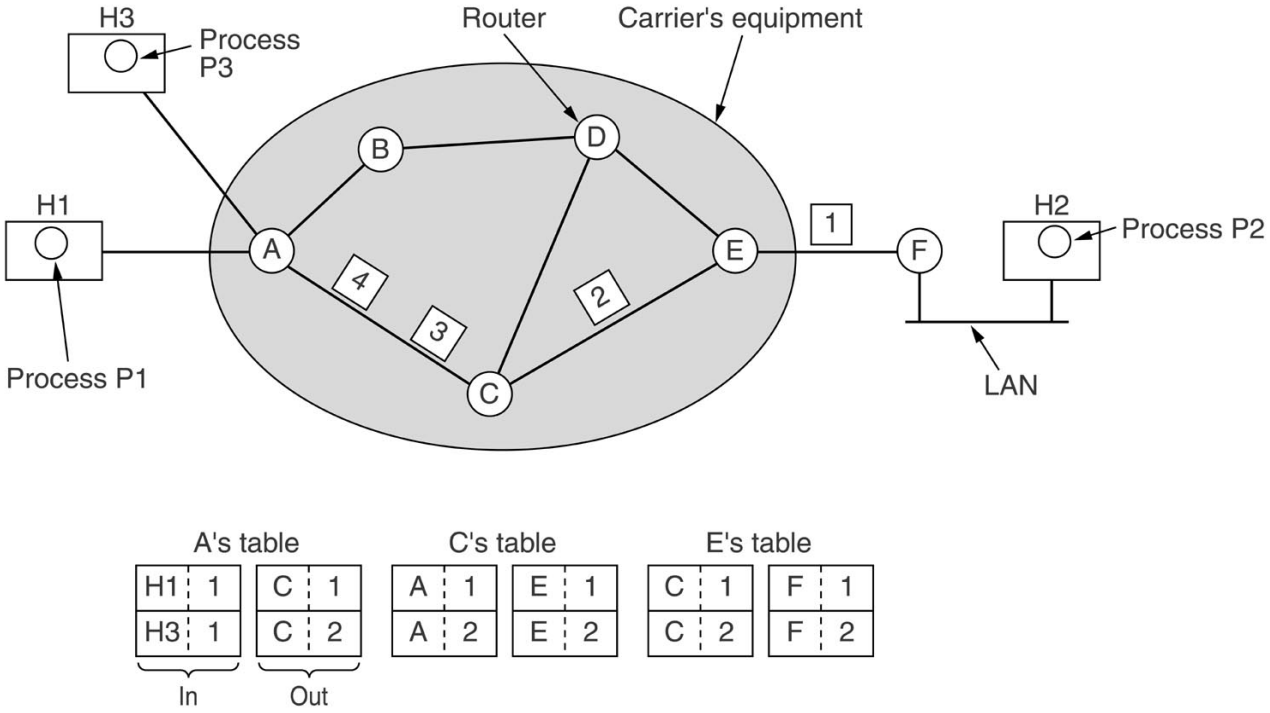
E's table

A	C
B	D
C	C
D	D
E	–
F	F

Mode avec connexion

Le mode avec connexion établie un chemin (circuit virtuel). Ce circuit virtuel est inscrit dans les tables de routage. Tous les paquets du message sont envoyés sur le même circuit virtuel grâce aux données contenues dans la table de routage (identifiant de connexion, port de sortie). A la fin du transfert, le circuit virtuel est supprimé.

Le schéma ci-dessous montre un exemple de transfert d'un message m de 4 paquets sur un circuit virtuel entre P1 et P2 et l'état des tables de routages lors du transfert.



Algorithmes de routage

Un algorithme de routage détermine le choix du port de sortie pour un paquet entrant. Il s'agit d'une fonction centrale de la couché réseau. Un algorithme de routage doit être robuste (éviter les pannes) et stable (évolutif). Il existe deux types d'algorithmes de routage :

- routage statique : les données des tables de routage sont calculées à la création du réseau en fonction de sa topologie
- routage adaptatif : les données des tables de routage sont recalculées dynamiquement en fonction des informations collectées/reçues. Des informations métriques (nombre de kilomètres), nombres de routeurs, délais, début,... sont nécessaires pour le calcul du plus court chemin (que ce soit en mode statique ou en mode dynamique).

Nous allons maintenant voir quelques exemples d'algorithmes de routage.

Algorithme de plus court chemin

Dans un algorithme de plus court chemin, le réseau est représenté par un graphe. Les arêtes sont munies d'une combinaison de métriques (km, routeurs, coûts, délais, perte, ...). On recherche alors le minimum de la fonction de coût entre chaque paire de sommets (voir le cours de graphe – licence 3).

Inondation

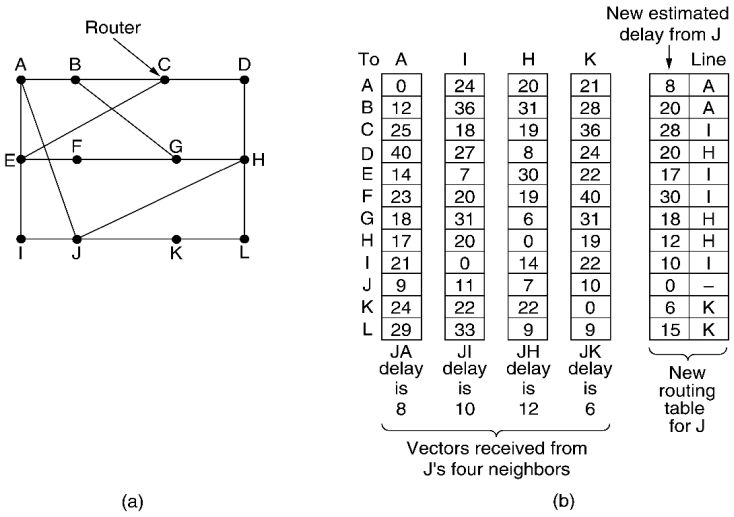
Un algorithme d'inondation envoi un paquet sur toutes les lignes pour regarder leur comportement : « mémoire » par stockage des numéros émis. L'inondation sélective inclut de plus une notion de direction des paquets.

Vecteur de distance

Il s'agit des algorithmes de Bellman – Ford ou de Ford – Fulkerson. Ce type d'algorithme dynamique a été utilisé par ARPA (1979). Chaque routeur indexe une table de routage définissant la ligne préférée et la distance évaluée. On tient aussi compte du nombre de paquets en attente, des délais d'acheminement et des distances. Les délais sont connus pour tous les noeuds adjacents et la table des délais d'un routeur est transmise à tous les voisins. Les tables de délais des voisins permet à un noeud de mettre à jour sa propre table de routage.

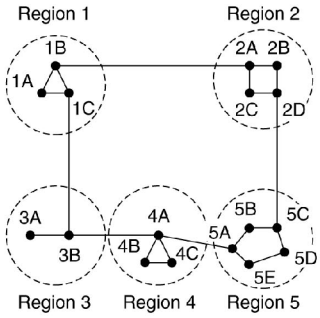
Le schéma ci-dessous montre un exemple d'algorithme des vecteurs de distances. (a) représente la topologie du réseau. En (b), on étudie à un moment donné les informations des délais des voisins de J. J mets alors à jour sa table de routage en sélectionnant les voisins par ordre de performance des délais. On obtient alors une nouvelle estimation des délais des transferts de A vers tous les autres sommets (en indiquant le chemin à suivre, i.e. le voisin par lequel passer et les délais

pour arriver à destination) :



Routage Hiérarchique

L'augmentation de la complexité des réseaux a entrainé une croissance des tables de routage et du trafic associé. Le routage hiérarchique apporte une notion de région (partie d'un réseau). Chaque région contient une table de routage interne (et indépendante des autres régions – si ce n'est pour les informations concernant le transfert vers une autre région). Il y a une seule entrée par région externes (permet la liaison entre région, de manière simplifiée). Le schéma ci-dessous montre le principe de fonctionnement d'un routage hiérarchique.



Full table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

(b)

Hierarchical table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

(c)

Contrôle de congestion

Quand il y a trop de paquets sur le réseau, il y a congestion (dégradation des performances) et des paquets sont perdus (effet accélérateur). Cela arrive s'il y a émission de plusieurs connexions sur une même ligne. On obtient alors une augmentation de la mémoire (file d'attente), entrainant une augmentation du délai de traitment et dépassement des timers, réémissions multiples,...

Traitements

On distingue deux approches :

- Approche préventive : lors de la conception du réseau (CF, acceptation, refus d'une ouverture,...)

- Approche dynamique : surveillance, ajustement local.

Ces approches sont baséees sur des compromis : 1 acquittement par paquet ou délai d'expiration des timers.

Les sous réseaux de circuits virtuels permettent :

- un routage évitant les zones de congestion
- un contrôle d'admission : plus d'ouverture de connexion dès l'apparition d'une congestion
- une préallocation de ressources à l'ouverture de la connexion (tables, buffers, bande passante) (robuste mais sous-utilisation des ressources).
- La suppression de paquets (analogie avec la production d'électricité)

Les stratégies de suppression :

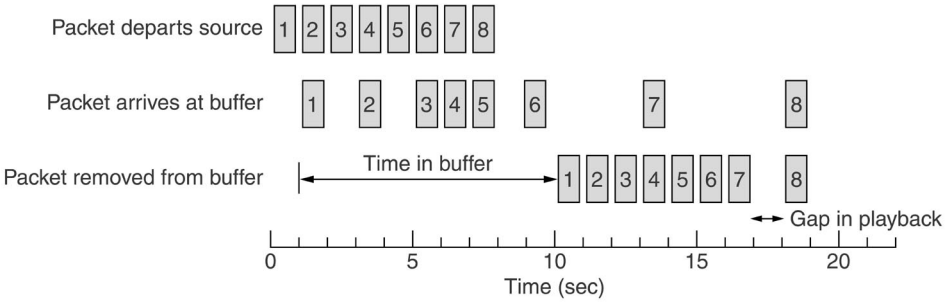
- transfert de fichiers : derniers paquets arrivés
- application multimédia : premiers paquets arrivés
- compression vidéo : images différentielles

Qualité de service

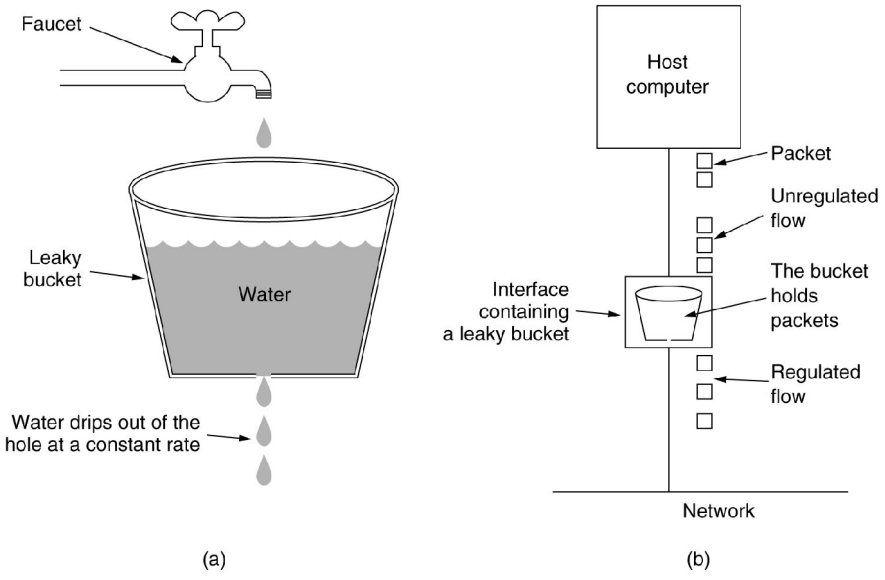
L'augmentation du nombre d'applciations réseaux, et par conséquent des attentes des utilisateurs à nécessité l'apparition d'une notion de qualité de service (QOSà pour définir une bonne adéquation services réseaux/besoins. Ici, il n'existe pas une bonne méthode mais un ensemble de techniques à combiner.

On distingue différentes stratégies :

- sur-allocation de ressources (et affinage dans le temps)
- mise en tampon de paquets audio, vidéo sur le web (le schéma ci dessous résume le principe de fonctionnement d'une mise en tampon)



- régulation de trafic : rafale, temps mort (principe identiquement au système d'un seau perçé, cf le schéma ci-dessous)



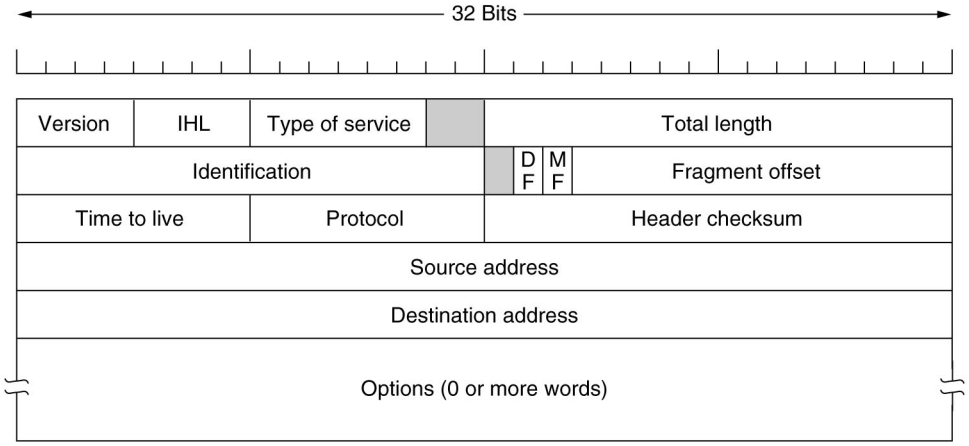
- contrôle d'admission

Interconnexion

Le but du travail des interconnexions est de permettre la liaison entre réseaux très hétérogènes. Pour cela, nous allons voir le fonctionnement de la couche réseau Internet via le procole « Internet Protocol » IP.

Le protocole IP existe maintenant en 2 versions : passage progressif de la version IPV4 (ou les adresses sont codés sur 4 octets) et la version IPV6 (adresses codés sur 16 octets). IP fonctionne en mode datagram. Format : entête + données.

Le format du datagramme IP est le suivant :



- version IP (4 ou 6)
- IHL : longueur de l'ente^te en mots de 32 bits (compris entre 5 et 15)
- type de service : classe (fiabilité, service)
- longueur totale (entête + données)
- identification : ID du datagramme (fragment)
- DF = don't fragment
- MF = More Fragment (fragments à suivre)
- Position de fragment (numéro du fragment)
- durée de vie (TTL) : décrémenté à chaque saut
- protocole de transport (TCP, UDP)
- total du contrôle d'entête (recalculé à chaque saut)
- adresses : système d'adressage global
- options : nouvelles versions, idées, ...

Le système d'adressage IP fonctionne ainsi :

- On identifie de façon unique un équipement (ordinateur, routeur) Internet. Cette identification est gérée par un numéro de réseau et un numéro d'hote d'une part, mais aussi par un numéro de carte réseau (éventuellement plusieurs sur une machine). Le numéro de carte réseau est unique (géré par l'ICANN).
- Les numéros d'adresses varient de 0.0.0.0 à 255.255.255.255. Ils sont initialement basés sur 5 classes (A : 128 * 16M, B : 16384 * 65536, C : 2M * 256, D : multicast).

Constat

IP fut conçu au départ pour connecter les universités et les DOD (Department of Défense). Mais, dans les années 90's, l'explosion du nombre de machines connectés et, dans les années 00's l'accélération des portables et des téléphones ont fait qu'il n'y a plus assez d'adresses IP.

Ainsi, l'IETF fait un appel à propositions visant à permettre des milliards d'hotes, réduire les tables de routages, optimiser le protocole (vitesse, sécurité) et faire coexister les deux versions. Une version retenue depuis 1992 est l'IPV6.

IPV6 :

- adresse plus longue (codée sur 16 octets) ce qui permet quelques millions d'adresses par m²
- entête simplifié, donc traitement plus rapide
- authentification, confidentialité incluse
- transition V4 vers V6 sur une dizaine d'années

Couche Transport

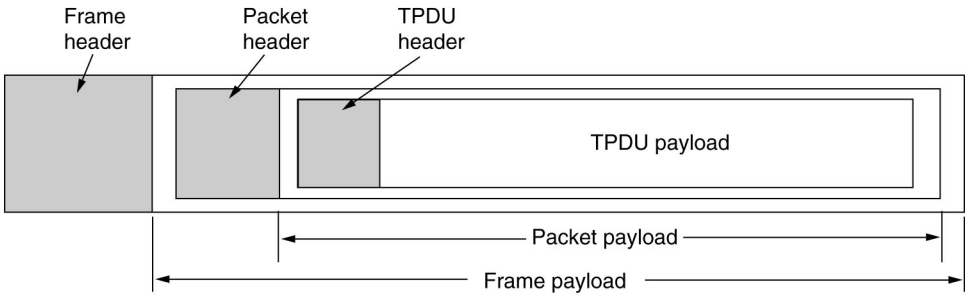
La couche transport s'occupe des services. C'est elle qui gère les sockets. On distingue deux grands protocole de la couche transport : UDP et TCP.

Les services proposés par la couche transport sont assez proches de ceux de la couche réseau, à savoir : modes avec ou sans connexion, adressage, contrôle de flux. Les différences essentielles sont :

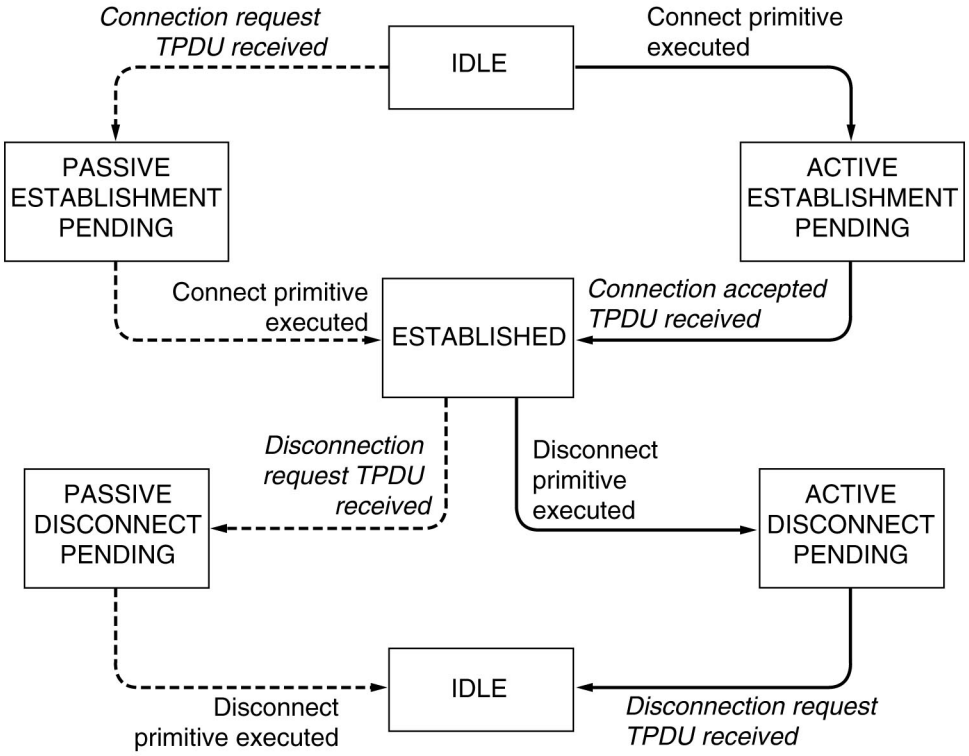
- le code transport fonctionne uniquement sur ordinateur (pas sur routeur ou opérateur)
- possibilité pour l'utilisateur d'améliorer la qualité de service (Quality of Service)
- les primitives de transport sont indépendantes du réseau, ce qui permet la portabilité des logiciels sur des réseaux différents
- la couche transport utilise des filtres pour les applications utilisateurs

Flux Transport

Echange de Transport Protocol Data Unit (TPDU) : paquet – trame :



Exemple de service transport



Sockets

Le tableau ci-dessous récapitule des primitives de TCP (Unix Berkeley) :

Primitive	Meaning
SOCKET	Create a new communication end point
BIND	Attach a local address to a socket
LISTEN	Announce willingness to accept connections; give queue size
ACCEPT	Block the caller until a connection attempt arrives
CONNECT	Actively attempt to establish a connection
SEND	Send some data over the connection
RECEIVE	Receive some data from the connection
CLOSE	Release the connection

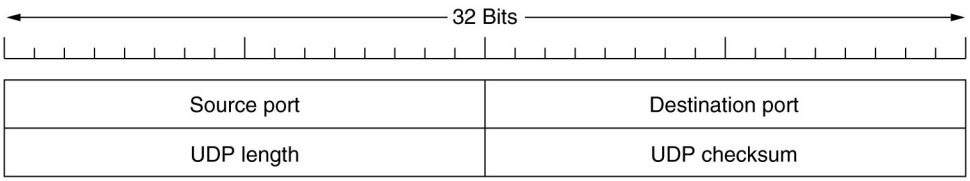
C'est à dire :

- Socket : création d'un point terminal et allocation de tables
- E : format d'adressage, type de service, protocole à utiliser
- S : descripteur de fichier (si succès)
- Bind : affectation d'une adresse réseau au socket
- Listen : allocation d'espace pour mettre en attente des appels entrant non bloquant
- Accept : création d'un socket bloquant

Exemple de protocoles de transport

UDP

UDP : User Datagram Protocol est un protocole de transport fonctionnant en mode sans connexion. Les applications encapsulent les datagrammes IP et les envoient sans établir de connexion. Un segment UDP est constitué d'un entête et des données. L'entête a le format suivant :



Analyse :

Dans ce protocole, il n'y a pas de contrôle d'erreur, de retransmission (laissés aux applications utilisateurs). UDP n'est donc juste qu'une interface pour IP adapté aux applications client – serveur (transfert simples et rapides).

Exemple DNS :

- Envoi d'un segment UDP pour connaître une adresse IP
- réponse par un segment UDP
- pas d'ouverture (fermeture, contrôle de flux, ...)

TCP

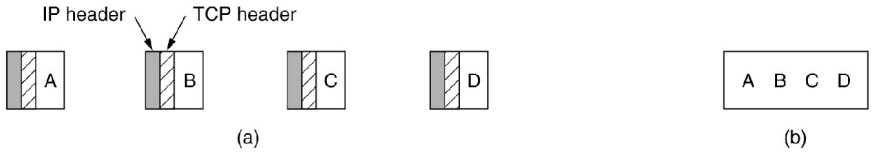
TCP : Transmission Control Protocol est un protocole de transmission fiable sur des réseaux non fiables. Il a été conçu pour s'adapter à des interconnexions de réseaux hétérogènes et évolutifs. Il est basé sur IP (datagramme) et utilise des timers pour retransmettre en cas de perte ou réassembler les données en un message ordonné. Les données sont envoyées immédiatement.

Fonctionnement :

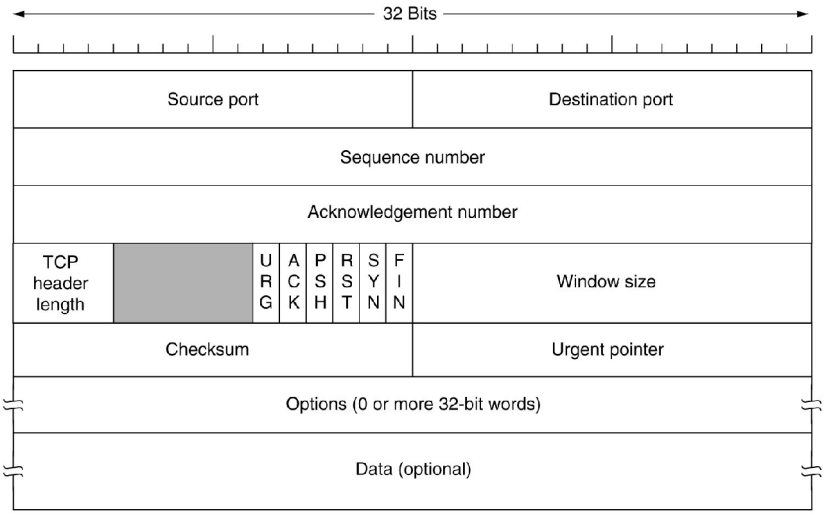
TCP fonctionne avec deux points de connexion E et R, le numéro de socket est déterminé par l'adresse IP de l'ordinateur et le numéro local (de port, sur 16 bits). Les ports inférieur à 1024 sont des ports réservés pour les services standards :

Port	Protocol	Use
21	FTP	File transfer
23	Telnet	Remote login
25	SMTP	E-mail
69	TFTP	Trivial file transfer protocol
79	Finger	Lookup information about a user
80	HTTP	World Wide Web
110	POP-3	Remote e-mail access
119	NNTP	USENET news

Un socket peut supporter n connexions. Une connexion TCP est identifiée par un coucle (socket1, socket2). On ne peut donc pas utiliser de multicast avec TCP (connexion bidirectionnelle). Enfin, comme le montre le schéma ci-dessous, TCP utilise un flux d'octets (pas de mémoire de décomposition) :



Entête :



L'entête TCP est composé d'une partie fixe (20 octets) et d'options :

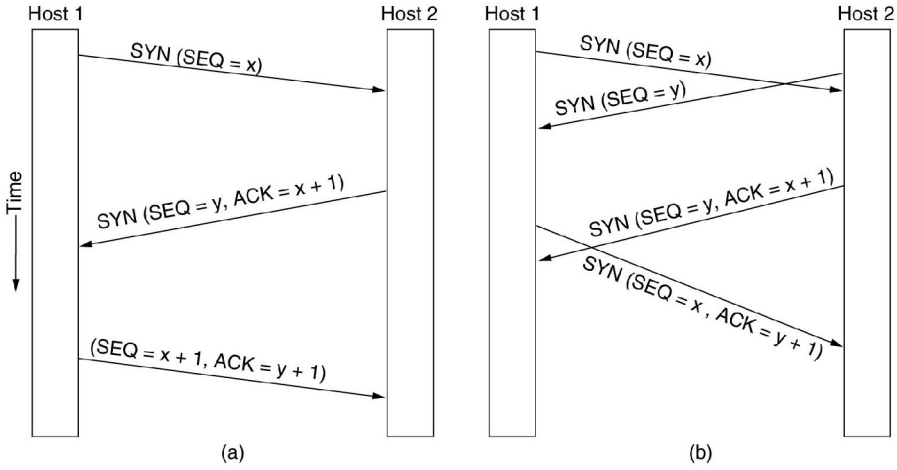
- Port source et destination : permettent d'identifier la connexion
- Le numéro de séquence : sert à l'ordonnancement
- Longueur de l'entête : mesurée en mots de 32 bits
- 6 drapeaux de 1 bits chacun :
  - URG : données urgentes à suivre
  - ACK : validité du numéro d'accusé
  - PSH : remise immédiate des données (pas de stockage)
  - RST : réinitialisation de la connexion
  - SYN : établissement de la connexion
  - FIN : libération de la connexion
- taille de la fenètre : nombre d'octets transmissibles après l'octet acquitté (peut être nulle)
- séparation acquittement / autorisation d'envoi

- fenêtre dynamique de taille variable
- checksum : entête + données

Etablissement d'une connexion :

Le serveur attend l'arrivée d'une communication (listen, accept). Le client initiateur envoie une demande de connexion (connect, IP-Port), segment TCP (SYN = 1, ACK = 0). Si l'application est en écoute (listen), alors fourniture du segment TCP, puis acceptation ou refus, sinon envoi d'un segment TCP (RST = 1).

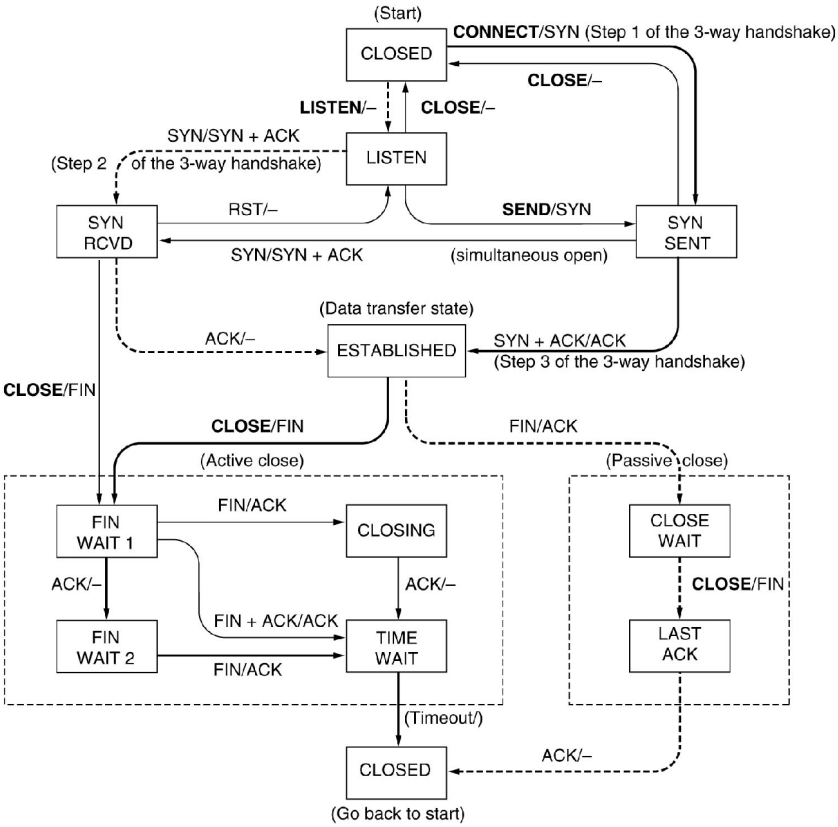
Le schéma ci-dessous résume le principe d'établissement de connexion avec TCP :



Libération d'une connexion :

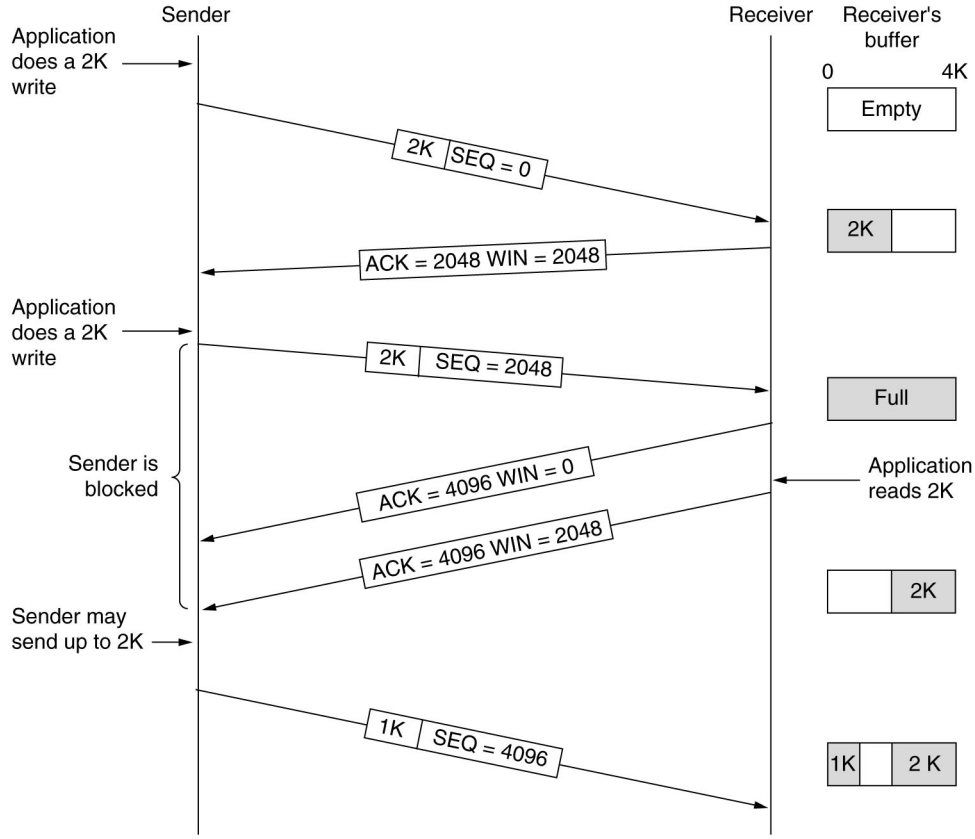
La connexion TCP étant bidirectionnelle, il est nécessaire de fermet la connexion dans les deux sens. La fermeture dans un sens se fait grace à l'envoi d'un segment (FIN = 1). Le timer déclenché à l'nvoi ferme alors la connexion si pas de ACK à l'échéance.

Le schéma ci-dessous résume le principe de libération d'une connexion :



Transmission de données :

Le schéma ci-dessous montre le fonctionnement d'une transmission de données entre un émetteur et un récepteur s'échangeant des données avec le protocole TCP :



Couche application

DNS et le Web sont des exemples concrets d'utilisations de la couche application.

DNS

Le DNS (Domain Name System) permet l'identifiation d'une adresse logique. Une adresse logique est plus facile à retenir qu'une adresse IP, et est plus stable que cette dernière. En effet, si on change de serveur, l'adresse IP change. L'adresse logique, en revanche, reste la même.

Il faut donc mettre en place un système de traduction.

A l'époque d'ARPAnet, un fichier host.txt servait de table de traduction. Cette table mise à jour chaque nuit fonctionnait convenablement car il y avait peu d'ordinateurs. La croissance du réseau, et la collision dans le choix des adresses ont impliqué la mise en place du mécanisme DNS.

Mécanisme

Le fonctionnement de DNS repose sur un système hiérarchique de domaines. Une BDR (Base de Données Répartie) implémente ce schéma et une interrogation est faite via UDP.

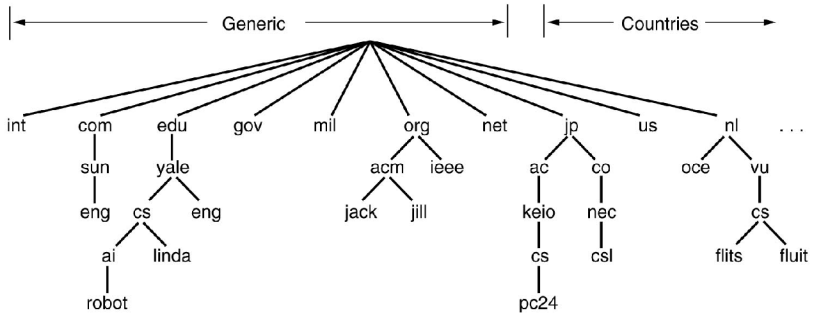
Espace de nom :

Au premier niveau, on distingue les Top Level Domain, qui sont environ au nombre de 200 et répartis parmis :

- les domaines génériques : .com (company), .edu (education), .gov (gouvernement), .org (organisation), ...
- les domaines nationaux : .au (Australie), .fr (France), .be (Belgique), ...

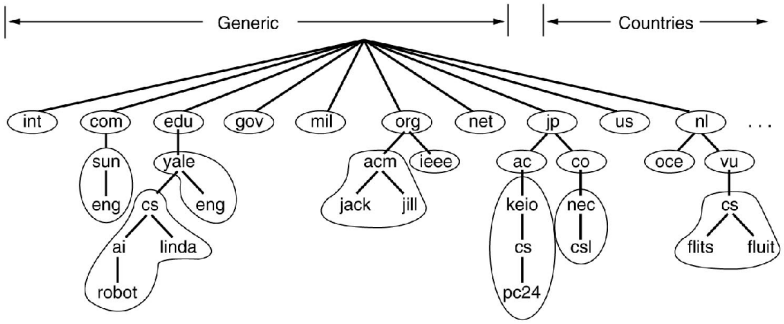
En 2000, l'ICANN a ajoutée les domaines .biz (business), .info, .name et .pro (professionnel)

Une feuille contient des machines. Une adresse est un chemin d'une feuille vers la racine. Le droit de création d'une nouvelle feuille est demandé au père immédiat. Le schéma ci-dessous synthétise une partie du Top Level Domain et de ses sous arbres :



Serveur de noms :

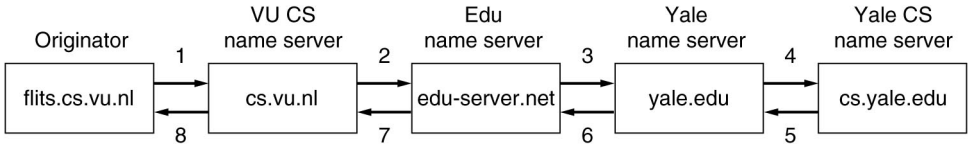
Le serveur de nom décompose l'espace des nomes en zone, comme le montre par exemple le schéma ci-dessous :



Fonctionnement

Un serveur primaire et n serveurs secondaires par zone. Une recherche locale (dans la zone) renvoie les informations directement si l'adresse se situe dans la même zone. Sinon, la recherche envoi le requête vers le serveur de premier niveau de zone, puis redescent dans l'arbre.

Le schéma ci-dessous explique le cheminement d'une requête :



Un stockage local sur le chaque permet une réutilisation de l'information si la durée de vie de celle ci n'est pas dépassée. Cela rend l'accès à l'information plus rapide et évite de charger le réseau.

Web

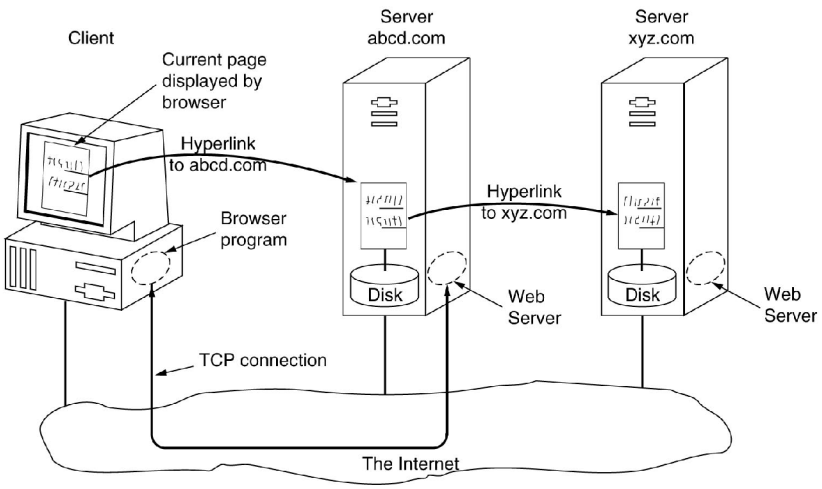
Historique

- 1989 : CERN (Centre d'Etude et de Recherche Nucléaire) – Tim Berners Lee à l'idée de présenter des informations de manière graphique sur le réseaux (=pages web), et ce, dans le but de permettre un dialogue entre physiciens de nationalités et de localisation différentes.
- 1990 : premier prototype en mode caractère
- 1991 : démo publique : Hypertext'91
- 1993 : distribution du premier navigateur Mosaïc mis au point par Marc Andreesen (Illinois)
- 1994 : Création de Netspace (MA)
- 1994 : Création du Wolrd Wide Web et Consortium (W3C) par le CERN et le MIT. Le directeur devient T. Berners Lee. Le W3 pour mission le développement, la normalisation de protocoles et leur promotion. On peut suivre son activité sur le site www.w3c.org
- 1995 : Entrée dans le domaine public et afflux des investisseurs (1.5 milliard $)
- 1998 : Rachat par AOL (4.2 milliards $)

Organisation

Le Web se découpe en deux parties, une partie navigateur qui permet d'accéder à des domaines qui sont mis en place sur des serveurs. Les serveurs hébergent des sites et des pages web, accessibles via le navigateur. Ces sites et pages peuvent contenir toute sorte d'information multimédia (texte, images, vidéos, son). La navigation entre les pages s'effectue grace à des hyperlien.

Fonctionnement



Fonctionnement coté client :

- récupération de l'URL (Uniform Ressource Locata), par exemple http://www.labri.fr/equipes/image.html
- le navigateur (N) interroge un serveur DNS pour connaître l'adresse IP de www.labri.fr
- le serveur répond 148.290.156.8
- N ouvre une connexion TCP sur l'adresse 148.290.156.8 sur le port 80
- N envoie une requête (equipes/image.html)
- le serveur www.labri.fr envoie les fichiers
- fermeture de la connexion TCP
- N affiche le texte puis les images contenues

Fonctionnement coté serveur :

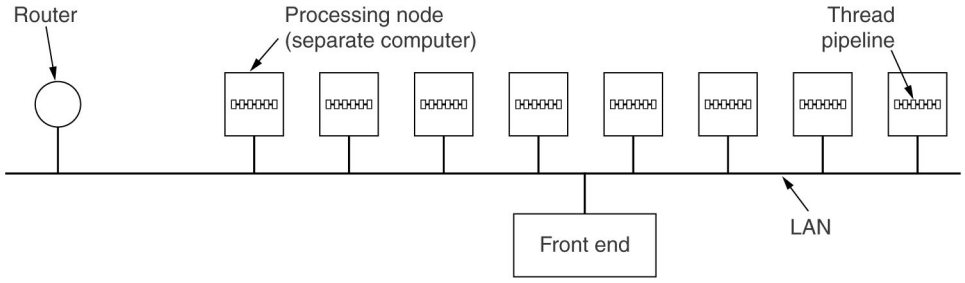
- Le serveur accepte une connexion TCP
- il récupère le fichier demandé sur le disque
- et renvoie le fichier au client

Problèmes

- Chaque requête implique, coté serveur, à un accès disque. Ce qui est trop couteux. On utilise alors le système de cache pour éviter de trop surcharger le réseau et les serveurs.
- Architecture multithread et multidisques.

Une solution optimale consiste à utiliser une architecture multiprocesseurs (grappe de PC). Un module d'entrée distribue les requêtes à différents processus. Pour la mémoire cache, il y a deux solutions : soit un accès à une mémoire partagée (un PC est utilisé uniquement pour le cache), soit les requêtes sont triées en fonction des pages demandées (spécialisation des processeurs pour un certain nombre de requête).

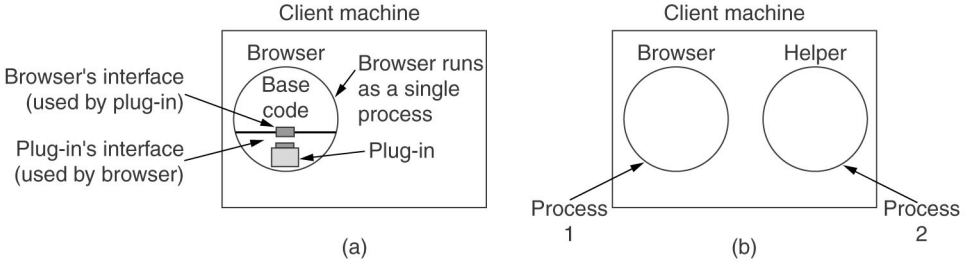
Le schéma ci-dessous résume le fonctionnement d'une architecture multiprocesseurs.



Navigateurs

Un navigateur est un interpréteur HTML (langage normalisé) offrant des boutons et fonctions de navigations. Il peut, de plus, charger des fichiers de type connu (PDF, GIF, JPEG, MPEG, MP3, ...) ou inconnu (MIME : informations autre que l'anglais ASCII, avec association d'un visualisateur à un type MIME). Il s'agit là du compromis entre le plug-in qui va charger un fichier à l'intérieur du naviagateur et une application interne (Acrobat pour PDF, Word, ...).

Le schéma ci-dessous résume les deux possibilités. On remarque toutefois que dans le cas d'un plug-in, un seul processus tourne sur la machine (a). Dans le deuxième cas, il y a un processus pour chaque application (le navigateur et les visualisateurs – ou applications – externes) (b).



URL

L'URL (Uniform Ressource Locator) permet de définir un protocole, un nom DNS de machine et un nom de fichier.

Par exemple, dans l'adresse http://www.labri.fr/image.html : http définit le protocole à utiliser (http), www.labri.fr est d'adresse DNS du serveur, et image.html est le nom du fichier (ici un fichier HTML) à charger.

Nous rappelons que si le nom de fichier n'est pas mentionné, le navigateur charge alors la page index.html par défaut. Un navigateur peut interpréter les protocoles suivants :

Name	Used for	Example
http	Hypertext (HTML)	http://www.cs.vu.nl/~ast/
ftp	FTP	ftp://ftp.cs.vu.nl/pub/minix/README
file	Local file	file:///usr/suzanne/prog.c
news	Newsgroup	news:comp.os.minix
news	News article	news:AA0134223112@cs.utah.edu
gopher	Gopher	gopher://gopher.tc.umn.edu/11/Libraries
mailto	Sending e-mail	mailto:JohnUser@acm.org
telnet	Remote login	telnet://www.w3.org:80

Optimisation des performances

L'afflux des utilisateurs sur un serveur va limiter les performances. Pour éviter cela, on utilise des proxy. Un proxy mets en cache les pages les plus demandées. Un proxy peut être hiérarchique (ST, entreprise, FAI).

Lorsqu'un client C demande une page au proxy P, P lui fournit s'il a la page en local. Sinon, il la demande au proxy « suivant », la stocke et la délivre.

Il faut alors trouver un compromis entre la durée de stockage et la péremption des données. On utilise pour cela un ensemble d'heuristiques telles que la date de dernière modification de la page, le non-stockage des pages contenant des scripts...

Sécurité

Introduction

Historique

Au départ, la notion de réseaux sécurisés était destinée aux universitaires, mais aujourd'hui elle touche le grand public (e – commerce) et les entreprises (données stratégiques). L'augmentation du nombre d'utilisateurs connectés et le transfert de données critiques ont donc créées une augmentation des problèmes de sécurité.

Nature

On distingue les dangers suivants :

- espionnage d'informations (courrier, déclarations d'impots, factures, ...)
- postage anonyme (forums, médias)
- vol d'informations (commerciales, militaires)
- escroqueries (virements, moyens de paiement)
- sabotage (par jeu, ou conflictuel)

Ces dangers pour la sécurité des données implique une confidentialité, authentification, non répudiation, et contrôle d'intégrité.

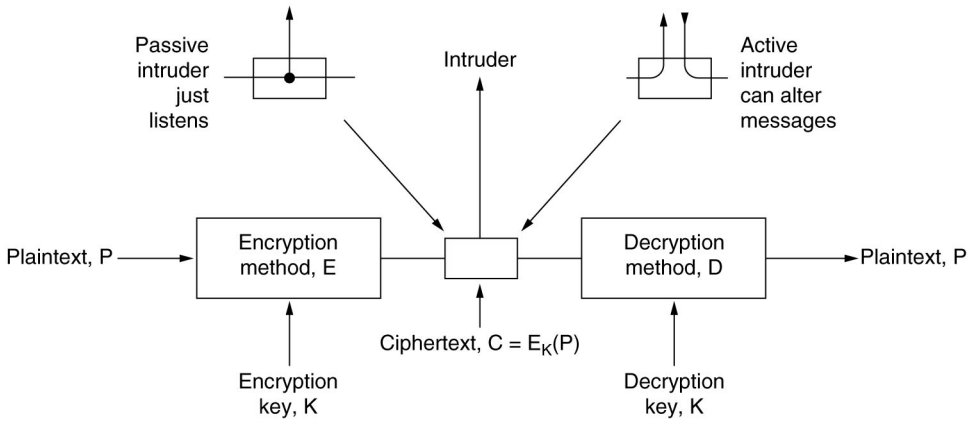
Une remarque toutefois, le SPAM n'est pas encore considéré comme un vrai problème de sécurité (mais pollution importante).

Cryptographie

(et merde! Voir cours de crypto de l'an dernier!)

Introduction

La cryptographie est une technique très ancienne consistant à coder une information à l'aide d'une clé (et décoder à la réception du message) :



Aujourd'hui, les méthodes de codage et décodage sont publiques :

- complexité de conception
- complexité de déploiement embarqué
- difficulté de garder le secret
- évaluation extérieure de l'algorithme

Ainsi, la méthode est connu, mais la clé n'est connue que de l'émetteur et du récepteur. En cas de découverte du code, il suffit d'utiliser une nouvelle clé.

La difficulté de décodage est exponentielle en fonction de la longueur de la clé (combinaison d'un cadenas) et en

fonction de l'importance de l'information.

Cryptographie : quelques exemples

Substitution

On distingue plusieurs types d'algorithmes de substitution :

- code de César : décalage de chaque lettre de 3 : a = D, b = E, ...
- dans le même esprit : décalage de k.
- bijection quelconque

Le déchiffrement exhaustif de ce type de cryptographie peut être très long, mais un déchiffrage statistique (fréquence d'apparition des lettres et des syllabes dans une langue, prise en compte du contexte sémantique pour rechercher les mots) peut rendre le décryptage très rapide.

Transposition

La transposition consiste à modifier l'ordre des symboles. Par exemple un texte écrit matriciellement, puis lu par colonne selon un ordre induit par la clé.

M	E	G	A	B	U	C	K
7	4	5	1	2	8	3	6
p	l	e	a	s	e	t	r
a	n	s	f	e	r	o	n
e	m	i	l	l	i	o	n
d	o	l	l	a	r	s	t
o	m	y	s	w	i	s	s
b	a	n	k	a	c	c	o
u	n	t	s	i	x	t	w
o	t	w	o	a	b	c	d

Plaintext

pleasetransferonemilliondollarsto
myswissbankaccountsixtwotwo

Ciphertext

AFLLSKSOSELAWAIATOOSSCTCLNMOMANT
ESILYNTWRNNTSOWDPAEDOBUOERIRICXB

Masques jetables

On choisit un masque (chaine de bits aléatoires) et on effectue un « ou exclusif » entre le mot (ASCII) et la clé (utilisé une seule fois).

Message 1:	1001001	0100000	1101100	1101111	1110110	1100101	0100000	1111001	1101111	1110101	0101110
Pad 1:	1010010	1001011	1110010	1010101	1010010	1100011	0001011	0101010	1010111	1100110	0101011
Ciphertext:	0011011	1101011	0011110	0111010	0100100	0000110	0101011	1010011	0111000	0010011	0000101
Pad 2:	1011110	0000111	1101000	1010011	1010111	0100110	1000111	0111010	1001110	1110110	1110110
Plaintext 2:	1000101	1101100	1110110	1101001	1110011	0100000	1101100	1101001	1110110	1100101	1110011

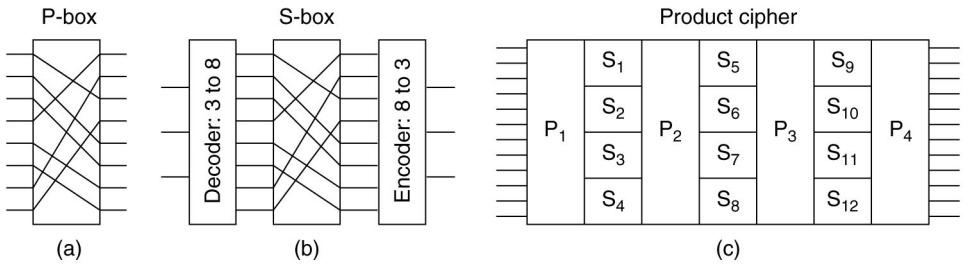
Avec ce style de clé aléatoire, il n'y a pas de fréquence d'apparition de symboles. Le message est donc quasi impossible à décrypter. Cependant, la difficulté réside dans le transport de la clé et au fait que la taille des données cryptables est limitée.

Cryptographie à clé symétrique

On peut distinguer deux implémentations pour un système cryptographique à clé symétrique :

- logicielle : souple, mais peu rapide
- matérielle :figée mais rapide

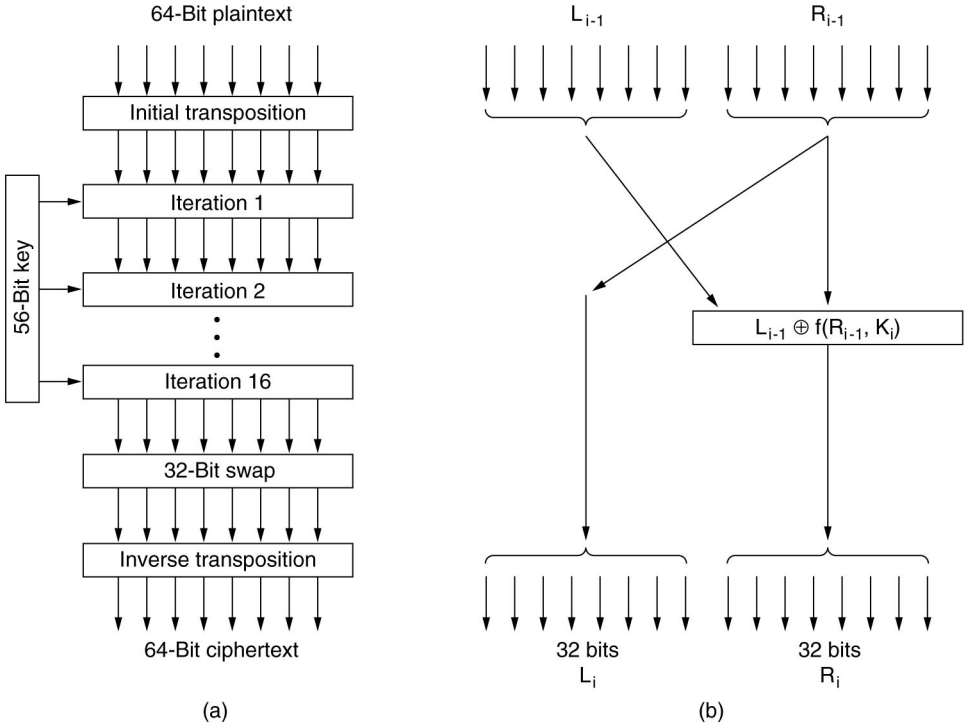
On utilise en fait un compromis entre les deux méthodes. Au niveau matériel, on code notamment les transpositions grâce à des boites de permutations P et les substitutions par des boites S. Le produit de déchiffrement (cf l'exemple ci-dessous) est une chaine de boites P et S.



DES

Le système cryptographique DES (Data Encryption Standard) est apparu en janvier 1977. Créé par IBM, il a d'abord été adopté par le gouvernement américain. L'entrée et la sortie se fait sur des mots de 64 bits. Le premier étage (respectivement dernier) est une transposition (respectivement inverse) indépendante de la clé. L'avant dernier étage est une permutation de 32 bits entre la partie Gauche et Droite. Les étages intermédiaires sont des ou exclusif, fonctions de la clé.

Le schéma ci-dessous résume les différentes étapes d'un cryptage DES :



Un problème est apparu au début du fonctionnement de DES. La clé était initialement de 128 bits (lorsqu'il a été développé par IBM). Cette clé a été ramené à 56 bits (NSA). En 1977, des chercheurs ont émis la conception d'une machine cassant DES en moins d'une journée (coût de la machine 20 M $).

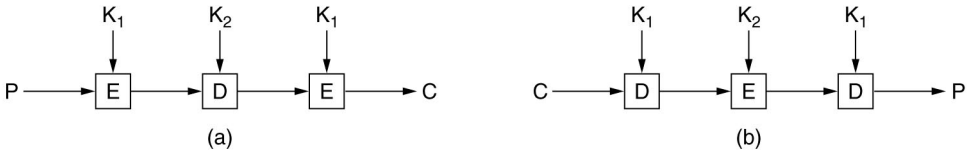
Triple DES

En 1979, triple DES (norme IS8732) apparaît. Il consiste a effectuer DES 3 fois. Les première et dernière étapes de calcul se font avec une même clé K1 (cryptage de l'entrée), l'étape du milieu est effectuée avec une autre clé K2 (décryptage de l'entrée).

L'avantage de ce système est qu'il est bien plus difficile à casser que DES (pas de machine de 20 M $ ici!) et reste compatible avec du matériel conçu uniquement pour DES. En effet, il suffit d'effectuer triple DES avec les deux clés K1 et K2 égales pour retrouver DES. En effet, on crypte avec K1 à la première étape, on décrypte avec K2 = K1 à la deuxième (on retrouve donc le message initial) et on recrypte avec K1 à la dernière étape.

Triple DES est donc un compromis entre la sureté des données, la lourdeur de transfert et la compatibilité avec DES.

Le schéma ci-dessous résume le fonctionnement de triple DES : (a) – lors du cryptage, (b) – lors du décryptage :



AES

AES a été demandé par le National Institute of Standard and Technology (NIST) en janvier 1997, sur concours avec cahier des charges : la conception doit être totalement publique, l'algorithme public, avec des clés de 128, 192 et 256 bits, implémentation peut être logicielle et matérielle.

En 1998, 15 propositions sont dans la compétition. En octobre 2000, c'est l'algorithme de Rijndael qui est considéré comme le nouveau standard américain (FIPS 197) pour son potentiel de sutur standard internationnal.

Cryptologie à clé publique

Le point faible de la cryptographie à clé publique est que la distribution des clés à tous les utilisateurs augmente le risque de vol/perte. En 1976, Diffie et Hellman, de l'université de Stanford, propose le système cryptographique à clé publique fonctionnant sur le principe de deux clés différentes. Une clé servant au chiffrement (encryption E), une autre au déchiffrement (decryption D) des données. Ainsi, pour tout message M : D(E(M)) = M.

Il est très difficile de déduire D à partir de E. La clé de chiffrement E_A est publique, la clé de déchiffrement D_A reste privée. L'utilisateur A calcul $P' = E_B(D_A(P))$ et l'envoie à B. B calcule $E_A(D_B(P')) = E_A(D_B(E_B(D_A(P)))) = P$.

RSA

Le système cryptographique RSA a été développé par Rivest, Shamir et Adelman (MIT). Ce système est basé sur la théorie des nombres (factorisation de très grand nombres). La taille de la clé est importante (1024 bits). Cette méthode est très robuste, mais assez lente.

Gestion des clés

Les clés publiques sont affichées sur le WEB : il y a donc risuqe d'interception. On doit donc utiliser un mécanisme de sécurisation. Une première solution consiste en un site gérant les clés de façon sécurisées. Mais on est alors confronté à un problème de saturation (goulot d'étranglement) et de robustesse.

Une deuxième solution consiste à utiliser les certificats.

Certificats

Le CA (organisme certificateur) délivre un certificat à un utilisateur identifié. Ce certificat est signé par la secrète du CA. Toute vérification se fait alors localement, sans accès à un serveur CA.

Organisation des CA

Une première solution consiste en un unique CA, centre de distribution de clé : mais même problème que précédemment (goulot d'étranglement)

Une deuxième solution consiste à dupliquer les CA, mais il y a alors rique de vol et problème de choix de l'organisme).

On a donc mis en place (troisième solution) le PKI (Public Key Infrastructure) : organisation arborescente des CA. Il s'agit en fait d'un CA père gérant les CA fils (clé signées) des certificats transmis dans les messages.

Réseaux sans fils

L'idée des réseaux sans fils est très ancienne. En 1901 : la transmission télégraphique (Morse), en 1997 : norme IEEE 802.11. Il existe toute sorte de nature de réseaux sans fils :

- interconnexion de systèmes : clavier, souris
- réseaux locaux : IEEE 802.11 WiFi
- grands réseaux : téléphonie, GSM, GPRS, UMTS.

WiFi : IEEE 802.11

Wifi (Wireless Fidelity) – normalisation 1990 à 1997 – est un principe de connexion pour réseaux locaux compatible avec Ethernet au dessus de la couche 2 (transmission de paquets IP). La vitesse de connexion est de l'ordre de 1 ou 2 Mbits par seconde à l'origine, mais en 1999, les normes 802.11a (5Mbits/s), 802.11b (11Mbits/s), puis en 2001 802.11g (54 Mbits/s) sont mises en place.

Il existe deux principes de fonctionnement :

- avec station de base (point d'accès)
- sans station de base (réseaux ad hoc)

Sous couche MAC

La sous couche MAC d'un réseau Ethernet classique écoute la connexion puis émet. Avec le système de réseaux sans fils, le problème de la portée de l'onde radio ne permet une écoute optimale de la « ligne ». Par conséquent, le CSMA n'est pas adapté.

On distingue deux modes :

- DCF (obligatoire)
- PCF (facultatif)

PCF Point Coordination Function)

Une station de base invite séquentiellement à émettre. Il y a donc une émission régulières d'informations nécessaires pour les émissions.

DFC (Distributed Coordinated Function)

Ce mode repose sur le protocole CSMA CA (Collision Avoidance), qui permet deux types d'écoutes :

- écoute avant d'émettre, émission complète, réémission complète en cas de collision et attentes aléatoires,
- écoute partielle et utilisation d'un canal virtuel représentant l'occupation.

Exemple

- A veut émettre vers B
- il émet une demande RTS
- B répond OK : CTS
- C entend RTS de A : NAV
- D entends CTS de B : NAV

Bluetooth

Le système de réseau sans fil Bluetooth a été conçu par Ericson en 1994 (souhait de connecter ses matériels mobiles). Il s'agit d'une technologie faible (portée, puissance, coût) qui concurrence les LAN sans fils (WiFi) En 1999, le comité IEEE adopte Bluetooth comme base de discution d'une norme (802.15) pour les PAN (Personal Area Network).

Son architecture repose sur un petit réseau constitué d'un maître et au plus 7 esclaves actifs. Les esclaves peut se mettre en veille (économie d'énergie).

FIN DU COURS – BONNES REVISIONS