Licence d'informatique – 3ème année

Cours de Réseaux

1998 - 2007

Zoubir MAMMERI

1ère Partie

Rappels

- Introduction aux réseaux
- Concepts de base du modèle OSI
- Couche Physique
- Protection contre les erreurs

Chapitre 1
Introduction aux réseaux

I. Notion de réseau informatique

Depuis la fin des années 1960, les réseaux informatiques n'ont cessé de se développer. Ils sont nés du besoin de faire communiquer des terminaux distants avec un ordinateur central. Dans un premier temps, les réseaux informatiques étaient réservés à l'interconnexion d'ordinateurs et terminaux des grands laboratoires et universités ainsi que des centres de commandement militaire. Aujourd'hui, avec le développement d'Internet, les réseaux informatiques sont partout. En effet, les réseaux informatiques se retrouvent dans les installations industrielles, les véhicules, les bâtiments, les hôpitaux, les administrations, les campus universitaires, etc.

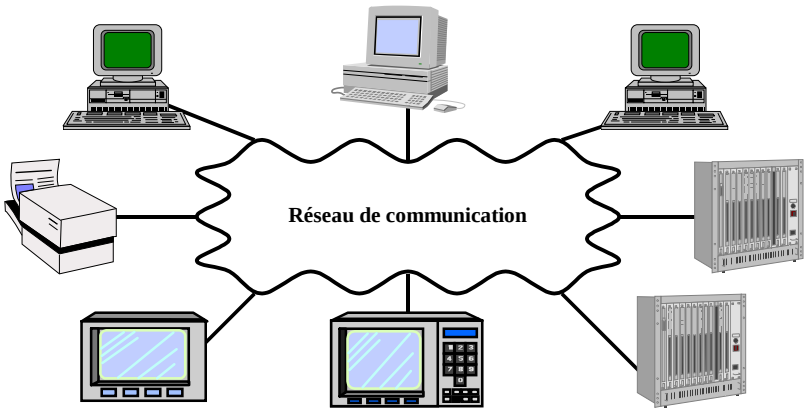
Un réseau informatique offre globalement deux groupes de fonctions : des services de transmission de données et des services applicatifs pour les utilisateurs (comme l'accès à la messagerie électronique, le transfert de fichiers à distance, la vidéo à la demande, etc.).

DEFINITIONS

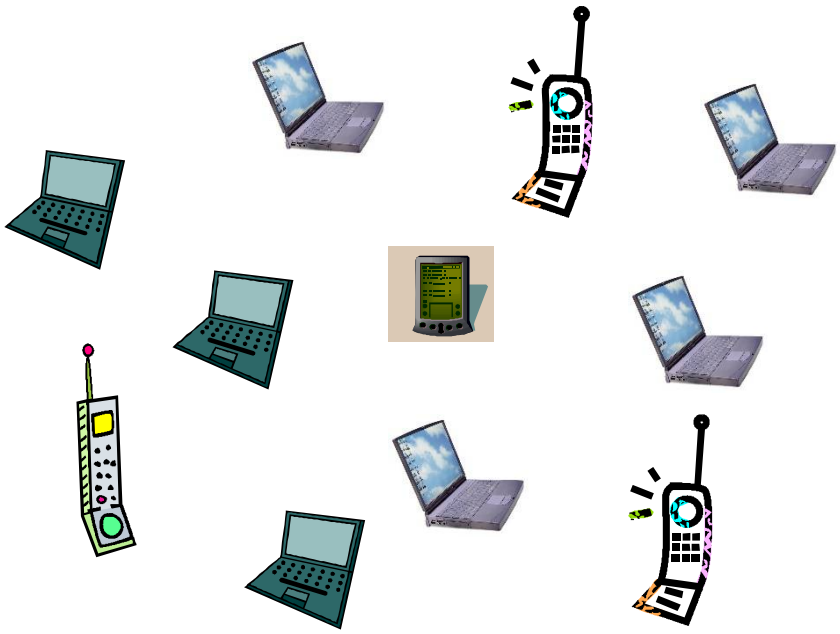
1. Un **réseau de transmission de données** peut être défini comme un ensemble de ressources (lignes de transmission, prises de raccordement, modems, etc.) permettant l'échange de données entre équipements distants.
2. Un **réseau informatique** est un ensemble de ressources matérielles et logicielles permettant à des utilisateurs distants de coopérer, d'utiliser des informations communes ou des logiciels communs, de partager des imprimantes, etc.

Un réseau est composé d'un ensemble de *nœuds*. On utilise aussi les termes *station* ou *site* à la place de nœud. Un nœud est composé d'un équipement de traitement de données et d'un équipement dédié à la communication. Ce dernier assure des fonctionnalités plus au moins complexes selon le type de réseau utilisé. Dans le cas des réseaux locaux en général, cet équipement est appelé *contrôleur de communication*. Un contrôleur de communication peut être intégré ou non à l'équipement qu'il relie au réseau. De nos jours, le contrôleur de communication est à l'intérieur du boîtier de l'ordinateur, donc le réseau est de plus en plus invisible.

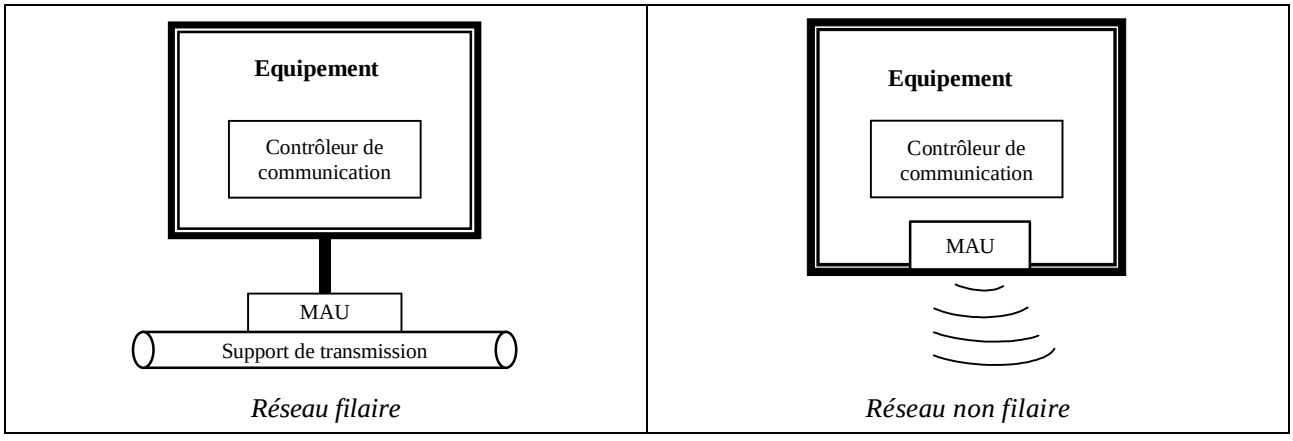
Chaque équipement est relié au support physique par l'intermédiaire d'une unité de raccordement au support appelée MAU ("Medium Access Unit"). Dans certains réseaux comme Ethernet, la MAU est appelée aussi *transceiver* ("transmitter-receiver").



Exemple de réseau filaire



Exemple de réseau sans fil



Connexion physique à un réseau

II. Classes de réseaux

Différentes catégories de réseaux peuvent être dénombrées et les critères de classification de réseaux abondent. Généralement, les réseaux sont classés selon les domaines d'application, les distances couvertes, les débits ou la mobilité.

II.1. Classement selon les domaines d'application

La prise en compte des grands domaines d'utilisation des réseaux, permet de distinguer les catégories de réseaux suivantes :

- *Réseaux bureautiques* : utilisés pour la communication de données au sein d'une entreprise, d'une université ou d'une administration pour l'échange d'informations, de documents, etc.
- *Réseaux industriels* : utilisés pour l'échange de données entre systèmes de commande d'une installation industrielle (un complexe pétrochimique, un laminoir ou une usine de montage de véhicules par exemple).
- *Réseaux embarqués* : utilisés dans les robots, les voitures, les trains, les avions, les engins spatiaux, etc.
- *Réseaux militaires* : utilisés par les systèmes de commandement militaire.
- *Réseaux de télécommunications* : utilisés pour la communication de la voix (réseaux téléphoniques).
- *Réseaux domotiques* : utilisés pour la communication dans les maisons et appartements (pour raccorder, par exemple, des appareils électroménagers, la centrale d'alarme, le système de chauffage, la télévision et le système d'éclairage).
- *Réseaux pour le bâtiment* : utilisés essentiellement pour la vidéosurveillance de bâtiments, la distribution de chaînes de télévision ou de radio, le relevé à distance des compteurs d'électricité et du gaz, etc.
- *Réseaux personnels*: utilisés par une personne pour le raccordement de plusieurs appareils appartenant à un même individu, famille ou groupe.
- *Réseaux de corps/vêtement (body/wear)* : réseaux portés par des personnes (raccordement d'appareils téléphoniques, de musique, RFID (Radio frequency Identification), implants…)

En général, les critères de choix pour tenir compte du domaine d'applications sont : la fiabilité et la disponibilité (exigées en particulier par les applications critiques), la possibilité de fonctionner dans des environnements difficiles (exigée particulièrement par les applications industrielles et embarquées), la mobilité, l'autonomie (en énergie), le débit et le coût de connexion (exigé surtout pour les applications grand public).

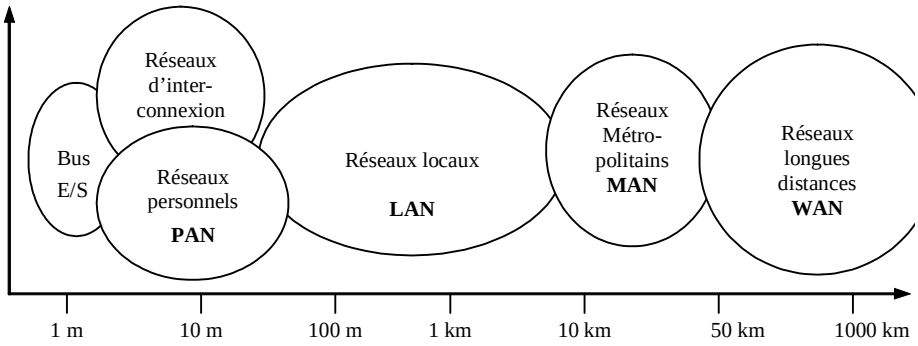
II.2. Classement selon les distances

Les réseaux peuvent interconnecter des équipements distants de quelques centimètres (ou moins) à quelques milliers de kilomètres (voire plus). Ainsi, on distingue les cinq classes de réseaux suivantes :

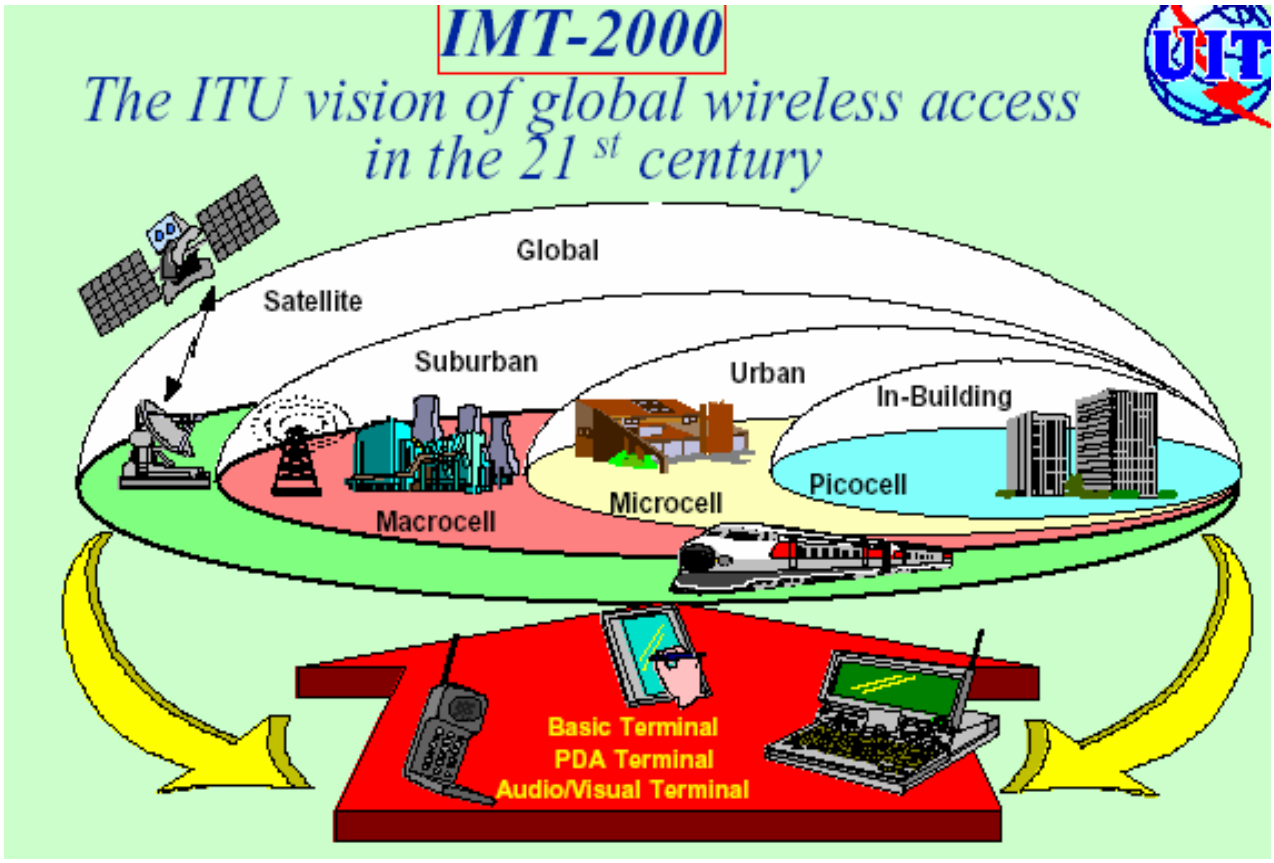
- *Bus de calculateur* : la communication entre l'unité centrale d'un calculateur et ses différents organes (clavier, imprimante, etc.) se fait via un bus. Même si le bus d'un calculateur n'est pas un réseau au sens communément utilisé pour parler de réseaux, c'est un moyen de communication avec ses propres protocoles.
- *Réseaux d'interconnexion dans les multiprocesseurs* : dans une architecture multiprocesseur, les différents processeurs (dont le nombre peut dépasser le millier) sont reliés les uns aux autres et aux blocs de mémoire par un réseau d'interconnexion. Même si leur longueur est relativement faible, les réseaux d'interconnexion s'apparentent (par leurs structures et leurs protocoles) aux autres réseaux.
- *Réseaux personnels* (appelés aussi PAN : Personal Area Networks) *:* ils permettent l'interconnexion d'un nombre réduit d'équipements appartenant à un même utilisateur. Un PAN (comme Bluetooth) permet le raccordement d'imprimante, unité de stockage... sur une distance de quelques mètres et permet ainsi d'éviter d'avoir des câbles partout.

- *Réseaux locaux* (appelés aussi LAN (“Local Area Networks”) : ils permettent d’assurer l’interconnexion d’équipements au sein d’un site géographiquement limité à quelques kilomètres (un campus universitaire, un hôpital, une usine ou un laboratoire, par exemple).
- *Réseaux métropolitains* : utilisés pour l’interconnexion au sein d’une agglomération urbaine (interconnexion de sites universitaires ou d’une entreprise par exemple).
- *Réseaux longue distance* : comme leur nom l’indique, ces réseaux permettent de transporter des données sur de longues distances, entre pays ou entre continents. Ces réseaux utilisent des infrastructures au niveau du sol et des satellites.

Pour permettre à un équipement raccordé physiquement à un réseau d’accéder à des équipements raccordés à d’autres réseaux, on utilise des équipements d’interconnexion (que l’on appelle des *ponts*, des *routeurs* ou des *passerelles* selon le type de service qu’ils offrent). Ainsi, aux yeux de l’utilisateur, un réseau peut avoir des limites qui dépassent de très loin le contexte de son réseau physique. L’exemple typique aujourd’hui est le réseau Internet où l’on peut accéder de n’importe où à un nombre quasi-infini d’informations. En réalité, Internet est un réseau de réseaux.



Différentes catégories de réseaux de communication.



Différentes catégories de réseaux de communication.

II.3. Classement selon les débits

Le débit d'un réseau désigne le nombre de bits qu'un équipement peut transmettre au maximum par seconde. Il est exprimé *kb/s* (kilo bits par seconde), *Mb/s* (méga bits par seconde) ou *Gb/s* (giga bits par seconde).

Comme pour la vitesse des processeurs, le débit des réseaux ne cesse d'augmenter. Il était de quelques centaines de bits/s fin des années 1960. Aujourd'hui, le débit de certains réseaux se chiffre en Gb/s.

Pour tenir compte du débit offert, les réseaux sont regroupés en trois catégories :

- *Réseaux à faible débit (ou basse vitesse) :* leur débit est de quelques dizaines ou centaines de kb/s. De tels réseaux tendent à disparaître, étant donné les exigences de plus en plus fortes en termes de débit dans tous les domaines.
- *Réseaux haut débit* : leur débit varie de quelques Mb/s à 100 Mb/s. Des réseaux comme FDDI, DQBD, Fast Ethernet font partie de cette classe.
- *Réseaux très haut débit* : leur débit dépasse la centaine de Mb/s et peut atteindre aujourd'hui des Gb/s. Ces réseaux sont essentiellement destinés à l'interconnexion d'agglomérations, à la distribution de chaînes de télévision et radio, à l'accès rapide à Internet. Des réseaux comme ATM entrent dans cette catégorie.

On notera que les applications les plus "gourmandes" en débit sont généralement celles où il y a des échanges d'images fixes ou animées.

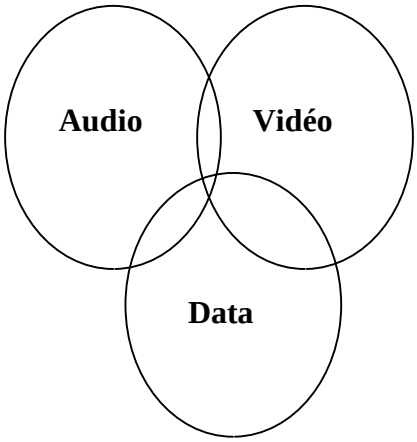
II.4. Classement selon la mobilité

La technologie du mobile a envahi le domaine des télécommunications et s'impose comme moyen performant pour la téléphonie. Elle s'installe petit à petit pour remplacer le câblage filaire dans les systèmes informatiques. Ce qui conduit à distinguer aujourd'hui deux classes de technologie de réseaux :

- *Réseaux filaires :* ils utilisent des câbles pour raccorder les équipements au réseau. La mobilité des équipements est extrêmement limitée, voire impossible.
- *Réseaux sans fil :* les équipements émettent et reçoivent en utilisant des antennes, ce qui leur permet de se déplacer (ou d'être déplacés), sans interruption de la communication. L'étendue de la mobilité dépend de l'infrastructure utilisée (types d'ondes utilisées, nombre de relais, puissance des relais, etc.). Dans les réseaux sans fil on distingue deux classes :
    - Les réseaux avec infrastructure (ceux sont les réseaux utilisant des points d'accès).
    - Les réseaux ad hoc ou sans infrastructure.

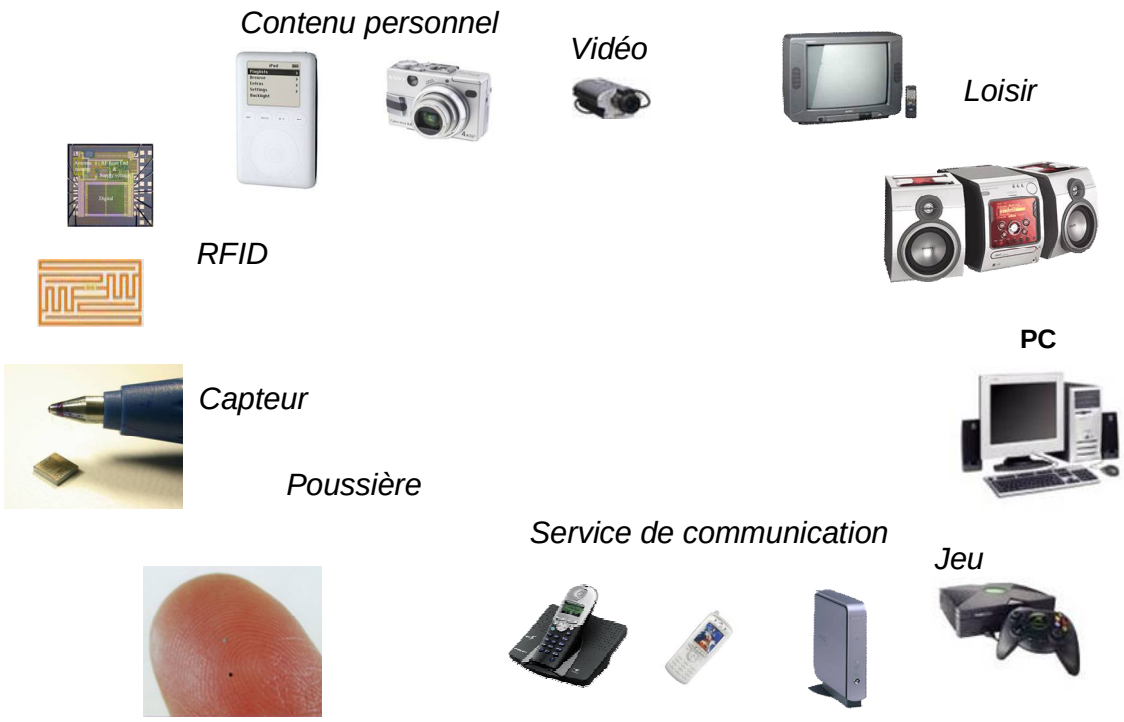
II.5. Nature des trafics et équipements connectés

Il y a quelques années, l'essentiel du trafic réseau était des données informatiques (des fichiers ou des mails). De nos jours, différents types de trafics cohabitent dans les réseaux. De plus en plus, dans un même réseau, il y a de l'audio, de la vidéo et des données informatiques. Ces trois types de trafic n'ont pas les mêmes besoins en termes de qualité de service (c'est-à-dire qu'ils exigent des débits, temps de transfert et taux d'erreur différents). Le réseau devra intégrer les mécanismes nécessaires pour supporter ces flux et répondre à leurs exigences.



Nature des trafics sur le réseau

Les équipements connectés aux réseaux sont très hétérogènes : PC, stations de travail, PDA, télévision, caméra, caméscope, ..., téléphone portable, baladeur, réfrigérateur, console de jeu, ...
Dans Internet, on parle même de *choses* (Internet des choses), tellement les types d'entités raccordés à Internet peuvent être infiniment nombreux.



Quelques uns des appareils communément raccordés aux réseaux

III. Un peu d'histoire

- 1965-1968 : lancement, par l'agence ARPA (Advanced Research Projects Agency), de projets de recherche sur la communication électroniques aux US
- 1969 : apparition du premier réseau au monde : *ARPANet* (Advanced Research Project Agency Network). C'est un réseau développé aux USA pour interconnecter des centres de calcul. ARPANet est considéré comme l'ancêtre d'Internet.
- Début des années 1970 : apparition du premier réseau français, *Cyclades*. Il a été développé pour relier des universités et laboratoires.
- Milieu des années 1970 : apparition des premiers réseaux locaux (en particulier, le réseau Ethernet).
- Années 1980 : percée des réseaux locaux. Des investissements considérables ont été faits pour équiper les laboratoires, universités, entreprises, etc. de réseaux locaux.
- 1981 : apparition de Bitnet, premier réseau d'universitaires aux US.
- 1986 : naissance de l'IETF (Internet Engineering Task Force).
- 1990 : invention du WWW (World Wide Web) par Tim Berners-Lee
- 1991 : apparition de Linux
- Début des années 1990 : début de percée d'Internet aux USA.
- 1994 : apparition de Netscape
- Milieu des années 1990 : début d'Internet en France.
- Vers la fin des années 1990 : quasi généralisation d'Internet dans le monde.
- En 2004, plus de 600 millions de personnes connectées à Internet.

Les réseaux informatiques sont aujourd'hui quasiment partout (entreprises, hôpitaux, immeubles, maisons, bureau, universités, avions, voitures, etc.).

Les réseaux couvrent des architectures qui vont des plus simples (comme deux PC reliés en point à point) aux plus complexes (comme Internet).

IV. Applications des réseaux

On distingue les applications générales et les applications spécifiques.

IV.1. Applications générales

- raccordement de terminaux à un ordinateur central (c'était l'une des premières utilisations des réseaux),
- messagerie électronique,
- accès à des bases de données distantes,
- transfert de fichiers,
- partage de ressources (logicielles et matérielles),
- applications dites CTI (Computer Telephony Integration) pour la gestion de centres d'appels téléphoniques.
- applications autour du multimédia pour le transport de la voix et de l'image.

IV.2. Applications spécifiques

- applications transactionnelles (dans les banques, compagnies aériennes, agences de voyage, etc.),
- applications en télé-x (télémédecine, télétravail, télé-enseignement, télé-achat, etc.),
- systèmes embarqués (trains, avions, voitures, chars, fusées, etc.),
- installations industrielles (pétrochimie, sidérurgie, montage de véhicules, etc.),
- gestion de la circulation en milieu urbain,
- système GPS (aide à la navigation)
- musique à distance,
- travail collaboratif (téléconférence, télé-édition de texte, télé-CAO, etc.)

V. Propriétés attendues des réseaux

1. Portée : pouvoir atteindre un maximum de sites (cela peut être contradictoire avec les besoins de sécurité),
2. Mobilité : pouvoir se déplacer sans interruption du service de communication
3. Consommation d'énergie faible pour les réseaux sans fil
4. Débits élevés (voire très élevés pour certains domaines d'applications)
5. Simplicité d'utilisation des services offerts,

6. Robustesse et disponibilité,
7. Sécurité,
8. Temps de réponse borné (essentiel pour les applications temps réel),
9. Facilité d'évolution et de migration (changement de versions de logiciels, extension du réseau, etc.),
10. Coûts faibles (coût d'installation, coût des équipements, coût des logiciels, coût de formation, etc.).

VI. Normalisation

L'objectif de la normalisation est de définir et promouvoir des normes pour permettre aux utilisateurs et fournisseurs de logiciels ou de matériels de travailler avec les mêmes concepts (au niveau syntaxique et sémantique).

La normalisation a aussi pour objectif de stimuler la concurrence entre fournisseurs de produits, ce qui devrait conduire à la baisse des coûts pour les utilisateurs.

La normalisation, dans le domaine des réseaux, s'avère de plus en plus importante vus les types d'accès existants aujourd'hui (n'importe qui peut accéder à des données se trouvant à l'autre bout de la planète, ce qui ne peut se faire que s'il y a un minimum de normes d'échange à respecter).

Il existe des organismes de normalisation :

- au niveau national (comme AFNOR en France, DIN en Allemagne, IEEE aux USA),
- au niveau continental (par exemple le CEN en Europe),
- au niveau international (ISO ou ITU).

ISO : International Organization for Standards
ITU : International Telecommunications Union.

Le processus de normalisation est souvent très lent :

- normalisation au niveau d'un pays (convaincre au niveau national),
- normalisation au niveau continental (convaincre au niveau européen, par exemple),
- normalisation au niveau mondial (arriver à imposer une idée au niveau international).

Parfois, la normalisation conduit à freiner le développement de certaines idées pour lesquelles les produits et concepts changent trop vite, comme c'est le cas des réseaux.

Il y a toujours les partisans et les adversaires de la normalisation tous azimuts. Les premiers mettent en avant les avantages de la normalisation et les seconds ont pour argument la difficulté de mettre en place des organisations efficaces, rapides et surtout vraiment indépendantes (souvent les attitudes partisanes de tel ou tel pays conduisent à geler pendant des années le programme d'une norme).

Certains considèrent que les réseaux propriétaires (c'est-à-dire non normalisés) constituent une arme efficace pour accroître la sécurité d'un système informatique (en effet, si on ne connaît pas les règles de communication utilisées au sein du réseau d'une organisation, il est plus difficile d'y pénétrer).

VI. Les RFC (Request For Comments)

Dans le domaine Internet, on ne parle pas de normes mais de standards. Comme les standards Internet sont censés évoluer dans le temps suite à des améliorations/remarques/critiques émanant de tout le monde, les documents officiels qui contiennent les standards sont appelés RFC. Ils sont publiés par un organisme appelé IETF (Internet Engineering Task Force).

La liste complète des RFC se trouve à : http://www.ietf.org/iesg/1rfc_index.txt

Il existe des milliers de RFC (presque 4800 en janvier 2007). Depuis avril 1969, des centaines de RFC sont rajoutés chaque année. Des RFC peuvent devenir obsolètes suite à l'apparition de nouveaux RFC.

Les RFC sont classés, selon cinq classifications : **obligatoire**, **recommandé**, **facultatif**, **limité**, **non recommandé** ainsi que trois niveaux de maturité qui sont **standard proposé**, **standard brouillon**, **standard internet**.

Voici quelques numéros de RFC très utilisés :

- IP (Internet Protocol) : RFC 791
- Assigned Numbers : RFC 1340
- The IP Network Address Translator (NAT) : RFC 1631
- ARP (Address Resolution Protocol) : RFC 826
- Icmp (Internet Control Message Protocol) : FRC 792
- TCP (Transmission Control Protocol) : RFC 793
- UDP (User Datagram Protocol) : RFC 768
- DHCP (Dynamic Host Configuration Protocol) : 2131
- Bootp (Bootstrap Protocol) : RFC 951
- Ipsec (IP Security) : RFC 2401, 2402, 2406, 2408, 2409, 3095…
- PPP (Point-to-Point Protocol) : RFC 1661
- DNS (Domain Name Server) : RFC 1033, 1034, 1035
- VoIP : RFC 3550, 2032, 3261

Chapitre 2
Concepts de base du modèle OSI

I. Introduction

Problèmes posés par la prolifération des réseaux

- hétérogénéité des matériels, langages, systèmes
- différents formats de codage et de représentation des données
- ajout et retrait de stations dynamiquement
- utilisation de supports variés (ligne téléphone, fibre optique, câble coaxial, …)
- …

Nécessité d'un modèle OSI pour l'interconnexion de systèmes ouverts
(ISO's Open Systems Inteconnection basic reference model)
Début des travaux 1977, norme ISO en 1984

Objectifs du modèle OSI

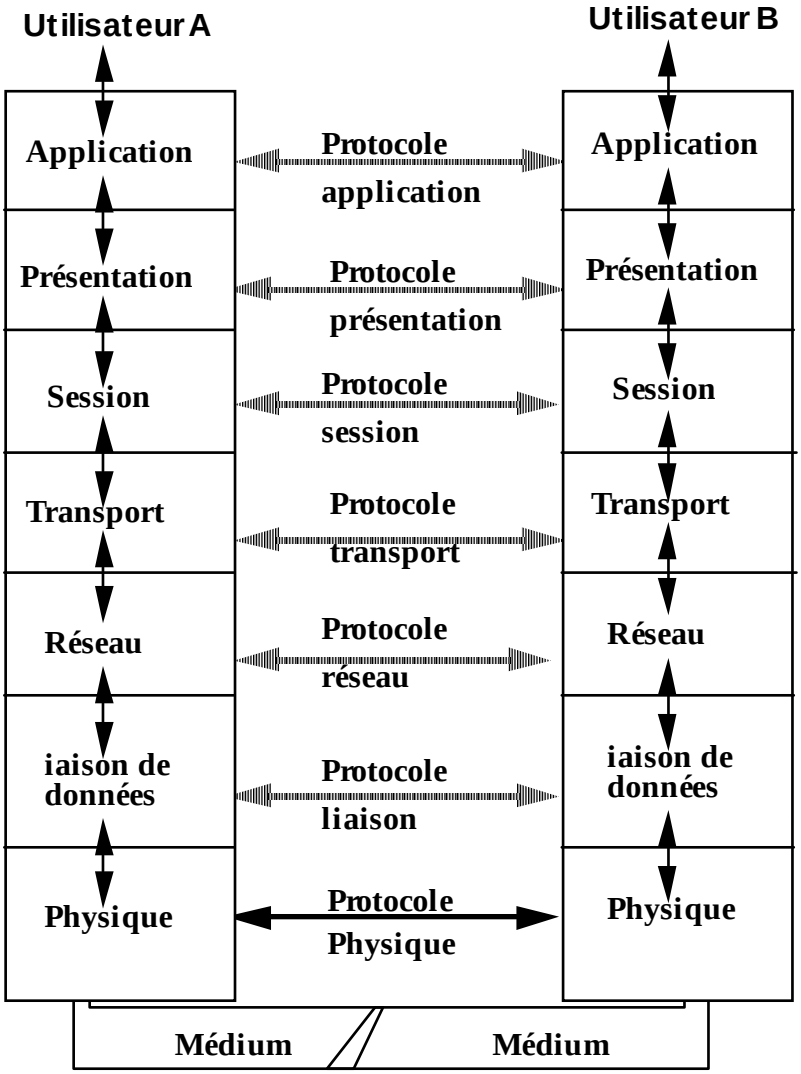
- fournir une base commune pour coordonner le développement des réseaux ;
- permettre un maximum d'indépendance vis à vis du matériel et des fournisseurs ;
- permettre plusieurs débouchés pour un même produit ;
- pérennité des investissements ;
- permettre plus de flexibilité d'utilisation des composants matériels/logiciels ;
- permettre l'interopérabilité et la portabilité des applications ;
- permettre plus de concurrence et donc plus de baisse des coûts.

Principe

- diviser les fonctions à réaliser en familles (notion de couches) ;
- spécifier, réaliser et tester séparément les couches ;
- faire évoluer une couche sans toucher à ses voisines.

Aujourd'hui : Il existe peu ou prou de systèmes compatibles ISO de 1 à 7
Le modèle OSI sert surtout de modèle conceptuel et de référence.

II. Architecture du modèle OSI



Architecture du modèle de référence OSI

Modèle à 7 couches :

- Les couches 1, 2, 3 et 4 sont orientées vers le transport d'informations.
- Les couches 5, 6 et 7 fournissent des services d'accès à la communication pour différents types d'applications.

1 Couche physique

Elle assure la transmission physique des bits, en tenant compte du type de support utilisé, du débit, …

2 Couche liaison de données

- Son rôle est d'établir, de maintenir et de terminer des connexions logiques entre deux entités communicantes.

- Elle permet de détecter les erreurs de transmission, de demander les retransmissions, de contrôler le flux, …

3 Couche réseau

- Elle assure l'acheminement de paquets dans les réseaux maillés.

- Elle détermine le chemin optimal pour les paquets de manière dynamique ou non.

- Exemples : X25 et IP.

4 Couche transport

- Elle sert d'intermédiaire entre les couches orientées traitement et celles orientées communication.

- Elle fournit un mécanisme fiable de communication.

- Elle assure le contrôle d'erreurs et de flux de bout en bout.

- Elle adapte les paquets selon les réseaux (fragmentation et réassemblage, multiplexage des connexions, …).

5 Couche session

- Elle offre les moyens d'organiser et de synchroniser le dialogue entre applications.

- Elle offre les mécanismes de reprise en cas d'anomalies.

6 Couche présentation

Elle est chargée des problèmes liés à la diversité de codage des informations, de formats de fichiers, …

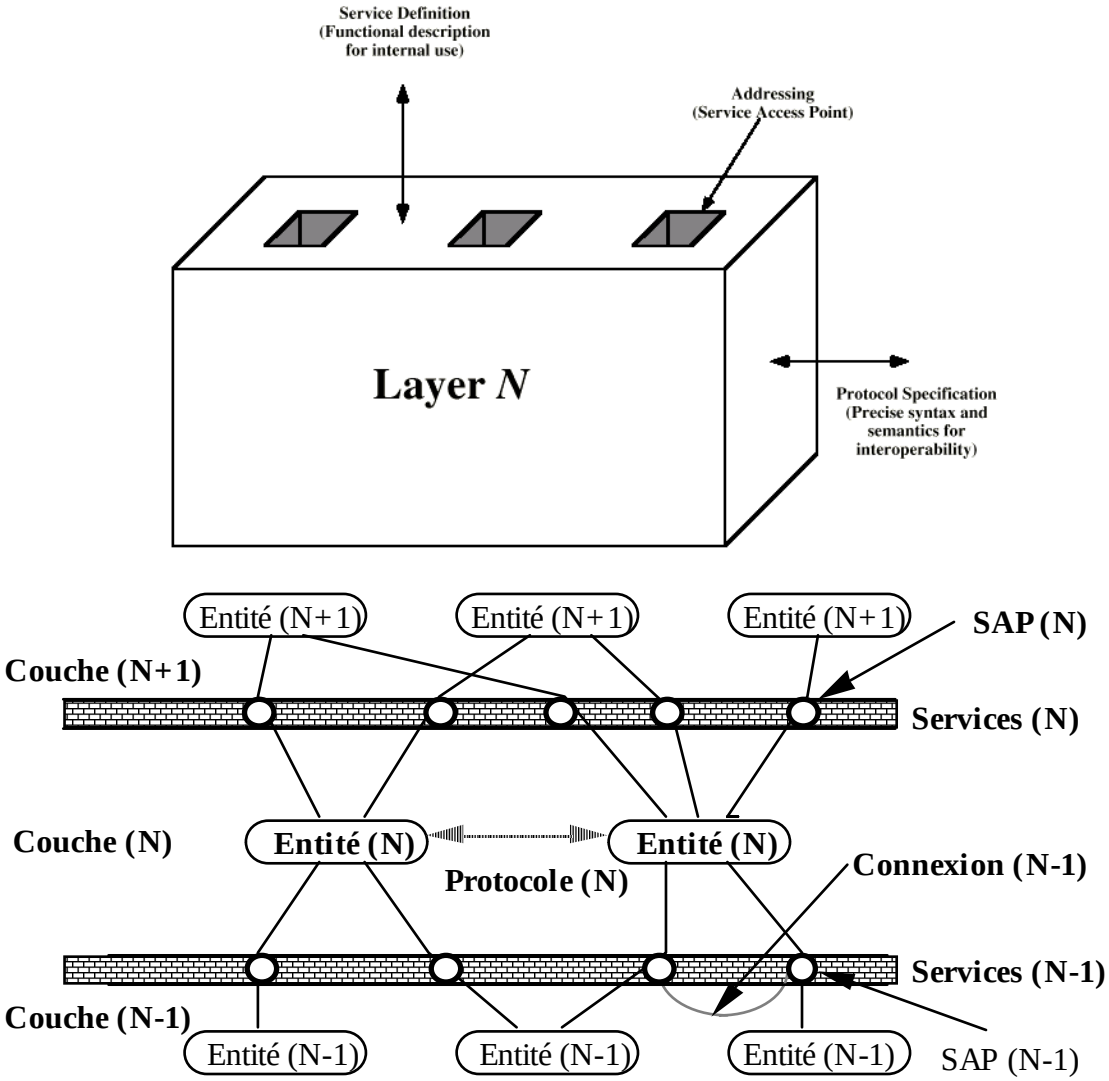
7 Couche application

- C'est la source et la destination des données, toutes les autres couches n'existent que pour la servir.

- Elle offre à l'usager les moyens lui permettant d'accéder à l'environnement OSI.

III. Principales caractéristiques communes aux couches OSI

Une couche peut être considérée comme une boîte noire avec son interface et son protocole.



Une couche et son environnement

Les termes suivants sont utilisés par différentes couches avec des sens (à peu près) identiques :

- services et protocoles,
- adressage,
- point à point ou multipoints ou diffusion,
- connexion,
- acquittement,
- contrôle de flux,
- multiplexage.

a) **Services, protocoles et PDU**

- **Service** (fonction)

Un service est défini par un ensemble de primitives paramétrées

Ex. `Connect(Adresse1, Adresse2,...),`

`OpenFile(NomFichier, Site,...),`

`SignalError(typeErreur, Adresse,...)`

`TtransferFile(NomFichier, Site,...) ...`

- **Protocole**

Si on considère par exemple, le service de transfert de fichier :

+ Le site source (où se trouve le fichier) doit activer la communication avec le site destinataire, il doit s'assurer que le destinataire est prêt à recevoir le fichier, il envoie le fichier morceau par morceau et doit s'assurer que tous les morceaux sont bien reçus.

+ Le site destinataire doit attendre l'établissement de la communication par la source, s'assurer que l'espace mémoire nécessaire pour stocker le fichier est disponible, recevoir le fichier morceau par morceau, acquitter les morceaux au fur et à mesure de leur réception,...

+ La source doit présenter les messages selon un format connu par le destinataire. Si les deux sites utilisent des formats internes de données différents, il faut prévoir des mécanismes de traduction des formats....

A travers l'exemple précédent, on voit l'importance de règles pour mener à bien une communication. Ces règles sont appelées **protocole**.

Dans le cas général, un protocole définit des règles sur

+ *Syntaxe* (formats des trames, paquets ou messages)

+ *Sémantique* : sens des messages, information de contrôle, gestion d'erreurs

+ *Timing* : définition des temporisateurs pour déclencher les opérations (d'émission de données ou d'acquittements, de retransmission...), séquencement des opérations et des PDU (dans quel ordre les PDU doivent être émis et traités).

- PDU

Les données échangées entre deux entités appartenant à une même couche sont structurées sous forme de PDU (Protocol Data Unit). Chaque couche définit un ou plusieurs formats de PDU. Sans la connaissance exacte de la forme des PDU, deux entités ne peuvent pas communiquer. Le format de PDU est donc un aspect important dans la définition du protocole d'une couche.

b) **Adressage** :

L'objectif d'un mécanisme d'adressage est d'identifier sans ambiguïté l'entité correspondante dans une communication. L'entité peut se trouver à n'importe quelle couche. En général, on utilise :

- un nom logique (ex. la machine `marine.cict.fr`, ou l'adresse IP `129.12.34.90,` le port TCP `25`)

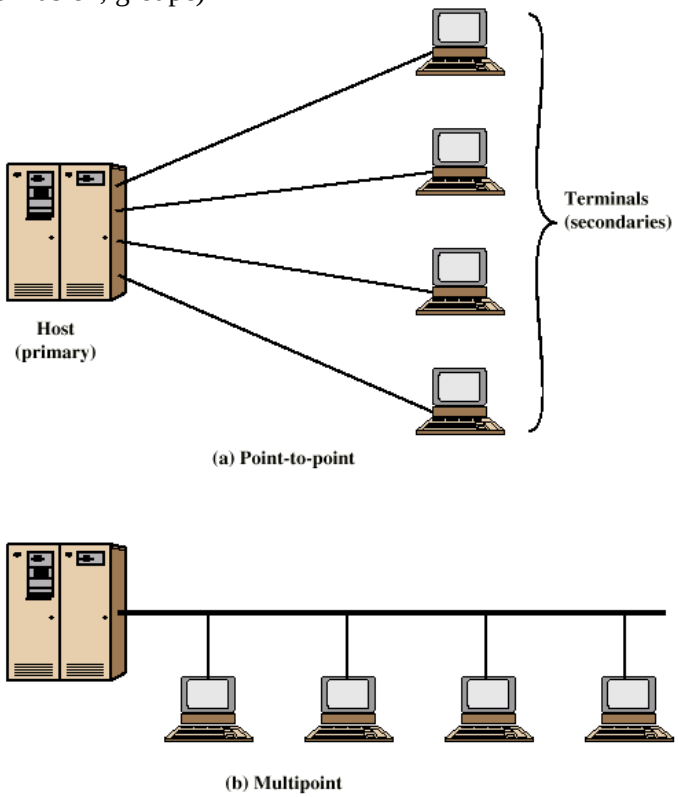
- nom physique (une adresse MAC enregistrée par le fabricant de la carte réseau ou par l'utilisateur via un outil de configuration).

- une combinaison de noms logiques ou une combinaison de nom(s) logique(s) et de nom(s) physique(s).

c) **Type de liaison**

- point à point

- multipoint (diffusion, groupe)



Liaison point à point et multipoint

Au niveau physique, sur une liaison point à point, le signal part d'une source et est reçu par un seul destinataire. Par contre, sur une liaison multipoint, un même signal transmis se propage sur le médium et est reçu par plusieurs destinataires différents (il n'y a pas besoin d'envoyer le message à chaque destinataire séparément). Même si c'est au niveau de la couche physique que l'on voit le mieux la notion de liaison point à point et multipoint, cette notion se trouve au niveau de toutes les couches. Dans les couches supérieures à la couche liaison de données, les liaisons multipoint se gèrent en utilisant éventuellement des liaisons point à point au niveau physique. Par exemple, on peut avoir des liaisons de niveau Réseau en multipoint en passant par un réseau où physiquement les liaisons sont en point à point. Dans ce cas, la couche Réseau va dupliquer les paquets sur toutes les liaisons inférieures en point à point sans que la couche transport ne soit au courant.

d) **Connexion**

- mode orienté connexion (mode connecté)
- mode non orienté connexion (mode non connecté)

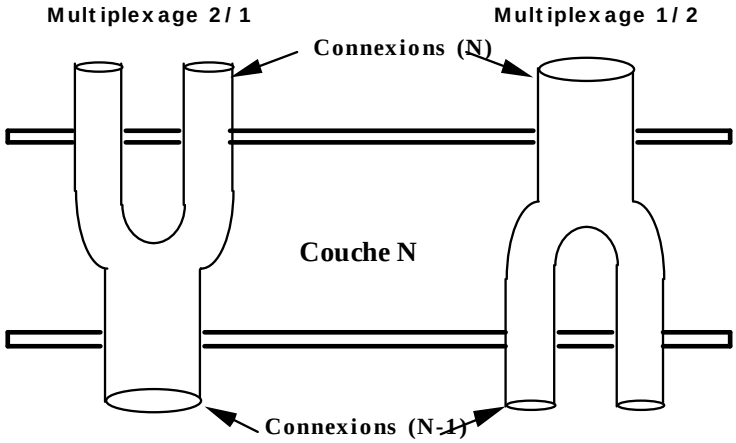
e) **Acquittement**

- avec acquittement : pour chaque échange | pour n échanges
- sans acquittement

f) **Contrôle de flux**

- sans contrôle de flux
- avec contrôle de flux
  - . stop-and-wait (1 message à la fois)
  - . fenêtre coulissante (n messages, anticipation sur les acquittements)

g) **Multiplexage** (fusion ou éclatement de connexion)



Principe du multiplexage en fusion et éclatement

IV. Accès aux services OSI

IV.1. Les quatre types d'opérations

Le service offert par une couche est accessible via des **primitives** ayant des **paramètres**. Plus on monte au niveau des couches, plus il y a de primitives.

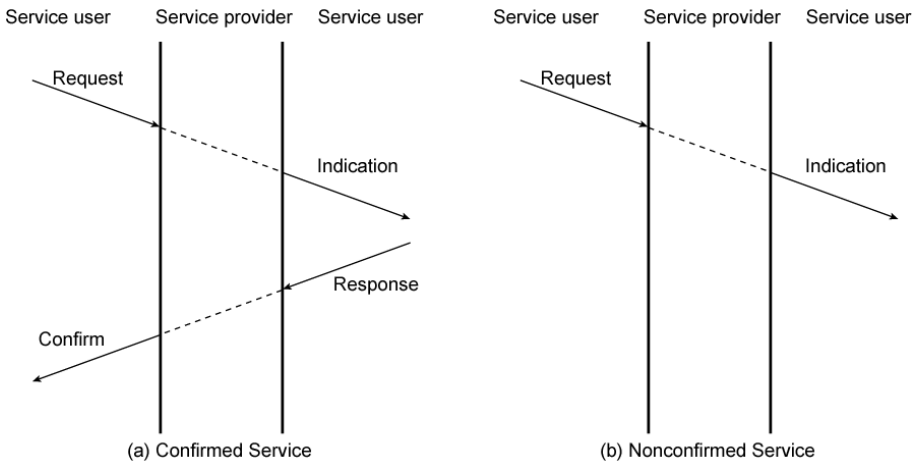
Pour maîtriser la chronologie de traitement des opérations liées à un service, on utilise une classification (abstraite) des primitives en quatre groupes :

- *Request* (requête) : primitive appelée par un utilisateur de service pour demander le service et passer les paramètres d'appel

- *Indication* (indication) : primitive déclenchée par le fournisseur de service (c'est-à-dire la couche inférieure) pour signaler soit une demande de service envoyée par le correspondant distant, soit pour signaler une situation anormale (erreur par exemple) détectée par le niveau inférieur.

- *Response* (réponse) : primitive appelée par l'utilisateur de service pour répondre à une demande de service en précisant, comme paramètres, les résultats qu'il souhaite envoyer à son correspondant.

- *Confirm* (confirmation) : primitive déclenchée par le fournisseur de service pour rendre compte et/ou retourner les résultats correspondant à la requête précédemment faite par l'utilisateur local de service.



Principe d'enchaînement des primitives de service (b)

IV.2. Types de fonctionnements possibles

a) **1-2-3-4-5-6-7** *: service avec réponse du correspondant*

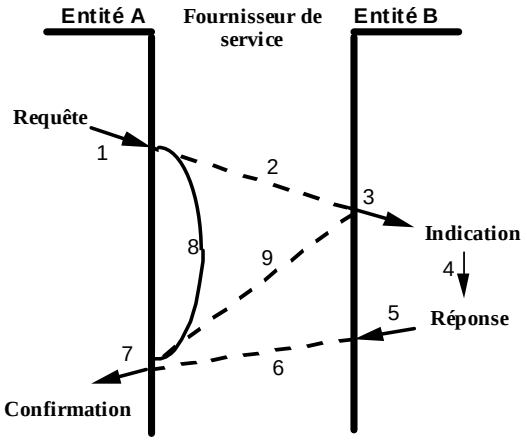
b) **1-2-9-7** *: service confirmé par le site distant*

c) **1-8-7** *: service localement confirmé*

d) **1-2-3** *: service non confirmé*

e) **3 :** *: invocation locale par le fournisseur de service*

f) **1 :** *: requête sans suite*



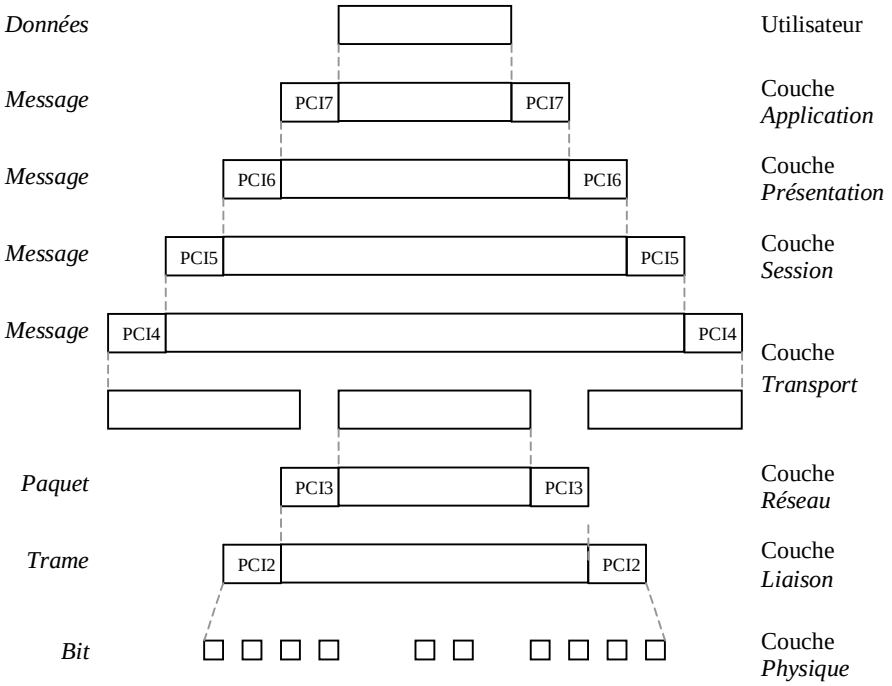
Principe d'enchaînement des primitives de service (b)

Exemples d'enchaînements de primitives

* `OpenFile.request -> OpenFile.indication -> OpenFile.response -> OpenFile.confirm`
* `Abort.request -> Abort.indication`
* `Network-failure.indication`

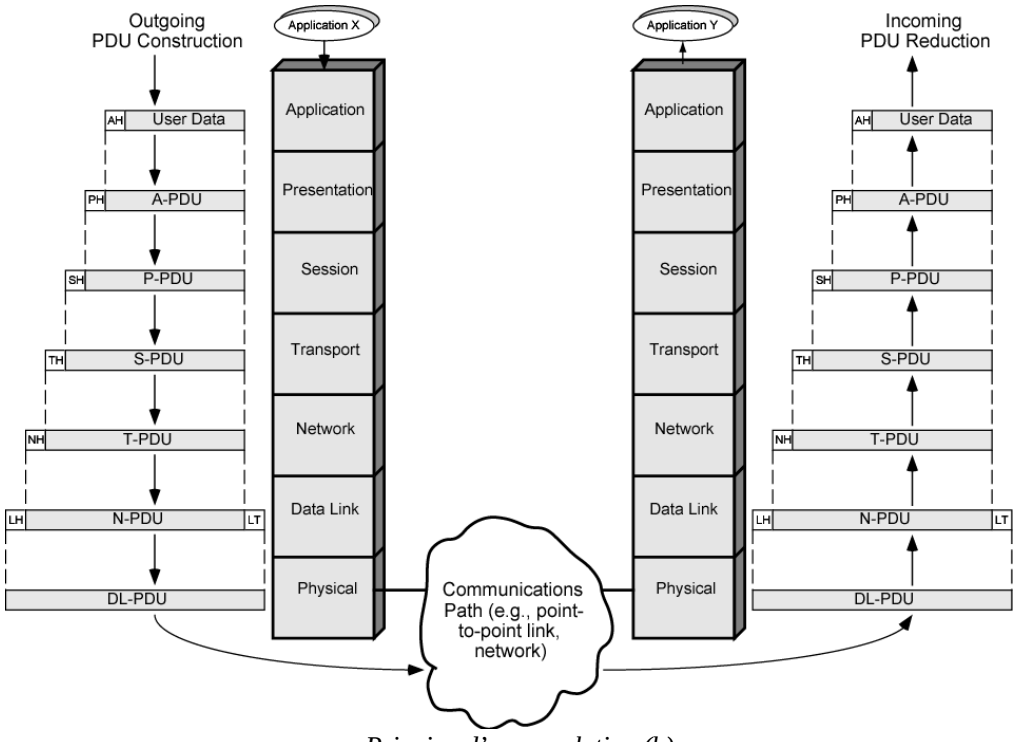
V. Principe d'encapsulation de données

Les deux figures suivantes montrent comment on applique le principe d'encapsulation pour envoyer des données d'un utilisateur à un autre.
Chaque couche a ses propres marqueurs (PCI : Protocol Control Information) pour identifier le rôle de chaque PDU



PCI : Protocol Control Information

Principe d'encapsulation (a)

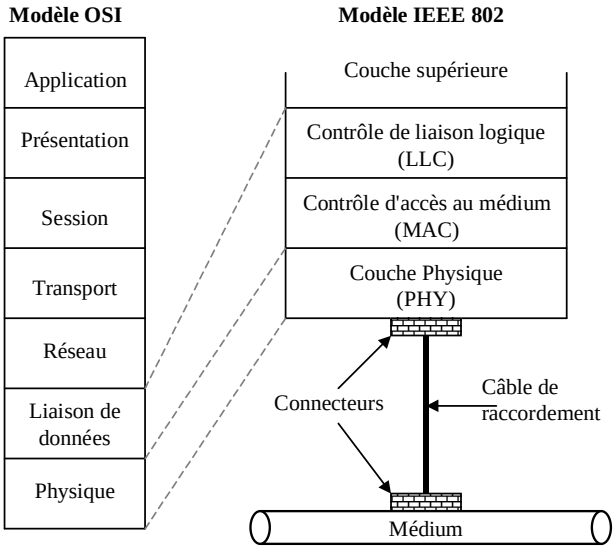


Principe d'encapsulation (b)

VI. Inconvénients du modèle OSI

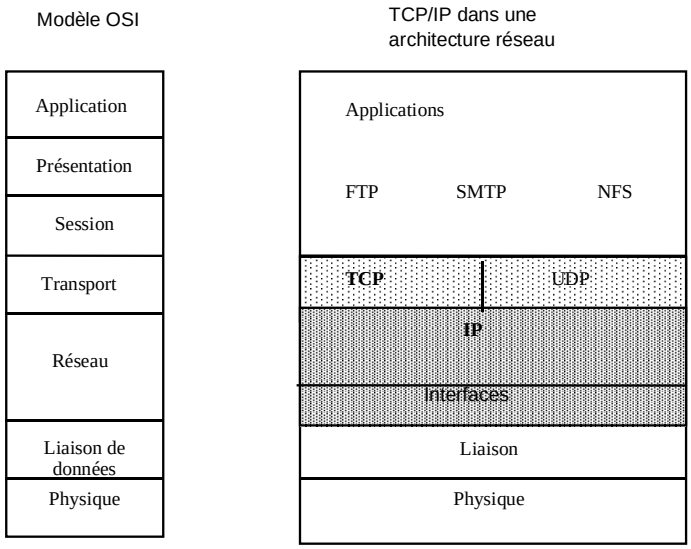
1. Ce n'est pas le bon moment : les normes apparaissent soit trop tôt soit trop tard par rapport aux besoins et aux produits.
2. Ce n'est pas la bonne technologie : l'ISO impose des concepts et mécanismes à un certain niveau et exclut ensuite toute nouvelle idée (par exemple l'occultation du mode orienté connexion)
3. Pas la bonne implantation : la complexité des modèles conduit souvent à des implantations lourdes et complexes. Les utilisateurs gardent toujours une image négative des produits lourds.
4. Pas la bonne politique : l'ISO est vue le plus souvent comme une administration.

VII. Modèle IEEE 802 (pour les réseaux locaux)

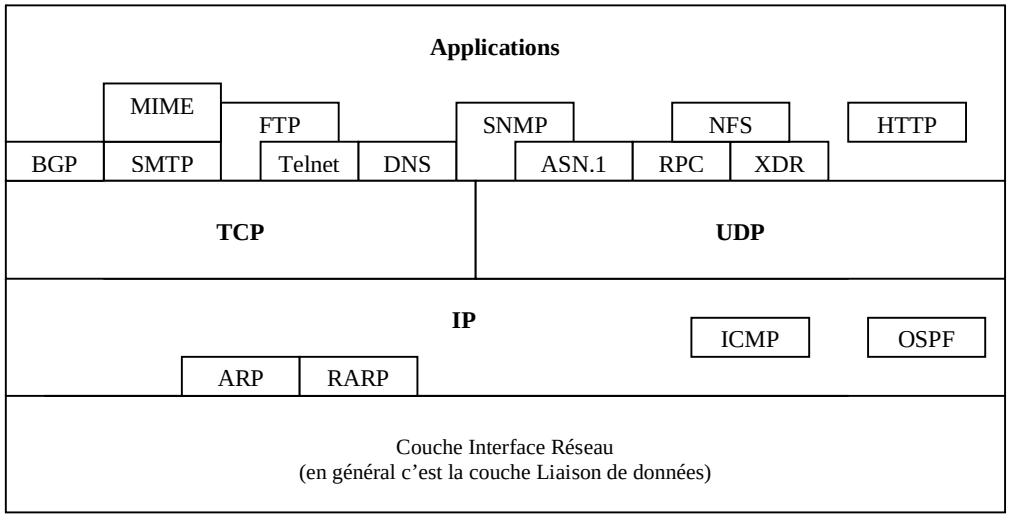


Modèle IEEE

VIII. Famille TCP/IP



Architecture générale de TCP/IP



ARP : Address Resolution Protocol
BGP: Border Gateway Protocol
FTP : File Transfer Protocol
IP : Internet Protocol
NFS : Network File System
RPC : Remote Procedure Call
SNMP : Simple Network Management Protocol
Telnet : Terminal NETwork
XDR : eXternal Data Representation
ASN.1 : Abstract Syntax Notation 1
DNS : Domain Name System
ICMP : Internet Control Message Protocol
MIME : Multi-purpose Internet Mail Extension
RARP : Reverse Address Resolution Protocol
SMTP : Simple Mail Transfer Protocol
TCP : Transmission Control Protocol
UDP : User Datagram Protocol

Principaux protocoles de la suite TCP/IP

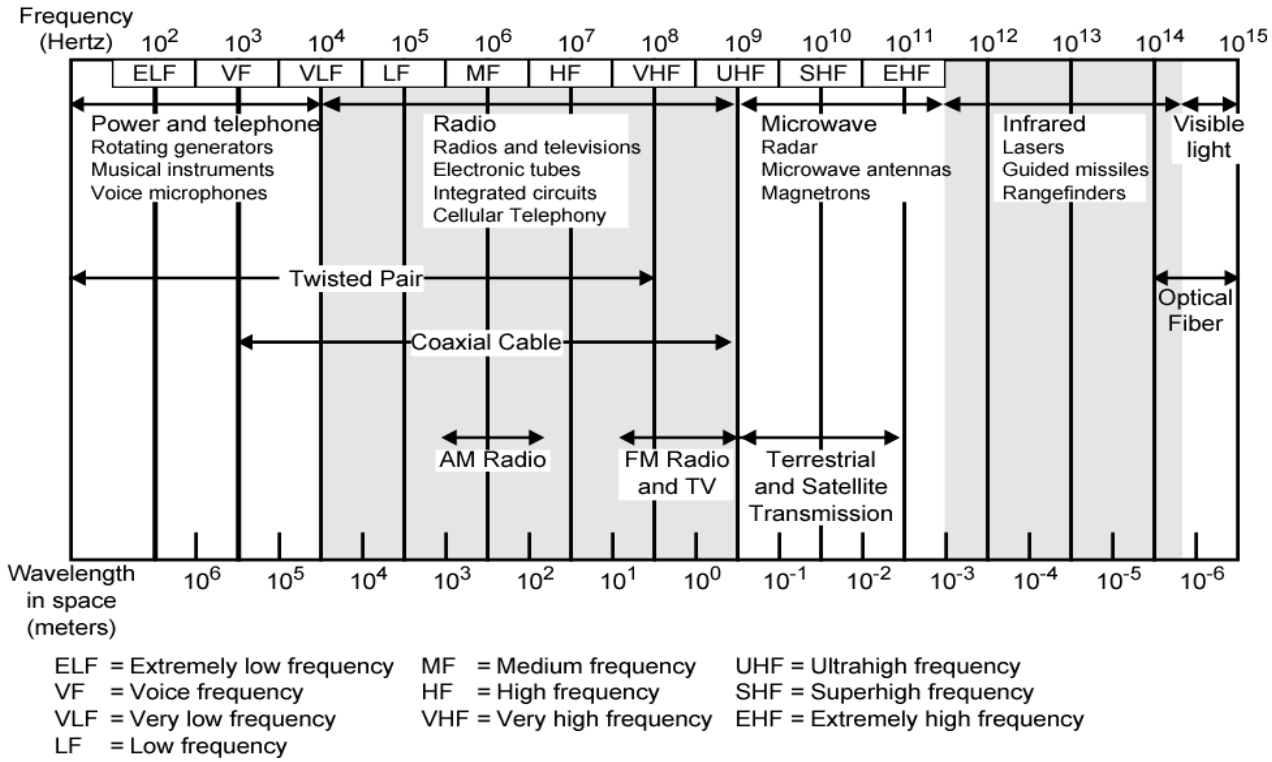
Exercice 1
Donner des cas de systèmes organisés en couches en utilisant des exemples de la vie courante (la circulation de document dans une entreprise, le commerce international, la poste, etc.).

Chapitre 3
Couche physique

I. Concepts liés aux signaux électromagnétiques

I.1 Spectre électromagnétique

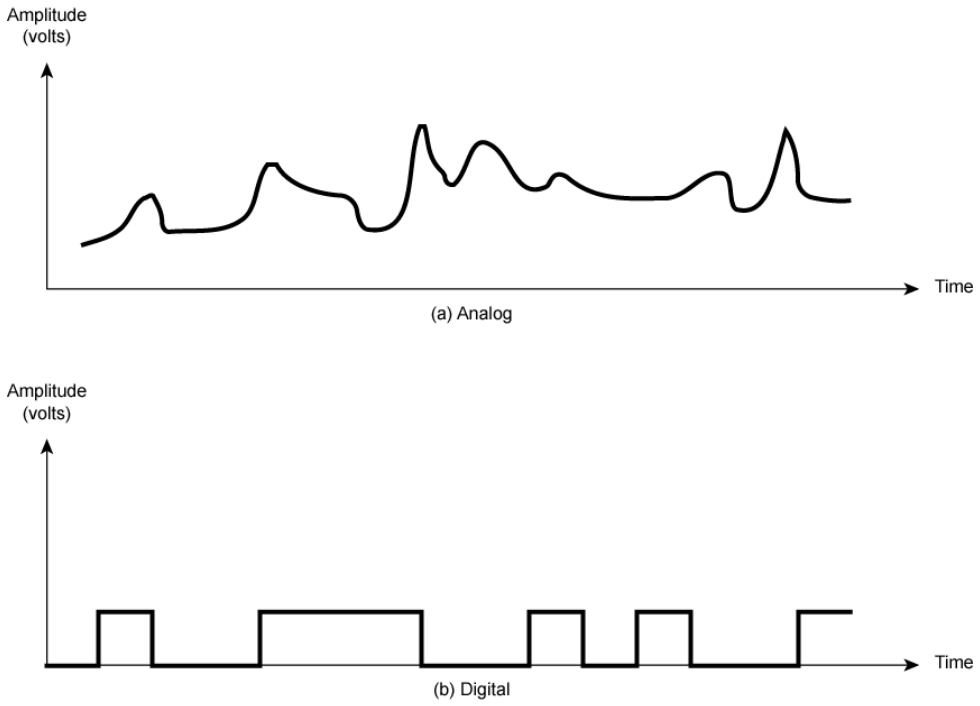
La transmission d'informations (analogique ou numériques) se fait en utilisant le spectre électromagnétique. Actuellement, le spectre des fréquences est géré comme suit (sachant qu'il y a des règles d'utilisation des fréquences selon les pays pour tenir compte en particulier de besoins militaires) :



Spectre électromagnétique

I.2 Echantillonnage et modulation par impulsions codées

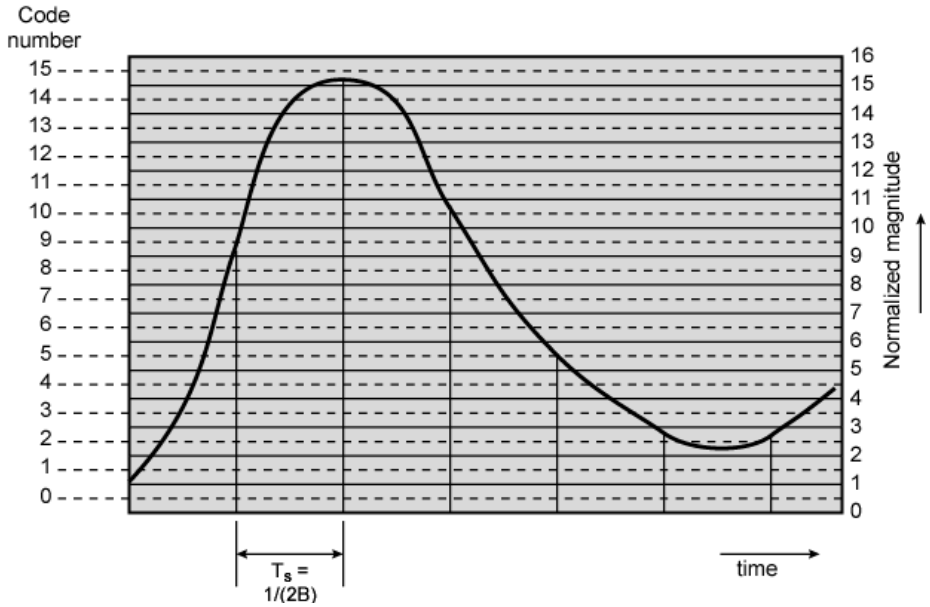
Les signaux transmis sur un canal peuvent être digitaux (binaires) ou analogiques (continus). Les domaines de la téléphonie et de la télévision étaient jadis dominés par des liaisons analogiques. De nos jours, la plupart de transmission d'informations se font en digital (numérique).



Forme des signaux analogiques et digitaux

Il faut noter qu'à l'origine, beaucoup de signaux (tels que la voix, le signal de température...) sont de nature analogique et il faut les échantillonner pour les transformer en numérique. De manière périodique, un signal analogique est échantillonné, ensuite une valeur numérique (sur *n* bits, dans l'exemple de la figure ci-dessous, $n = 4$) est retenue pour représenter le signal analogique.

L'échantillonnage d'un signal analogique s'effectue à une fréquence double de la plus élevée des fréquences du signal analogique (sur la figure ci-dessous, *B* désigne la fréquence la plus élevée du signal analogique). Cette règle est dite Règle d'échantillonnage de Shannon.



Principe de l'échantillonnage de signal analogique

Cas de la voix (ligne téléphonique)

La voix (humaine) utilise une bande de 100 Hz à 7 kHz. L'oreille humaine est sensible aux signaux ayant des fréquences allant de 20 Hz à 20 kHz.

Une ligne téléphonique normale a une bande passante de 300 Hz à 3400 Hz. Cela signifie que certaines fréquences (dites harmoniques) de la voix ne sont pas véhiculées par la ligne téléphonique normale. On dit dans ce cas que la ligne téléphonique constitue un filtre qui élimine certaines fréquences.

La voix véhiculée par une ligne téléphonique doit donc être échantillonnées au double de la fréquence la plus élevée (c'est à dire 6800 Hz). Pour simplifier les composants, on échantillonne à 8 kHz. Ensuite chaque échantillon est représenté sur m bits (m = 8 pour l'Europe et 7 pour les USA). On notera qu'avec m=8, on a 256 niveaux de signaux différents, ce qui permet théoriquement d'avoir une meilleure qualité du son en Europe qu'aux USA. En pratique, l'oreille humaine est insensible (pour beaucoup de personnes) aux deux types d'échantillonnage. Ainsi, une ligne téléphonique correspond à un débit défini comme suit :

```
8 bits * 8 kHz = 64 kb/s  /* Pour l'Europe */
7 bits * 8 kHz = 56 kb/s  /* Pour les USA */
```

C'est de cette manière qu'on explique les 64 kb/s pour les modems gérés par le réseau téléphonique commuté (RTC).

I.3. Eléments de la théorie du signal

Théorème de Shannon (1948)

$C = H \times \log_2(1 + S/N)$

C : Capacité maximale d'un canal (en bits par seconde)

H : Bande passante (en Hertz)

S/N : rapport signal sur bruit (en décibels)

Le rapport ***S/N*** est exprimé par ***$10\log_{10}(S/N)$*** décibels.

Rapidité de modulation = nombre (n) d'éléments binaires transmis par seconde

La rapidité de modulation s'exprime en Bauds. Si n = 1, on peut confondre Baud et bit/s.

II. Supports de transmission

Les supports de transmission peuvent être filaires ou non filaires. Les supports filaires englobent la paire métallique, le câble coaxial et la fibre optique.

L'installation de supports filaires et leur maintenance coûte cher (creuser des tronchées en zone urbaine ou montagneuse, réparer les centaines de câbles cassés chaque année à cause chantiers ou par des chalutiers par exemple...). Après leur installation, la maintenance des fils coûte cher.

L'installation des antennes et des satellites coûte cher aussi. Les désagréments des chantiers du filaire et du non filaires ne sont pas les mêmes. Les deux types de supports sont parfois complémentaires, parfois concurrents. Le débat 'câble ou sans fil' n'est pas encore tranché aujourd'hui.

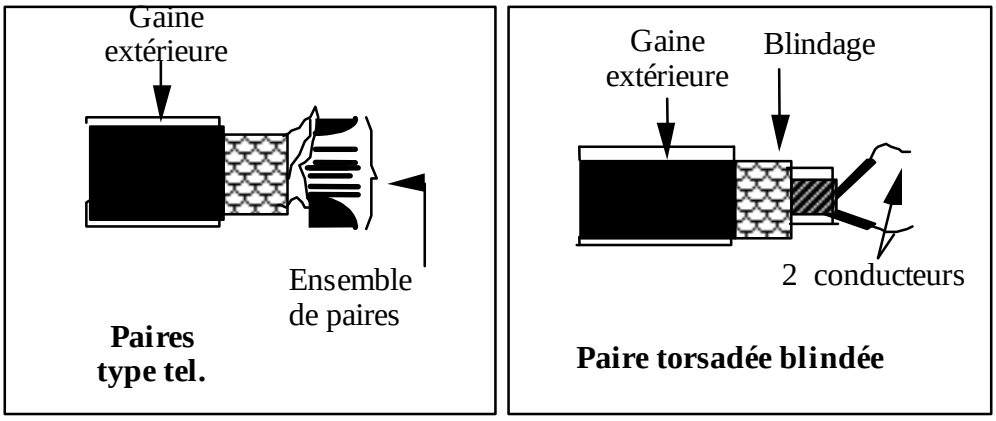
II.1 Supports filaires

1. Paire torsadée

On distingue :

- Les paires torsadées non blindées (Unshielded Twisted Pairs) : les moins chères et les plus faciles à installer, mais les plus sensibles aux bruits. Il existe plusieurs catégorie de paires non torsadées selon la bande de fréquences : Catégorie 3 (jusqu'à 16 MHz), Catégorie 4 (jusqu'à 20 MHz), Catégorie 5 (jusqu'à 100 MHz), Catégories 5E, 6 et 7.

- Les paires torsadées blindées (Shielded Twisted Pairs) : un peu plus complexes que les premières, mais plus résistantes aux bruits.



Paire métallique

a) Avantages :
- C'est le support le plus commun
- Le réseau téléphonique filaire est à base de paire torsadée (essentiellement pour le raccordement des abonnés)
- Très utilisé dans les réseaux locaux pour avoir des débits allant de 10 Mb/s à 100 Mb/s.
- Peu cher et facile à installer

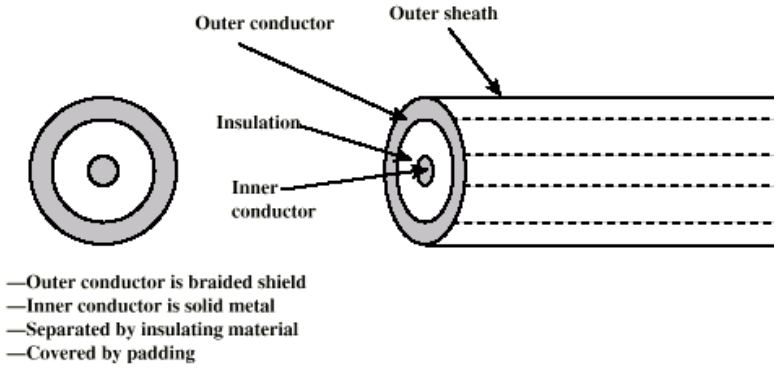
- Inconvénients :
  + Débits et bande passante faibles comparés aux autres supports
  + Sensible aux bruits.
  + Distances limitées.

2. Câble coaxial

Parmi les caractéristiques du câble coaxial, on peut citer :

- C'est le plus versatile des supports.
- Il offre une bande passante pouvant dépasser les 500 MHz.
- Il est utilisé surtout pour la distribution de télévision.

- Un câble coaxial peut transporter jusqu'à 10000 appels téléphoniques simultanés.
- Actuellement, le câble coaxial est de plus en plus remplacé par la fibre optique.
- Il est utilisé aussi pour faire des réseaux locaux.



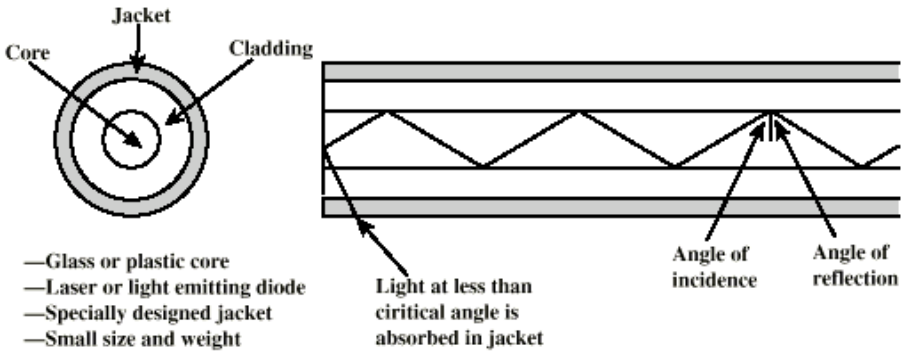
Câble coaxial

3. Fibre optique

Une fibre optique agit comme un guide d'ondes lumineuses ayant des fréquences de 10^{14} à 10^{15} Hz

Il existe deux types de fibres optiques :

- Les "Light Emitting Diode" (LED) : les moins chères, plus résistantes et qui durent longtemps
- Les "Injection Laser Diode" (ILD) : offrent les meilleurs débits, plus chères.



Fibre optique

Parmi les caractéristiques de la fibre optique, on peut citer :

- Très hauts débits pouvant atteindre plusieurs centaines de G b/s ;
- Peu encombrantes en taille et poids ;
- Résistantes aux bruits (très bonne isolation) ;
- Les distances entre répéteurs peuvent atteindre plusieurs dizaines de Km ;
- La fibre optique est surtout utilisée pour les réseaux métropolitains et l'interconnexion de réseaux locaux.

4. Résumé des caractéristiques des supports filaires :

Support	Débit (M b/s)	Résistance aux bruits	Délai de propagation	Distance entre répéteurs	Remarques
Paire tél.	≤100	Faible	50 µs/km	2 km	Très répandue. Connexion facile
Paire torsadée blindée	100	Bonne	5 µs/km	2 km	Connexion facile
Câble coaxial	100 ou plus	Très bonne	4 µs/km	1 à 9 km	Répandue. Connexion facile
Fibre optique	des dizaines de milliers	Excellente	5 µs/km	40 km	Répandue. Connexion difficile

Récapitulatif des caractéristiques de supports filaires

II.2 Supports non filaires (monde du Wireless)

1. Principes

Il y a quelques années, on parlait très peu des supports non filaires pour l'interconnexion d'équipements informatiques. Les réseaux sans fil étaient réservés à quelques initiés et/ou à quelques applications (on trouvait les réseaux sans fil pour la communication d'urgence, pour la gestion d'entrepôts, gestion de flottes de véhicules...). De nos jours, le mode du sans fil a tout envahi et probablement la majorité des réseaux de demain seront tous non filaires.

Le non filaire permet de s'affranchir du cordon que constitue le câble de raccordement. Sans fil, l'utilisateur est plus mobile et plus libre dans ses mouvements tout en restant connecté.

Actuellement, les bandes de fréquence en wireless sont utilisées comme suit :

- 2 GHz à 40 GHz : bandes utilisées par les micro ondes et les liaisons point à point et les satellites.
- 30 MHz à 1 GHz : bandes utilisés pour la radio et liaisons unidirectionnelles
- 3×10^{11} Hz à 2×10^{14} Hz : bandes utilisées par l'infrarouge et les communications locales

Il y a deux manières d'implanter le non filaire : l'infrarouge et les ondes radio. Les moyens physiques de transmission et réception englobent les antennes terrestres et les satellites avec leurs antennes associées.

Exemple de Hub sans fil

2. Antennes

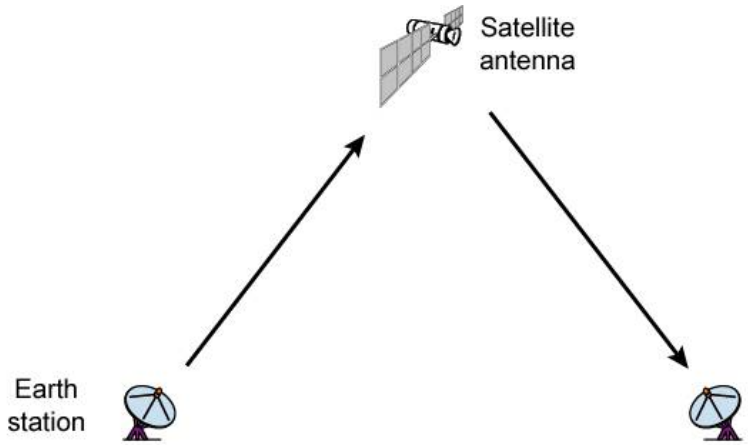
Les communications sans fil utilisent des antennes paraboliques ou non pour la transmission et réception de signaux.

Un émetteur peut diffuser soit dans toutes les directions (cas des satellites et des relais de radio) ou vers un récepteur unique localisé dans une zone bien délimitée.

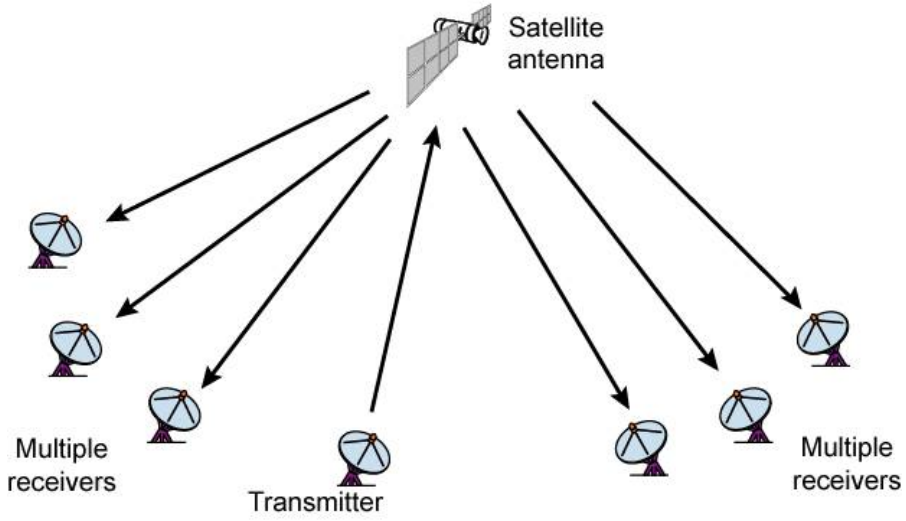
Selon les systèmes de télécommunications, pour recevoir un signal, le récepteur doit se trouver dans le champ de vision de l'émetteur ou non. Dans le premier cas les communications se font en ligne directe dans l'autre les signaux se propagent sur différents chemins avant d'atteindre le récepteur.

3. Communications par satellite

Les satellites constituent des stations relais de signaux. Un satellite reçoit un signal sur une fréquence, amplifie ce signal et le retransmet sur une autre fréquence. Les satellites utilisés sont souvent géostationnaires (à 35784 Km de la terre). Les satellites sont utilisés surtout pour la télévision et la téléphonie, mais aussi de plus en plus pour les communications de données (applications Internet ou autres).



Liaison satellite point-à-point



Liaison satellite en diffusion

4. Communications par ondes radio terrestres

Ces communications sont :

- Unidirectionnelles ;
- Utilisées généralement pour la radio et la télévision mais aussi pour la transmission de données ;
- Sujettes aux interférences causées par le relief.
- Les ondes se propagent malgré les obstacles.

5. Communications par ondes infrarouges

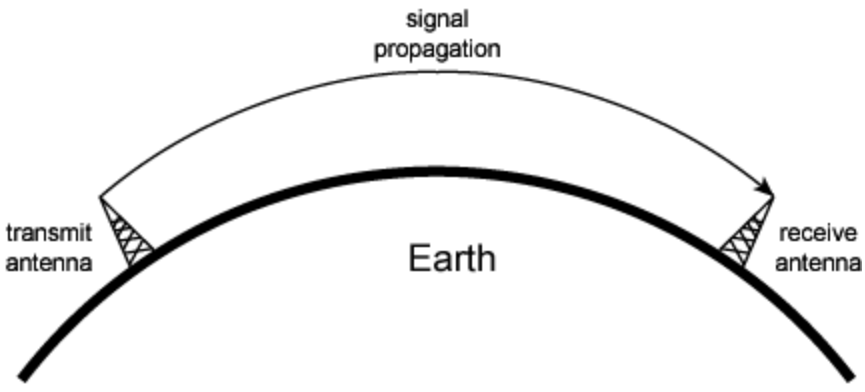
Ces communications sont caractérisées par ce qui suit :

- Elles sont utilisées pour la télécommande et par certains réseaux LAN.
- Emetteur et récepteur doivent être visibles l'un de l'autre.
- Les ondes sont bloquées par les obstacles.
- Les ondes se propagent sur de courtes distances.

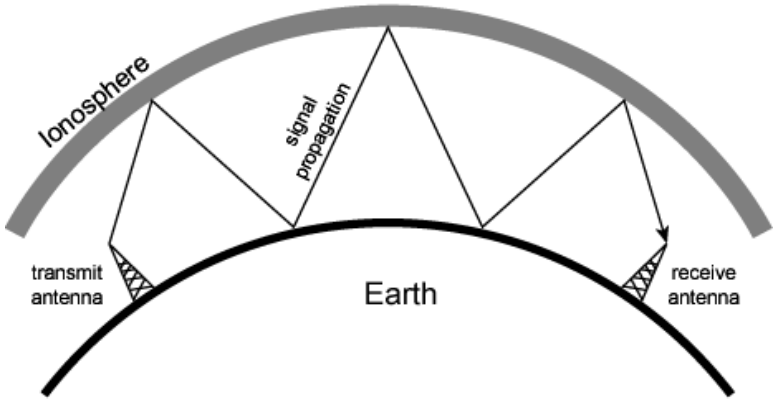
6. Modes de propagation

Le mode de propagation des ondes est lié aux débits fournis et à la puissance nécessaire à transmission. Les ondes se propagent selon les trois modes suivants :

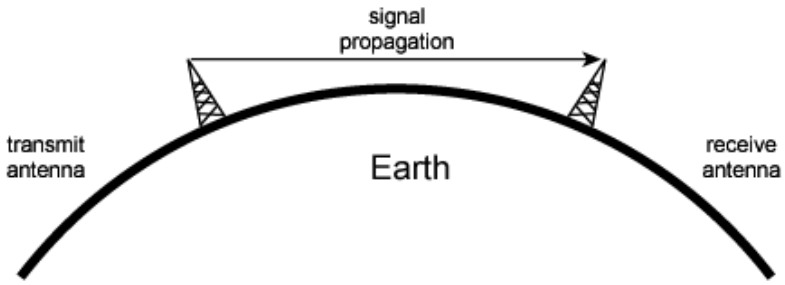
- propagation en surface ('ground-wave propagation) utilisant la troposphère (0-10 Km) ;
- propagation au niveau de l'ionosphère (60-800 Km) ;
- propagation en ligne directe.



Propagation en surface



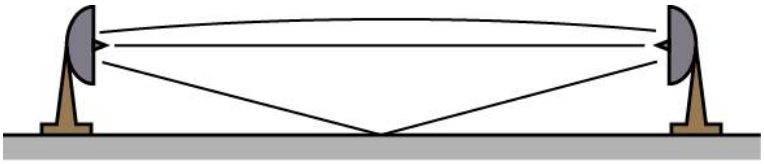
Propagation via ionosphère



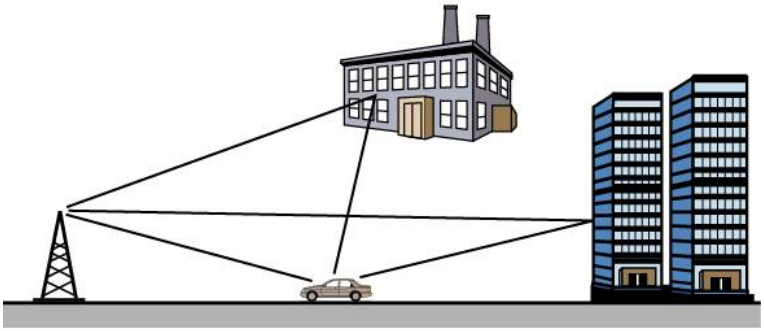
Propagation en ligne directe

7. Réflexion de signaux

Un signal se propage en suivant éventuellement plusieurs chemins. Lorsqu'un signal est réfléchi par les obstacles qui se trouvent entre l'émetteur et le récepteur, le récepteur peut recevoir plusieurs copies du même signal (ces copies sont légèrement déphasées les unes par rapport aux autres). Les obstacles peuvent aussi conduire à la perte du signal. Les deux figures ci-dessous donnent des exemples de signaux réfléchis :



(a) Microwave line of sight

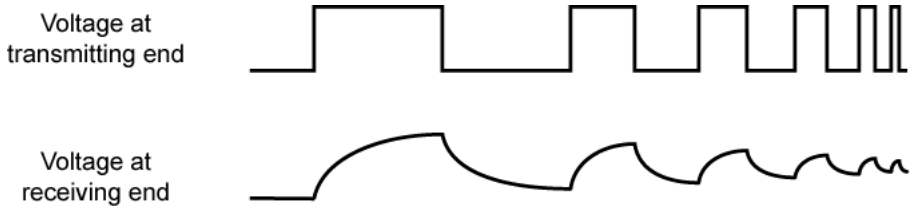


(b) Mobile radio

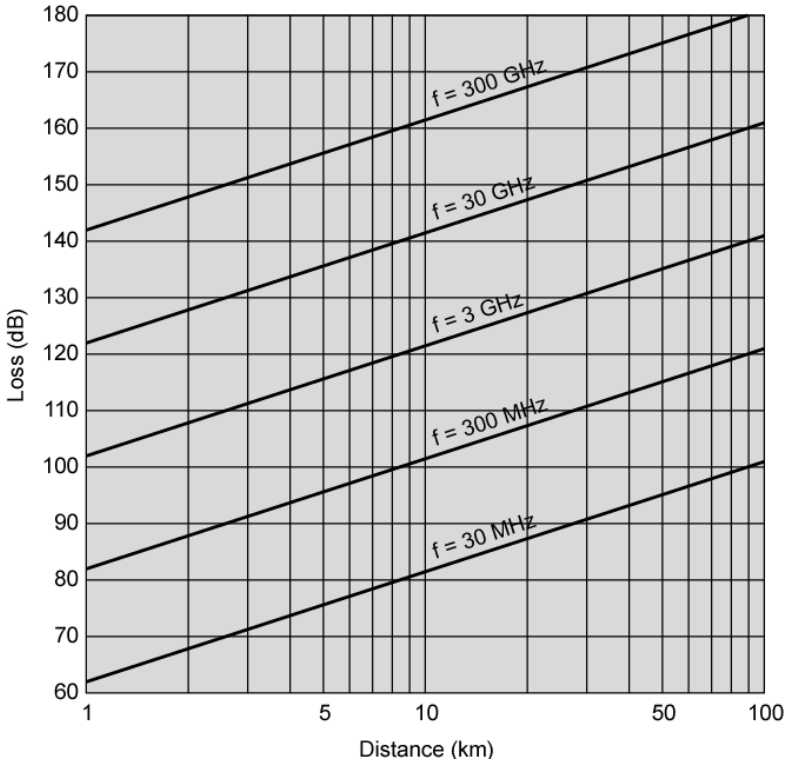
Réflexion de signaux

8. Atténuation (affaiblissement) de signaux

Les signaux électromagnétiques s'affaiblissent avec les distances. Ils sont absorbés par les particules (eau, oxygène...) qu'ils traversent. Par conséquent, les moyens à mettre en oeuvre pour transmettre sur de longues distances avec des débits élevés doivent être appropriés. Plus on veut transmettre loin, plus les équipements sont chers. La figure suivante donne une idée sur la perte (et bruit) en fonction des distances et fréquences pour les hautes fréquences utilisées en propagation directe.

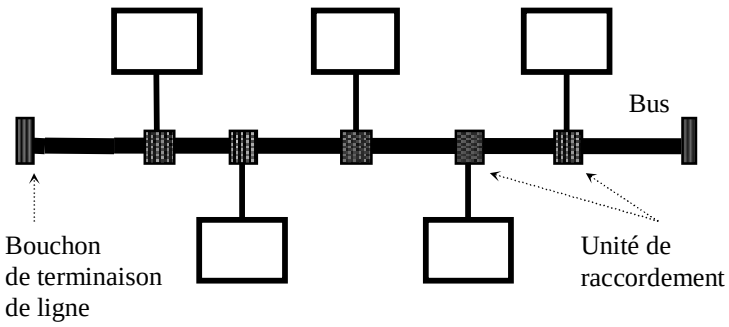


Atténuation de signaux digitaux

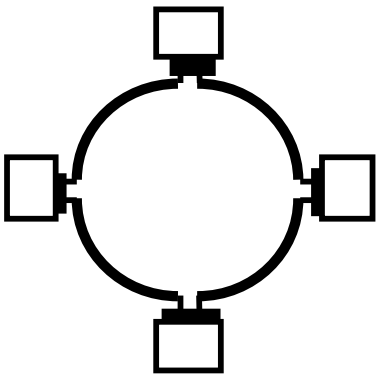


Affaiblissement des signaux en fonction des distances

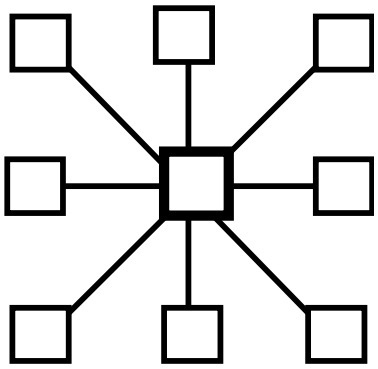
III. Topologies des réseaux



Topologie en bus

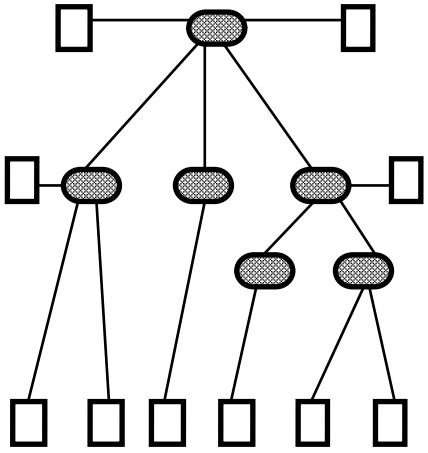


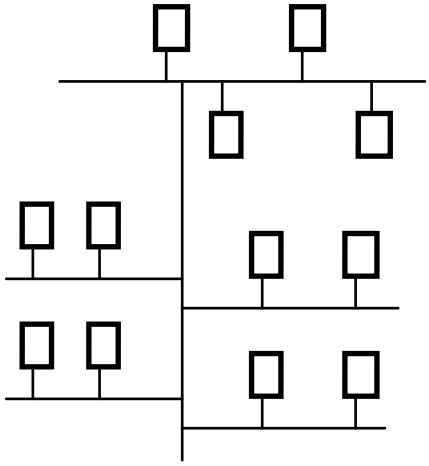
Boucle



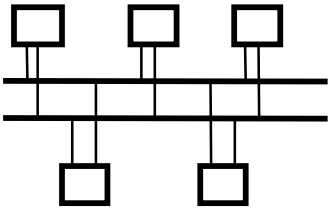
Etoile

Topologie en boucle et étoile

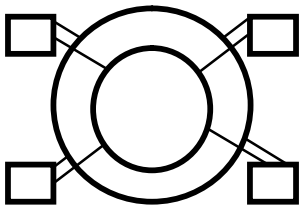




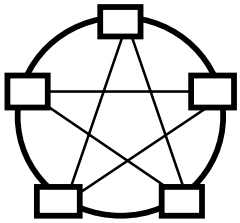
Structure arborescente



Bus doublé

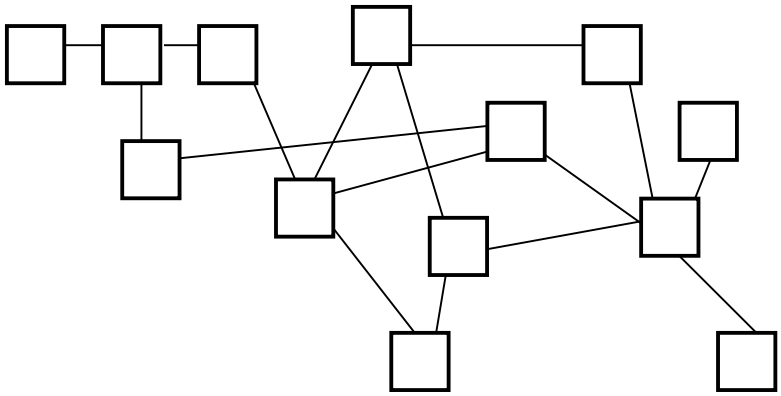


Boucle doublée



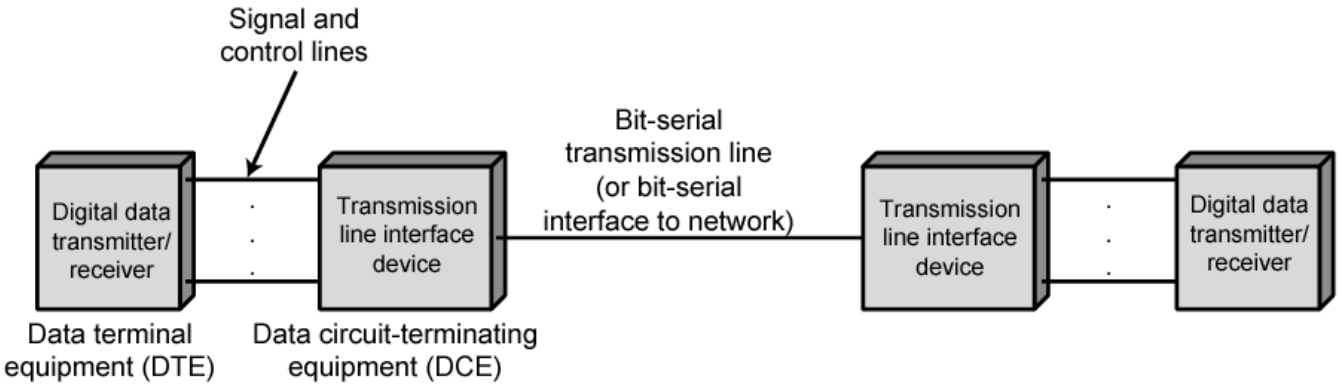
Architecture en chaîne

Topologies redondantes



Topologie maillée

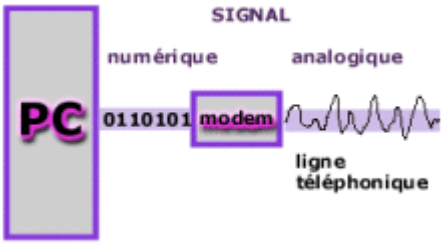
IV. Modèle de transmission



(a) Generic interface to transmission medium

(b) Typical configuration

Chaîne de transmission (a)



Chaîne de transmission (b)

DCE (Data Communication Equipment) externe dit modem

V. Codage des informations

Les bits envoyés vers le modem (ou en direct sur la ligne s'il n'y a pas de modem) sont codés selon un des codes suivants :

- NRZ (Non Return to Zero)
- NRZI (Non Return to Zero Inverted)
- Manchester (ou Manchester 1)
- Manchester différentiel (ou Manchester 2)
- Bipolar –AMI,
- B8ZS, HDB3
- Autres

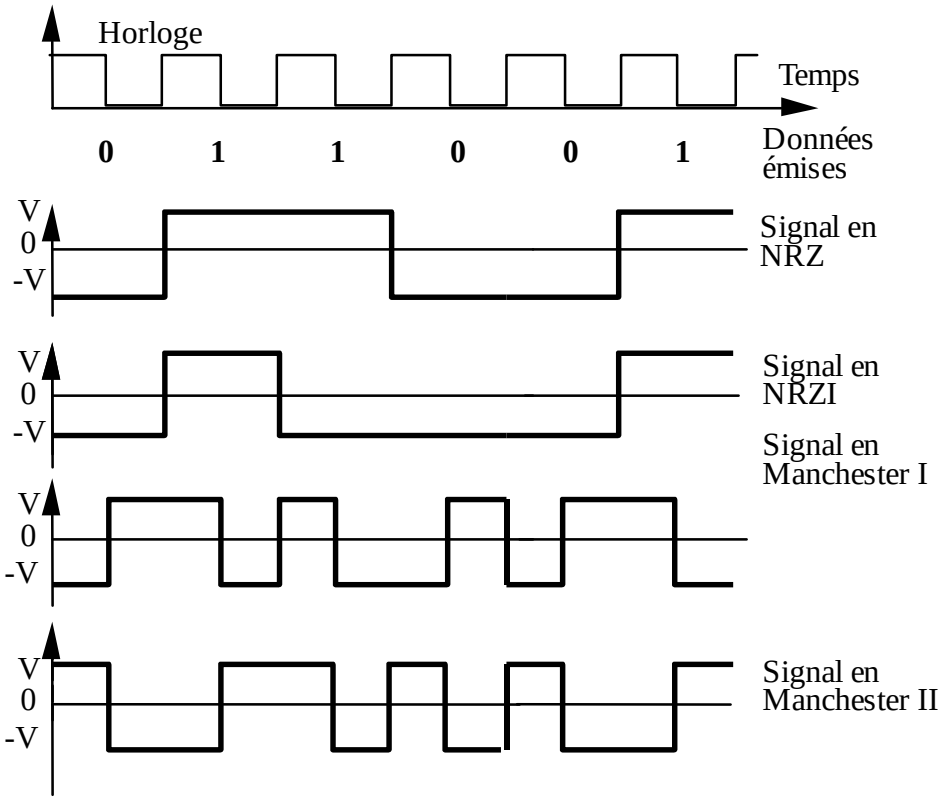
Nous détaillons seulement les codes les plus employés dans les réseaux, à savoir : NRZ, NRZI, Manchester 1 et 2.

- *Code NRZ (Non Return to Zero) :* un bit *0* (respectivement *1*) est représenté par un signal de tension -V (respectivement +V) pendant une durée égale à *1/f*.
- *Code NRZI (Non Return to Zero Inverted) :* a été introduit pour pallier certaines faiblesses du code NRZ. Ainsi, pour éviter les successions de 0, le signal change d'état pour coder un *0* et reste dans le même état pour coder un *1*.
- *Code Manchester* (appelé aussi Manchester I) : chaque bit b_i est transmis sous formes de deux états, a_i' et a_i'', qui dure chacun *1/2f*. Il y a toujours une transition par intervalle *1/f*.

  Si $b_i = 0$, alors $a_i' = -V$ et $a_i'' = +V$. Si $b_i = 1$, alors $a_i' = +V$ et $a_i'' = -V$.
- *Code Manchester différentiel* (appelé aussi Manchester II) : comme pour le code Manchester I, un bit b_i est codé par deux états a_i' et a_i'', mais le codage du bit b_i tient compte du bit b_{i-1}. On obtient alors les règles de codage suivantes :

  Si $b_i = 0$ alors $a_i' = a_{i-1}'$ et $a_i'' = - a_i'$

  Si $b_i = 1$ alors $a_i' = - a_{i-1}'$ et $a_i'' = - a_i'$.



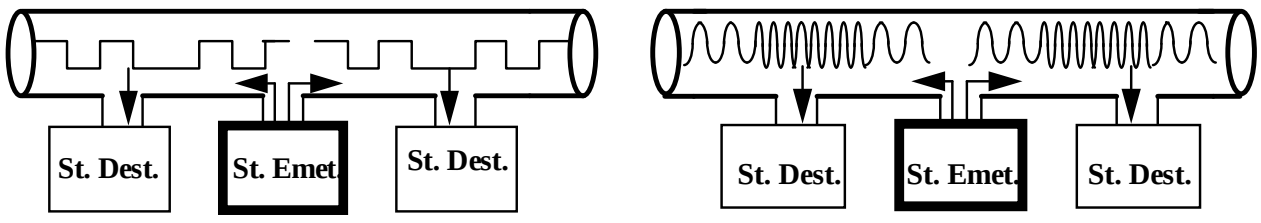
Exemple de codage de bits

Critères de comparaison des codes :

- résistance aux bruits
- bande passante requise
- énergie
- horloge de synchronisation

VI. Modes de transmission

VI.1. Deux modes de transmission



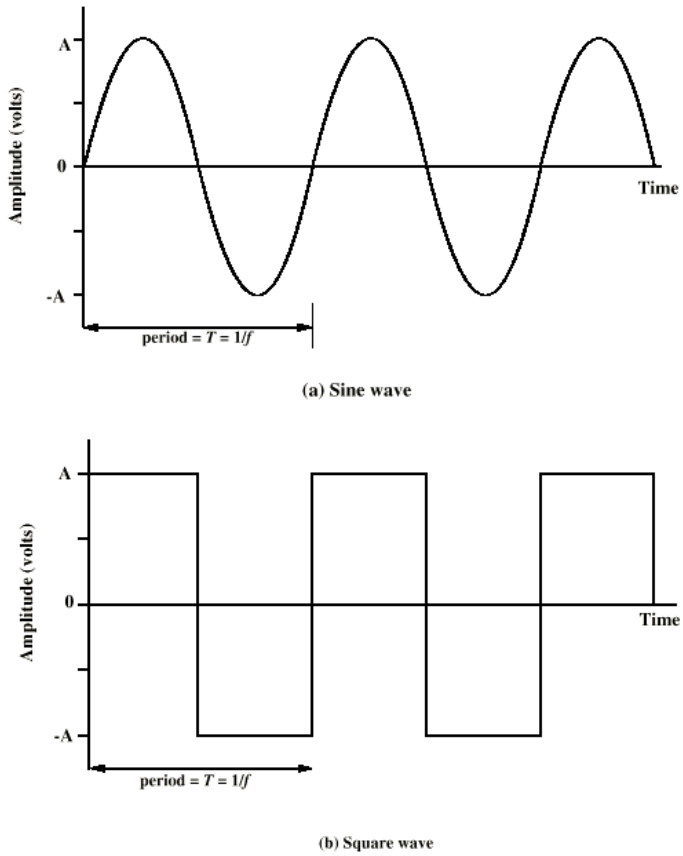
a) Transmission en bande de base

b) Transmission en bande porteuse avec modulation de fréquence

Modes de transmission de signaux digitaux

VI.2. Modulation

$S(t) = A(t)\cos(2\pi F(t)t + p(t))$
A(t) : amplitude, F(t) : fréquence, p(t) : phase



Signal sinusoïdal vs signal carré

1. Modulation de base

Il existe trois types de modulation de base

- **Modulation d'amplitude** (ASK : Amplitude Shift Keying) : On modifie l'amplitude pour transmettre des bits différents. Ce mode de modulation est souvent utilisé sur les fibres optiques. La modulation ASK est plus sensible aux atténuations de signal que les autres modulations.

- **Modulation de fréquence** (FSK : Frequency Shift Keying) : On modifie la fréquence pour transmettre des bits différents. La modulation de fréquence est utilisée fréquemment sur le câble coaxial. Elle est moins sensible aux atténuations. Il y a plusieurs formes de modulation de FSK :

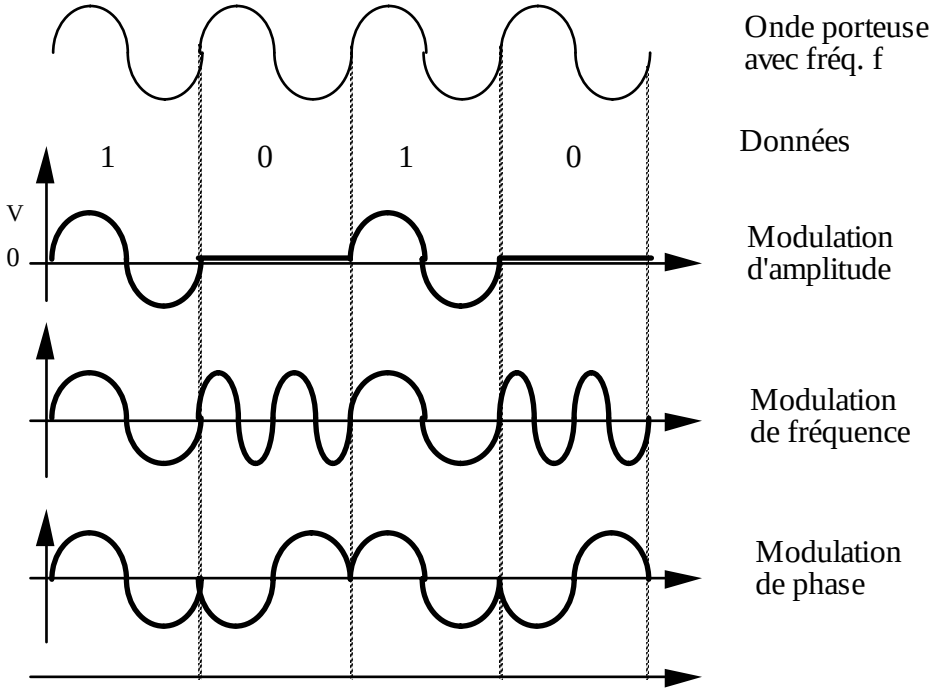
  + Binary FSK (par défaut FSK veut dire Binary FSK) : on utilise deux fréquences pour distinguer deux valeurs 0 et 1.

  + Multiple FSK : on utilise *n* fréquences différentes pour transmettre *n* bits simultanément.

  + Phase-Continuous FSK : on change la fréquence de manière graduelle en fonction des bits à transmettre.

- **Modulation de phase** (PSK : Phase Shift Keying) : On modifie la phase pour transmettre des bits différents.

Un moment élémentaire = une période de l'horloge d'émission (il peut correspondre à 1 ou plusieurs bits)



Exemple de modulation de bits(b)

Conventions de représentation relatives à l'exemple

Modulation	**Représentation des bits**	
	0	1
Amplitude	0	V
Fréquence	2f	f
Phase	π	0

Exemple de modulation de bits(b)

2. Modulation mixte

On peut combiner deux ou trois modes de modulation de base pour transmettre plusieurs bits en même temps.
Par exemple :

AM + FM pour transmettre deux bits simultanément
AM + FM +PM pour transmettre trois bits simultanément.

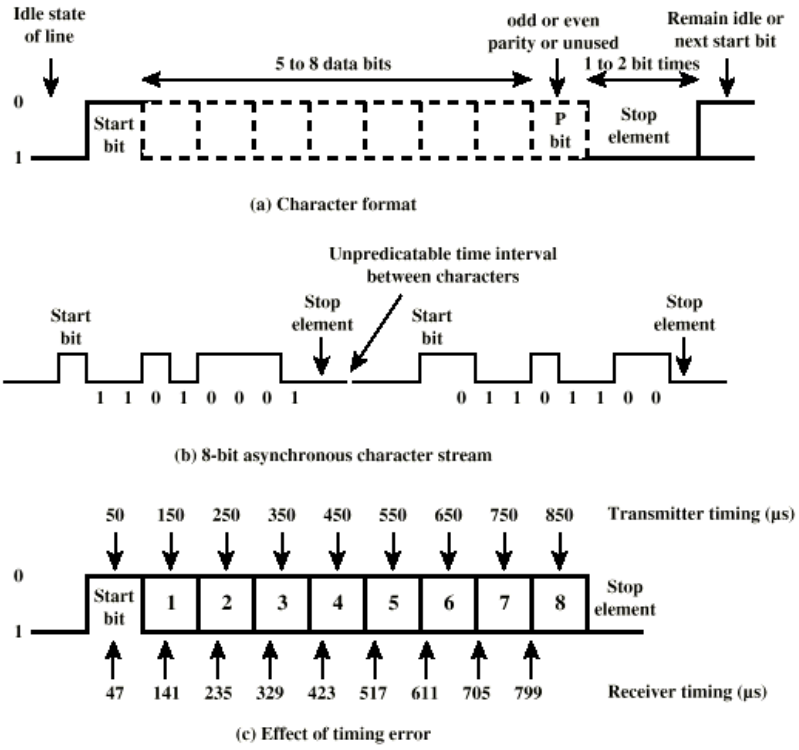
La modulation mixte est peu utilisée actuellement, car elle nécessite des composants électroniques plus complexes que ceux utilisés dans le cas de modulation simple.

VI.3. Synchronisation : transmission synchrone et asynchrone

Pour assurer la synchronisation entre émetteur et récepteur on utilise de l'information redondante avec les informations transmises. La synchronisation signifie la manière de caler le récepteur sur le signal de l'émetteur pour reconnaître les bits transmis. Il existe deux manières de transmettre : transmission synchrone et transmission asynchrone. Il faut noter qu'aujourd'hui la plupart des transmissions sont asynchrones.

1. Transmission synchrone

La transmission synchrone consiste à transmettre un nombre fixe de bits (8 en général, parfois 5) encadrés par des bits spéciaux au début et à la fin, ces bits sont dits *bit start* et *bit stop*. On parle aussi de communication par caractère. La synchronisation se fait caractère par caractère. Le récepteur reconnaît le bit start ensuite il commence à mesurer la valeur du signal pour reconnaître les bits utiles. Lorsqu'il termine de reconnaître le bit stop, il arrête de se synchroniser, jusqu'au prochain bit start. La figure suivante montre le principe de la transmission synchrone. La troisième sous-figure montre les effets d'une mauvaise synchronisation. Le bit start dure 50 µs et chaque bit de données dure 100 µs. Le récepteur se cale mal car il échantillonne tous les 94 µs.



Principe de la transmission synchrone

2. Transmission asynchrone

La transmission asynchrone consiste à transmettre à chaque fois un bloc de bits de taille variable et délimité par des octets de contrôle (qui composent le préambule et la fin de trame). Il y a deux façons d'assurer la synchronisation émetteur/récepteur :

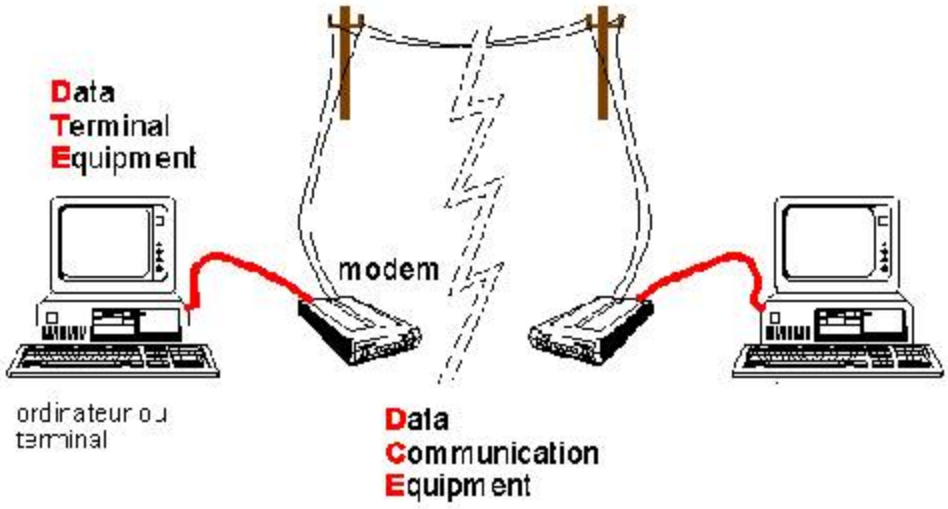
- Une ligne dédiée à l'horloge relie l'émetteur et le récepteur (cette ligne transporte uniquement les tops d'horloge de l'émetteur). Cette une solution très chère et ne marche que pour de courtes distances.

- Déduction de l'horloge de l'émetteur à partir du signal transmis. C'est le cas notamment du code Manchester où il y a un changement de niveau à l'intérieur de chaque bit et c'est ce changement qui permet au récepteur de se caler sur l'émetteur. En général, chaque trame commence par un préambule (plusieurs bits, ayant une forme fixe, qui ne transportent aucune donné), ce qui permet de laisser un peu de temps pour que le récepteur se cale correctement sur l'émetteur avant de commencer à reconnaître les bits de la trame proprement dite.

VII. Interfaces, connecteurs et modems

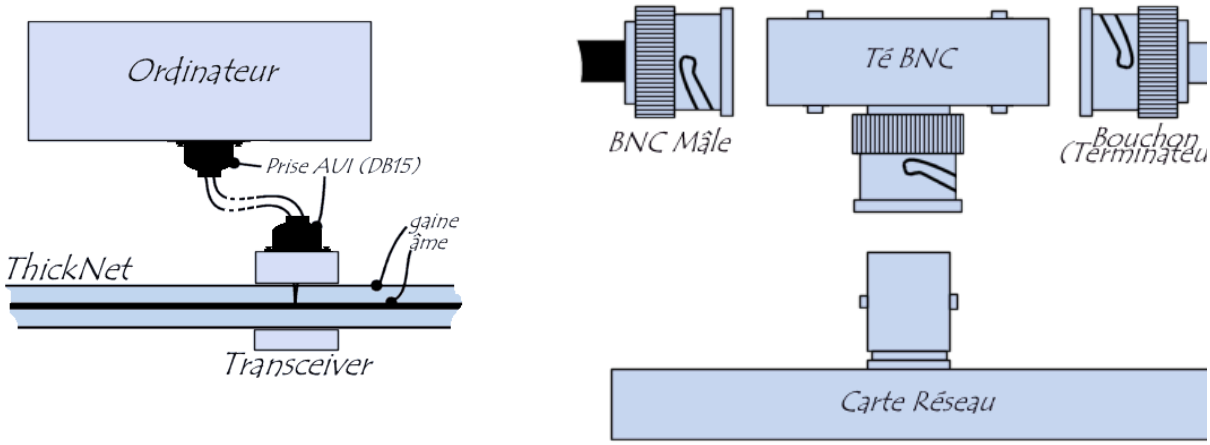
Au niveau physique, les interfaces définissent des règles de raccordement liées aux aspects suivants :

- mécaniques (forme et taille des connecteurs),
- électriques (voltage, timing des signaux...),
- fonctionnels (lignes de données, de contrôle, de masse, de courant),
- procéduraux (séquencement des événements de niveau physique) ?

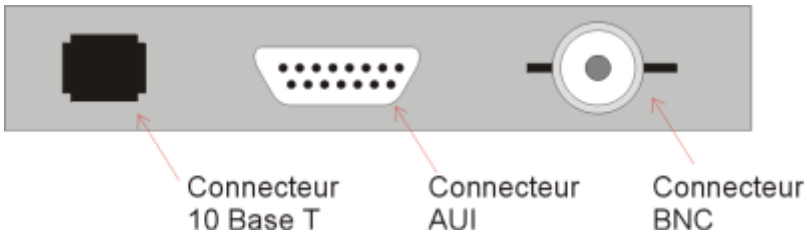


Connexion via modem.

1. Exemple de raccordement



Exemple de raccordement à un réseau



Principaux types de connecteurs pour le réseau Ethernet.

2. Jonctions

Il existe un nombre important de normes de jonctions. La plus connue s'appelle RS 232 (ou V24).

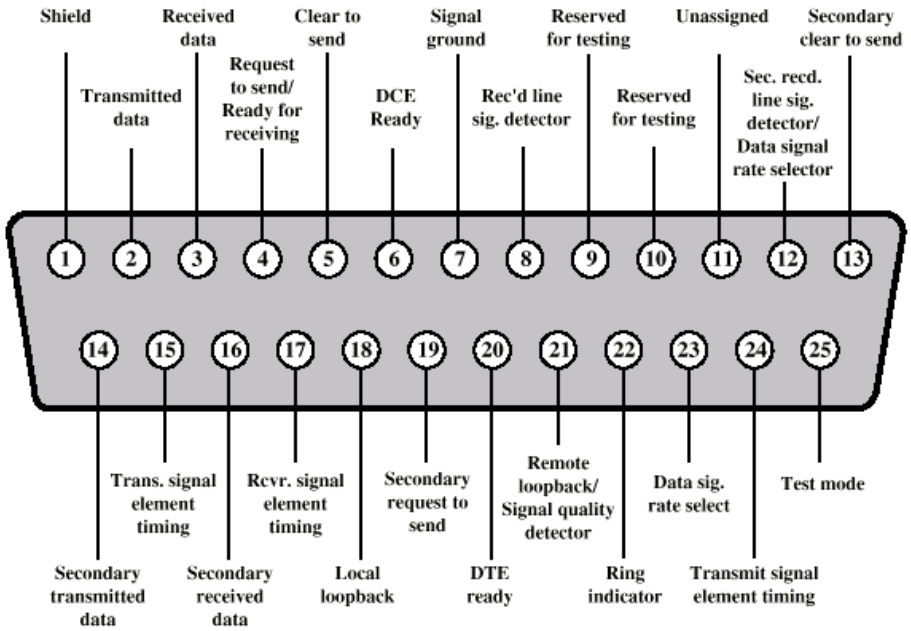
Les connecteurs V24 peuvent avoir 25 (DB25) ou 9 (DB9) broches. Les noms des broches varient selon que l'on utilise la documentation V24 (ITU-T) ou la documentation RS 232.

3. Liaison V24 (RS 232 ou EIA-232)

Connecteur DB9 - V24 (RS 232)

Connecteur DB25 - V24 (RS 232)

Signaux de la norme V.24 (a)

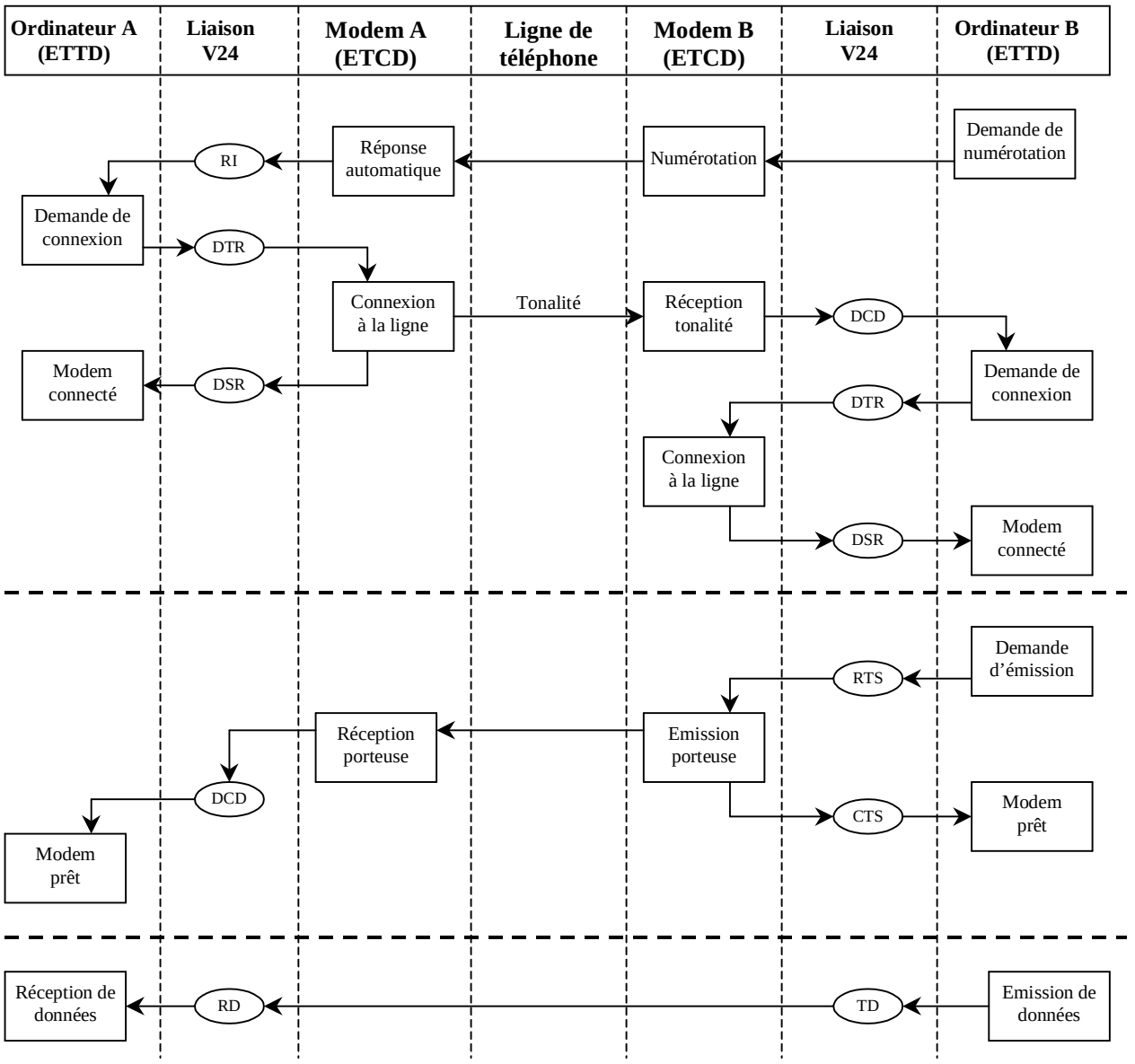


Signaux de la norme V.24 (b)

N° de broche (DB25)	N° de broche (DB9)	Description RS232	Description
1		PG : Protective Ground	Terre de protection
2	3	TD : Transmitted Data	Emission de données
3	2	RD : Received Data	Réception de données
4	7	RTS : Request To Send	Demande pour émettre
5	8	CTS : Clear To Send	Prêt à émettre
6	6	DSR : Data Set Ready	Poste de données prêt
7	5	SG : Signal Ground	Terre de signal
8	1	DCD : Data Carrier Detect	Détection porteuse
9			
10			
11			
12		SRLSD : Secondary RLSD	Demande pour émettre sur voie secondaire
13		SCTS : Secondary CTS	Prêt à émettre sur voie secondaire
14		STD : Secondary TD	Emission de données sur voie secondaire
15		TC : Transmit Clock (DCE)	Horloge d'émission du modem
16		SRD : Secondary RD	Réception de données sur voie secondaire
17		RC : Receive Clock (DCE)	Horloge de réception du modem
18		LL : Local Loopback	Bouclage local
19		SRTS : Secondary RTS	Demande pour émettre sur voie secondaire
20	4	DTR : Data Terminal Ready	Equipement terminal de données prêt
21		RM : Remote loopback	Bouclage/essai de maintenance
22	9	RI : Ring Indicator	Indicateur d'appel
23		DSRS : Data Signal Rate Selector	Sélecteur de débit binaire
24		TC : Transmit Clock (DTE)	Horloge d'émission du terminal
25		TI : Test Indicator	Indicateur d'essai

Etablissement d'une liaison V24 entre PC et Modem

L'ordinateur ou le terminal est appelé ETTD (Equipement Terminal de Traitement de Données) et le modem est appelé ETCD (Equipement de Terminaison de Circuit de Données)



Principe d'établissement de connexion physique via un modem

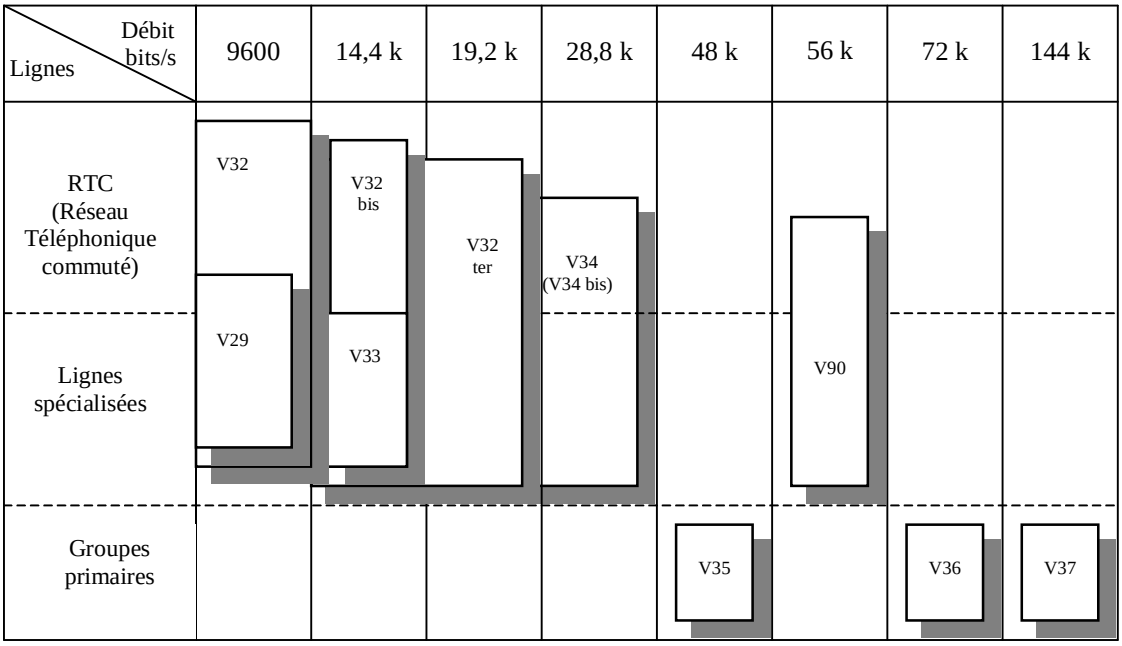
4. Connecteurs Ethernet

Carte réseau Ethernet (prise RJ45)

Connecteur RJ45

Connecteurs Ethernet

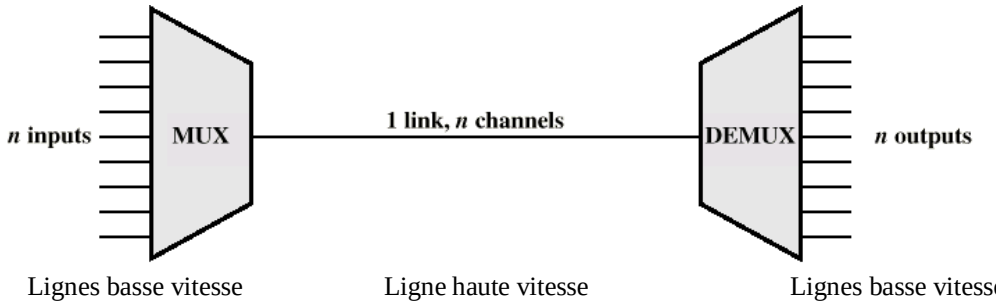
5. Normalisation des modems (par l'UIT-T)



Normes de modem

VIII. Multiplexeurs

L'objectif des multiplexeurs est de minimiser le nombre de lignes physiques.

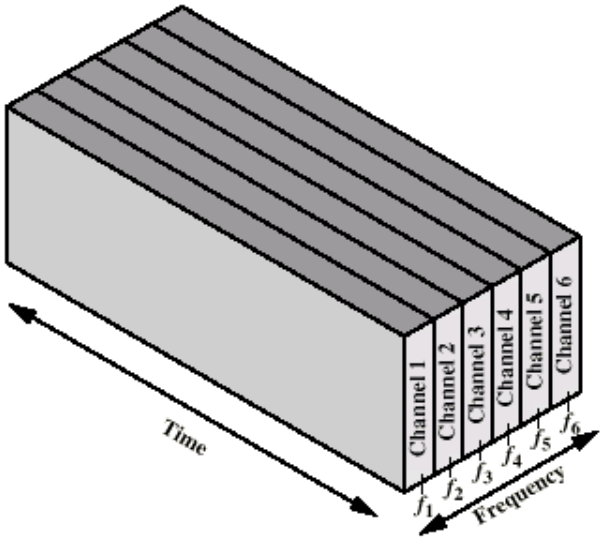


Principe général d'un multiplexeur

VIII.1. Multiplexeur fréquentiel

Une bande de fréquences (c'est-à-dire un sous-canal) est allouée à chaque station reliée par une ligne basse vitesse. Les sous-canaux sont séparés par des bandes de fréquence pour éviter les interférences entre eux.

C'est une technique largement répandue dans le domaine de la radio et télévision.

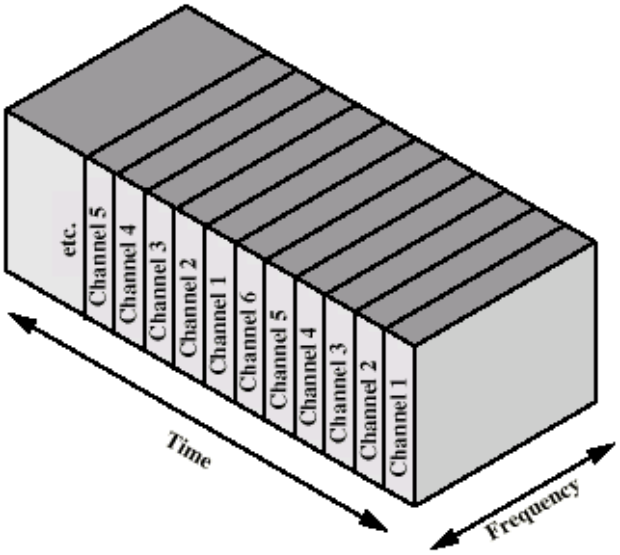


Multiplexage fréquentiel

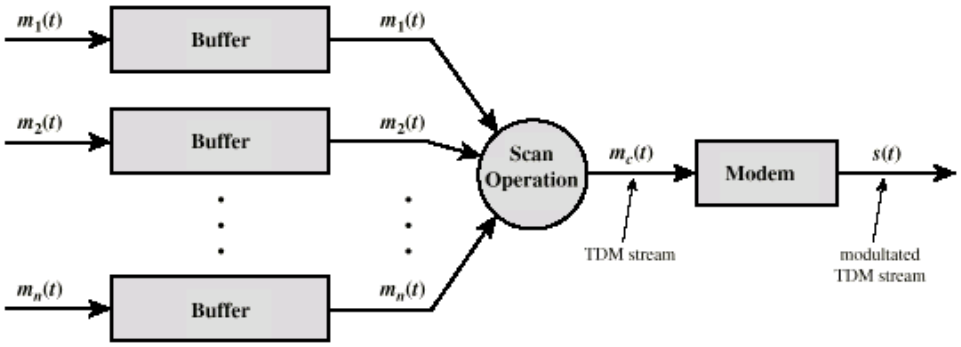
VIII.2. Multiplexeur temporel

Il existe deux types de multiplexage temporel :

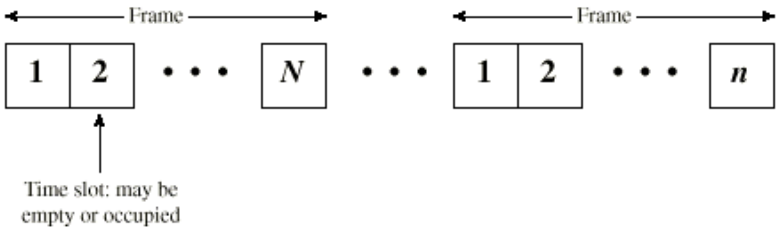
- Synchrone (une même quantité de temps est allouée à chaque ligne basse vitesse de manière périodique)
- Asynchrone (la quantité de temps allouée aux lignes basses vitesses sont différentes)



Multiplexage temporel (TDMA) - a



(a) Transmitter



Multiplexage temporel (TDMA) - b

VIII.3. Multiplexeur d'ondes (WDM : Wave division multiplexing)

Ce multiplexage n'est possible que pour les réseaux où la transmission ce fait par ondes lumineuses, c'est le cas essentiellement des fibres optiques. Il s'agit d'utiliser des faisceaux lumineux avec des bandes de fréquences différentes (dans ce sens, WDM est un cas particulier du multiplexage fréquentiel). Chaque couleur de faisceau constitue un canal.

Exercices

Exercice 1
Soit la chaîne de bits 110101001.
a) Appliquer les trois types de modulation à la chaîne précédente.
b) Transmettre la chaîne précédente avec un seul mode de modulation, mais avec trois bits par moment élémentaire.
c) Représenter la chaîne précédente avec le code NRZI puis le code Manchester II.

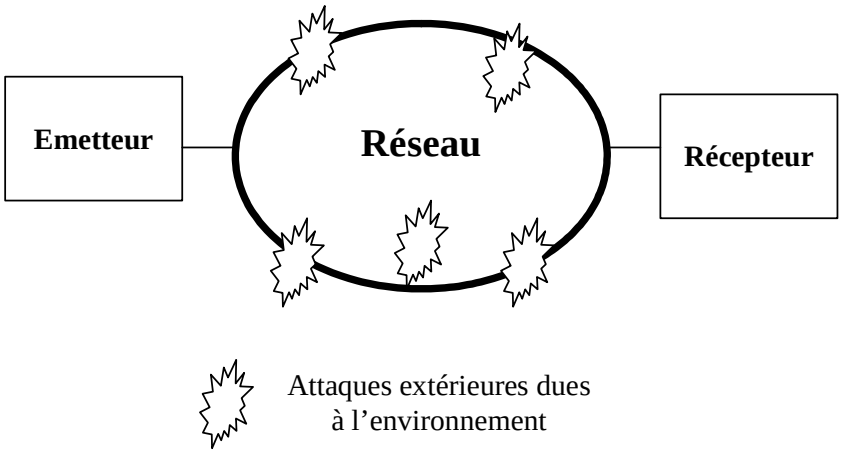
Exercice 2
Quelle est la valeur du rapport S/N (signal sur bruit) permettant de transmettre à 1544 k b/s sur un canal offrant une bande passante de 50 kHz.

Exercice 3
Un son Hi-Fi de bande passante de 0 à 30 kHz a été numérisé par une technique dite MIC (modulation par impulsion codée). La technique MIC consiste à échantillonner un signal analogique et le représenter sous la forme d'échantillons de valeurs numériques sur 10 bits. La fréquence d'échantillonnage est le double de la fréquence la plus élevée du signal (c'est un théorème de Shannon qui le stipule).
a) Quel est le débit binaire nécessaire à la transmission des données du signal ainsi numérisé ?
b) On désire multiplexer 8 canaux de ce type de voie (basse vitesse) à l'aide d'un multiplexeur temporel synchrone. Quel est le débit binaire de la voie haute vitesse du multiplexeur ?

Chapitre 4
Protection contre les erreurs de transmission

I. Notion d'erreur de transmission

Les voies de communication sont imparfaites :
- à cause des bruits :
  + bruit aléatoire
  + bruit en provenance de canaux voisins
  + bruit dû aux équipements de raccordement ou autres
- à cause de la qualité de la ligne.



Réseau et de communication son environnement

Conséquences des bruits

- perte d'information durant la transmission,
- altération de l'information,
- adjonction d'information.

Les situations d'erreurs peuvent être réduites selon les types de supports, mais pas éliminées totalement.

⇒ Nécessité d'un mécanisme de protection contre les erreurs

- Codes détecteurs
- Codes correcteurs d'erreurs

II. Codes cycliques (CRC : cyclic redundancy check)

a) Association d'un polynôme à une chaîne de caractères

Soit *S* une chaîne de *n* bits à transmettre : $a_{n-1}a_{n-2} \ldots a_1a_0$

Les a_i sont des valeurs binaires *0* ou *1*.

A la chaîne *S*, on associe un polynôme *P(x)* défini à partir des bits de *S* :

$P(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \ldots + a_1x + a_0$.

b) Utilisation d'un polynôme générateur

Un code cyclique est défini par un polynôme générateur fixe connu par tous les équipements du réseau. Ce polynôme, noté *g(x)*, est de degré *k*. *k* est le nombre de bits de contrôle généré par le code.

Exemples de g(x) très utilisés :

CRC-32 = $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

CRC-16 = $x^{16} + x^{15} + x^2 + 1$

CRC-12 = $x^{12} + x^{11} + x^3 + x^2 + x + 1$

c) Codage à l'émission

Soit *C(x)* = reste de la division de $x^kP(x)$ par *g(x)*.

$C(x) = c_{k-1}x^{k-1} + c_{k-2}x^{k-2} + \ldots + c_1x + c_0$.

Les *k* bits $c_{k-1}, c_{k-2}, \ldots, c_0$ constituent les bits de contrôle (appelés aussi CRC)

On transmet les *n* bits de données (de a_{n-1} à a_0) suivis des bits de contrôle (de c_{k-1} à c_0)

Décodage à la réception

Soit *V* la chaîne de *m* bits reçue : $v_{m-1}v_{m-2}\ldots v_1 + v_0$.

On associe le polynôme *V(x)* à la chaîne *V* :

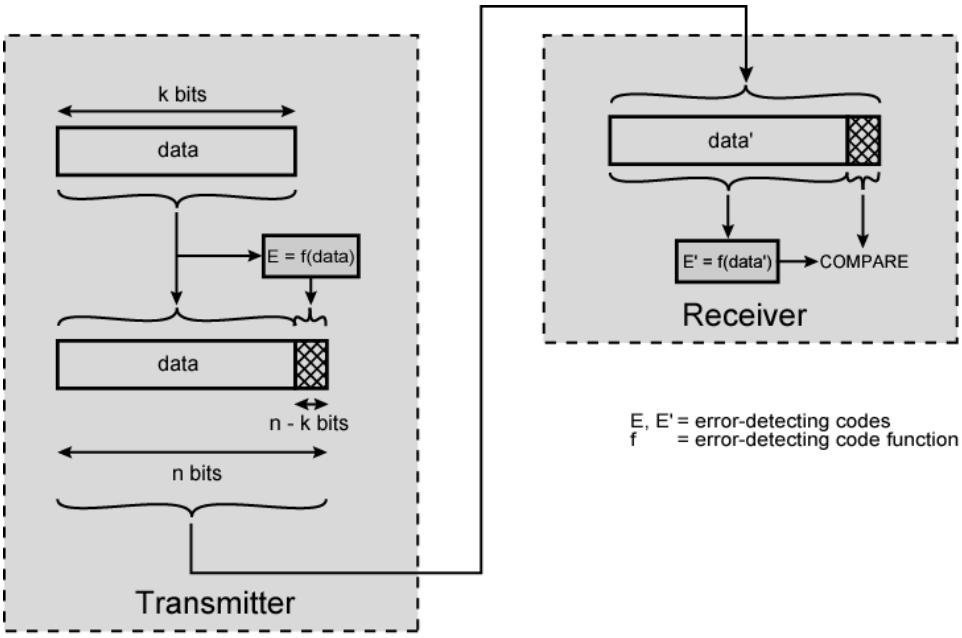
$V(x) = v_{m-1}x^{m-1} + v_{m-2}x^{m-2} + \ldots + v_1x + v_0$.

R(x) = reste de la division de *V(x)* par *g(x)*.

Si *R(x)* est nul, il n'y a pas de détection d'erreur. Sinon il y a erreur.

Remarques :

- L'opération de division se fait modulo 2 dans le corps (0, 1, +). L'opération + est un *ou exclusif*. Il n'a pas de différence entre + et -.
- Le principe des codes cycliques décrit précédemment ne signifie pas que toutes les erreurs sont détectables par les codes cycliques.



Récapitulatif du principe de codage/décodage

III. Correction d'erreurs (FEC : Forward error correction)

III.1. Quelques définitions

Rendement d'un code (R) = rapport entre l'information utile sur l'information transmise

$R = n/(n+k)$

n : nombre de bits de l'information utile), k : nombre de bits de contrôle

Mot du code = chaîne de bit obtenue en appliquant le code.

Poids de Hamming d'une chaîne *S* (noté *PH(S)*) = nombre de bits à 1 dans une chaîne

Distance de Hamming entre deux chaînes *S1* et *S2* (noté *DH(S1, S2)*) = nombre de bits à 1 du vecteur somme de *S1* et *S2*.

Exemple :

S1 = 10110011

S2 = 01100010

PH(S1) = 5

PH(S2) = 3

S1 + S2 = 11010001

DH(S1, S2) = 4

Distance de Hamming d'un code = minimum des distances de tous les couples de mots appartenant à ce code.

Correction

- Par retransmission
- Automatique

III.2. Correction par retransmission

- Dans cette approche, on demande à l'émetteur de retransmettre toute trame jugée erronée.
- Elle est liée au mécanisme de contrôle de flux

  + Un acquittement pour chaque trame (technique Stop and wait)

  + Un acquittement pour plusieurs trames (technique avec anticipation ou technique de la fenêtre coulissante).
- Nombre de retransmissions maximum : limité à la configuration.

III.3. Correction automatique

- On essaie de "deviner" la trame émise à partir de la trame reçue.
- On substitue la chaîne reçue par la chaîne qui est un mot du code dont la distance de Hamming est la plus petite
- **Théorème** : si la distance de Hamming d'un code est supérieure ou égal à *2e +1*, alors on peut corriger *e* erreurs.

Quand corriger ?

- Quand le récepteur n'a pas de moyen pour notifier les erreurs à l'émetteur (case des sondes)
- Quand le temps de retransmission est trop important et rend l'information invalide après sa retransmission (cas des applications temps réel).

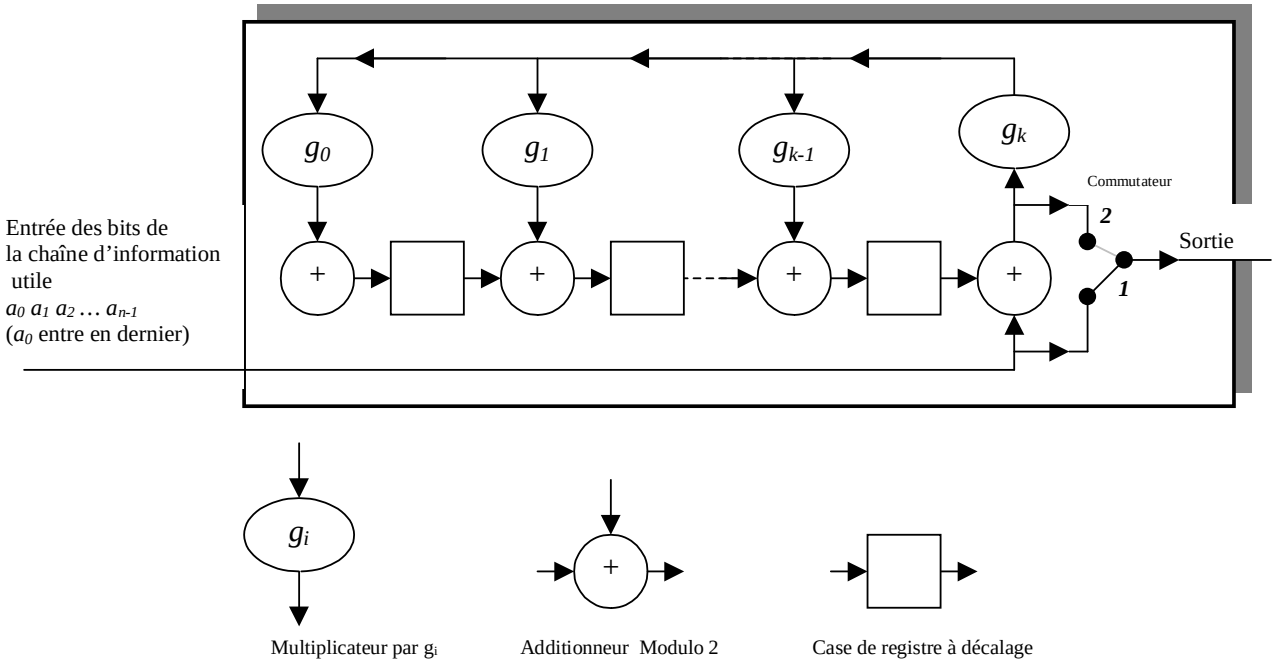
IV. Mise en œuvre des mécanismes de détection d'erreur

Les opérations de codage et décodage sont déclenchées pour chaque transmission de trame. Pour que le temps pris par ces opérations ne crée pas de ralentissement de la communication, elles sont réalisées par câblage. Pour cela on utilise un circuit de codage et un circuit de décodage.
Le circuit de codage (ou de décodage) est constitué d'un registre à décalage à *k* cases (*k* est le degré du polynôme générateur *g(x)*, d'additionneurs modulo 2 et de multiplicateurs binaires formés par les coefficients de *g(x)*.

Principe du codage :

- Les cases du registre à décalage sont remises à zéro au début de chaque transmission et le commutateur du circuit est mis en position 1.
- Les bits d'information utile entrent dans le circuit de codage en commençant par le bit a_{n-1} et en terminant par le bit a_0.
- A chaque top d'horloge, un bit d'information utile entre dans le circuit, ensuite les opérations d'addition et de multiplication sont exécutées simultanément et les cases du registre à décalage sont décalées d'une position vers la droite. Le commutateur du circuit se trouve en position 1, pour laisser sortir chaque bit d'information utile directement sur la sortie du circuit de sorte que tous les bits d'information utile se retrouvent dans le bon ordre sur la ligne de communication.

- Quand tous les bits d'information utile ont traversé le circuit, les cases du registre à décalage contiennent les bits du CRC. On met alors le commutateur sur la position 2, et les *k* cases du registre à décalage sont envoyées sur la sortie au bout de *k* tops d'horloges. Ainsi, les bits du CRC suivent ceux des données (le bit c_0 est le dernier bit transmis)



Circuit pour le codage

Principe du décodage :
Le décodage utilise un circuit analogue à celui du codage. Lorsque tous les bits de la chaîne reçue ont traversé le circuit de décodage, les cases du registre à décalage contiennent le reste de la division de *Q(x)* (qui est le polynôme correspondant à la chaîne reçue) divisé par *g(x)*. Si le registre à décalage n'est pas nul, un signal est généré pour signaler une erreur de transmission.

Exercices

Exercice 1

On considère un code cyclique avec un polynôme générateur égal à $g(x) = x + 1$.

a) Calculer les bits de contrôle pour la chaîne de bits *10011001*.
b) Introduire une erreur et montrer que le code choisi permet de la détecter.
c) Introduire deux erreurs et montrer que le code choisi ne permet pas de les détecter.
d) Introduire trois erreurs et montrer que le code choisi permet de les détecter.
e) Généraliser le raisonnement de détection à *m* erreurs (*m* pair ou impair)

Exercice 2

Est-ce que le code de parité impaire est un code cyclique ? justifier la réponse.

Exercice 3

Soit une transmission de caractères à 3 bits. On transmet un seul caractère à la fois. La détection des erreurs se fait en utilisant un CRC3 dont le polynôme générateur est $G(x) = x^3 + 1$. Les caractères à transmettre sont 110 et 101.

a) Quelle est la chaîne de bits réellement transmise pour chacun des deux caractères ?

b) Introduire une erreur et montrer que cette erreur est détectée par le code.
c) Quels sont les types d'erreurs non détectées par le code choisi ?

Exercice 4

a) Calculer les bits de contrôle de la chaîne *011011* avec un polynôme $g(x) = x^3 + x^2 + 1$.
b) Donner une classe (ou forme) d'erreur non détectable par un code avec $g(x) = x^3 + x^2 + 1$.

Exercice 5

a) Donner le polynôme générateur d'un code cyclique permettant de détecter tout paquet de trois erreurs consécutives (c'est-à-dire avec trois bits consécutifs erronés). On s'intéresse à un seul paquet d'erreurs à la fois.
b) Donner le polynôme générateur d'un code cyclique permettant de détecter tout couple de paquets de trois erreurs consécutives (c'est-à-dire avec trois bits consécutifs erronés). On s'intéresse à deux couples de paquets d'erreurs à la fois dans des chaînes ne pouvant excéder 8 bits.

Exercice 6

On s'intéresse à une source d'informations qui transmet des données sous forme d'octet (8 bits par trame). Chaque octet est contrôlé par un bit de parité paire. Si on souhaite corriger des erreurs, tous les octets avec un bit de parité paire ne peuvent pas être des mots du code.

a) proposer un vocabulaire (c'est-à-dire un ensemble de mots) pour un code permettant de corriger une erreur.
b) proposer un vocabulaire pour un code permettant de corriger deux erreurs.

Exercice 7

Donner le circuit de code, puis le circuit de décodage pour :

a) le polynôme $g(x) = x + 1$.
b) le polynôme $g(x) = x^2 + x + 1$.

Utiliser des exemples de chaînes pour tester les quatre circuits.

Exercice 8

Soit une source qui ne peut avoir comme information utile à transmettre que les chaînes suivantes :

1000000	0001000
1000010	1000001

On utilise un polynôme $g(x) = x^3 + 1$ pour élaborer le CRC.
Quel nombre d'erreurs peut-on corriger avec le code obtenu ?

Exercice 9

Calculer le nombre (minimum, puis maximum) de trames qu'il faut pour transmettre un fichier de 2 M octets à raison de 1 k octets par trame et avec une fenêtre d'anticipation égale à 100.

a) sans erreur de transmission
b) avec une seule trame sur 100 qui est erronée (et avec retransmission uniquement de la trame erronée).

Exercice 10

Un capteur raccordé à un réseau produit des données correspondant à la position verticale d'un objet. La position est graduée de 0 à 15 (la mesure de position est une valeur entière). On transmet une seule position par message. Pour transmettre les valeurs de position, le capteur utilise un polynôme générateur $G(x) = x^4 + x + 1$. Soit M un message codé qui contient la position la plus haute.

a) Quelle est la chaîne de bits contenue dans le message M ?

b) Introduire deux erreurs sur le message M et montrer que ces erreurs sont détectées par le code.

c) Quels sont les types d'erreurs non détectées par le code choisi ?

c) Proposer les circuits de codage et de décodage d'erreurs pour le code choisi et montrer le fonctionnement de ces circuits dans le cas où la position est égale à 5.

Exercice 11

Définir et tester le circuit de codage et décodage pour

a) $g(x) = x + 1$
b) $g(x) = x^2 + 1$

2ème Partie

- **Couche Liaison de données**
- **Couche Réseau**
- **Sécurité**

Chapitre 5
Couche liaison de données

I. Introduction

Le rôle de la couche liaison de données (LdD) est de gérer l'accès au support de transmission et les connexions logiques (quand celles-ci existent). Ainsi la couche LdD est composée de deux sous-couches :

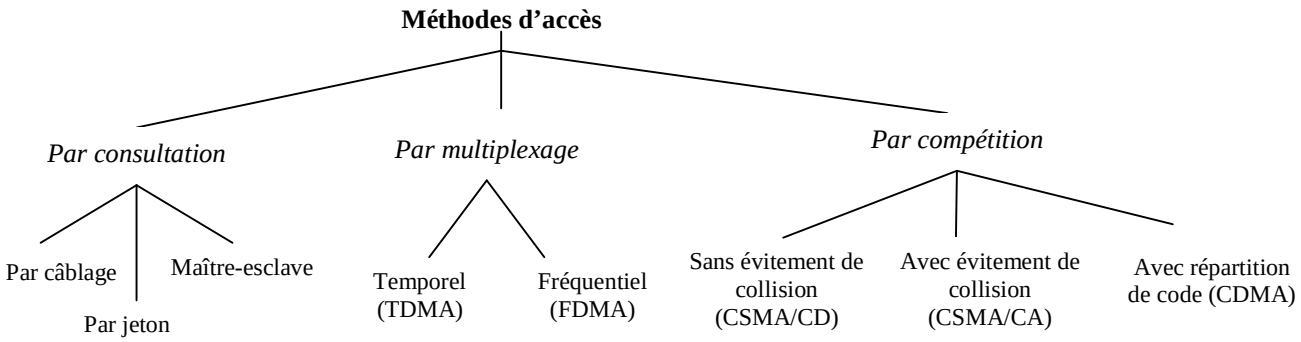
- sous-couche MAC (Medium Access Control),
- sous-couche LLC (Logical Link Control).

II. Techniques d'accès au support (MAC)

La méthode ou technique d'accès au médium (souvent abrégée par MAC : "Medium Access Control") est une composante fondamentale des réseaux. C'est le protocole de la méthode d'accès d'un réseau qui permet de définir les règles d'attribution du support de transmission aux nœuds du réseau. En d'autres termes, la technique MAC définit les règles de partage du médium entre nœuds du réseau.

Beaucoup de méthodes d'accès au support de transmission ont été proposées et expérimentées pour répondre à différents besoins. Certaines de ces méthodes sont devenues des normes internationales et d'autres sont limitées à quelques réseaux propriétaires. Dans ce paragraphe, nous allons présenter les principes généraux des techniques MAC les plus connues.

Les techniques MAC peuvent être divisées en trois groupes, selon les règles auxquelles doivent se conformer les nœuds pour accéder au support de transmission : méthodes par consultation, méthodes par compétition et méthodes par multiplexage. Le choix d'une méthode d'accès dépend de plusieurs critères, notamment : la topologie du réseau, le type de support (filaire ou non filaire), le déterminisme d'accès, la possibilité de privilégier certains nœuds et la tolérance aux fautes.



Classes de méthodes d'accès dans les réseaux

II.1. Méthodes d'accès par consultation

Le principe de base des méthodes par consultation est que les nœuds "se consultent" pour décider de celui qui a le droit d'utiliser le support de transmission. Un nœud accède au support s'il y est autorisé. L'autorisation peut être gérée de plusieurs manières, notamment :

- **Par échange d'informations** : les nœuds s'échangent des informations de manière permanente pour connaître celui qui a le droit d'émettre. Généralement, le droit d'utiliser le support est géré soit de manière centralisée par un site privilégié (on parle dans ce cas d'une technique de type maître-esclave), soit par le passage d'un jeton.
- **Par utilisation d'éléments physiques dédiés :** les nœuds utilisent des lignes spéciales pour marquer leur intention d'utiliser le médium et un circuit permet d'autoriser les nœud à émettre. Cette technique est rarement utilisée dans les réseaux où le nombre de nœuds est variable ou important.

Il faut souligner que les techniques par consultation sont surtout utilisées dans les réseaux locaux et/ou filaires.

a) Méthodes d'accès maître-esclave

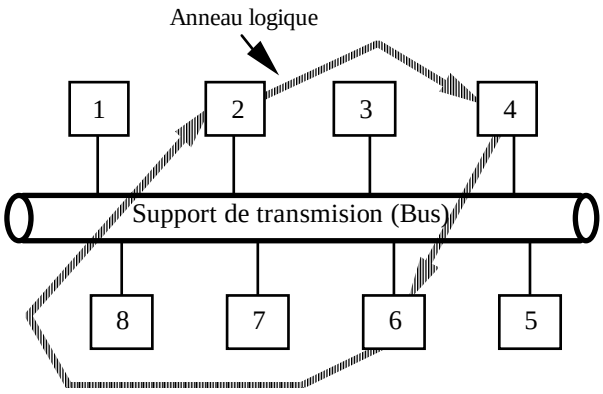
Dans les méthodes maître-esclave, les nœuds sont regroupés en deux catégories : un nœud maître (ou nœud primaire) et des nœuds esclaves (ou nœuds secondaires). Un nœud esclave n'a le droit d'émettre que si le site maître l'autorise. Les règles appliquées par le nœud maître pour autoriser les nœuds esclaves à émettre sont diverses. En particulier, les deux manières suivantes d'autoriser les nœuds esclaves sont utilisées dans les réseaux que l'on trouve dans les installations industrielles automatisées (comme les laminoirs, les complexes pétrochimiques, les usines de montages de voitures...) :

- *Méthode par scrutation régulière* : le nœud maître scrute un par un les nœuds esclaves selon un ordre préétabli ; lorsque le dernier nœud est scruté, le nœud maître reprend la scrutation à partir du premier nœud. Il donne le droit à émettre à un site esclave qui l'utilise le temps de transmettre une ou plusieurs trames, ensuite le nœud esclave rend le droit d'émettre au nœud maître. Si un site esclave n'a pas de trame à transmettre, il rend immédiatement le droit d'émettre au nœud maître. L'inconvénient de cette méthode est qu'il y a parfois une perte de temps à scruter des nœuds qui n'ont rien à transmettre. Différentes solutions ont été proposées pour atténuer ce défaut.
- *Méthode utilisant une table d'arbitre* : dans cette méthode, le nœud maître est appelé *arbitre du réseau* et possède une table qui lui indique à quel moment exactement il faut scruter chaque nœud esclave. Dans ce cas, la scrutation est interprétée par le nœud esclave comme une demande à émettre (et non une invitation à émettre, comme dans le cas de la méthode à scrutation). Un des réseaux de terrain qui utilise cette méthode est le réseau *WorldFIP* développé en France.

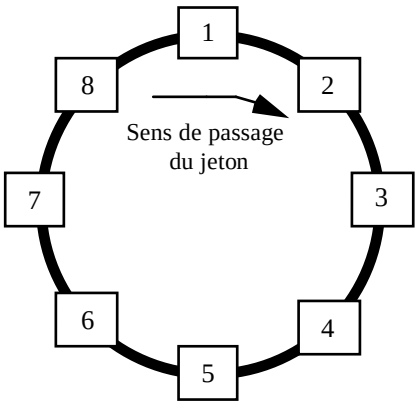
Pour des raisons de tolérance aux fautes, certains (ou tous les) nœuds esclaves ont le statut de nœud maître de secours et ils peuvent reprendre (mais un seul nœud à la fois) le contrôle du réseau lorsque le nœud maître courant tombe en panne.

b) Méthodes d'accès à jeton

Les méthodes à jeton fonctionnent essentiellement sur des topologies en boucle ou en bus. Sur une boucle, l'ordre de passage du jeton est défini par l'ordre physique de connexion des nœuds. Sur une topologie en bus, on définit un anneau logique pour déterminer l'ordre de passage du jeton. Construire un anneau logique revient à indiquer (par un protocole d'initialisation de réseau) les adresses du prédécesseur et successeur de chaque nœud sur l'anneau logique. Lorsque le jeton fonctionne sur un bus, on parle de *bus à jeton* ("token bus") et de *boucle à jeton* ("token ring"), dans le cas d'une topologie en boucle. Une fois que l'anneau physique ou logique est établi, un jeton (un jeton est une trame qui a un format spécial) est créé (par une procédure de création du jeton), ensuite le jeton passe de nœud en nœud selon l'ordre de l'anneau. Chaque nœud n'a le droit d'émettre que s'il reçoit le jeton, après quoi il peut transmettre pendant un certain temps limité et passe ensuite le jeton à son successeur. Le jeton tourne sur l'anneau tant qu'il y a au moins deux nœuds actifs.



Exemple de bus à jeton



Exemple de boucle à jeton

Techniques à jeton

II.2. Méthodes d'accès par compétition

Dans les méthodes d'accès par compétition, dites aussi *méthodes d'accès aléatoire,* il n'y a pas d'élément logique ou physique qui permet aux nœuds de se mettre d'accord pour utiliser le support ; chaque nœud peut commencer à transmettre dès qu'il le souhaite (à quelques exceptions près). Cela conduit évidemment à des situations de conflit d'accès quand deux ou plusieurs nœuds transmettent simultanément. Plusieurs solutions ont été proposées pour gérer les situations de conflit (*collision*). Dans le contexte des réseaux locaux et réseaux sans fil, c'est essentiellement la technique CSMA et ses variantes qui sont utilisées.

a) Méthode CSMA/CD

Dans la méthode CSMA/CD ("Carrier Sense Multiple Access with Collision Detection"), un nœud peut émettre dès qu'il le souhaite à condition de détecter que le support est libre. Si le support n'est pas libre, le nœud ajourne sa tentative jusqu'à la prochaine libération du support. Si le bus est libre, le nœud commence sa transmission et compare ce qu'il émet par rapport à ce qu'il reçoit, s'il y a une différence entre les deux signaux, il est fort probable qu'au moins un autre nœud soit en train de transmettre en même temps que lui. Dans ce cas, il arrête sa tentative de transmission, envoie un signal de brouillage pour signaler la collision aux autres nœuds, attend pendant un certain délai aléatoire avant de tenter une nouvelle fois sa transmission. L'inconvénient de cette méthode est que si le nombre de nœuds qui souhaitent transmettre est important le nombre de collisions devient tel qu'aucun de ces nœuds n'arrivent à transmettre sa trame. Par conséquent, la méthode CSMA/CD n'est pas adaptée aux applications temps réel. Il faut signaler que de par sa simplicité, la méthode CSMA/CD est utilisée par le réseau local le plus répandu au monde, à savoir le réseau Ethernet.

Pour éviter les collisions en chaîne conduisant à des temps de réponse excessifs, des améliorations de la méthode CSMA/CD ont été proposées, notamment les méthodes CSMA/CA, CSMA/CR et CSMA/DCR.

b) Méthode CSMA/DCR ("CSMA with Deterministic Collision Resolution")

La méthode CSMA/DCR intègre un algorithme de résolution de collision lancé par tous les nœuds qui détectent une collision. Cet algorithme utilise une technique de résolution en arbre binaire : en cas de collision, les nœuds sont partagés en deux groupes selon leurs adresses : un groupe des gagnants et un groupe des perdants. Les nœuds appartenant au groupe des perdants cessent d'émettre, les autres tentent de transmettre. Si le groupe des gagnants contient plus d'un nœud, il y a de forte chance qu'une nouvelle collision soit détectée, auquel cas le groupe est à nouveau scindé en deux, et ainsi de suite jusqu'à ce que le groupe des gagnants ne contienne qu'un seul nœud qui peut alors transmettre tranquillement sa trame. La méthode CSMA/DCR permet de borner le temps d'attente pour transmettre une trame.

c) Méthode CSMA/CA (“CSMA with Collision Avoidance”)

Dans la méthode CSMA/CA, chaque nœud utilise les informations qu’il possède sur l’état d’activité du support pour calculer la probabilité d’entrer en collision s’il tente une transmission. Le nœud évite de transmettre pendant les instants où la probabilité de collision est jugée élevée. Cela conduit un nœud à attendre pendant un nombre (qui varie selon la charge du support) d’unités de temps même si le support est libre. La chance d’avoir deux nœuds qui tentent de transmettre en même temps est réduite par rapport à CSMA/CD. Les situations de collision ne sont pas complètement écartées en utilisant CSMA/CA. Alors, les nœuds utilisent les acquittements pour savoir s’il y a eu collision ; la non-réception d’acquittement au bout d’un certain temps conduit un nœud émetteur à considérer qu’il y a eu collision.

d) Méthode CSMA/CR (“CSMA with Collision Resolution”)

Avant de commencer sa transmission, chaque station doit tester l’état du support et elle ne peut transmettre que si le support est libre. Pour éviter les collisions en chaîne, un nœud qui transmet une trame (sachant qu’une trame commence par une adresse unique), cesse d’émettre s’il reçoit un bit différent du sien. Ainsi, un nœud qui émet un bit à *1* s’arrête s’il voit passer sur le support un bit à *0*. En revanche un nœud qui reçoit un bit identique à celui qu’il a émis continue de transmettre. Comme les adresses diffèrent au moins par un bit, un seul nœud poursuit la transmission de sa trame jusqu’à sa fin. Cette technique est mise en œuvre sur le réseau CAN (Controler Area Network) qui est le réseau le plus utilisé dans le domaine de l’automobile.

e) Méthode à répartition de code : CDMA (“Code Division Multiple Access”)

C’est l’une des techniques d’accès utilisées dans le domaine des réseaux sans fil, notamment dans les réseaux UMTS et dans les réseaux locaux sans fil. Elle est fondée sur un principe qui permet l’accès multiple mais aussi la sécurité des communications au niveau physique. L’idée de base de CDMA c’est comme si dans une salle, plusieurs personnes parlent en même temps mais dans des langues différentes. Pour chaque personne qui parle, seul son correspondant connaît la langue et arrive à extraire du ‘vacarme ambiant’ ce que dit son interlocuteur. Celui qui ne connaît pas une langue, reçoit le signal lié à cette langue mais ne peut pas le comprendre.

Techniquement parlant, CDMA est basée sur la répartition par codes qui permet à plusieurs sources d’émettre sur les mêmes fréquences. Chaque utilisateur est différencié du reste des utilisateurs par un code *C* qui lui a été alloué au début de sa communication et qui est orthogonal au reste de codes liés à d’autres utilisateurs. Dans ce cas, pour écouter l’utilisateur ayant le code *C*, le récepteur n’a qu’à multiplier le signal reçu par le code *C* associé à cet utilisateur. Chaque code est représenté sur *k* éléments. A chaque bit à transmettre, CDMA crée une séquence de *k* bits (on parle d’étalement de spectre) obtenu à partir de la valeur du bit initial et du code. Le signal ainsi transmis s’apparente à du bruit (car seul celui qui connaît le code de la source est capable de retrouver la chaîne de bits initiale). Dans ce sens, CDMA permet de renforcer la sécurité au niveau physique. A noter de CDMA été longtemps utilisée par les militaires.

Exemple :
Une source A utilise un code égal à <1, -1, 1, 1, -1, -1>. Pour transmettre un bit 1, elle transmet six bits <1 0 1 1 0 0> et pour transmettre un bit 0, elle transmet six bits <0 1 0 0 1 1>. Cela signifie que pour transmettre un bit initial égal à 1, la source génère 6 bits où le bit i (i=1, ..., 6) est égal à 1 si le ième élément du code est égal à 1 et 0 s’il est égal à -1. La séquence de bits transmis dans le cas où le bit initial à transmettre est 0 est obtenue en faisant le complément à 2 de la séquence obtenue pour un bit initial égal à 1.
La méthode CDMA repose sur l’orthogonalité des codes attribués aux sources. Mathématiquement parlant, si on a *n* utilisateurs avec *n* codes, alors tout ensemble de vecteurs dans le *n*-espace sont orthogonaux si tout point dans le *n*-espace peut être exprimé seulement avec une combinaison linéaire de ces vecteurs. On rappelle qu’un produit scalaire de deux vecteurs U et V de composantes u_1, u_2... u_n et v_1, v_2,... v_n est la somme $u_1v_1 + u_2v_2 + ... u_nv_n$.
Par exemple, si on affecte les codes suivants à 3 stations (A, B, C), alors les codes sont orthogonaux :

Code de A = <1, -1, -1, 1, -1, 1>
Code de B = <1, 1, -1, -1, 1, 1>
Code de C = <1, 1, -1, 1, 1, -1>

Lorsque les signaux arrivent en provenance de plusieurs sources qui transmettent simultanément, ces signaux s'additionnent chez le récepteur. Le récepteur calcule la corrélation du signal avec le code de l'émetteur, ce qui permet de retrouver les bits du message (s'il n'y a pas eu d'erreur de transmission). Ci-dessous, nous expliquons à travers un exemple le principe simplifié de CDMA (attention les séquences composées de 1 et -1 sont utilisées pour comprendre le principe de CDMA, la transmission effective ne considère que des 1 et 0) :

- On considère deux sources S_A et S_B qui transmettent en même temps.
- La source S_A transmet le message A ayant pour information la chaîne binaire '100' mais codée par la séquence <1 -1 -1> où un bit initial à 1 est remplacé par un élément à 1 dans la séquence et un bit initial à 0 est remplacé par -1. La source S_B transmet le message B ayant pour information la chaîne binaire '001' mais codée par la séquence <-1 -1 1> avec la même règle que pour le message A.
- La source S_A utilise comme code la séquence C_A = <1 -1 -1 1>. La source S_B utilise comme code la séquence C_B = <1 -1 1 -1>. Les deux codes sont choisis pour être orthogonaux, c'est-à-dire que leur produit scalaire C_A*C_B est nul et leur produit scalaire C_A*C_A est maximum.
- Le message A est multiplié par le code C_A pour obtenir le produit A*C_A :
  A*C_A = {<1 -1 -1 1>, <-1 1 1 -1>, <-1 1 1 -1>}.
- Le message B est multiplié par le code C_B pour obtenir le produit B*C_B
  B*C_B = {<-1 1 -1 1>, <-1 1 -1 1>, <1 -1 1 -1>}.
- Les séquences correspondant aux deux produits A*C_A et B*C_B traduites en termes de bits sont transmises. Une fois transmise simultanément, ces deux séquences produits, A*C_A et B*C_B, sont additionnées car les signaux simultanés s'additionnent.
  A*C_A + B*C_B = {<0 0 -2 2>, <-2 2 0 0>, <0 0 2 -2>
- A la réception, le destinataire du message A (et qui connaît le code C_A) multiplie la séquence reçue par le code C_A. On a : (A*C_A + B*C_B)*C_A = {<0 0 2 2>, <-2 -2 0 0>, <0 0 -2 -2>}. On prend la moyenne des signaux reçus sur la durée d'un bit initial. C'est-à-dire (0+0+2+2)/4 = 1, (-2-2+0+0)/4 = -1, (0+0-2-2)/4 = -1. Ce qui permet de retrouver la séquence <1 -1 -1> et ensuite la chaîne initiale '100'.
- A la réception, le destinataire du message B (et qui connaît le code C_B) multiplie la séquence reçue par le code C_B. On a : (A*C_A + B*C_B)*C_B = {<0 0 -2 -2>, <-2 -2 0 0>, <0 0 2 2>}. Avec la même procédure de calcul de la moyenne effectué pour le message A, on retrouve la chaîne '001'.

Le CDMA peut être combiné aux techniques de multiplexage temporel et fréquentiel pour donner lieu au WCDMA (wideband CDMA), TD-CDMA (Time Division CDMA)...

II.3. Méthodes d'accès par multiplexage

Il y a trois catégories de multiplexage : multiplexage temporel, multiplexage fréquentiel et multiplexage d'ondes.

a) Multiplexage temporel (TDMA : Time Division Multiple Access)

Cette technique est aussi connue sous le sigle TDMA ("Time Division Multiplexing Access"). L'allocation du support de communication fonctionne de manière cyclique. On fixe, selon les besoins des nœuds connectés, la durée d'un tour d'allocation du support (soit *TT* cette durée). Chaque nœud *i* connaît sa position exacte dans un tour et a la droit d'émettre au maximum pendant un temps H_i. La somme des H_i est égale à *TT*. Les valeurs de temps (quanta) des H_i alloués aux nœuds peuvent être identiques ou différentes selon l'importante et la quantité du flux de données généré par chaque nœud. Dans le premier cas, on parle de **TDMA synchrone** et dans le second, de **TDMA statistique**.

L'inconvénient majeur du TDMA synchrone est que lorsqu'un nœud n'a pas de données à émettre, le support reste libre, même si d'autres nœuds ont beaucoup de trames à transmettre. On notera que cette technique est utilisée dans le domaine de la téléphonie.

Le TDMA statistique améliore l'utilisation de la bande passante en fixant de manière dynamique la durée d'utilisation du médium par chaque station. La signalisation (gestion des demandes, notification à chaque station de la durée quelle peut utiliser) rend ce TDMA plus complexe à mettre en oeuvre.

b) Multiplexage fréquentiel (FDMA : Frequency Division Multiple Access)

Dans ce cas, la bande de fréquences du réseau est subdivisée en sous-canaux et chaque nœud n'a le droit d'émettre que sur un seul sous-canal qui lui est réservé. On notera que cette technique est très utilisée dans le domaine de la radio (où chaque chaîne de radio émet sur une bande de fréquences qui lui est réservée).

c) Multiplexage d'ondes (WDM : Wavelength Division Multiplexing)

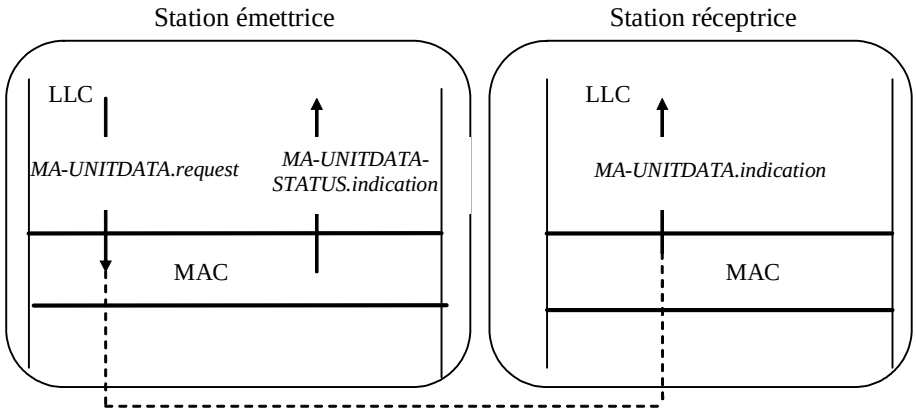
Ce multiplexage n'est possible que pour les réseaux où la transmission ce fait par ondes lumineuses, c'est le cas essentiellement des fibres optiques. Il s'agit d'utiliser des faisceaux lumineux avec des bandes de fréquences différentes (dans ce sens, WDM est un cas particulier du multiplexage fréquentiel). Chaque couleur de faisceau constitue un canal. Actuellement, on peut avoir, sur une même fibre optique disponible dans le commerce, plusieurs centaines de faisceaux lumineux offrant chacun un débit de plusieurs dizaines de Gb/s sur plusieurs dizaines (centaines) de Km.

II.4. Interface MAC/LLC

L'interface entre MAC et LLC offre, en général, les services suivants :

1. *MA-UNITDATA.request* : permet à la sous-couche LLC de demander à la sous-couche MAC de transmettre une trame. Les paramètres de cette primitive englobent au moins : l'adresse source, l'adresse de destination et les données.
2. *MA-UNITDATA.indication* : permet à la sous-couche MAC d'indiquer à la sous-couche LLC qu'une trame est arrivée. Les paramètres de cette primitive englobent au moins : l'adresse source, l'adresse de destination et les données.
3. *MA-UNITDATA-STATUS.indication* : permet à la sous-couche MAC de rendre un compte à la sous-couche LLC concernant sa demande de transmission (réussite ou échec de la demande transmission). Les paramètres de cette primitive englobent au moins : l'adresse source, l'adresse de destination et le résultat d'émission. Un résultat positif signifie seulement que la trame a pu être envoyée, cela ne signifie pas nécessairement que la trame a été effectivement reçue par son destinataire. Nous verrons que c'est la sous-couche LLC qui gère les acquittements.

Les initiales '*MA*' sont rajoutées aux primitives précédentes pour indiquer qu'il s'agit de primitives de niveau MAC.



Primitives de l'interface MAC/LLC

III. Contrôle de liaison logique (LLC)

Dans sa version complète, la sous-couche LLC gère les connexions, le contrôle de flux et les erreurs. Pour tenir compte des besoins de certains domaines d'applications (telles que certaines applications temps réel où on privilégie le temps de réponse à la perte de messages) où la gestion des connexion ou des acquittements n'est pas souhaitée, trois types de LLC existent :

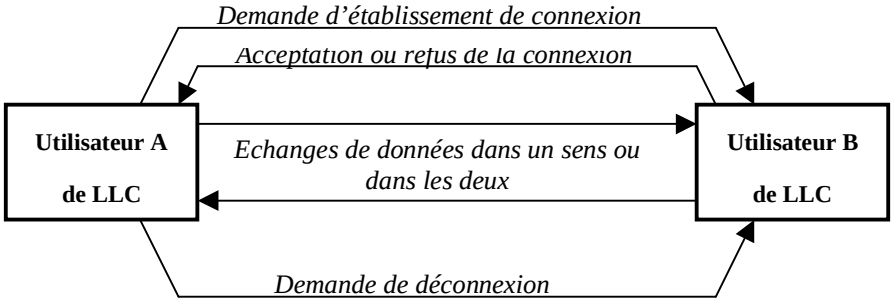
- *LLC1* : service sans gestion de connexion et sans gestion d'acquittement ;
- *LLC2* : service avec gestion de connexion et avec gestion d'acquittement ;
- *LLC3* : service sans gestion de connexion et avec gestion d'acquittement.

Nous nous intéressons dans ce chapitre au LLC le plus complet, c'est-à-dire LLC2.

III.1. Phases du service LLC2

Trois phases sont à distinguer :

- Phase d'établissement de connexion,
- Phase de transfert de données,
- Phase de déconnexion.



Phases du modèle LLC2

Dans les figures suivantes les trames échangées entre entités de niveau LLC sont placées dans des rectangles :

- *SABME* (Set Asynchronous Balanced Mode Extended) : trame utilisée pour demander l'établissement de connexion.
- *DISC* (Disconnect) : trame envoyée par une station pour informer son correspondant qu'elle arrête la connexion.
- *UA* (Unnumbered Acknowledgement) : trame envoyée par une station pour indiquer à son correspondant qu'elle accepte sa demande de connexion ou de déconnexion.
- *I* (Information) : trame de données (c'est-à-dire contenant des données en provenance de la couche supérieure).
- *RR* (Ready to Receive) : trame qui indique que le récepteur a bien reçu les trames dont le numéro précède celui qui est contenu dans la trame RR.
- *RNR* (Not Ready to Receive) : trame envoyée par une station pour indiquer à son correspondant qu'elle a bien reçu les trames dont le numéro précède celui contenu dans la trame RNR, mais qu'elle ne peut plus momentanément recevoir des données.

- *REJ* (Reject) : trame qui indique que les trames de données sont rejetées à partir du numéro spécifié dans la trame REJ.
- *SREJ* (Selective Reject) *:* trame qui indique que la trame de données dont le numéro est spécifié est rejetée (une seule trame est rejetée).
- *FRMR* (FRaMe Reject) : trame envoyée par une station pour indiquer à son correspondant un rejet de trame pour des raisons d'invalidité de format.

Le format de ces trames sera détaillé dans le paragraphe III.9.

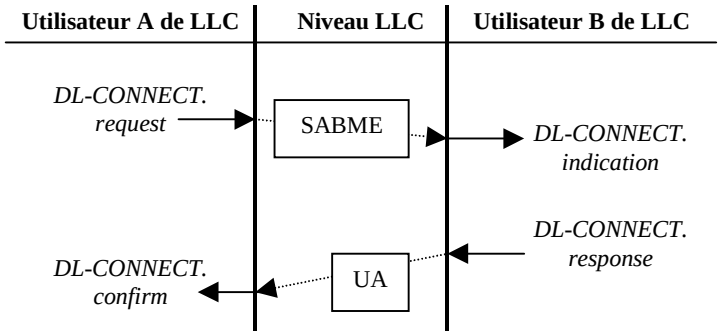
III.2. Primitives d'établissement de connexion

Le service d'établissement de connexion est un service confirmé qui contient les quatre primitives suivantes :

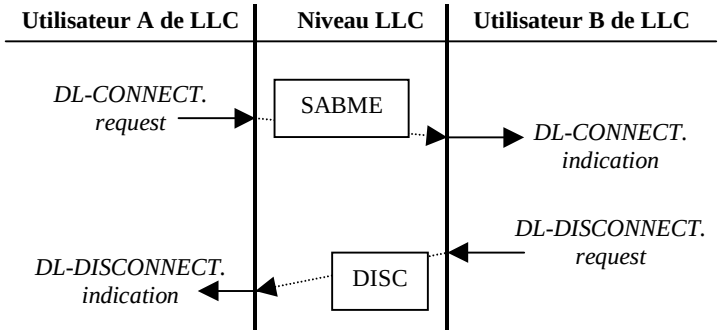
- *DL-CONNECT.request*
- *DL-CONNECT.indication*
- *DL-CONNECT.Response*
- *DL-CONNECT.Confirm*

La demande d'établissement de connexion (*DL-CONNECT.request*) peut être demandée par l'un des deux correspondants ou par les deux en même temps.

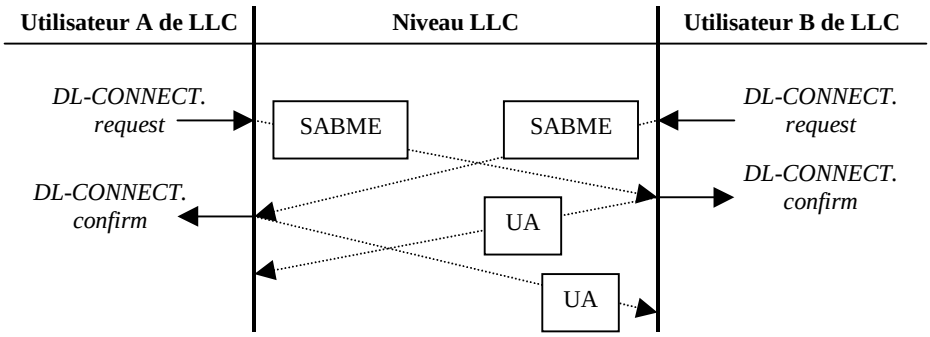
Noter que les initiales '*DL'* signifient Data Link.



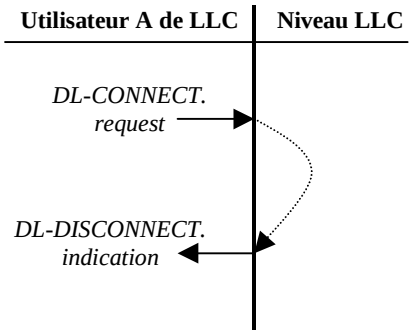
Cas 1 : Demande d'établissement de connexion avec acceptation du correspondant.



Cas 2 : Demande d'établissement de connexion avec refus du correspondant.



Cas 3 : Demandes simultanées d'établissement de connexion.



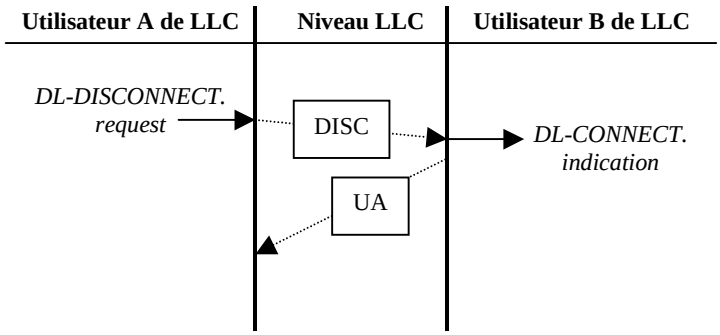
Cas 4 : Refus local d'établissement de connexion
(par exemple, par ce que l'adresse de destination est inexistante
ou que la ligne physique est coupée)

III.3. Primitives de déconnexion

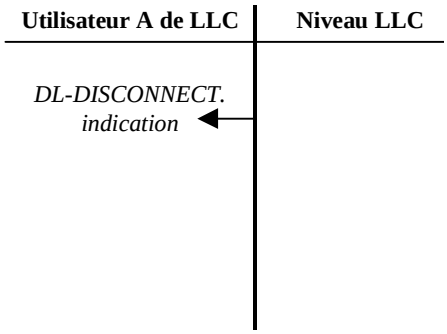
Le service de déconnexion est un service non confirmé qui contient les deux primitives suivantes :

- *DL-DISCONNECT.request*
- *DL-DISCONNECT.indication*

Lorsqu'un utilisateur demande une déconnexion de niveau liaison de données, il n'attend pas de réponse de son correspondant même si la sous-couche LLC du correspondant envoie une trame pour notifier à son homologue que la connexion a bien été libérée.



Cas 1 : Demande de déconnexion par un des deux utilisateurs de LLC.

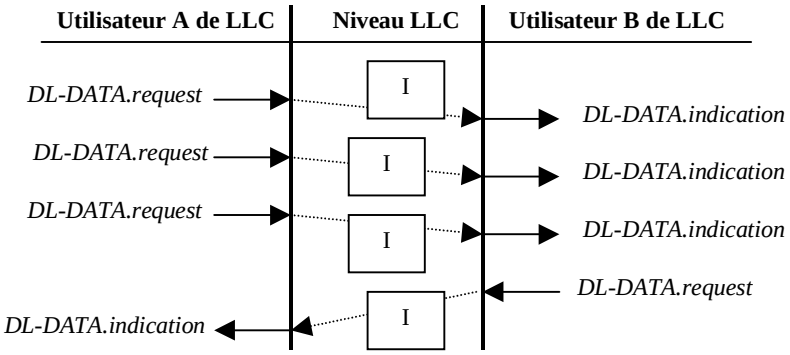


Cas 2 : Déconnexion imposée par le LLC local suite à des anomalies par exemple.

III.4. Primitives d'échange de données

Le service d'échange de données permet aux deux utilisateurs de LLC de s'échanger des données. Il s'agit d'un service non confirmé qui contient les deux primitives suivantes :

- *DL-DATA.request*
- *DL-DATA.indication*

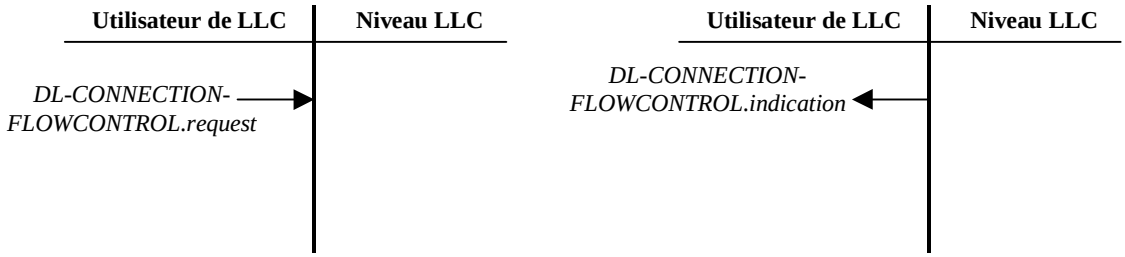


Exemple d'échanges de données.

III.5. Primitives contrôle de flux

Le service de contrôle de flux de niveau LLC est un contrôle de flux local qui permet soit à un utilisateur soit à LLC de fixer la taille de la fenêtre d'anticipation. Il ne conduit donc à aucun échange sur le réseau. Il contient les deux primitives suivantes :

- *DL-CONNECTION-FLOWCONTROL.request*
- *DL-CONNECTION -FLOWCONTROL.indication*

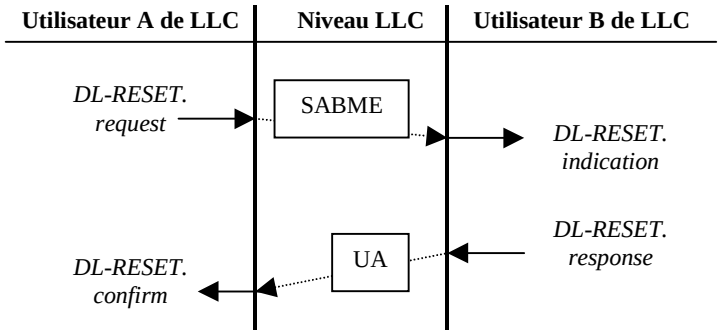


Contrôle de flux.

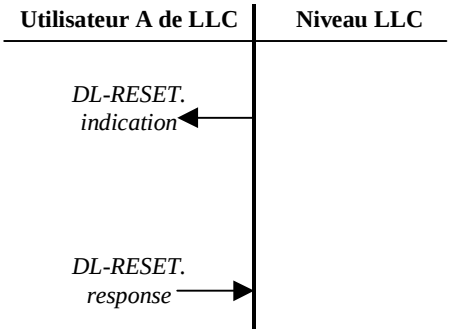
III.6. Primitives de réinitialisation

Le service de réinitialisation permet de purger une connexion en cas d'anomalies ; toutes les trames en attente de réception ou d'émission sont détruites. C'est donc un service à utiliser avec précaution. Il s'agit d'un service confirmé qui contient les primitives suivantes :

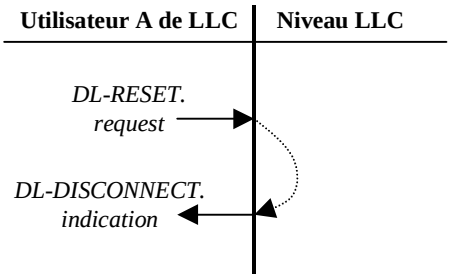
- *DL-RESET.request*
- *DL- RESET.indication*
- *DL-RESET.response*
- *DL- RESET.confirm*



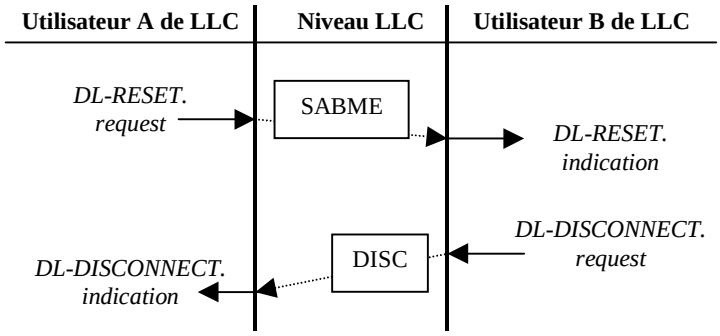
Cas 1 : Demande de réinitialisation avec acceptation du correspondant.



Cas 2 : Demande de réinitialisation locale (interne).



Cas 3 : Refus local de réinitialisation.



Cas 4 : Refus de réinitialisation par le correspondant.

III.8. Contrôle de flux et contrôle d'erreurs

a) Objectifs du contrôle de flux

Le destinataire de trames de données en provenance d'une source peut ne pas avoir les capacités nécessaires à la réception des trames. Les raisons sont diverses : le récepteur est plus lent que l'émetteur, le récepteur n'a pas assez de mémoire, le récepteur est pris par d'autres communications (il faut signaler qu'une station peut être en communication avec plusieurs autres stations en même temps)... L'objectif du contrôle de flux est d'assujettir le fonctionnement de la source aux capacités de réception du destinataire afin d'éviter la perte de trames. En d'autres termes, le contrôle de flux sur un liaison logique permet d'éviter le débordement du destinataire et donc la perte de données.

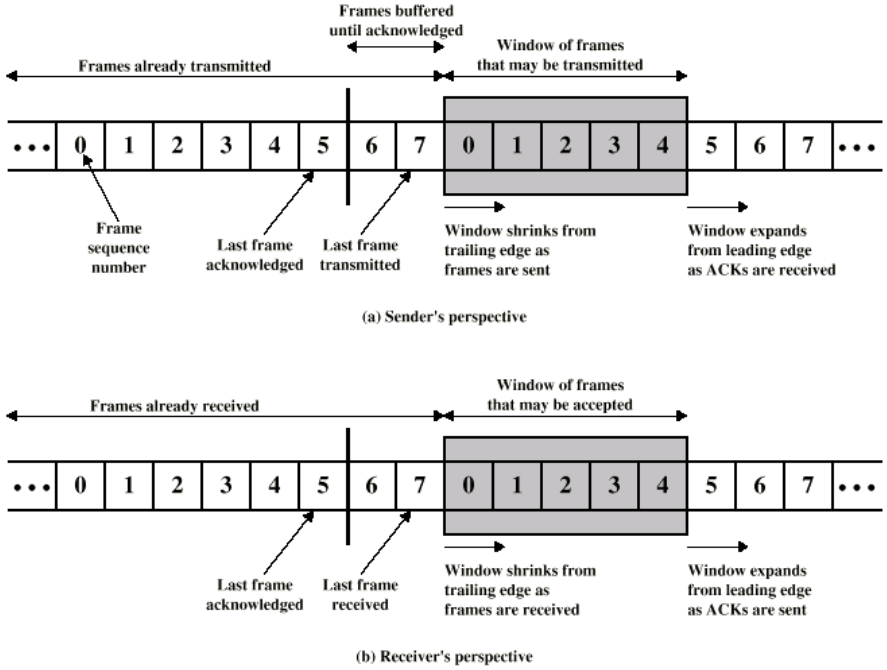
b) Techniques de contrôle de flux

Le contrôle de flux au niveau liaison logique se fait selon deux techniques : la technique **stop and wait** et la technique **avec anticipation** :

- Technique *stop and wait* : l'émetteur envoie une trame et se bloque en attente d'un acquittement. Si l'acquittement lui parvient au bout d'un délai fixé (la valeur de ce délai dépend du débit de transmission, de la distance entre source et destination et d'autres critères), il peut envoyer la trame suivante, sinon il renvoie la même trame jusqu'à ce qu'il reçoive l'acquittement positif ou bien jusqu'à ce que le nombre de retransmissions soit épuisé. On notera que cette technique génère une surcharge du réseau due aux trames d'acquittement, ce qui a comme conséquence une baisse du rendement du réseau.
- Technique *avec anticipation* (dite aussi avec *fenêtre coulissante*) : l'émetteur peut envoyer jusqu'à 2^n (*n* est le nombre de bits utilisés pour coder le numéro de trame, il est généralement égal à 8) trames avant d'attendre un acquittement. Les trames sont numérotées de *0* à $2^n - 1$, (les numéros sont répétés modulo 2^n). Lorsqu'une source reçoit un acquittement contenant le numéro *v* cela signifie que toutes les trames numérotées de *v'* (*v'* c'est le numéro contenu dans la trame d'acquittement précédente ou *0* s'il n'y a pas eu précédemment d'acquittement) jusqu'à à *v-1* ont été bien reçues et que le destinataire acquittera

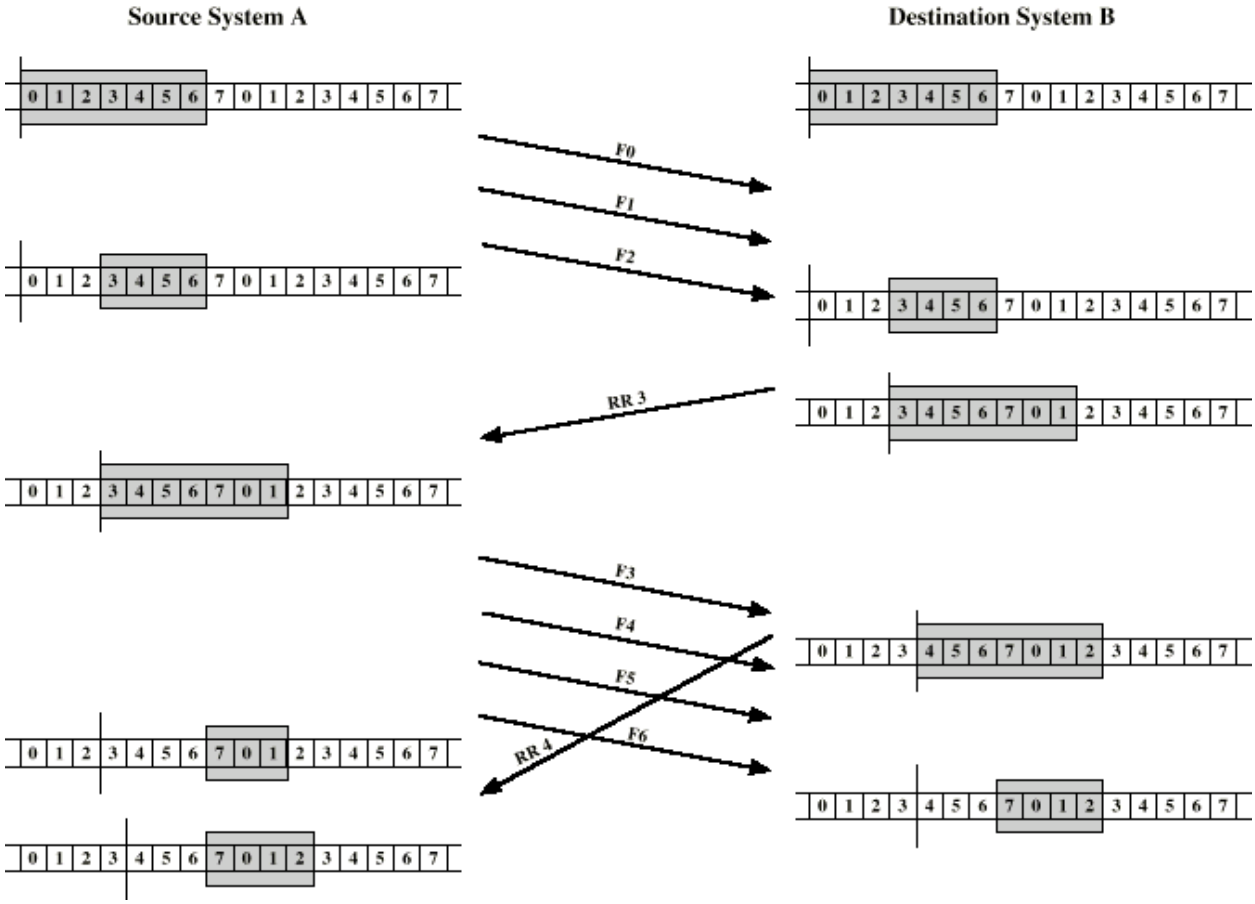
prochainement à partir du numéro *v*. Attention lorsqu'une source reçoit un acquittement pour 5 trames par exemple alors qu'elle en a envoyées 10, cela ne signifie ni que les 5 trames restantes sont bien reçues ni qu'elles sont perdues. Un acquittement ne concerne que les trames reçues et validées par le récepteur. Lorsqu'une trame est reçue au niveau de la couche liaison de données, elle peut être mise en file d'attente sans traitement, car le récepteur est occupé par d'autres activités (traitement de requêtes, ouvertures de fichiers, calculs, affichage...). Ainsi, les différentes activités effectuées par le récepteur, peuvent le conduire à acquitter un sous ensemble des trames qu'il a reçues. En général, l'instant où le récepteur acquitte une trame reçue est indéterministe. Cependant, le récepteur doit respecter le temps maximum entre l'instant de réception d'une trame et l'acquittement de celle-ci, de plus le récepteur doit savoir que s'il n'acquitte pas les trames reçues, la source peut être bloquée et qu'elle sera amenée à retransmettre le cas échéant.

La figure suivante montre les informations gérées par la source et le récepteur pour gérer les acquittements.



Fenêtre vue par la source et par la destination

La figure suivante montre un exemple d'enchaînement d'échange de trames de données (F0, F1 ...F7) et d'acquittement (RRx = « ready to receive frame number x »)



Exemple d'enchaînement

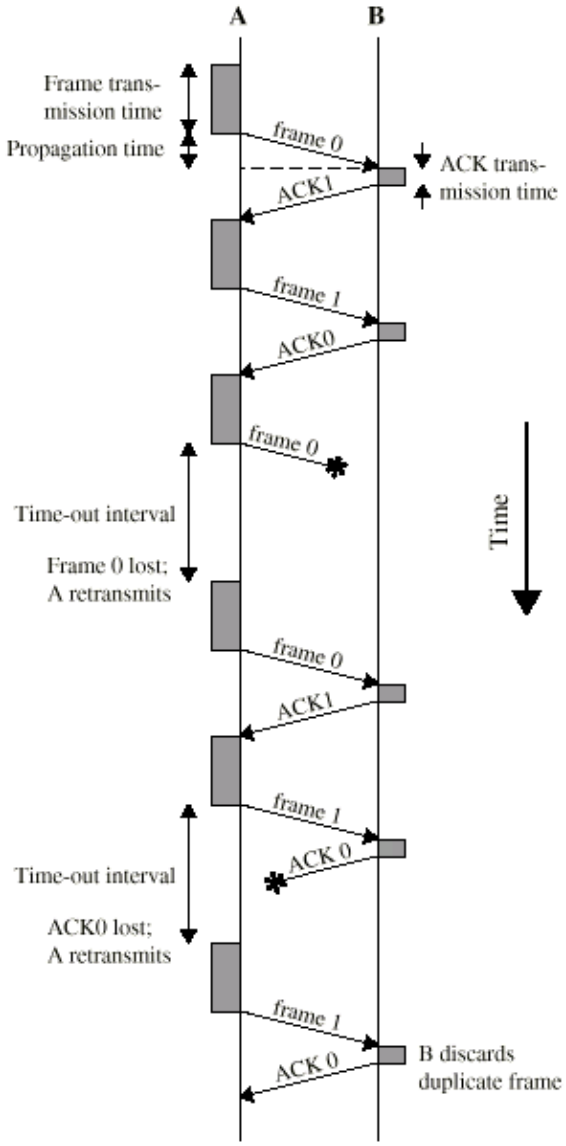
c) Contrôle d'erreur

L'objectif du contrôle d'erreur est de s'assurer qu'aucune trame ne se perde. Lorsqu'une trame est détectée comme erronée, elle est soit retransmise par la source soit corrigée par le destinataire. Lorsqu'une trame est perdue, la numérotation de trame doit permettre de détecter cette perte en remarquant qu'il y a un 'trou' dans la numérotation. Dans la couche liaison de données, il n'y a que de la retransmission automatique (ARQ : Automatic repeat request).

Il existe deux techniques de retransmission qui fonctionnent conjointement avec les techniques de contrôle de flux : technique stop and wait et technique avec anticipation.

- Technique *stop and wait* : la source émet une trame et attend un acquittement. Lorsque la trame reçue est correcte, un acquittement positif (Ack : acknowledgement) est émis par le destinataire. Quand la trame est erronée, un acquittement négatif (Nack : Negative acknowledgement) est émis par le destinataire. En cas de Nack, la source retransmet sa trame. Pour ne pas attendre indéfiniment l'acquittement en cas de perte de celui-ci ou de la trame de données émise, la source arme un temporisateur. Quand se dernier se déclenche, la source retransmet. A noter que si la source retransmet suite à une perte d'acquittement, le destinataire va se rendre compte (en utilisant le numéro de séquence d'émission) que la trame qu'il reçoit est une copie de la trame précédente. Dans ce cas, il transmet un acquittement (avec le même numéro que celui qu'il a précédemment envoyé) mais ignore les données contenues dans la trame. Pour gérer ces aspects, on numérote de manière alternative 0 et 1 (c'est-à-dire en modulo 2) les trames de données et les trames d'acquittement. Les trames de données seront notées F0, F1, F0, F1, ... (attention même si deux trames de données portent le même numéro de séquence, elles contiennent des données qui sont différentes). Les acquittements sont notés ACK0, ACK1, NACK0, NACK1,... Dans beaucoup de cas, on n'utilise que des acquittements positifs. Dans ces cas, s'il y a une erreur, le destinataire rejette la trame et

ne signale rien à la source. L'absence d'acquittement (détectée par la source à l'aide d'un temporisateur armé juste après la transmission de la trame de données) est équivalente à une erreur ou perte. La figure suivante donne un exemple d'enchaînement en utilisant la technique stop and wait



Exemple d'enchaînement avec la technique Stop and wait

- Technique *avec anticipation* : la source émet un bloc de plusieurs trames (sachant que le nombre maximum de trames en attente d'acquittement ne doit pas dépasser la taille de la fenêtre d'anticipation). Quand une trame est détectée erronée ou perdue, le destinataire peut appliquer l'une des deux stratégies (sachant qu'une seule stratégie est applicable pour un même réseau) :

  + *Rejet en bloc* : le destinataire indique un numéro à partir duquel toutes les trames sont rejetées et dans ce cas la source retransmet toutes les trames (même si certaines d'entre elles n'ont pas été touchées par les erreurs lors de la tentative précédente). Cette stratégie est simple mais elle peut conduire à une surcharge du réseau.

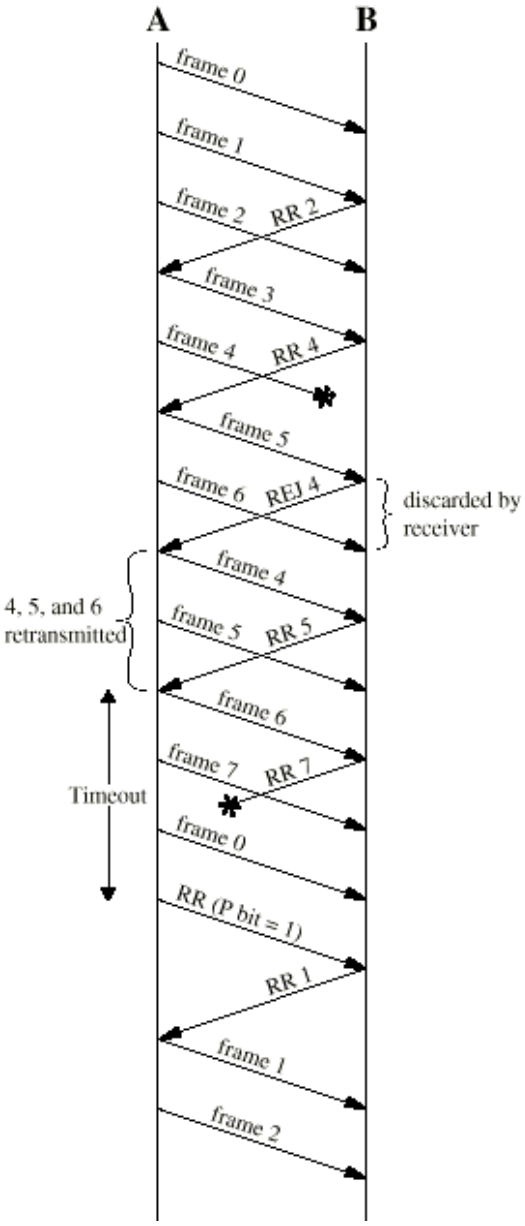
  + *Rejet sélectif* : le destinataire indique le numéro de la trame erronée ou perdue. Dans ce cas, la source retransmet seulement la trame signalée. Cette stratégie permet d'utiliser de manière plus efficace la bande passante, mais elle est plus complexe à implanter car elle nécessite des structures de données plus complexes que celles requises par la première stratégie.

Les deux situations suivantes sont rares, mais à considérer lors de l'implantation de stratégie avec anticipation :

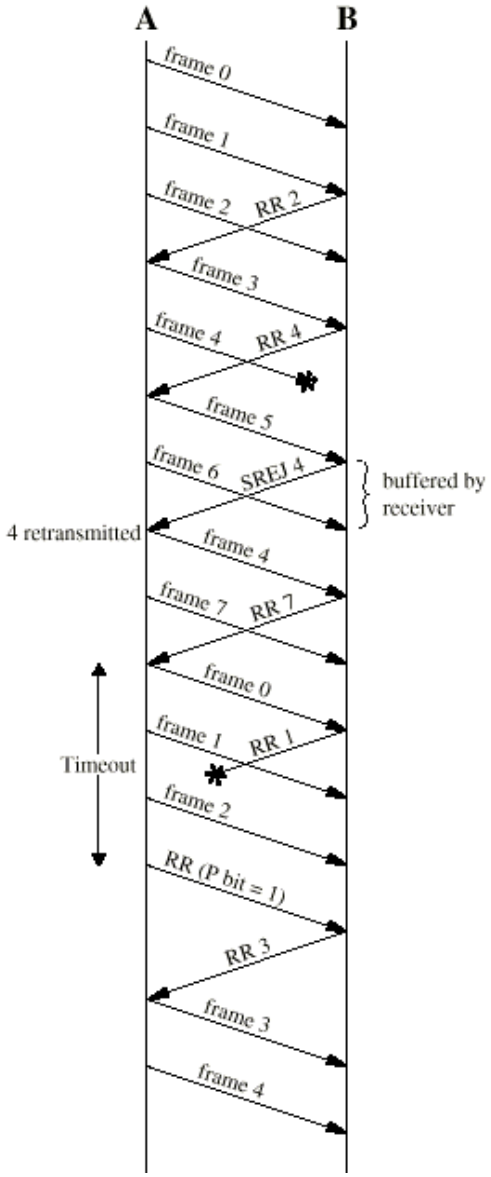
+ Situation 1 : Lorsque la dernière trame émise est perdue (et que la source ne va plus envoyer de trame de données), le destinataire ne fait rien (car il n'a pas vu cette dernière trame). Dans ce cas, le réveil de la source se déclenche, ensuite la source peut soit envoyer un acquittement avec le bit P=1 (dans ce cas le destinataire comprend qu'il faut que lui aussi transmette un acquittement et comme il envoie un acquittement qui n'intègre pas la dernière trame, la source comprend qu'il faut retransmettre la trame), soit il retransmet sa dernière trame.

+ Situation 2 : Lorsqu'une trame d'acquittement (cet Ack peut correspondre à toutes les trames en attente d'acquittement ou à certaines d'entre elles) est perdue, la source retransmet (après déclenchement de son réveil) toutes les trames pour lesquelles elle attendait un acquittement (alors qu'elle peut recevoir un acquittement pour les trames non encore acquittées dans l'acquittement précédent). Dans ce cas à partir des numéros de trames, la source et la destination peuvent retrouver un état cohérent où les trames de données en double sont éliminées, sinon elles réinitialisent la connexion en cas d'impossibilité de recouvrement.

Les figures suivantes montrent deux enchaînements en utilisant la technique de la fenêtre d'anticipation avec rejet en bloc ou sélectif.



Exemple d'enchaînement avec la technique d'anticipation (avec rejet en bloc)



Exemple d'enchaînement avec la technique d'anticipation (avec rejet sélectif)

Dans les deux techniques de retransmission, le nombre de retransmissions est borné. On fixe un seuil pour le nombre de retransmissions, car si le nombre d'erreurs ou de perte pour une même trame dépasse ce seuil c'est qu'il y a de graves problèmes sur la ligne de transmission (par exemple, coupure de la ligne ou parasites persistants sur la ligne) et cela ne sert à rien de continuer à transmettre de ces conditions.

III.8. Procédure HDLC

L'un des protocoles les plus répandus dans le domaine des réseaux, au niveau de la couche 2, est celui connu par le nom de HDLC (High Data Link Control). D'ailleurs, la quasi-totalité des protocoles LLC utilisés dans la pratique sont des sous-ensembles de HDLC.

HDLC distingue deux types de stations :

- *station primaire* : c'est la station qui initie les échanges. Les trames qu'elle envoie sont dites trames de commande. La station primaire maintient un état séparé pour chaque connexion logique avec une station secondaire.
- *station secondaire* : envoie des trames de réponse quand elle est sollicitée par la station primaire.

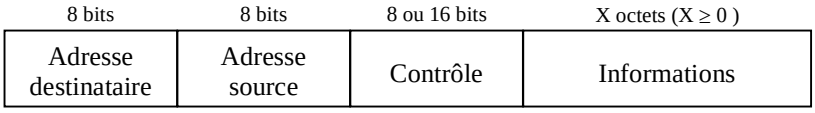
Par ailleurs, HDLC distingue deux modes de configurations de réseau :

- *Mode non balancé* (ou asymétrique) : il y a une seule station primaire et une ou plusieurs stations secondaires.
- *Mode balancé* (ou symétrique) : toutes les stations ont le même statut. En général, ce mode est plus approprié que le mode précédent pour la gestion de liaisons full duplex ou semi duplex. Dans ce mode, on identifie un sous-mode dit mode asynchrone balancé (ABM : asynchronous Balanced Mode) où chaque correspondant peut commencer les échanges sans permission de l'autre. C'est ce mode ABM qui est le plus utilisé dans les réseaux existants.

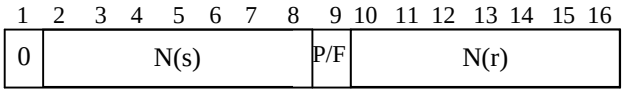
III.9. Format des trames LLC

Il existe trois types de trames échangées au niveau LLC :

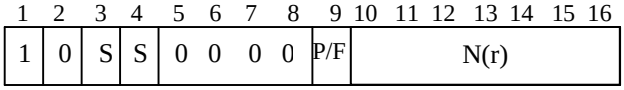
- Trames de type *I* (Information) qui véhiculent des données utilisateur et des acquittements
- Trames de type *S* (Supervision) qui permettent principalement de véhiculer des acquittements.
- Trame de type *U* (Unnumbered) qui permettent d'établir et de rompre les connexions.



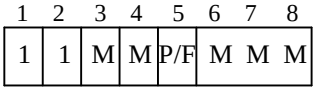
Format général de trame LLC.



Format du champ de contrôle des trames de type I (Information).



Format du champ de contrôle des trames de type S (Supervision).



Format du champ de contrôle des trames de type U (non numérotée).

Les trames d'information (type I) et de supervision (type S) contiennent des numéros de séquence :

- *N(s)* : numéro de séquence en émission qui spécifie le numéro de séquence d'émission de la trame de données (attention les trames de type *S* ou *U* ne sont pas numérotées à l'émission). Ce numéro est incrémenté de 0 à 127 (modulo 128).
- *N(r)* : numéro de séquence en réception qui spécifie le numéro de séquence en réception. Ce numéro indique au correspondant que l'émetteur de la trame est prêt à recevoir la trame de données numéro N(r), cela signifie qu'il a bien reçu toutes les trames de données qui précédent ce numéro. Ce numéro est incrémenté de 0 à 127 (modulo 128).

Le bit *P/F* dans les trames de supervision et non numérotées est dit bit *P* (polling) dans la trame du demandeur (d'établissement de connexion par exemple) et bit *F* (Final) dans la trame de celui qui répond (par exemple celui qui accepte une demande de connexion).

Dans les trames de supervision, le champ *SS* peut prendre les valeurs suivantes :

- *00* : désigne une trame *RR* (Ready to Receive) qui est une trame envoyée par une station pour indiquer à son correspondant qu'elle est prête à recevoir la trame numéro N(r). Il s'agit donc d'une trame qui véhicule un acquittement (mais pas de données).
- *10* : désigne une trame *RNR* (Not Ready to Receive) qui est une trame envoyée par une station pour indiquer à son correspondant qu'elle a bien reçu les trames de données jusqu'au numéro *N(r) – 1*, mais (pour des raisons internes) elle ne peut plus momentanément recevoir des données. Lorsqu'elle devient prête à recevoir des données, elle enverra une trame *RR*.
- *01 :* désigne une trame *REJ* (Reject) qui indique que les trames de données sont rejetées à partir du numéro *N(r)*. L'émetteur des trames de données rejetées (à cause d'erreurs de transmission) devra retransmettre ses trames.
- *11 :* désigne une trame *SREJ* (Selective Reject) qui indique que la trames de données ayant le numéro *N(r)* est rejetée. L'émetteur devra retransmettre la trame de données rejetée.

Dans les trames non numérotées, le champ *MM* peut prendre les valeurs suivantes :

- *11P110* : désigne une trame *SABME* (Set Asynchronous Balanced Mode Extended). La trame *SABME* est utilisée pour demander l'établissement de connexion.
- *00P010* : désigne une trame *DISC* (Disconnect). La trame *DISC* est envoyée par une station pour informer son correspondant qu'elle arrête la connexion.
- *00F010 :* désigne une trame *UA* (Unnumbered Acknowledgement). La trame *UA* est envoyée par une station pour indiquer à son correspondant qu'elle accepte sa demande de connexion ou de déconnexion.
- *11F001 :* indique une trame *FRMR* (FRaMe Reject). La trame *FRMR* est envoyée par une station pour indiquer à son correspondant un rejet de trame (car le champ contrôle n'est pas valide, la longueur de données dépasse la longueur maximale, un champ *N(r)* non valide, etc.).

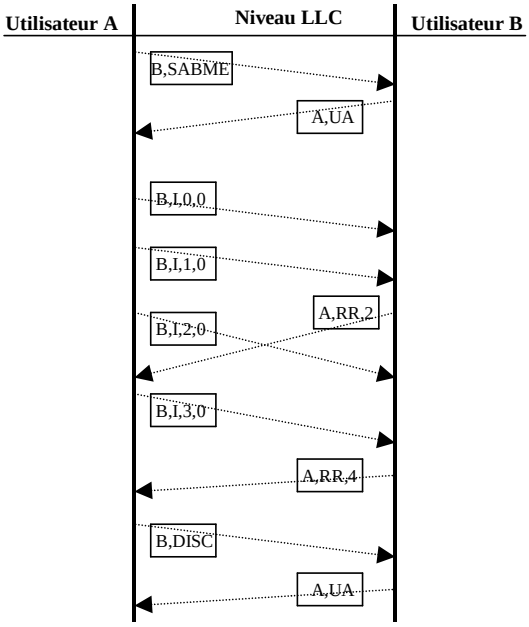
Exemples d'échange

Dans les enchaînements de trames montrés par les figures suivantes, les notations suivantes sont utilisées :

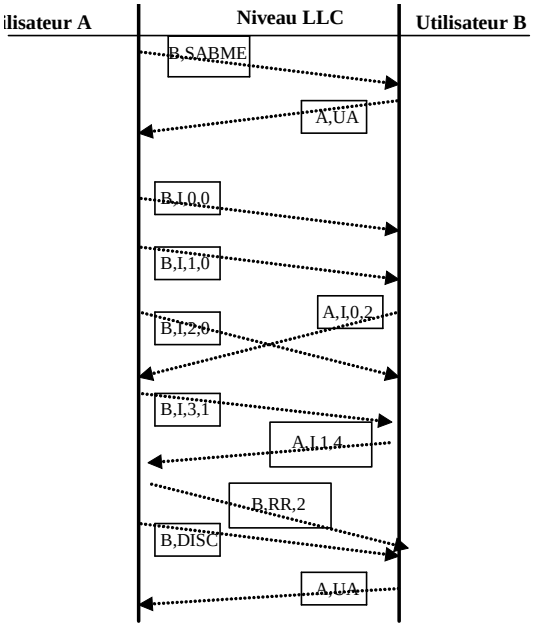
- Une trame sous la forme ***X,I,s,r*** désigne une trame *I* envoyée vers la station *X* et qui contient un numéro de séquence en émission égal à *s* et un numéro de séquence en réception égal à *r*.
- Une trame sous la forme ***X,R,r*** désigne une trame *S* envoyée vers la station *X* avec *R* qui peut être égal à *RR, RNR* ou *REJ*. *r* désigne un numéro de séquence en réception.

- Une trame sous la forme ***X,U*** désigne une trame *U* envoyée vers la station *X* avec *U* qui peut être égal à *UA*, *FRMR* ou *DISC*.

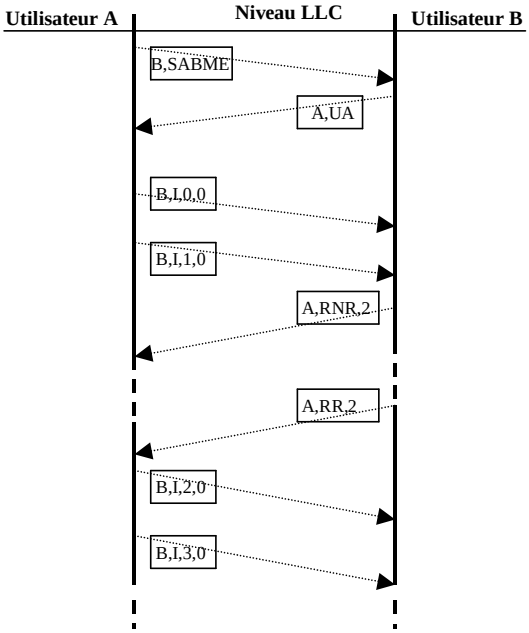
On notera que sur les figures suivantes, les primitives de service qui conduisent à la création des trames échangées sont omises pour ne pas surcharger les figures.



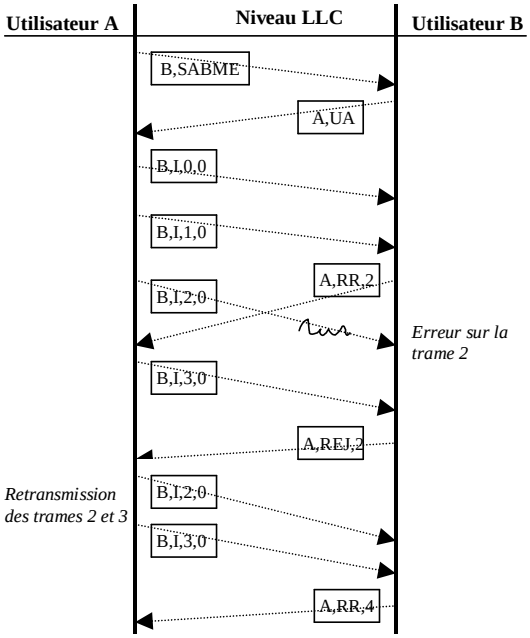
Cas 1 : Echange de données unidirectionnel.



Cas 2 : Echange simultané de données



Cas 3 : Echange de données avec arrêt et reprise.



Cas 4 : Echange de données avec des erreurs.

Exemples d'enchaînement de trames de niveau Liaison de données.

Exercices

Exercice 1
Reprendre l'exemple de CDMA présenté dans le cours.

Exercice 2
On considère trois stations A, B et C qui s'échangent des données de la manière suivante :

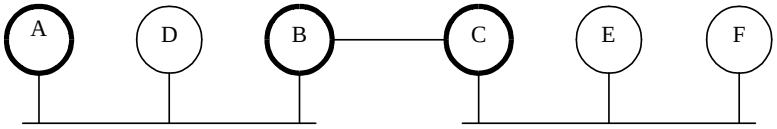
1. La station A ouvre une connexion avec la station B, puis une autre connexion avec la station C.
2. La station B accepte la connexion et envoie sa réponse.
3. La station C accepte la connexion et envoie sa réponse.
4. La station A envoie sa première trame de données vers B.
5. La station A envoie sa deuxième trame de données vers B.
6. La station C envoie sa première trame de données vers A
7. La station C envoie sa deuxième trame de données vers A.
8. La station A envoie sa première trame de données vers C.
9. La station A envoie sa deuxième trame de données vers C.
10. La station A envoie sa troisième trame de données vers C qui sera endommagée par des erreurs.
11. La station B envoie sa première trame de données vers A.
12. La station A envoie sa troisième trame de données vers B qui sera endommagée par des erreurs.
13. La station A envoie sa quatrième trame de données vers C.
14. Les trois stations n'ont plus de données à transmettre. Elles doivent acquitter toutes les trames en attente de réception. Cette étape nécessite plusieurs trames d'acquittement et de retransmission.
15. La station A ferme la connexion avec la station C.
16. La station B ferme la connexion avec la station A.

Proposer un diagramme d'échange de trames (selon le style vu en cours) en respectant la chronologie des échanges ci-dessus.

Exercice 3
Soit un réseau dans lequel trois stations A, B et C ne sont pas reliées directement (voir figure suivante). La station A souhaite émettre 5 trames vers la station C et la station C souhaite émettre 6 trames vers la station A. La station B n'a pas de données à émettre mais elle relaye les trames des deux autres stations. Pour communiquer les trois stations sont obligées d'établir, deux à deux, des connexions de niveau liaison de données. Les mécanismes de contrôle de flux et d'erreurs utilisés sur les deux connexions sont les suivants :

- la connexion entre les stations A et B fonctionne en mode acquittement trame par trame (technique « stop and wait»)
- la connexion entre les stations B et C fonctionne avec une fenêtre d'anticipation de taille égale à 3.

On suppose qu'une trame de données sur deux est affectée par les erreurs de transmission sur la ligne entre A et B et une trame sur trois sur la ligne entre B et C.
Donner le diagramme d'appels de primitives de service de liaison de données et d'échanges de trames pour que les stations A et C arrivent à émettre et recevoir correctement leurs trames.



Topologie du réseau étudié

Exercice 4 (Etude d'implantation de protocoles)

On veut définir et programmer des protocoles de communication de niveau liaison de données. Les protocoles à étudier sont relativement simples par rapport à ceux qui existent dans la pratique.
On suppose que l'on dispose des fonctions suivantes :

`Prendre_de_la_couche_reseau(AS, AD, T):` permet de chercher les données à transmettre qui se trouvent dans une zone mémoire fixée dans la couche réseau. Au retour, les données sont placées dans `T`, sachant que les deux premiers octets du tampon `T` contiennent la longueur des données. `AS` et `AD` sont deux entiers qui contiennent les adresses source et destination.

`Construire_trame(AS, AD, T, Trame)` : à partir des données qui se trouvent dans `T` et des adresses `AS` et `AD`, construire une trame en respectant le format de trame (adresse source, adresse destination, longueur des données, données, CRC). Le format des trames dépend du protocole implanté. `Trame` contient la trame à émettre. On suppose que le CRC est un simple bit de parité sur tous les octets qui composent la trame.

`Passer_vers_couche_physique(Trame)` : transmettre bit par bit une trame.

`Prendre_de_la_couche_physique(Trame) :` permet de chercher les données reçues qui se trouvent dans une zone mémoire fixée dans la couche physique. Au retour, la trame est placée dans `Trame`.

`Récupérer_données(Trame, T, AS, AD) :` permet d'extraire les données contenues dans la trame et les placer dans le tampon `T` (les deux premiers octets du tampon `T` contiennent la longueur des données reçues). Les paramètres `AS` et `AD` reçoivent les adresses source et destination de la trame.

`Passer_vers_couche_réseau(TamponRéception, ASR, ADR) :` permet de placer les données reçues dans une zone mémoire fixée dans la couche réseau. `ASR` et `ADR` contiennent les adresses source et destination de la trame reçue par la couche liaison de données.

`Attendre(Sig1, Sig2, ... Sign) :` attendre qu'un signal parmi *n* soit signalé.

`Signale(Sig) :` fonction booléenne qui indique si un signal `Sig` a été signalé.

`Effacer(Sig) :` met le signal `Sig` dans l'état non arrivé (efface une occurrence du signal `Sig`).

Protocole 1

Le premier protocole est utopique et est défini de la manière suivante :
- les deux correspondants (émetteur et récepteur) sont toujours opérationnels
- on dispose d'une mémoire infinie chez l'émetteur et le récepteur
- il n'y a jamais d'erreurs de transmission
- les données circulent dans un seul sens
- la couche réseau de l'émetteur est toujours prête à envoyer de nouvelles données
- la couche réseau du récepteur est toujours prête à recevoir de nouvelles données

Avec les hypothèses précédentes le premier protocole ne nécessite ni contrôle d'erreurs, ni contrôle de flux.
On peut décrire les processus émetteur et récepteur de la manière suivante :

```
Emetteur 1 :
     Tampon  TamponEmission /* Tampon pour stocker les données à émettre */
     TTrame  Trame /* structure de données pour stocker la trame à émettre */
     Tant que (1==1)
       {  Prendre_de_la_couche_reseau(TamponEmission, AS, AD)
```

```
          Construire_trame(AS, AD, TamponEmission, Trame)
          Passer_vers_couche_physique(Trame)
        }


Récepteur 1 :
     Tampon  TamponRéception /* Tampon pour stocker les données reçues */
     TTrame  Trame /* structure de données pour stocker la trame reçue */
     Signal SigRecPHY  /* le signal SigRecPHY est notifié par la couche physique
                          quand une trame est reçue. Il s'agit d'une manière
                          d'implanter la notion d'indication de service */
     Tant que (1==1)
       {  Attendre (SigRecPHY)
          Prendre_de_la_couche_physique(Trame)
          Effacer(SigRecPhy)
          Récupérer_données(Trame, TamponRéception, AS, AD)
          Passer_vers_couche_réseau(TamponRéception, AS, AD)
        }
```

Protocoles à étudier

Protocole 2 : On ne suppose plus que la couche réseau est toujours prête :

- Du côté émetteur, la couche réseau notifie un signal `SigEmis` quand il y a des données à émettre.
- Du côté récepteur, la couche réseau notifie un signal `SigRec` quand elle a fini de traiter la dernière trame qui lui a été transmise.

Protocole 3 : protocole avec gestion des erreurs de transmission

- L'émetteur envoie une trame et attend un acquittement. Pour ne pas attendre indéfiniment un acquittement, l'émetteur arme un temporisateur (en utilisant une fonction `tempo(dt)`). On suppose que la fonction `Tempo` renvoie un signal `SigTempo` quand la temporisation arrive à échéance.
- Si l'émetteur reçoit un acquittement avant l'expiration du délai, il peut envoyer la prochaine trame, sinon il retransmet une nouvelle fois la trame. Si le nombre de retransmissions dépasse `10`, la communication est arrêtée.
- On suppose que le récepteur n'envoie que des acquittements positifs.

Protocole 4 : protocole avec contrôle de flux

- Le récepteur ne dispose pas d'une mémoire infinie et l'émetteur ne peut pas avoir plus de *n* (*n* est fixé) trames en attente d'acquittement.
- Lorsque le récepteur détecte une erreur de numérotation (car une ou plusieurs trames ont été rejetées à cause d'erreurs de transmission), il rejette toutes les trames qui lui arrivent tant qu'il n'a pas reçu la trame avec le numéro de séquence attendu.

Chapitre 6

Couche Réseau

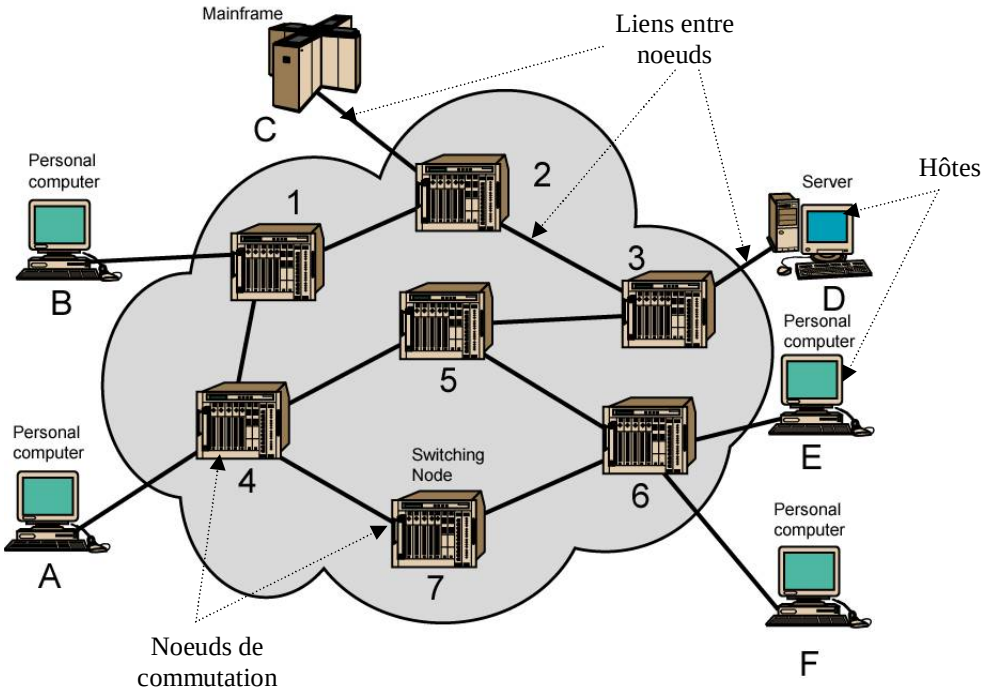
I. Introduction

Objectifs

Le chemin entre une source et une destination, dans un réseau, peut passer par plusieurs noeuds intermédiaires. Ces noeuds constituent un chemin (une route) entre les deux correspondants. Un chemin est constitué de noeuds et des liens (ou liaisons physiques) entre ces noeuds. Le rôle de la couche Réseau est :

1) Sélectionner le meilleur chemin entre une source et une destination. Changer de chemin si le chemin courant n'est plus possible (à cause d'un tronçon en panne) ou parce qu'un meilleur chemin est possible.
2) Acheminer les paquets d'un bout à l'autre le long du chemin choisi.

Par exemple, sur la figure suivante, le chemin entre le 'mainframe' (gros calculateur) C et l'ordinateur D peut être établi de la manière suivante {(C, 2), (2, 3), (3, D)}, qui est le plus court chemin si on compte le nombre de sauts ou {(C, 2), (2, 1), (1, 4), (4, 5), (5, 3), (3, D)} si le lien (2, 3) est en panne. Attention, la notion de meilleur chemin a différentes interprétations selon le critère à optimiser (nombre de sauts, prix à payer, débit offert, temps de transit...).



Exemple de réseau de commutation

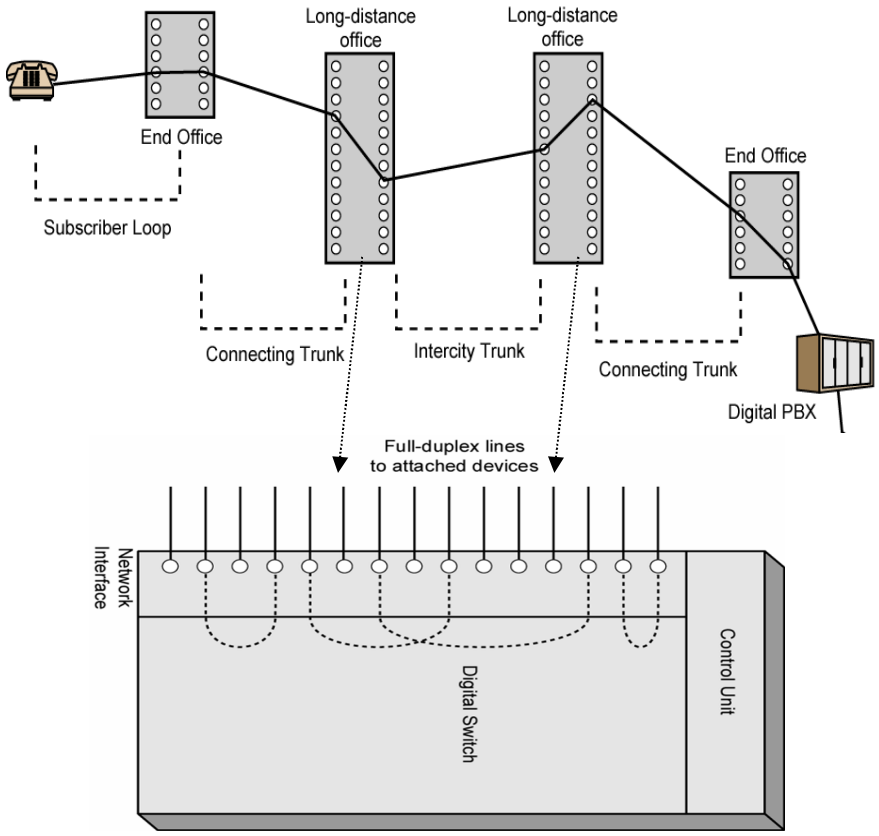
II. Commutation de circuits vs commutation de paquets

On parle de commutation (switching) pour désigner la fonction qui consiste à faire passer un bloc de données d'un noeud intermédiaire à un autre. Deux technologies de commutation sont possibles : commutation de circuit et commutation de paquet. Aujourd'hui la commutation de circuits (jadis utilisée en téléphonie) est en passe de disparaître.

II.1. Commutation de circuits

C'est une technologie utilisée en téléphonie. La route entre deux abonnés du téléphone est établie tronçon (on dit aussi segment) par tronçon. Des switches électromagnétiques sont utilisés dans les commutateurs et centraux téléphoniques pour raccorder les tronçons. Comme le montre la figure suivante, les tronçons sont physiquement connectés et réservés pendant toute la durée de communication (même si les deux correspondants ne parlent pas). L'établissement de connexion réseau se fait de manière physique en raccordant les différents tronçons qui composent la route entre la source et destination.
Cette solution est peu efficace surtout pour la transmission de données (qui sont souvent de nature sporadique) mais elle bien adaptée à un environnement de télécommunication peu informatisé. Depuis quelques années, les centraux à commutation de circuits sont de plus en plus remplacés par les centraux à commutation de paquets car elle est plus flexible.

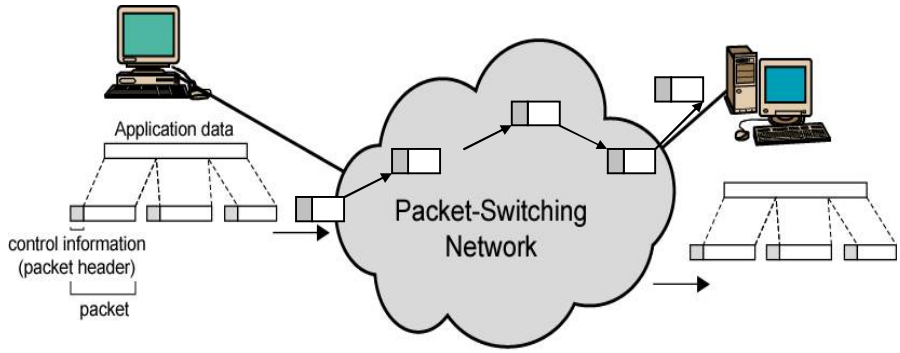
Pour remédier aux insuffisantes de la commutation de circuits par réservation durant toute la communication, des technologies basées sur la commutation temporelle de circuits peuvent être utilisées. Dans ces technologies, un tronçon est alloué pendant un quantum de temps à une source ensuite il est affecté à une autre source. Le temps pendant lequel le tronçon n'est pas affecté à une source ne doit pas remettre en cause la qualité de sa transmission (on fait du temps partagé entre les abonnés du téléphone). Par ailleurs, des techniques sophistiquées peuvent être utilisées pour réduire les temps d'établissement de circuits en pré-connectant certaines parties des chemins, en utilisant plusieurs niveaux de tronçons...



Principe de la commutation de circuits

II.2. Commutation de paquets

La commutation de paquets est une technologie basée sur l'informatique dans laquelle les connexions entre abonnés utilisent les mêmes lignes physiques. Lorsqu'un paquet arrive dans un commutateur, il est stocké en mémoire (en file d'attente) tant que la voie de sortie dont il a besoin est occupée ; lorsque celle-ci devient libre le paquet est envoyé vers le noeud suivant et de proche en proche, ce paquet atteint sa destination. C'est un mécanisme de partage de lignes plus efficace mais qui fait appel à beaucoup de traitements (gestion de tables de routage, stockage des paquets, reconnaissance des adresses de destination, test de la disponibilité de voie de sortie, gestion de file d'attente...) qui ne sont possibles que grâce à l'utilisation de processeurs. Parfois la commutation de paquets peut conduire à la perte de paquets suite au débordement de file d'attente dans les nœuds de commutation. Ce problème ne se pose pas pour la commutation de circuits car tout est réservé à l'avance.



Principe de la commutation de paquets

Circuit virtuel : une connexion de niveau Réseau est parfois appelée Circuit virtuel. Le terme de *Circuit virtue*l a été introduit par les opérateurs de télécommunications pour faire l'analogie entre la commutation de circuit (où le circuit a une existence physique puisqu'il est composé de segments matériels raccordés les uns aux autres) et la commutation de paquet où le circuit est logique ou virtuel (obtenu en utilisant les files d'attente et liens des routeurs traversés pas le chemin entre la source et destination).

Remarque : dans la suite nous nous intéressons uniquement à la commutation de paquets.

II.3 Mode orienté connexion vs mode non orienté connexion

La communication entre une source est une destination peut être gérée selon un des deux modes : orienté connexion et non orienté connexion.

1. Non connecté ou non orienté connexion (communauté Internet) :

- Pas d'établissement de circuit virtuel (d'où un gain de temps).
- Service non fiable : pas de garantie de livraison de paquets.
- Pas de connexion, pas de contrôle de flux.
- Les PDU échangés sont appelés datagrammes.
- Tous les datagrammes, entre une source et une destination, n'empruntent pas le même chemin.
- Les datagrammes peuvent arriver dans le désordre (il faut les ordonner).
- L'utilisateur de la couche réseau doit tout contrôler lui-même.

2. Connecté ou orienté connexion (opérateurs Télécom : service payant) :

- Nécessité d'établissement de circuit virtuel explicite (qui prend un certain temps)
- Service fiable : garantie de livraison de paquets.
- Garantie que les paquets sont livrés dans leur ordre d'émission.
- Contrôle de flux assuré.
- Trafic urgent privilégié.
- Tous les paquets d'une source vers une même destination empruntent le même chemin sauf en cas de changement de chemin (changement peu fréquent).

II.4. Signalisation

On appelle signalisation toutes les fonctions nécessaires à la gestion des connexions autres que celles dédiées à la transmission de données de l'utilisateur. Il s'agit de tous les messages nécessaires à l'établissement, contrôle et maintenance des connexions. Ces messages de service peuvent utiliser les mêmes canaux que les données de l'utilisateur (on parle dans ce cas d'une signalisation *dans la bande* – inband) ou utiliser des canaux séparés (on parle dans ce cas d'une signalisation *hors bande* – out of band). A titre indicatif, actuellement, la couche Réseau des opérateurs de réseaux utilisent plutôt la signalisation hors bande (pour que leurs messages n'affectent pas et ne soient pas affectés par les messages des abonnés) et le réseau Internet utilise une signalisation dans la bande.

III. Algorithmes de routage

L'algorithme de routage est la composante de la couche réseau qui a la responsabilité de sélectionner le chemin par lequel un paquet doit transiter. On cherche souvent un chemin "optimal". Cependant, la notion d'optimal peut prendre plusieurs sens : le chemin le plus court en temps de transfert, le chemin avec le moins de routeurs, le chemin ne passant pas par tel ou tel routeur, …

Dans le cas où la couche réseau offre un service connecté, tous les paquets suivent le même chemin (sauf en cas d'anomalie et de reconfiguration). Dans le cas où elle utilise un service non connecté, les datagrammes suivent (a priori) des chemins différents.

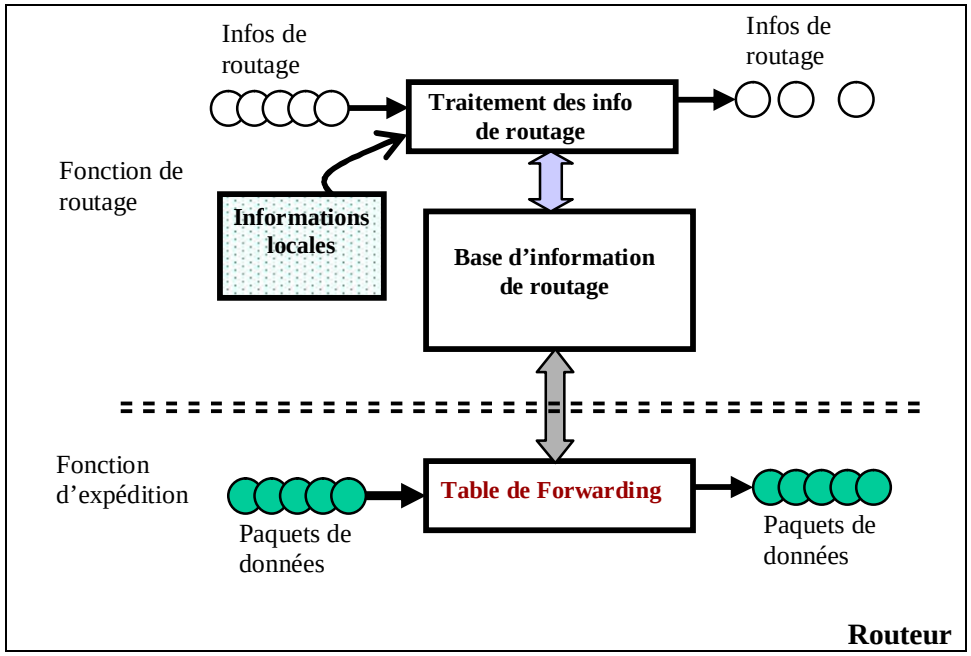
Les informations qui décrivent les chemins entre les différents noeuds d'un réseau sont structurées sous forme de tables dites ***tables de routage***.

III.1. Routage vs Expédition ('Routing vs Forwarding')

Pour que les paquets puissent 'voyager' de noeud en noeud jusqu'à leur destination, chaque noeud dispose de deux fonctions complémentaires :

- une ***fonction de routage*** : qui construit la table de routage à partir des informations échangées entre les noeuds de manière à optimiser le chemin pour transporter les paquets. La fonction de routage est exécutée de temps en temps pour mettre à jour la table de route suite des changements d'état significatifs dans le réseau. Dans le cas d'un routage fixe, cette fonction peut être absente et l'administrateur de réseau configure manuellement la table de routage.
- une ***fonction d'expédition*** ('*forwarding*') : pour chaque paquet reçu, cette fonction consulte la table de routage pour savoir vers quel noeud suivant ('*next hop*') il faut expédier le paquet entrant en fonction de son adresse source et destination. C'est une fonction dont l'exécution est très fréquente et donc son implantation doit être étudiée minutieusement (parfois on la réalise de manière câblée).

Souvent, les gens confondent ces deux fonctions. Encore plus grave, certains considèrent que l'expédition c'est le routage. Par ailleurs, les deux termes *table de routage* ('routing table') et *table d'expédition* ('forwarding table') sont souvent utilisés de manière interchangeable.



Routage et Expédition

III.2. Types de routage

Plusieurs algorithmes de routage ont été définis. On peut classer ces algorithmes selon différents points de vue. En particulier, la classification suivante est couramment retenue :

- Routage fixe,
- Routage par inondation,
- Routage aléatoire,
- Routage adaptatif (complètement adaptatif ou semi adaptatif).

Une autre classification consiste à distinguer les algorithmes de routage selon l'endroit où sont prises les décisions :

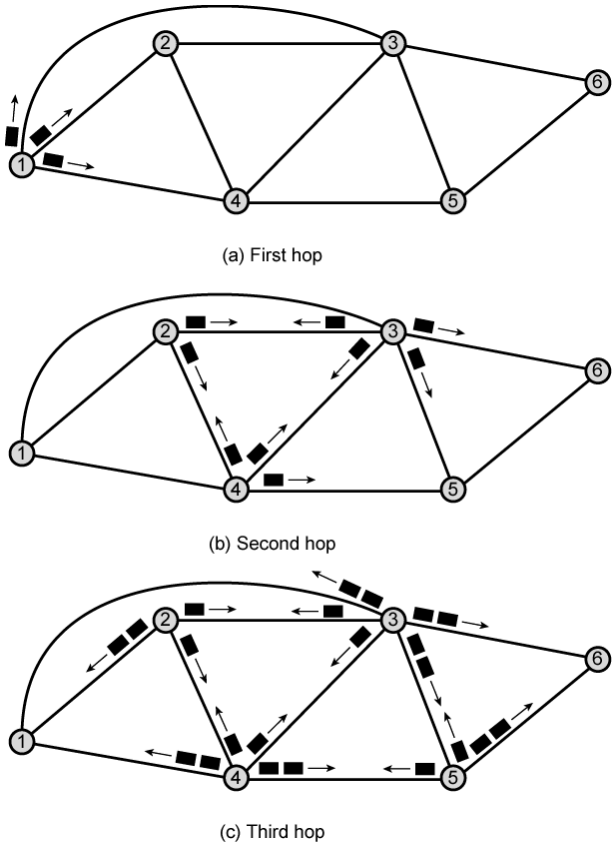
- Routage par la source : c'est la source d'un paquet qui détermine le chemin que doit suivre ce paquet. Le fonctionnement de ce routage suppose que chaque source a une vue complète (topologie et éventuellement charge) du réseau. Ce type de routage est mal adapté aux réseaux de grandes tailles.
- Routage pas à pas (dit aussi hop-by-hop) : la décision du chemin est prise par chaque noeud. En effet, chaque noeud détermine, selon ses informations, le prochain noeud auquel il devra envoyer chaque paquet qui lui parvient. C'est ce routage qui est le plus déployé dans les réseaux Internet.
- Routage hiérarchique : le réseau est subdivisé en zones et chaque zone a son routage et elle n'est vue, par les autres zones, qu'à travers son représentant. Nous reviendrons sur ce routage plus loin.

a) Routage fixe

Dans chaque routeur, on calcule le meilleur chemin pour atteindre chaque destination puis on fige les tables de routage. Ces tables ne changent que si la topologie du réseau change suite à des pannes, retrait ou adjonction d'équipement. Cette solution a l'avantage d'être simple à implanter. Elle permet aussi de trouver un chemin optimal, même moyennant un algorithme complexe, maos cette complexité ne gène pas car l'algorithme est exécuté rarement. Malheureusement, ce type de routage ne prend pas en compte l'évolution dynamique du réseau, notamment la surcharge momentanée de liens et routeurs. Dans beaucoup de réseaux, il est difficile, voire impossible, de tout savoir à l'avance et donc de fixer définitivement la table de routage. En réalité, même si on parle de routage fixe, les tables de routage sont modifiées de temps en temps pour tenir compte du fonctionnement du réseau (pannes de noeuds ou lignes, etc.).

b) Routage par inondation (flooding routing)

Si on ne connaît pas la topologie du réseau, on peut procéder par inondation. Le même paquet sera transmis sur tous les liens et routeurs. Ainsi, on est sûr que s'il y a un chemin pour atteindre le destinataire, le paquet est remis à son destinataire. C'est le type de routage le plus sûr pour livrer un paquet si on ne désire pas gérer les états des liens et routeurs.



Exemple de routage par inondation

Malheureusement, cette robustesse a un coût. En effet, on imagine aisément les problèmes de congestion que crée un tel type de routage. Chaque paquet est émis sur toutes les lignes de sortie. Cela pourrait conduire évidemment à des duplications de paquets, à des bouclages de paquets et la congestion du réseau. Pour réduire ces inconvénients plusieurs stratégies ont été proposées :

- Un compteur de sauts est placé dans chaque paquet. Ce compteur est initialisé au moment de la soumission du paquet par sa source. Il est décrémenté par chaque routeur. Quand la valeur du compteur arrive à zéro, le paquet est éliminé.
- Les paquets sont numérotés ce qui permet d'éliminer les doubles. Inconvénient, il faut garder en mémoire les numéros associés à toutes les sources.
- Inondation sélective : envoyer le paquet vers les nœuds censés être les mieux placés pour acheminer le paquet. Il faut donc une connaissance a priori de la topologie du réseau ou des réseaux utilisés(s).

Même s'il existe des solutions pour éviter les paquets en double dans un même routeur ou que les paquets bouclent dans le réseau, le routage par inondation est (très) peu utilisé dans des réseaux de tailles moyennes ou importantes.

c) Routage aléatoire

Lorsqu'un paquet arrive, le routeur choisit un lien de sortie (sauf celui par lequel le paquet est arrivé) et transmet le paquet. De cette manière, le routeur n'a pas besoin d'informations d'état sur les autres routeurs. C'est un algorithme simple, mais peu efficace lorsque le réseau est fortement maillé et/ou avec beaucoup de routeurs. Il ne permet pas de trouver le chemin optimal. Par ailleurs, un paquet peut boucler indéfiniment dans le réseau.

d) Routage adaptatif

Le routage adaptatif est utilisé aujourd'hui par presque tous les réseaux. Les décisions de routage sont prises en fonction de l'état des liens et routeurs. Les informations d'état sont échangées (périodiquement ou à la demande) entre les noeuds du réseau. L'échange d'informations d'état conduit à une surcharge du réseau et il faut trouver un compris entre la fréquence des échanges des informations d'état et le surcoût engendré. Le routage adaptatif conduit à une certaine complexité dans les réseaux et les différences entre réseaux, au niveau routage, réside dans la manière de gérer les informations d'états.

III.3 Critères de sélection de chemin

Dans les premiers réseaux, on cherchait surtout à assurer la connectivité (tout utilisateur peut atteindre d'autres utilisateurs) et les premiers algorithmes de routage avaient un seul critère important pour sélectionner un chemin, c'est celui du nombre de sauts (nombre de noeuds traversés). Avec l'apparition d'applications (telles que la téléconférence, la téléphonie, le commerce électronique...) ayant des contraintes notamment en termes de débit, de délai de transit et de taux d'erreurs, des nouveaux algorithmes de routage ont été introduits pour tenir compte de ces critères (que l'on appelle souvent paramètres de qualité de service).

On sait que plus il y a de critères à satisfaire, plus l'algorithme de routage est complexe. En particulier, on sait qu'à partir de deux critères de sélection, le problème de routage devient NP-complet.

III.4. Algorithmes de base pour le calcul du plus court chemin (shortest path routing)

Nous présentons ici les deux algorithmes de routage de base qui ont servi de point de départ à la proposition de nombreux algorithmes de routage déployés actuellement dans les réseaux. Il s'agit des algorithmes de Dijkstra proposé en 1959 et de Bellman-Ford proposé en 1962 connus tous les deux dans la recherche opérationnelle. L'algorithme de Dijkstra a une complexité de $O(n^2)$, n étant le nombre de sommets du graphe. L'algorithme de Bellman-Ford a une complexité de $O(nm)$, n étant le nombre de sommets du graphe et m le nombre d'arcs.

III.4.1. Algorithme de Dijkstra

Principe : L'algorithme de Dijkstra résout le problème de la recherche d'un plus court chemin à origine unique *s*, vers n'importe quel autre sommet, pour un graphe orienté pondéré $G = (S, A)$ dans le cas où tous les arcs ont un poids positif ou nul : $\forall (u,v) \in A,\ w(u,v) \geq 0$. *A* c'est l'ensemble des arcs et *S* l'ensemble des sommets.

L'algorithme de Dijkstra maintient à jour un ensemble *E* des sommets de *G* dont le plus court chemin à partir de l'origine *s* est connu et calculé. À chaque itération, l'algorithme choisit parmi les sommets de $S \setminus E$ (c'est-à-dire parmi les sommets dont le plus court chemin à partir de l'origine n'est pas connu) le sommet *u* dont l'estimation de plus court chemin est minimale. Une fois un sommet *u* choisi, l'algorithme met à jour, si besoin est, les estimations des plus courts chemins de ses successeurs (les sommets qui peuvent être atteints directement à partir de *u*).

***Algorithme** DIKSTRA(G, w, s):*

```
G : graphe (A, S). (A : ensemble des arcs. S : ensemble des sommets)
s : Nœud source.
w(i,j) : poids de l'arc entre les sommets i et j.
Pred(u) : prédécesseur de u sur le chemin.
L(u): poids du chemin de s à u.

/* Initialisation */
Pour chaque sommet v de G faire
      L(v) := +∞ ;
      Pred(v) := Nil ;
FinPour
L(s) := 0 ;
E := ∅ ; /* A la fin de l'exécution de l'algorithme, l'ensemble E contient
           les nœuds par lesquels passent les plus courts chemins de s
           vers tous les autres nœuds du graphe. */
F := S ;

/* Phase d'exploration des sommets */
Tant que F ≠ ∅ faire
      u := Extraire-Min(F) ; /* u tel que : L[u] = min{L[y], ∀y ∈ F} */
      E := E ∪ {u} ;
      Pour chaque arc (u,v) de G faire
            Si L(v) > L(u) + w(u,v) alors
                                        L(v) := L(u) + w(u,v) ;
                                        Pred(v) := u ;
            FinSi
      FinPour
FinTantQue
```

A la fin de l'exécution de *DIKSTRA(G, w, s)*, on peut reconstituer le plus court chemin de *s* vers une destination *d* par le pseudo-code suivant :

```
S = ∅ ; /* S contiendra le chemin de s vers d */
x := d ;
S = S ∪ {d}  /* insère d au début de S */
Tant que x != s
    x = prédecesseur[x]  /*On continue de suivre le chemin*/
    S = S ∪ {x}
FinTantQue
```

III.4.2. Algorithme de Bellman-Ford

L'algorithme de Bellman-Ford résout le problème des plus courts chemins avec une origine unique *s* dans le cas général où le poids des arcs peut être négatif. Appelé sur un graphe $G = (S, A)$, cet algorithme renvoie un booléen indiquant si le graphe contient ou non un circuit de poids strictement négatif accessible à partir de l'origine.

Algorithme Bellman-Ford (*G*, *s*, *w*)

```
G : graphe (A, S). (A : ensemble des arcs. S : ensemble des sommets)
s : Nœud source.
w(i,j) : poids de l'arc entre les sommets i et j.
Pred(u) : prédécesseur de u sur le chemin.
L(u): poids du chemin de s à u.


/* Initialisation */
Pour chaque sommet v de G faire
      L(v) := +∞ ;
      Pred(v) := Nil ;
FinPour
L(s) := 0 ;

/* Exploration des arcs */
pour i:=1 à Cardinal(S) – 1 faire
      pour chaque arc (u,v) ∈ A faire
            Si L(v) > L(u) + w(u,v) alors
                                          L(v) := L(u) + w(u,v);
                                          Pred(v) := u ;
            FinSi
pour chaque arc (u,v) ∈ A faire
      si L(v) > L(u) + w(u,v) alors renvoyer FAUX
renvoyer VRAI
```

III.5. Routage à vecteur de délai/distance (delay/distance vector routing)

Principe de l'algorithme

Cet algorithme est basé sur le principe suivant : "si mon voisin atteint le site Y avec un coût de n et si je peux atteindre mon voisin avec un coût m. Alors le coût pour atteindre le site Y est de m+n".
Le critère de performance ici est le délai estimé.
Chaque site *i* maintient deux vecteurs : D_i et S_i

d_{i1}
d_{i2}
d_{i3}
…
….
d_{in}

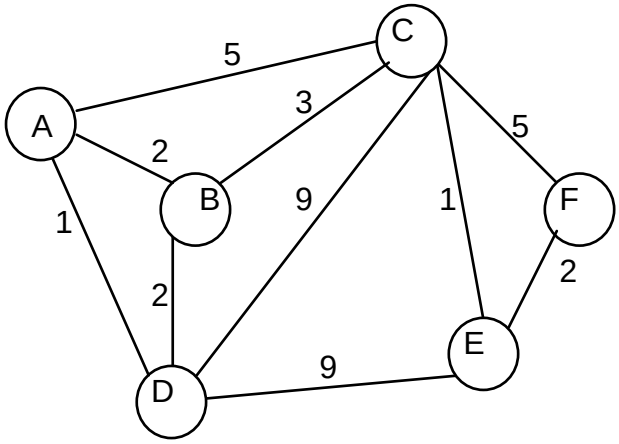
D_i , le vecteur délai pour le nœud i.

S_{i1}
S_{i2}
S_{i3}
…
….
S_{in}

S_i , le vecteur délai pour le nœud i.

Avec d_{ij} = estimation actuelle du délai minimum de i vers j (d_{ii}= 0).
N= Nombre de sites
S_i= vecteurs des sites successeurs de i.
S_{ij} = site voisin sur le chemin de coût minimum de i vers j.
Périodiquement (toutes les 128 ms), chaque site échange son vecteur de délai avec tous ses voisins. Sur la base des délais vecteurs entrants, un site k met à jour ses propres vecteurs de la façon suivante :
$d_{kj} = \min_{I \in A} [l_{ki} + d_{ij}]$
A = ensemble des voisins du site k ;
L_{ki} = estimation actuelle de délai de k à i.

Exemple



Un exemple de réseau

Destination	Délai	Nœud voisin
A	0	-
B	2	B
C	5	C
D	1	D
E	6	C
F	8	C

Table de routage de A à l'instant t.

En provenance de B	En provenance de C	En provenance de D
2	3	1
0	3	2
3	0	2
2	2	0
3	1	1
5	3	3

Informations en provenant des autres sites 128 ms plus tard.

Destination	Délai	Nœud voisin
A	0	-
B	2	B
C	3	C/D
D	1	D
E	2	D
F	4	D

Table de routage de A à l'instant t + 128 ms.

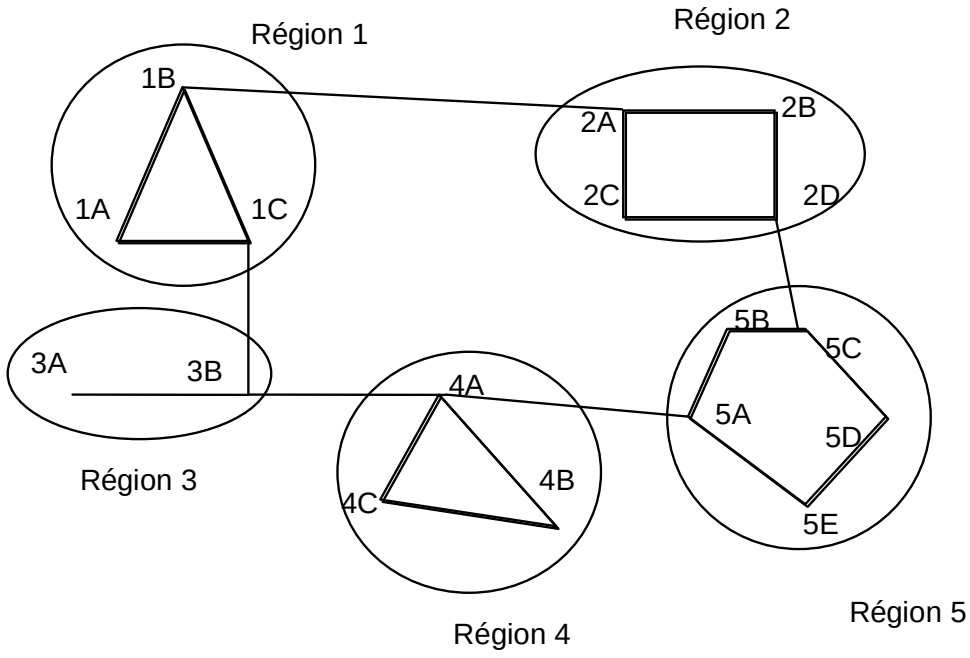
III.6. Routage par informations d'état de lien (link state routing)

Cet algorithme a remplacé celui du routage à vecteur de distance dans Arpanet. Son principe est le suivant :

- découvrir ses voisins de manière dynamique à l'initialisation ;
- mesurer les temps d'acheminement vers chacun des voisins en utilisant des sondes d'écho ;
- construire un paquet spécial disant tout ce qui vient d'être appris par le routeur et envoyer ce paquet à tous les autres routeurs du réseau ;
- calculer les plus courts chemins vers tous les autres routeurs à partir des informations collectées.

III.7. Routage hiérarchique

Les réseaux grandissent de plus en plus et les tables de routage croissent proportionnellement. Il faut alors plus de mémoire, de CPU, de bande passante pour la transmission des données. Une façon de faire face à ce problème est de procéder par niveaux hiérarchiques.
Le routage est fait par région ; chaque région a une connaissance des détails des informations de sa région et pas des autres régions.



Exemple de partitionnement d'un réseau en région

Normalement, la table de routage de chaque site contient les entrées de toutes les autres stations dans un routage non hiérarchique.

Destination	Lien	Nombre de sauts
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

Table de routage non hiérarchique pour le nœud 1A.

Le routage hiérarchique permet de réduire considérablement la taille des tables de routage.

Destination	Lien	Nombre de sauts
1A	-	-
1B	1B	1
1C	1C	1
Région 2	1B	2
Région 3	1C	2
Région 4	1C	3
Région 5	1C	4

Table de routage hiérarchique pour le nœud 1A

On remarque avec le routage hiérarchique qu'il y a une entrée pour chaque routeur local, mais une seule entrée pour les autres régions. De cette façon, le trafic de la région 2 suit la ligne 1B-2A et le trafic des autres régions passe par : 1C-3B. Une réduction du nombre d'entrées de la table de A est à noter. Mais le chemin est plus long. Par exemple : 1A vers 5C en passant par le région 2 est le meilleur chemin. Mais le routage se fait par la région 3 (3, 4 et 5).

III.8. Autres algorithmes

- routage dans les réseaux de mobiles,
- routage par diffusion et multidestinataires,
- routage avec garantie de qualité de service.

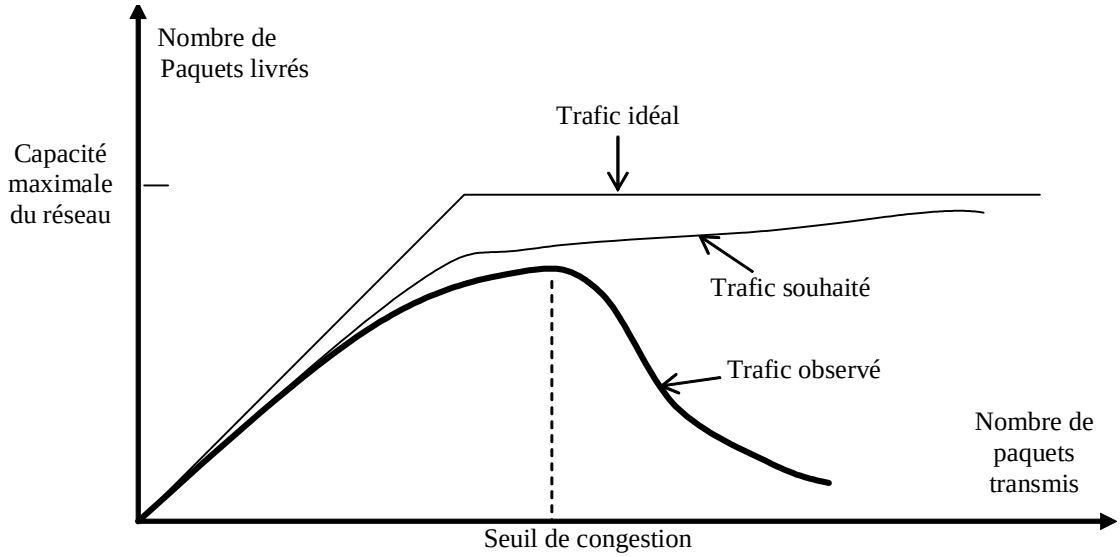
IV. Contrôle de congestion de réseau

IV.1. Notion de congestion

Dans un réseau routier, quand le nombre de véhicules augmente, les temps d'attente aux gares de péage et aux feux tricolores deviennent importants. De la même manière, dans un réseau informatique, quand le nombre de paquets dépasse un certain seuil, les performances du réseau (temps de réponse et taux de paquets livrés à leur destinataire) se dégradent. Cette situation est appelée congestion.

Dans un réseau, les paquets arrivant dans un routeur sont stockés dans des files d'attente avant de poursuivre leur chemin. L'attente dans la file d'attente dépend du nombre de paquets stockés dans cette file et de la capacité de transmission de la ligne à utiliser en sortie. La congestion est une situation où certaines (ou exceptionnellement toutes) files d'attente sont pleines et ne permettent pas de stocker des paquets. Cela conduit donc à des rejets de paquets donc à leur perte pour le destinataire.

Comme le montre la figure suivante, lorsque le trafic dépasse un certain seuil, les performances du réseau commencent à se dégrader. Si les demandes d'émission se poursuivent, le réseau peut se bloquer complètement. Lorsqu'un noeud à sa file de paquets en attente d'émission pleine, il ne peut plus recevoir de paquets. Il ne peut recevoir que s'il émet quelques paquets, mais si les noeuds voisins ont le même problème, toutes ces noeuds se retrouvent bloquées. Par phénomène de boule de neige, on peut bloquer tout un réseau si on n'arrête pas les sources en amont qui transmettent les paquets.



Performance de réseau en fonction de la charge

On peut mesurer la congestion d'un réseau à l'aide de différents paramètres :

- pourcentage de messages éliminés à cause d'un manque de mémoire,
- longueur des files d'attente de paquets,
- nombre de paquets dépassant leur délai de transmission,
- temps d'acheminement de paquet,
- autres.

IV.2. Contrôle de congestion

Principes de base :

- Surveiller le réseau afin de détecter quand et où une congestion apparaît.
- Envoyer l'information relative à une congestion aux endroits où des actions correctives peuvent être prises.
- Ajuster le comportement du réseau pour régler le problème :
  . éliminer des paquets,
  . démarrer des routeurs ou des lignes de secours.

IV.3. Prévention de congestion

Elle doit se faire à tous les niveaux :

Au niveau liaison de données :

- Instaurer un contrôle de flux et d'acquittement cela permet de réguler la communication entre deux points du réseau.

Au niveau réseau :

- Choisir un algorithme de routage :
  . peu encombrant en termes de paquets (éviter l'inondation)
  . adaptatif (pour tenir compte de l'évolution du réseau)
- Définir une politique d'élimination de paquets
- Limiter la durée de vie de paquets (attention, les sources ont tendance à fixer la durée de vie de leurs paquets à la valeur maximale).

Au niveau supérieur :

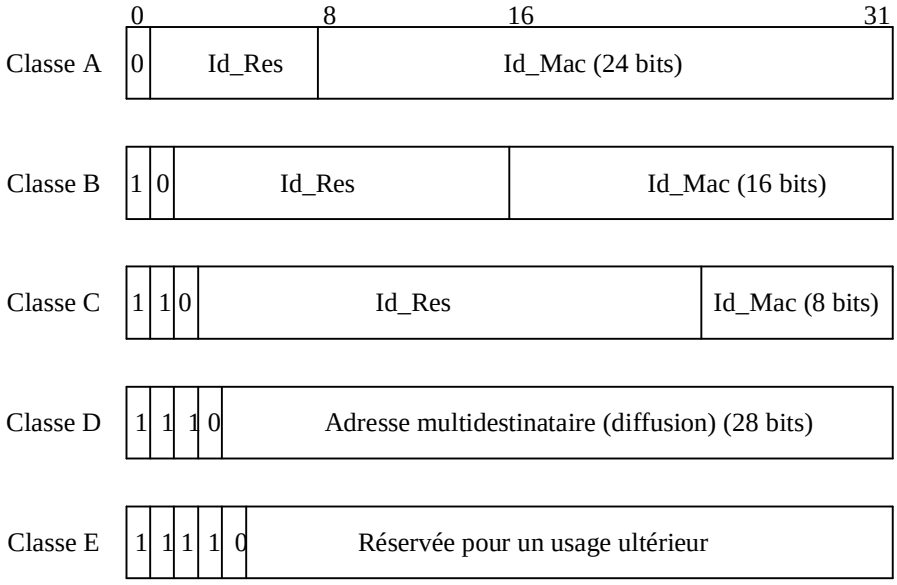
- canaliser le trafic conduisant à des avalanches ;
- écouler régulièrement le trafic (un paquet au maximum par unité de temps).

Le contrôle et la prévention de congestion sont difficiles à gérer dans un réseau non administré par une seule entité (comme c'est le cas d'Internet)

V. Routage dans IP

V.1. Adresses IP

Chaque ordinateur ou routeur connecté à Internet dispose d'une adresse sur 32 bits (ceci est valable pour la version 4 de IP, dans IP v6 le nombre des bits est de 128). Une adresse IP v4 est constituée d'une paire <*Id_Res*, *Id_Mac*>. *Id_Res* est un numéro de réseau unique sur Internet et *Id_Mac* est un numéro de machine (ordinateur ou routeur) à l'intérieur du réseau *Id_Res*. Il y a plus plusieurs façons de coder les adresse IP ; on parle de classes d'adresses IP. Comme le montre la figure suivante, il y a 5 façons de coder les adresses (conduisant ainsi à différents domaines de valeur des identificateurs de réseaux et de machines). Les premiers bits de l'adresse permettent de déterminer la classe. Les adresses de classe A sont allouées aux réseaux qui contiennent beaucoup de machines. Les adresses de classe C sont allouées aux réseaux de petites tailles (qui contiennent peu de machines).



Classes d'adresses IP

Comme le montre la figure suivante, les valeurs des adresses IP s'écrivent sous forme de quatre valeurs d'octets séparées par un point.

	Adresse la plus basse	Adresse la plus haute
Classe A	0.1.0.0	126.0.0.0
Classe B	128.0.0.0	191.255.0.0
Classe C	192.0.1.0	223.255.255.0
Classe D	224.0.0.0	239.255.255.255
Classe E	240.0.0.0	247.255.255.255

Valeurs des adresses IP

Certaines valeurs adresses particulières sont utilisées comme suit :

- Adresse égale à `0.0.0.0` : indique « cet ordinateur ». (valeur utilisée au démarrage seulement).
- Adresse de la forme `0.0.x.y` désigne « ordinateur sur ce réseau » (valable au démarrage seulement). `x.y` est différent de `0.0`
- Adresse de la forme `x.y.0.0` : diffusion dirigée vers le réseau d'Id_Res = `x.y`.
- Adresse de la forme `127.x.y.z` : adresse de bouclage (utilisée lors de test sur un même ordinateur).

V.2. Taille des datagramme, MTU et fragmentation

Un datagramme IP peut avoir une taille très importante (pouvant aller jusqu'à 65535 octets). Mais avec une telle taille il arrive souvent qu'un datagramme est fragmenté à plusieurs endroits sur le chemin de la source à la destination. En effet, un datagramme IP est encapsulé dans une trame niveau liaison de données pour passer

d'une station à un routeur, d'un routeur à un autre routeur ou d'un routeur à une station. Généralement, la connexion entre station et routeur se fait via un réseau local Ethernet ou autre. Or, pour gérer de manière efficace, ses tampons d'émission et de réception de trames, chaque réseau local limite la taille maximale de trame. Par exemple, le réseau Ethernet limite la taille à 1500 octets le réseau FDDI la limite à 4500 (environ). On nomme cette limite MTU (Maximum Transfer Unit). Ainsi, pour traverser un réseau local, un datagramme doit être découpé en segments ne dépassant pas la MTU. Une fois un datagramme est fragmenté en plusieurs morceaux, ces morceaux sont acheminés, éventuellement sur des chemins différents, jusqu'à leur destination finale où ils seront rassemblés pour former le datagramme initial. Si un des fragments se perd, le datagramme ne peut être reconstitué par le destinataire et tous les segments correspondants à un datagramme sont rejetés à la réception. Les trois champs *Identification*, *Drapeaux* et *Offset de fragment*, contenus dans l'entête de paquet IP, permettent de contrôler la reconstitution du datagramme à partir de ses morceaux (voir le format de paquet IP).

V.3. Protocoles de résolution d'adresse

Protocole ARP (Address Resolution Protocol) : il permet à une station Ethernet de connaître l'adresse MAC d'une autre station connectée au même réseau Ethernet dont elle connaît l'adresse IP. Pour cela, elle diffuse une trame contenant l'adresse IP qu'elle connaît et la station ayant l'adresse IP reçoit (en même temps que toutes les autres stations du réseau Ethernet) la demande de résolution d'adresse et répond. Ce protocole est surtout utile pour le démarrage de réseau et pour tenir compte des changements d'adresses MAC sans changer les adresses IP.

Protocole RARP (Reverse ARP) : l'adresse IP de station est souvent enregistrée sur disque ce qui permet de la retrouver après le démarrage de la station. Pour un ordinateur sans disque, le protocole RARP est conçu pour permettre à une station de demander son adresse IP à un serveur d'adresse se trouvant sur le même réseau Ethernet.

Il existe aussi d'autres protocoles plus complexes pour obtenir une adresse IP de manière dynamique, de changer d'adresse lorsqu'un ordinateur mobile change de réseau...

V.4. Principe de base du routage dans IP

Le routage IP est un routage qui s'effectue sur un réseau non fiable et en mode non connecté. Le standard IP ne mentionne pas un algorithme de routage obligatoire, mais en préconise certains (le plus souvent utilisant l'approche dynamique et saut par saut). Le fait de ne pas imposer d'algorithme de routage permet l'interopérabilité entre les réseaux : chaque réseau utilise l'algorithme qu'il souhaite.

Le routage IP insiste seulement sur la notion table de routage et la manière de l'utiliser qui repose sur : la remise directe et la remise indirecte. Comme aucun algorithme de routage précis n'est obligatoire, IP n'impose pas la fréquence à laquelle la table de routage est mise à jour. Chaque administrateur de réseau veillera à ce que sa table soit mise à jour selon ses convenances et besoins.

Remise directe :

- C'est le dernier routeur sur le chemin du datagramme qui transmet directement le datagramme au destinataire.
- Il s'agit d'une transmission d'un paquet sur un même réseau.
- On utilise l'adresse physique (adresse MAC) du destinataire.
- Le datagramme est encapsulé dans une trame de niveau liaison de données.

Remise indirecte :

Le datagramme passe de routeur en routeur avant d'atteindre sa destination. Chaque routeur intermédiaire choisit le prochain routeur auquel il faut envoyer le datagramme en fonction de l'adresse de destination et de la table de routage de ce routeur.

V.5. Table de routage

Elle contient pour chaque numéro de réseau :

- le prochain routeur auquel il faut envoyer le datagramme (next hop),
- une indication pour signaler une remise directe si le destinataire se trouve sur le même réseau (local) que le routeur,
- un routeur par défaut,
- rien : pour signaler une erreur de routage.

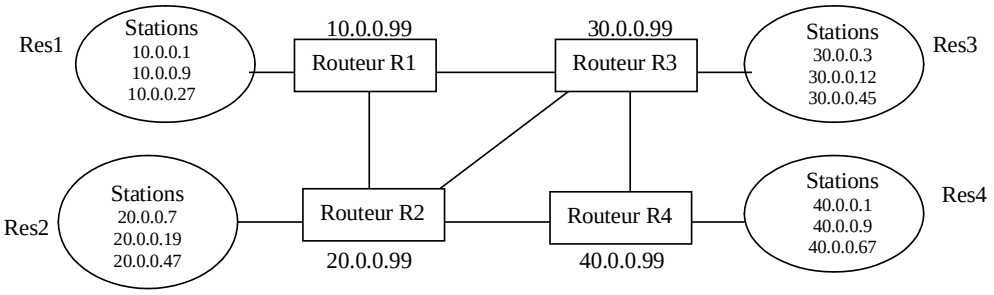
Exemple :

Réseau 1 (avec l'adresse réseau = 10.0.0.0) : constitué des stations 10.0.0.1, 10.0.0.9, 10.0.0.27

Réseau 2 (avec l'adresse réseau = 20.0.0.0) : constitué des stations 20.0.0.7, 20.0.0.19, 20.0.0.47

Réseau 3 (avec l'adresse réseau = 30.0.0.0 : constitué des stations 30.0.0.3, 30.0.0.12, 30.0.0.45

Réseau 4 (avec l'adresse réseau = 40.0.0.0) : constitué des stations 40.0.0.1, 40.0.0.9, 40.0.0.67



Structure du réseau

Numéro de réseau de destination	Adresse de remise
10.0	remise directe
20.0	20.0.0.99
30.0	30.0.0.99
40.0	30.0.0.99

Table de routage du routeur R1

V.6. Algorithme d'expédition (forwarding) de IP

L'algorithme que nous présentons ci-dessous décrit le comportement du routeur IP quand il reçoit un paquet. Par abus de langage, cet algorithme est appelé aussi algorithme de routage (alors qu'il ne s'intéresse qu'à l'expédition).

1. On regarde *AdIPDest* (Adresse IP de Destination). A partir de *AdIPDest*, on détermine *SuffRes* (le Suffixe du Réseau). Le nombre de bits du suffixe réseau dépend de la classe d'adresse IP.

2. Si *SuffRes* correspond à l'ID de réseau d'un des réseaux connecté directement, l'hôte de destination se trouve sur un réseau directement connecté. Dans ce cas, on effectue une remise directe du datagramme. Le datagramme IP est encapsulé dans une trame de la couche liaison de données et est envoyée directement vers le destinataire.

3. Si on ne trouve pas de correspondance à l'étape 2, on examine la table de routage à la recherche d'une entrée spécifique d'un hôte (un ordinateur (**)), correspondant à *AdIPDest*. Si une telle entrée existe, on transmet le datagramme IP, comme indiqué par l'entrée de routeur de prochain pas correspondante.

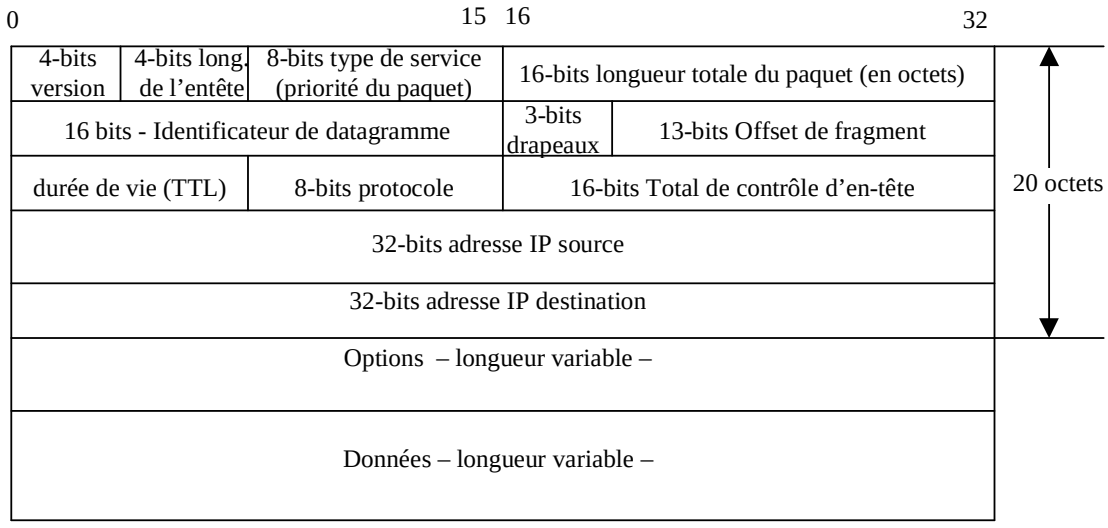
4. Si on ne trouve pas de correspondance à l'étape 3, on examine la table de routage à la recherche d'une entrée de réseau correspondant à *SuffRes*. Si une telle entrée existe, on transmet le datagramme de la manière indiquée par l'entrée du routeur de prochain pas correspondante.

5. Si on ne trouve pas de correspondance à l'étape 4, on regarde dans la table de routage la valeur correspondant à 0.0.0.0 qui est l'entrée par défaut. Si cette entrée existe, on transmet le datagramme de la manière indiquée par l'entrée du routeur par défaut correspondante.

6. Si on ne trouve rien à l'étape 5, le datagramme ne peut pas être routé et on déclare une erreur de routage.

(**) Le routage d'ordinateur à ordinateur peut aider à mieux sécuriser les communications, en imposant que le routage soit effectué par des ordinateurs et non des routeurs. Ce mécanisme est rarement utilisé.

V.7. Format de datagramme IP v4



Format de paquet IPv4

Version : version IPv4, IPv6, autres (TUBA, protocole P, non affecté, réservé). Ici on traite IP v4.

Long de l'entête : longueur de l'en-tête en mots de 32 bits (ce champ est obligatoire à cause du champ *Options* variable).

Type de service : composé de six champs qui indiquent la qualité de service

- Valeur de préséance associée au datagramme (Priorité maximale, immédiat, le même jour, d'ici au lendemain, etc.). Ce champ de 3 bits vaut par défaut 000. Il a était à l'origine destiné à des applications militaires aux USA.
- Délai : 1 bit (normal, faible)
- Débit : 1 bit (normal, élevé)
- Fiabilité : 1 bit (normal, élevé)
- Coût : 1 bit (normal, élevé)
- MBZ : 1 bit (Must Be Zero)

Longueur totale du paquet : contient la longueur de l'en-tête et des données en octets. Ce qui conduit à des datagrammes de longueur maximale égale à 65 535. Dans la pratique, les réseaux acceptent des longueurs plus petites indiquées par un paramètre appelé MTU (Maximum Transfer Unit). Dans le cas où un datagramme dépasse MTU octets, il est fragmenté.

Identificateur de datagramme : contient le numéro du datagramme assigné par la source. Ce champ sert principalement à identifier les fragments IP afin de les rassembler.

Drapeaux : DF (Don't fragment) = 0 si fragmentation non autorisée et 1 si elle l'est.

MF (More fragments) = 0 si dernier fragment du datagramme originel et 1 sinon.

Offset de fragment : indique la position (en octets) des données du fragment par rapport au début du datagramme originel.

Durée de vie : se mesure en secondes et représente la durée maximale de vie d'un datagramme. En réalité, la valeur est traitée comme un compteur décrémenté à chaque fois de 1 et non comme une quantité temporelle. Cette valeur est décrémentée à chaque routeur de la durée nécessaire au traitement du datagramme. Lorsque la durée de vie d'un datagramme tombe à 0, il est éliminé. La valeur par défaut est 32 ou 64. Beaucoup d'implantations d'IP ne permettent que la valeur par défaut.

Protocole : désigne le protocole de niveau supérieur auquel est destiné le datagramme (TCP, UDP,…).

Total de contrôle d'en-tête : Le complément à 1 de chaque valeur 16 bits constituant l'en-tête (excepté le champ contrôle d'en-tête lui-même est rajouté et on prend ensuite le complément à 1 de la somme).

Adresse source : adresse IP de la source sur 32 bits.

Adresse destination : adresse IP de la destination sur 32 bits.

Options : indique le niveau de sécurité, les informations du chemin emprunté par le datagramme, les informations d'estampillage (date de réception du datagramme par chaque routeur mesurée en millisecondes écoulées depuis minuit GMT).

Données : données du datagramme destinées à la couche supérieure.

VI. Protocoles

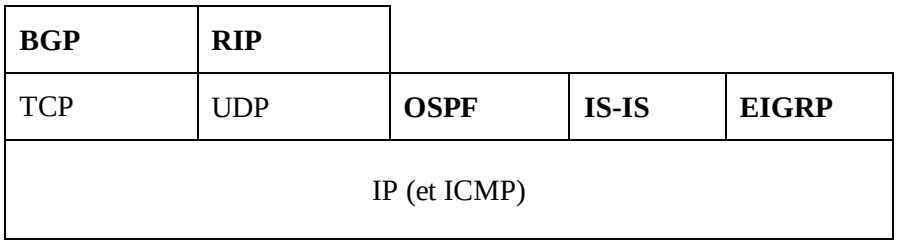
VI.1 Protocoles de routage

Le rôle d'un protocole de routage dans un réseau, ce n'est pas de calculer le plus court chemin (il ne remplace donc pas l'algorithme de routage), mais il permet de gérer les échanges d'informations entre noeuds pour mettre à jour les tables de routage. Il existe de nombreux protocoles de routage, pour les réseaux IP, pour les réseaux ATM, pour les réseaux mobiles, pour les réseaux ad hoc... Nous présentons brièvement les protocoles les plus utilisés en particulier ceux déployés dans Internet.

V.5.1. Protocoles de routage dans Internet

Internet est composé de réseaux autonomes que l'on appelle *domaines*. Actuellement, il existe plusieurs (dizaines de) milliers de domaines qui composent la toile Internet. Chaque domaine a son propre administrateur qui choisit selon ses besoins le routage à appliquer à l'intérieur de son domaine. Pour que les ordinateurs appartenant à des domaines différents puissent communiquer, il faut aussi avoir des protocoles de routage inter-domaines. Ainsi, on a deux types de protocoles de routage :

- protocoles pour le routage intra domaine (appelés 'IGP : Interior Gateway Protocols') : ces protocoles incluent notamment RIP, OSPF, IS-IS et EIGRP.
- protocoles pour le routage inter domaine (appelés 'EGP : Exterior Gateway Protocols') : ces protocoles incluent notamment BGP.



Protocoles de routage les plus utilisés

Plusieurs protocoles de routage peuvent être implantés sur un même routeur (l'interaction entre ces protocoles est généralement gérée manuellement).

RIP (Routing Information Protocol) : RIP est protocole de type vecteur de distance. Une seule métrique est utilisée pour calculer le plus court chemin (le nombre de sauts). Chaque noeud diffuse à ses voisins le nombre de sauts qu'il lui faut pour atteindre les autres noeuds du réseau. Les vecteurs de distance sont échangés toutes les 30 secondes par des paquets (appelés *Advertisements*). Chaque *advertisement* peut contenir jusqu'à 25 destinations. Si aucun *advertisement* n'est reçu en provenance d'un voisin au bout de 180 secondes, on considère ce voisin non atteignable, et on essaie de changer de route et des messages sont envoyés aux autres voisins pour les informer. RIP utilise l'algorithme de Bellman-Ford pour calculer le plus court chemin. Il est facile à implanter et utilisable seulement pour des réseaux de petite taille. Aujourd'hui, RIP est très peu utilisé.

OSPF (Open Shortest Path First) : OSPF est un protocole de type état de lien. Chaque routeur maintient une base d'informations sur les états des autres noeuds. OSPF utilise cette base et l'algorithme de Dijkstra pour déterminer les chemins les plus courts. C'est le protocole le plus déployé actuellement sur Internet. Par rapport à RIP, il offre des mécanismes d'authentification, il peut calculer le plus court chemin selon plusieurs métriques. Pour un domaine de grande taille, OSPF peut fonctionner en mode hiérarchique (en structurant le domaine en deux niveaux) pour minimiser les échanges liés aux états de lien. OSPF est plus complexe que RIP (à titre indicatif, les spécifications de RIP tiennent sur une trentaine de pages et celles de OSPF sur plus de 240 pages). La complexité est due à la gestion distribuée des informations d'état.

IS-IS (Intermediate System to Intermediate System). C'est un protocole à états de lien initialement proposé par l'ISO pour le mode non orienté connexion. Il a été repris par le monde IP pour permettre de mixer dans un même domaine des routeurs IP et des routeurs selon la couche ISO. IS-IS utilise l'algorithme de Dijkstra pour construire les chemins. Dans la pratique, il est peu employé.

EIGRP (Enhanced Interior Gateway Routing Protocol) : c'est un protocole Cisco à vecteur de distance proposé par Cisco's. EIGRP peut utiliser différentes métriques (délai, bande passante, fiabilité ou charge) pour sélectionner un chemin.

BGP (Border Gateway Protocol) : C'est actuellement le standard de facto (utilisé quasiment entre tous les domaines) dans Internet. Il est similaire au protocole de vecteur de distance. Chaque noeud de bordure (celui

qui représente un domaine) envoie à ses voisins (les autres noeuds de bordure) des chemins sous forme de domaines (et non sous forme de noeuds individuels). BGP utilise TCP pour échanger messages contenant les vecteurs de distance.

Remarque :
Il existe de nombreux autres protocoles de routage dans le monde IP notamment pour le multicast, la gestion de la QoS... La plupart de ces protocoles sont soit au stade expérimental, soit déployés au sein d'entreprises particulières.

V.5.2. Autres protocoles de routage

Les réseaux, autres que ceux fondés sur IP, ont aussi leurs protocoles de routage. On peut citer notamment :

- protocole PNNI (Private Network-to-Network Interface) largement utilisé dans les réseaux ATM ;
- protocoles WRP (Wireless Routing Protocol), AODV (Ad hoc On-Demand Distance Vector), DSDV (Destination-Sequeced Distance Vector), DSR (Dynamic Source Routing), etc. utilisés dans les réseaux mobiles ad hoc,

VI.2. Protocole ICMP (Internet Control and error Message Control)

Objectifs :

- aider la source d'un datagramme à comprendre les causes de non livraison de son datagramme ;
- fournir diverses informations sur le fonctionnement du réseau.

Le routeur qui détecte une erreur (hôte inaccessible, hôte inexistant, …) envoie un datagramme ICMP à la source. Il n'y a pas de notification d'erreur aux routeurs intermédiaires (ceux par lesquels le datagramme non livré est passé).

Quelques types d'erreurs :

- réseau inaccessible,
- réseau de destination inconnu,
- communication avec le réseau de destination interdite par l'administrateur réseau,
- ordinateur inaccessible,
- ordinateur inaccessible pour le service demandé,
- ordinateur destinataire inconnu,
- protocole inaccessible,
- fragmentation nécessaire et bit DF positionné à 1.

Autres fonctions de ICMP :

- test d'accessibilité d'un hôte ;
- un routeur en situation de congestion détruit des datagrammes et avertit leur source de réduire son débit ;
- horodatage de datagramme (pour synchroniser des horloges) ;
- demander à un ordinateur de changer sa table de routage car elle n'est pas optimale ;
- etc.

VII. Eléments de comparaison

VII.1. Points de différences entre les réseaux

Aspect	Quelques possibilités
Service offert	Connecté, non connecté
Protocole réseau	IP, X25, AppleTalk, Decnet, …
Adressage	Uniforme, hiérarchique (IP)
Diffusion	Possible ou non
Qualité de service	Garantie ou non
Gestion des erreurs	transmission fiable ou non, ordonnée ou non
Contrôle de flux	Fenêtre coulissante, contrôle de débit, …
Contrôle de congestion	Réalisée ou non
Sécurité	Chiffrement, authentification, …, tout en clair
Coût de communication	temps de connexion, délai de transit, facturation, …

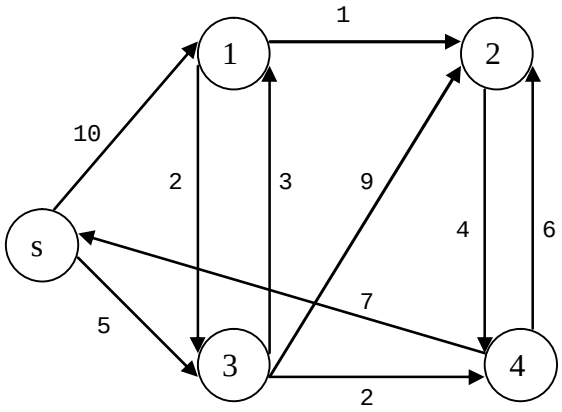
VII.2. Comparaison entre mode datagramme et circuit virtuel

Caractéristique	Mode datagramme	Mode connecté
Etablissement de circuit	Inexistante	Nécessaire
Adressage	Chaque datagramme contient les adresses de la source et destination	Chaque paquet contient un numéro de circuit virtuel
Informations de routage	On ne conserve aucune information de routage	Chaque CV établi requiert de la place dans les tables de routage
Routage	Chaque datagramme a un chemin indépendant	Tous les paquets d'un même CV suivent la même route
Conséquence d'une défaillance d'un routeur	Aucune (sauf perte des datagrammes présents dans le routeur défaillant)	Tous les CV traversant le routeur défaillant sont détruits. Il faut en établir d'autres
Contrôle d'erreur, de flux et congestion	Difficile et complexe	Facile à mettre en œuvre

Exercices

Exercice 1

Q1. Etude l'algorithme de routage de Dijkstra sur le réseau décrit par le graphe suivant (le noeud *s* est le point de départ).



Exemple de réseau modélisé à l'aide de graphe.

Q2. Etude l'algorithme de routage de Bellman-Ford sur le réseau décrit par le graphe précédent.

Exercice 2

Soit un réseau constitué de deux sous-réseaux *R1* et *R2* reliés entre eux par un routeur *B*. On s'intéresse à l'échange de paquets entre une station A située sur le sous-réseau *R1* et une autre station *C* reliée au sous-réseau *R2*.

On veut étudier le nombre de trames échangées sur le réseau pour permettre à la station *A* d'envoyer *n* paquets à la station *C*. On se place au niveau réseau et on suppose que la couche liaison de données est orientée connexion. Des acquittements sont nécessaires avant toute fermeture de connexion.

a) Cas 1 : On suppose que l'intervalle de temps entre deux paquets successifs est trop important pour garder la liaison logique entre deux stations. Ainsi, la liaison logique entre deux stations n'est maintenue que pendant la transmission d'un seul paquet.

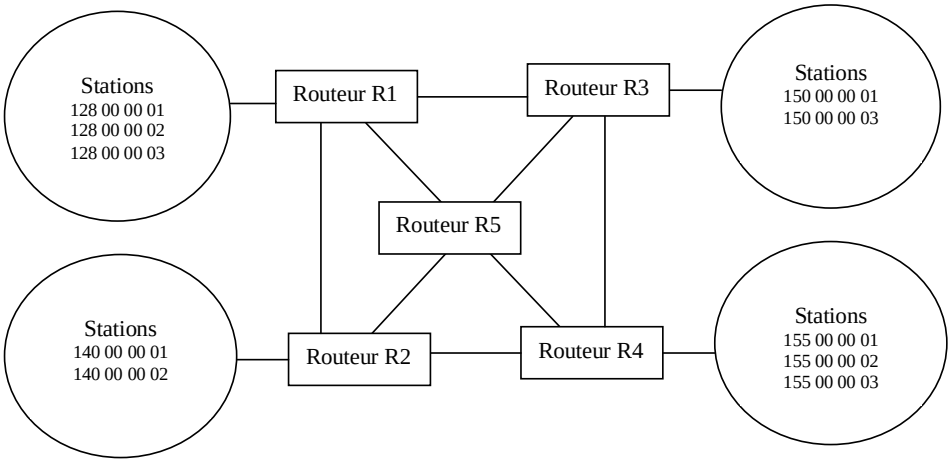
b) Cas 2 : On suppose que les *n* paquets sont transmis par la station *A* dans un intervalle de temps suffisamment court pour ne demander qu'une seule liaison logique par paire de stations.

c) Cas 3 : on reprend l'hypothèse du cas 1, mais on suppose cette fois que les paquets émis par la station *A* ont une même taille de *20 K* octets alors que la station *C* ne peut recevoir que des trames de longueur maximale égale à *10 K* octets. Les stations *A* et *B* peuvent gérer des trames de n'importe quelle taille.

d) Reprendre les hypothèses du cas a et montrer tous les appels de primitives de service de niveau liaison de données et réseau qui conduisent à la génération des trames véhiculées sur le réseau. On suppose que la couche réseau n'est pas orientée connexion.

Exercice 3

Soit un ensemble de quatre réseaux de stations interconnectées via cinq routeurs. Un cercle désigne un réseau local dans lequel les stations désignées sont directement connectées entre elles et avec un routeur.



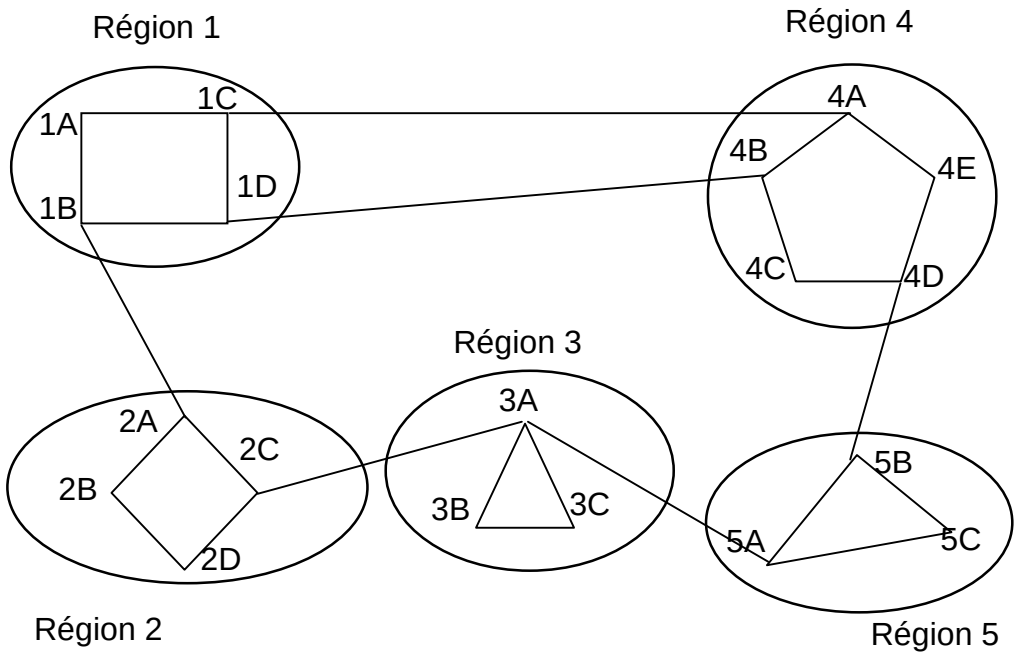
Question 3.1 : Donner les tables des cinq routeurs IP, dans le cas où le nombre de sauts effectués par chaque paquet est optimal. Le nombre de sauts correspond au nombre de routeurs traversés par un paquet.

Question 3.2 : Donner les tables des cinq routeurs qui offrent le pire cas pour le nombre de sauts pour un paquet.

Question 3.3 : On suppose que certains routeurs ou certaines lignes de communication peuvent tomber en panne. Proposez un algorithme simple qui permet à un routeur de changer dynamiquement sa table de routage quand il détecte une panne.

Exercice 4

Soit le réseau suivant, constitué de régions autonomes (aussi appelés systèmes autonomes – AS autonomous systems dans Internet). Etudions les avantages et inconvénients de la mise en place d'un routage hiérarchique.



Q1) Dans un premier temps on fait comme si le routage n'était pas hiérarchique. Donnez la table de routage du routeur 1A. On s'attachera à calculer les plus courts chemins dans ce cas.

Q2) Donnez les tables de routages des routeurs 1A, 1B, 2A, 3A, 4A et 5A dans le cas d'un routage hiérarchique. Ici on calculera les plus courts chemins entre les régions. Y a-t-il des allongements de chemins depuis 1A ?

Q3) On suppose que le lien 1C-4A tombe en panne. Qu'est ce qui doit changer dans les tables. Attention on ne doit pas recalculer les tables mais seulement les mettre à jour en fonction de la découverte des problèmes et de la propagation de la connaissance du problème.

Exercice 5

La probabilité de perte estimée pour un réseau est *PP*. La probabilité d'erreur de transmission est *PE*. Estimer le nombre de transmissions nécessaires pour recevoir correctement un ensemble de *N* messages avec une probabilité égale à *PC*. On suppose que les probabilités *PP*, *PE* et *PC* sont petites.
Application numérique : PP = 10^{-6}, PE = 10^{-5}, N = 1000, PC = 1 - 10^{-10}

Exercice 6

On suppose qu'il existe une fonction *Plus_court_chemin* qui fournit le plus court chemin entre deux nœuds donnés (voir algorithme de Dijkstra par exemple).

Etudier par raffinement successif les fonctions du routeur qui s'occupent du routage adaptatif.

Chapitre 7
Sécurité dans les réseaux

I. Introduction

Un système informatique relié à un réseau devient vulnérable aux attaques extérieures (piratage de données, effacement/modification de données, introduction de virus, …). Il faut donc protéger et contrôler les informations qui circulent dans les réseaux.

I.1. Termes utilisés en sécurité informatique

La sécurité informatique englobe trois aspects différents pour protéger les informations :

- ***Confidentialité*** : c'est la propriété d'une information de ne pas être révélée à des personnes non autorisées à la connaître. Cela signifie que le système doit mettre en place les mécanismes pour empêcher ces personnes de lire l'information confidentielle.
- ***Intégrité*** : c'est la propriété d'une information de ne pas être altérée. Cela signifie que le système doit empêcher toute modification indue ou incorrecte de l'information et assurer que les modifications légitimes soient réalisées.
- ***Disponibilité*** : c'est la propriété d'une information (ou service) d'être accessible aux utilisateurs autorisés à la manipuler. Lorsqu'une entité malicieuse empêche l'utilisation d'un service (c'est-à-dire le rend inaccessible) par des personnes autorisées, on parle de déni de service.

D'autres facettes (propriétés) de la sécurité peuvent être considérées : intimité (vie privée et anonymat), authenticité, non répudiation, pérennité, exclusivité, protection contre la copie illicite de logiciel, etc. Ces propriétés peuvent aussi s'exprimer en termes de confidentialité, intégrité et disponibilité.

Les trois propriétés (confidentialité, intégrité et disponibilité) peuvent êtres assurées totalement ou partiellement par des mécanismes de sécurité. Il ne faut pas donc confondre les buts (les propriétés) et les mécanismes mis en oeuvre pour les atteindre. Nous définissons ci-dessous les mécanismes les plus importants de la sécurité :

- ***Autorisation*** : action par laquelle on permet, ou on interdit, à un utilisateur un ensemble d'actions sur des informations.
- ***Authentification*** : reconnaître (avec certitude) un utilisateur.
- ***Cryptographie*** : l'art d'écrire des secrets pour les rendre inintelligibles à des tiers.
- ***Cryptanalyse*** : l'art de retrouver les secrets cachés dans des informations inintelligibles.
- ***Cryptologie*** : se compose de la cryptographie et cryptanalyse.
- ***Chiffrement*** : il vise à assurer la confidentialité d'informations en transformant un texte en clair en un cryptogramme (ou texte chiffré) à l'aide d'une clé de chiffrement. Le déchiffrement et l'opération inverse du chiffrement.
- ***Clé*** : information (secrète) utilisée pour chiffrer ou déchiffrer un texte chiffré.
- ***Signature*** : information contenue dans un message servant à garantir son authenticité et intégrité.

I.2. Types d'attaques

Les attaques de système informatique peuvent être plus au moins naïves, plus au moins dangereuses. Un étudiant qui cherche à rentrer dans un système car il veut tester les limites de ses compétences, un hacker qui cherche à déranger et un espion qui cherche à dérober des informations stratégiques n'ont pas les mêmes objectifs et sont à distinguer. Les intrus peuvent être externes (non connus du système qu'ils attaquent) ou internes (connus du système, mais qui cherchent à étendre leurs droits ou à abuser de leurs privilèges).

On distingue deux catégories d'attaques : attaques passives et attaques actives. Dans le cas d'une attaque passive, l'objectif pour l'attaquant c'est de connaître des informations jugées utiles pour lui malgré l'opposition du propriétaire de ces informations. Ce type d'attaques est très difficile à détecter. Par contre, on peut les prévenir en mettant en place des mécanismes de cryptographie, d'authentification... Dans le cas d'une attaque active, l'objectif de l'attaquant est de modifier ou détruire des informations, d'empêcher une partie du réseau de fonctionner, de rejouer un scénario après en avoir modifié certains aspects... Ces attaques sont plus dangereuses, elles sont difficiles à prévenir, mais il est possible de les détecter.

Les attaques peuvent se situer à plusieurs niveaux et prendre plusieurs formes, en voici les plus connues :

- *Ecoute passive* : écouter les messages émis ou reçus, savoir sur quels sites un utilisateur se connecte, ses correspondants en termes de message électroniques... Il s'agit d'attaques qui touchent à la confidentialité et vie privée.
- *Interception* : intercepter et modifier un message sur les voies de communication. Il s'agit d'attaques qui touchent à la confidentialité et l'intégrité.
- *Cryptanalyse* : il s'agit pour l'attaquant d'obtenir des informations sur les clés ou les algorithmes de chiffrement à partir d'informations non confidentielles pour s'en servir plus tard.
- *Répudiation* : consiste pour un utilisateur à refuser ou reconnaître une opération qu'il a effectuée. Ce type d'attaque est particulièrement nocif pour les applications de commerce électronique.
- *Déguisement* : consiste à tromper les mécanismes d'authentification (en usurpant une adresse ou nom d'utilisateur par exemple) pour se faire passer pour un utilisateur autorisé de façon à obtenir des droits d'accès illégitimes et ainsi compromettre la confidentialité, intégrité ou disponibilité.
- *Déduction par inférence* : consiste pour un attaquant à recouper des informations auxquelles il a légitimement accès pour en déduire d'autres auxquelles il ne devrait pas avoir accès. Par exemple, à partir des médicaments administrés à des patients, un médecin peut déduire le nombre de patients qui ont une certaine pathologie (même si ce médecin n'est censé que suivre des patients individuellement).
- *Porte dérobée* : utiliser des failles (dues à une mauvaise conception) dans un système pour contourner les mécanismes de contrôle d'accès.
- *Cheval de Troie* : c'est un programme effectuant une fonction illicite tout en donnant l'apparence d'effectuer une fonction légitime. Ce type de programme se répand de plus en plus sur Internet. On donne gratuitement (ou non) un logiciel destiné à réaliser un certain travail mais en plus de ce travail, le logiciel contient (mais ce n'est pas dit à celui qui l'utilise) un petit programme qui va par exemple espionner l'utilisateur.
- *Virus* : il s'agit de programmes (télé-)chargés sans le savoir par un utilisateur qui vont effectuer des opérations plus au moins néfastes pour son système. Aujourd'hui, les virus constituent un des fléaux dans le monde Internet.
- *Ver* : c'est un programme autonome qui se propage sur Internet (via le mail généralement) et se reproduit à l'insu des utilisateurs normaux. Il peut être porteur d'un virus qui peut être activé ou non selon l'attaquant.
- *Déni de service* : il consiste à empêcher des utilisateurs légitimes d'accéder aux informations/services ; il s'agit d'une attaque contre la disponibilité. En général, il suffit d'émettre des requêtes en très grand nombre vers un site pour le bloquer. Ainsi, une personne se trouvant à Bora bora peut empêcher un serveur à New York ou à Paris de fonctionner correctement.

II. Cryptographie

Les fonctions de base de la cryptographie sont le chiffrement et déchiffrement. Le chiffrement de message consiste à transformer le message initial en appliquant une forme déduite à partir d'une clé de chiffrement. Pour cela différents aspects doivent être étudiés, notamment : type de clé à utiliser (symétrique, publique...), algorithme de chiffrement et son implantation, protocole de communication et partage de clé.

Le chiffrement d'un message *M* par une clé *Ks* (Key of sender) fournit un message chiffré (ou crypté) *C* :

$$Ks : M \rightarrow C$$

Le déchiffrement d'un message *M* par une clé *Kr* (Key of receiver) fournit un message clair *M* :

$$Kr : C \rightarrow M$$

Quand la clé de chiffrement est identique à la clé de déchiffrement (*Ks = Kr*), on parle de ***chiffrement symétrique***.

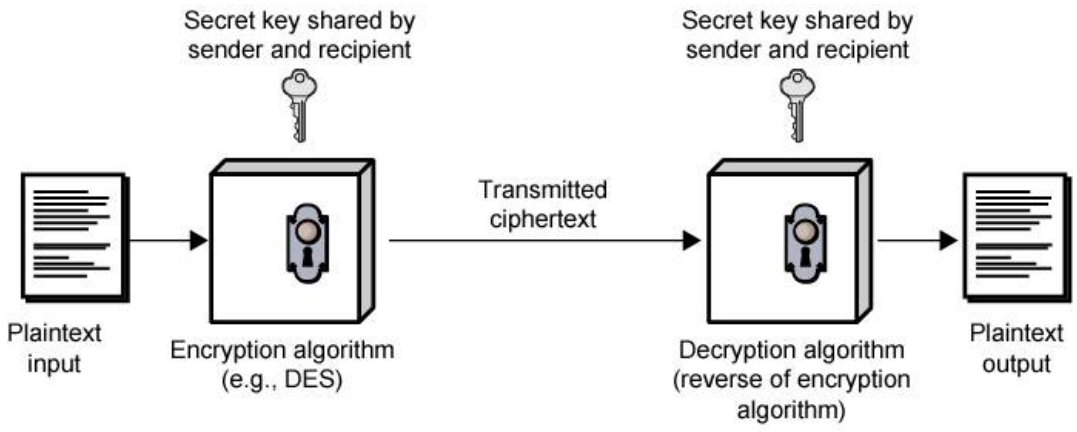
Jusque vers la fin des années 1970, toutes les stratégies de chiffrement étaient symétriques. Par exemple, les systèmes DES (Data Encryption Standard) et AES (Advanced Encryption Standard) sont deux des chiffres (c'est-à-dire deux systèmes de chiffrement/déchiffrement) symétriques les plus connus.

Si les deux clés *Ks* et *Kr* sont différentes, et si on connaît l'une des clés il est impossible (ou plus exactement, il est peu probable, moyennant une puissance de calcul raisonnable) de trouver l'autre, le chiffre est dit à ***clé publique***. Le chiffre à clé publique le plus courant est basé sur l'algorithme RSA.

II.1. Systèmes de chiffrement symétrique

II.1.1. Principe de fonctionnement du chiffrement symétrique

Le principe de chiffrement symétrique est illustré par la figure suivante : l'émetteur et le récepteur connaissent la clé de chiffrement. L'émetteur chiffre le message en utilisant la clé et le récepteur effectue l'opération inverse en utilisant la même clé.



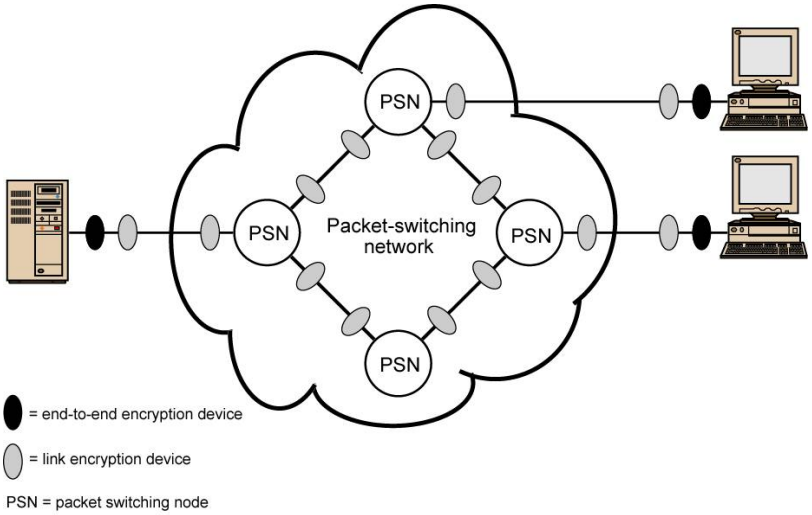
Principe simplifié de communication chiffrée par un système symétrique.

Pour garantir la sécurité, certains aspects doivent être pris en compte avec le plus d'attention :

- L'émetteur et le récepteur doivent avoir un moyen sûr pour échanger la clé (en dehors de la phase de communication des données à chiffrer). Noter qu'un des points de vulnérabilité des stratégies de sécurité réside dans la stratégie de partage de la clé.
- L'algorithme de chiffrement doit être robuste :
  + Même si on connaît l'algorithme, on ne peut pas déchiffrer sans clé.
  + Même si l'on dispose de plusieurs messages chiffrés et de leurs déchiffrements, on ne pas déchiffrer les autres messages.

Cela signifie qu'un bon mécanisme de chiffrement rend impossible (ou quasi impossible) le déchiffrement sans connaître la clé.

Le chiffrement peut se faire en tous points du réseau (comme le montre la figure ci-dessous) ou aux extrémités seulement. Il faut faire attention, le chiffrement est une opération qui a un coût. La mettre partout rend le réseau plus complexe, et éventuellement plus lent.



Points d'implantation des mécanismes de chiffrement.

Il existe différentes formes d'algorithmes de chiffrement qui traitent des blocs de texte d'origine pour produire des blocs chiffrés de même taille. On trouve dans cette catégorie : le DES (Data encryption standard), le Triple DES (3DES) et le AES (Advanced Encryption Standard).

DES : Data Encryption standard

- C'est un standard US.
- Il traite des blocs de 64 bits.
- Il utilise une clé de 56 bits.
- Il a été cassé en 1998 (en moins de trois jours par 'Electronic Frontier Foundation' avec une architecture matérielle puissante).
- Il n'est plus utilisé actuellement.

Triple DES :

- Introduit en 1999.
- Il utilise 3 clés et 3 exécutions de l'algorithme utilisé par DES.
- Il utilise des clés de 112 ou 128 bits.
- Il traite des blocs de 64 bits.
- Il est lent en termes d'exécution.

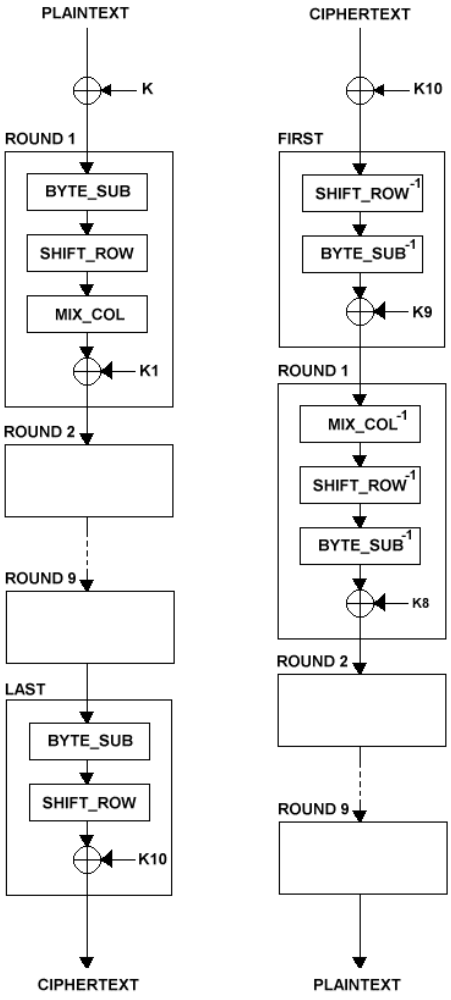
EAS : Advanced Encryption Standard

- Introduit par le National Institute of Standards and Technology (NIST) en 1997.
- Il a des performances meilleures ou équivalentes à celles de Triple DES.
- Il traite des blocs de 128 bits.
- Il utilise des clés de 128, 192 et 256 bits.
- En 2001, AES a été adopté comme “federal information processing standard” (FIPS #197)

II.1.2. Principe de fonctionnement de EAS

La plupart des algorithmes de chiffrement symétrique utilisent le chiffrement en utilisant deux types de transformation : substitution et permutation. Un round consiste à effectuer ces deux transformations. Plus le nombre de rounds est élevé, plus la sécurité apportée est grande. Mais attention, plus il y a de rounds plus il y a de consommation de ressources, de plus le nombre de rounds n’est pas le seul critère pour distinguer les algorithmes.
Par exemple, DES applique 16 rounds. 3DES applique 48 rounds. AES applique un nombre de rounds qui dépend de la taille de la clé : 10 rounds pour une clé de 128 bits, 14 rounds pour une clé de 256 bits.

La figure suivante montre le principe simplifié de AES à 10 rounds. L'AES opère sur des blocs de 128 bits (plaintext P) qu'il transforme en blocs cryptés de 128 bits (C) par une séquence d’opérations, en 10 rounds, à partir d'une clé de 128 bits. Dans chaque round :



Principe de fonctionnement de AES.

- BYTE_SUB (Byte Substitution) est une fonction non linéaire opérant indépendamment sur chaque bloc à partir d'une table dite de substitution.
- SHIFT_ROW est une fonction opérant des décalages (typiquement elle prend l'entrée en 4 morceaux de 4 octets et opère des décalages vers la gauche de 0, 1, 2 et 3 octets pour les morceaux 1, 2, 3 et 4 respectivement).
- MIX_COL est une fonction qui transforme chaque octet d'entrée en une combinaison linéaire d'octets d'entrée et qui peut être exprimée mathématiquement par un produit matriciel.
- le $\oplus$ désigne le OU exclusif.
- Ki est la ième sous-clé calculée par un algorithme à partir de la clé principale K.

Le déchiffrement consiste à appliquer les opérations inverses, dans l'ordre inverse et avec des sous-clés également dans l'ordre inverse.

Si on n'a pas de clé, même si l'implantation des fonctions BYTE_SUB, SHIFT_ROW et MIX_COL est connue, chercher la décrypter un message en tâtonnant devient impossible à partir d'une certaine taille de clé.

II.1.3. Distribution des clés dans les systèmes symétriques

Il y a plusieurs manières de distribuer les clés, chacune a ses avantages et inconvénients :

- La clé est sélectionnée par la source et communiquée au destinataire.
- Une tierce partie sélectionne la clé et la communique aux deux correspondants.
- Utiliser une ancienne clé pour chiffrer (par la source) et transmettre la nouvelle clé au destinataire.
- Utiliser une ancienne clé pour chiffrer (par une tierce partie) et communiquer la clé aux deux correspondants.
- ...

Lorsque la clé est distribuée par une tierce partie (souvent appelé distributeur de clé), au moment de l'établissement de connexion pour échanger des données, les deux correspondants s'adressent à cette tierce partie pour obtenir la clé. A la fin de la session, la clé est détruite. Cette technique est utilisée dans le commerce électronique pour effectuer des paiements par carte de crédit. La difficulté réside ici dans la procédure de notification de la clé aux deux correspondants (et si un intrus intercepte la clé).

II.2. Système de chiffrement asymétrique : clés privée et publique

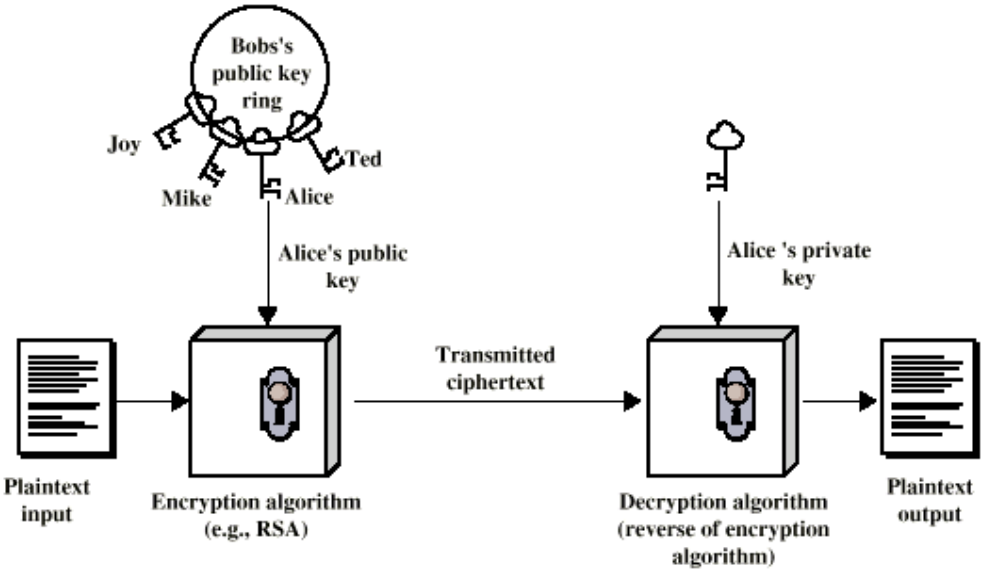
II.2.1. Principe de fonctionnement

Il s'agit de stratégies basées sur des algorithmes mathématiques qui évitent le problème de communication de clé (qui est le point faible du chiffrement à base de clé symétrique). Il s'agit de chiffrement dit asymétrique.

Le système fonctionne à deux clés : une clé publique (connue de tous) et une clé privée (secrète). Pour donner une image (voir figure suivante), la clé publique verrouille le cadenas (donc tout le monde peut le faire moyennant un minimum d'apprentissage selon le type de cadenas) et la clé secrète le déverrouille (seul le propriétaire peut le faire).

Le chiffrement à deux clés a un très grand avantage du point de vue de l'échange des clés. Car on ne publie que les clés publiques sans avoir à se cacher (tout le monde, ou presque, sait comment fermer un cadenas) alors que l'échange des clés de chiffrement symétrique doit se faire dans le plus grand secret (la même clé sert à chiffrer et déchiffrer), et si votre correspondant est à l'autre bout du monde ce n'est pas évident de lui transmettre la clé.

a) Chiffrement asymétrique pour garantir la confidentialité
Pour qu'un seul un récepteur puisse lire le contenu d'un message chiffré, l'émetteur utilise la clé publique de du récepteur pour chiffrer le message à transmettre, ensuite le récepteur utilise sa clé privée pour déchiffrer.



Principe du chiffrement asymétrique pour garantir la confidentialité

b) Chiffrement asymétrique pour garantir l'intégrité
Pour qu'un message ne soit pas modifié, l'émetteur utilise sa clé privée pour chiffrer le message, ensuite le ou les récepteurs utilisent la clé publique pour déchiffrer. Comme la clé privée n'est pas connue par les récepteurs, aucun site (autre que la source) ne peut modifier le contenu des messages émis, les chiffrer et les transmettre ensuite.

II.2.2. Algorithme RSA

RSA (du nom de ses auteurs Rivest, Shamir et Adleman) a été proposé en 1978. Il est devenu un standard de fait dans le monde Internet (le RFC 2437 de l'IETF définit en détail la spécification de RSA). Actuellement, RSA est l'algorithme le plus connu et le plus utilisé en cryptographie.

Son principe est le suivant :

1. Génération de clés privée et publiques
   - On choisit deux nombres premiers entiers très grands p et q.
   - On calcule $n = p*q$.
   - On choisit un nombre entier d premier avec $(p-1)*(q-1)$, tel que $1 < d < (p-1)*(q-1)$
   - On choisit un nombre entier e tel que : $e*d \equiv 1 \bmod[(p-1)*(q-1)]$.
   - <d, n> représente la clé privée, qui sert à décrypter, et qui doit rester secrète.
   - <e, n> représente la clé publique qui sert à crypter
   - Une fois les clés générées, p et q ne sont plus utiles.

2. Chiffrement :
   Considérons un message M tel que sa valeur numérique est strictement inférieure à n.

C, le message chiffré à partir de *M*, est obtenu par :

$C = M^e \text{ modulo } n$

Si la valeur numérique du message à chiffrer est supérieure à n, il faut découper le message en plusieurs blocs (ayant des valeurs inférieures à n) et chiffrer les blocs séparément.

3. Déchiffrement :

Considérons un message *C*, chiffré selon le calcul ci-dessus, alors on démontre que le message clair M est obtenu par la formule suivante par celui qui connaît *d* (*n* est déjà publique) :

$M = C^d \text{ modulo } n = (M^e \text{ modulo } n)^d \text{ modulo } n$

La formule précédente se démontre en utilisant une propriété mathématique de Euler et Fermat sur les nombres premiers.

La sécurité du RSA repose sur le fait que l'on ne connaît pas d'algorithme efficace (qui trouve la solution au bout de quelques minutes, voire même quelques semaines) de décomposition en facteur premier lorsque les nombres sont grands (plusieurs centaines de bits). Ainsi, même si on connaît *n* et *e*, on ne peut pas trouver rapidement *p*, *q* et *d*. A titre indicatif, un des bons algorithmes de référence pour factoriser un nombre *n* en nombres premiers a une complexité de l'ordre de $O(n*ln(n))$; $ln(x)$ désigne le log népérien de x). Ainsi, en utilisant un ordinateur de bas de gamme, pour factoriser un nombre de (en chiffres décimaux) :

- 50 chiffres, il faut 4 heures,
- 74 chiffres, il faut 104 jours,
- 100 chiffres, il faut 74 années
- 200 chiffres, il faut 4 milliards d'années
- 300 chiffres, $5*10^{15}$ années.

Même en tenant compte de l'évolution de la vitesse des processeurs, factoriser un nombre ayant plus de 100 chiffres devient une opération infaisable pour le commun des mortels. Plus on augmente la taille de *n* plus on rend impossible la factorisation de *n* même avec les calculateurs de la Nasa.

Exemple d'utilisation de RSA

Considérons le cas suivant :

1) choix des clés

- p = 3 et q = 11 ==> n = 33
- (p-1)*(q-1) = 20
- On peut choisir d = 3, il est premier avec 20.
- On peut choisir e = 7, car 3*7 est premier avec 20.

2) Chiffrement :

- Par exemple, on veut envoyer un message clair contenant la valeur 18 (remarquez que cette valeur est inférieure à n=33, sinon l'algorithme RSA n'est pas applicable).
- 18^7 modulo 33 = 612 220 032 modulo 33 = 06.
- On transmet le message chiffré égal à 06.

3) Déchiffrement :

- On reçoit la valeur 06.
- 06^3 modulo 33 = 216 modulo 33 = 18

III. Authentification

L'authentification des utilisateurs est l'un des prérequis de la sécurité des systèmes informatiques. En effet, il est nécessaire de reconnaître et distinguer les différents utilisateurs pour savoir à quelles informations ils ont le droit d'accès.

L'authentification d'un utilisateur consiste à l'identifier par une identité (unique) et à vérifier son identité. En général, l'identité (par exemple le nom de login) n'est pas une information confidentielle. Pour vérifier l'identité affichée par un utilisateur (et contrôler ainsi l'usurpation d'identité), le système doit demander à l'utilisateur une information qu'il connaît et que les autres utilisateurs ne connaissent pas. La vérification d'identité peut se faire par divers mécanismes plus au moins sophistiqués, plus au moins sûrs, plus au moins coûteux ... La complexité et le coût d'une stratégie d'authentification sont importants et doivent être pris en compte dès la conception de cette stratégie, pour éviter d'être confronté à des problèmes du type : le système est trop lent, le système nous coûte très cher en maintenance...

III.1. Vérification par mot de passe

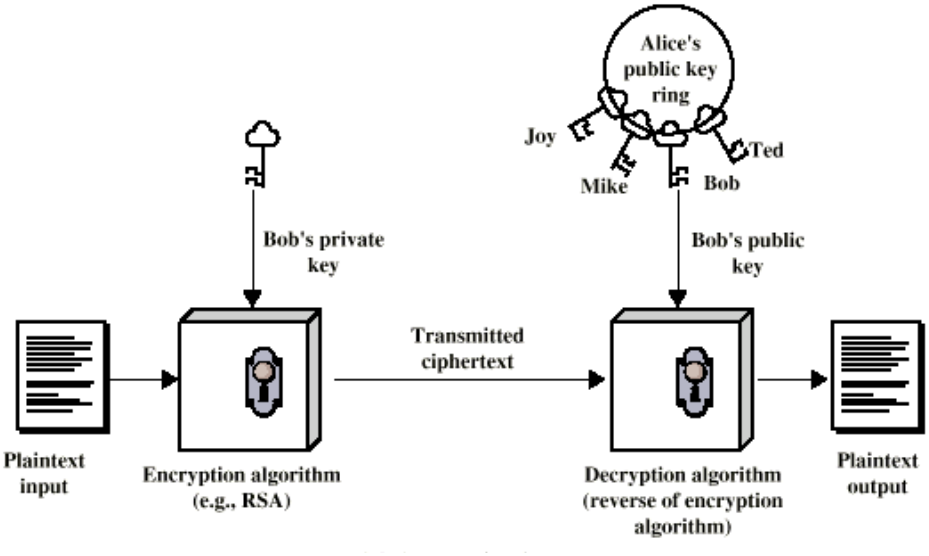
Dans ce cas, on parle d'authentification faible). La faiblesse de cette stratégie réside dans le choix et la longueur des mots de passe. En effet, au bout d'un délai raisonnable, on peut trouver un mot de passe. Par ailleurs, le système de vérification doit protéger le fichier qui contient les mots de passe. Par le passé, des fichiers de mots de passe de certains ordinateurs ont été piratés. Les principales attaques sur les mots de passe sont de trois types : rejeu de mot de passe (après l'avoir usurpé moyennant une écoute ou autre), recherche exhaustive de mots de passe (c'est une longue procédure, mais si l'attaquant est patient, il finira par trouver), attaque par dictionnaire (connaissant un utilisateur, son chien, ses goûts, les sports qu'il pratique, les films qu'il aime le plus ... on peut trouver son mot de passe). Une façon de ne pas se faire voler son mot de passe, c'est de le changer fréquemment. Certains systèmes vont même jusqu'à fournir un mot de passe aléatoire valable pour une seule connexion, avant de se connecter l'utilisateur demande un nouveau mot de passe (c'est ce que l'on appelle One-time password qui un standard de fait préconisé par l'IETF).

III.2. Vérification par secret partagé (question-réponse)

Le vérificateur pose une question dont (normalement) seul celui qui affiche une identité connaît la réponse. Si la réponse envoyée est la bonne, le système considère que l'authentification a réussi. Les questions et réponses associées dépendent du contexte (par exemple, on peut poser des questions de mécanique quantique à un physicien et le titre de son dessin animé favori à un enfant). Il est important de protéger la communication entre le vérificateur à la personne à authentifier afin d'assurer la protection contre le rejeu. Dans la pratique, se pose ici le problème de la distribution du secret entre le vérificateur et la personne à authentifier. Pour résoudre ce problème, on peut faire appel à un serveur d'authentification qui gère les secrets (au lieu de gérer des secrets entre chaque utilisateur et tous les autres utilisateurs, chaque utilisateur gère son secret avec le serveur d'authentification seulement). Quand une personne veut s'authentifier, cette dernière contacte le serveur d'authentification. Le système Kerberos est un exemple très utilisé de serveur d'authentification.

III.3. Vérification à l'aide d'identité signée par une clé privée

Chaque utilisateur a une clé privée qui lui permet de crypter ses messages et il donne sa clé publique à tous ceux avec qui il peut communiquer. Lorsqu'il veut communiquer avec un correspondant, il lui envoie son identité chiffrée (signée) avec sa clé privée. Son correspondant décrypte le message reçu avec la clé publique et trouve l'identité correcte. Voir figure ci-dessous.



Principe d'authentification par signature à l'aide de clé privée.

III.4. Vérification à l'aide de carte à puce

On utilise une carte à puce qui contient un code secret (dans ce cas, il ne faut pas que cette carte tombe entre de mauvaises mains). C'est une solution peu pratique, car elle nécessite un matériel supplémentaire pour lire la carte à puce.

III.5. Vérification en utilisant des informations biométriques

Il semble que les mécanismes précédents n'offrent pas de sécurité absolue (certains écrivent leur code secret sur un papier, comme le PIN de la carte bancaire). Si le contexte l'exige (c'est le cas des systèmes hautement critiques), on peur passer à un niveau plus personnel de l'authentification en utilisant des informations physiologiques ou des traits de comportement des individus : signature manuelle, vitesse de frappe, forme de la main, empreintes digitales, empreintes rétiniennes, empreintes vocales, forme du visage... Ce type de vérification est très coûteux et peut aussi être mis à mal par des attaquants ('criminels') à la 'James Bond'.

IV. Intégrité, signature, tiers de confiance

IV.1. Intégrité

L'intégrité signifie la prévention de modification non autorisée de l'information. Elle ne signifie pas nécessairement la confidentialité, par conséquent l'intégrité ne se gère pas nécessairement en faisant appel aux mécanismes de chiffrement. Par exemple, votre date de naissance n'est pas confidentielle, mais vous ne souhaitez pas qu'elle soit modifiée (de manière volontaire ou involontaire).

Pour assurer l'intégrité d'information, on peut procéder de différentes manières (non nécessairement exclusives) :

- ***Redondance*** : on rajoute de l'information (comme le bit de parité, ou écrire un nombre en chiffres et en lettres...) pour détecter les altérations. Mais ce mécanisme a une faiblesse : toute personne qui connaît le mécanisme de contrôle de redondance utilisé, peut transformer le message en faisant attention aux informations de redondance.

- ***Chiffrement*** : on chiffre l'information claire à l'aide d'une clé, ainsi pour modifier le message en clair il faut connaître la clé. Le problème c'est qu'il faut s'entendre sur la clé avec les personnes autorisées à lire l'information initiale. Si on utilise une clé symétrique, une personne autorisée (mais malicieuse) peut changer l'information initiale et enverra des messages modifiés sans que les autres s'en rendent compte. Résoudre le problème de l'intégrité par le chiffrement n'est pas nécessairement la meilleure solution.
- ***Techniques de hachage*** : une fonction de hachage a sens unique permet de générer une empreinte de taille fixe de n bits (n = 128 par exemple) à partir d'un message de taille quelconque. La fonction doit être choisie de sorte que la probabilité que deux messages différents aient la même empreinte est quasiment nulle. De cette façon, une fois que l'empreinte est fixée, il est impossible de trouver un autre message mais en gardant la même empreinte. Les fonctions de hachage ne reposent sur aucun secret. Des fonctions de hachage telles que MD5, SHA-1 et SH-2 sont d'usage courant.

IV.2. Fonctions de hachage

IV.2.1. Principe du hachage

Une fonction de hachage (parfois appelée *fonction de condensation*) est une fonction permettant d'obtenir un condensé[1] (appelé aussi *haché* ou *empreinte*) d'un message, c'est-à-dire une suite de bits (le plus souvent nettement plus courte que le message initial) représentant le message qu'il condense. La fonction de hachage doit être conçue de manière à ce que les deux propriétés suivantes soient garanties :

- Détection des altérations de message : la fonction de hachage associe un et un seul haché à chaque message. Un bon système de hachage devrait garantir que la moindre modification du message entraîne la modification de son haché. En réalité, cette propriété dépend de la taille du haché. En effet, on peut avoir des messages différents qui ont le même haché. Si un attaquant substitue un message *M'* à un message *M*, ayant tous les deux le même haché, l'altération n'est pas détectée. En effet, si la taille du haché est de n bits, il y a 2^n hachés différents. Par conséquent, si le nombre de valeurs possibles de messages est supérieur à 2^n, des messages différents peuvent avoir le même haché. Plus le nombre de bits d'un haché est élevé, plus faible est la probabilité d'avoir deux messages avec le même haché (et donc plus faible est la probabilité d'avoir des altérations de messages non détectées). C'est pour cette raison que les standards de sécurité sont revus régulièrement (en changeant de version) pour augmenter les tailles des hachés afin parer les attaques rendues possibles grâce aux progrès technologiques (notamment de calcul). Trouver, à l'aide d'un programme de recherche exhaustive, deux messages différents qui ont un même haché de 512 bits est quasiment impossible aujourd'hui.
- Impossibilité de reconstitution de message : la fonction de hachage doit être une fonction à sens unique afin qu'il soit impossible de retrouver le message original à partir du haché. C'est un peu comme pour l'empreinte digitale, la connaissance de cette empreinte ne permet pas à elle seule de remonter à l'individu, ni de reconstituer cet individu. Il faut que la correspondance ait été préalablement établie dans une base de données pour que l'identification puisse avoir lieu par comparaison.

Les algorithmes de hachage les plus utilisés actuellement sont :

- MD5 (*Message Digest 5*) qui génère une empreinte digitale de 128 bits. Il est courant de voir des documents en téléchargement sur Internet accompagnés d'un fichier MD5, il s'agit du condensé du document permettant de vérifier l'intégrité de ce dernier.
- SHA (*Secure Hash Algorithm*) qui génère des empreintes d'une longueur de 160 bits (pour le protocole SHA-1) et 256, 384 ou 512 bits (pour le protocole SHA-2).

[1] Le condensé devrait vous rappeler la notion de CRC utilisée pour protéger les trames contre les erreurs de transmission. Dans le cas de la sécurité, les hachés sont utilisés pour se protéger contre les altérations volontaires effectuées par des attaquants, alors que dans la protection contre les erreurs de transmission, il s'agit de détecter les altérations (involontaires) dues à des bruits et parasites sur les voies de communication.

Les fonctions de hachage se distinguent sur la taille des empreintes, mais aussi sur le nombre d'opérations qu'elles nécessitent (donc leur temps de calcul).

IV.2.2. Utilisation du hachage en sécurité

Il existe de nombreuses applications du hachage dans les systèmes de sécurité ; les plus répandues sont la protection de l'intégrité des données et l'authentification.

a) Utilisation d'empreinte pour l'intégrité de message

En transmettant un message accompagné de son haché, il est possible de garantir l'intégrité du message, c'est-à-dire que le destinataire peut vérifier que le message n'a pas été altéré (intentionnellement ou de manière fortuite) durant la communication. A la réception du message, il suffit au destinataire de calculer le haché du message reçu et de le comparer avec le haché accompagnant le document. Si le message (ou le haché) a été falsifié durant la communication, les deux empreintes ne sont plus égales.

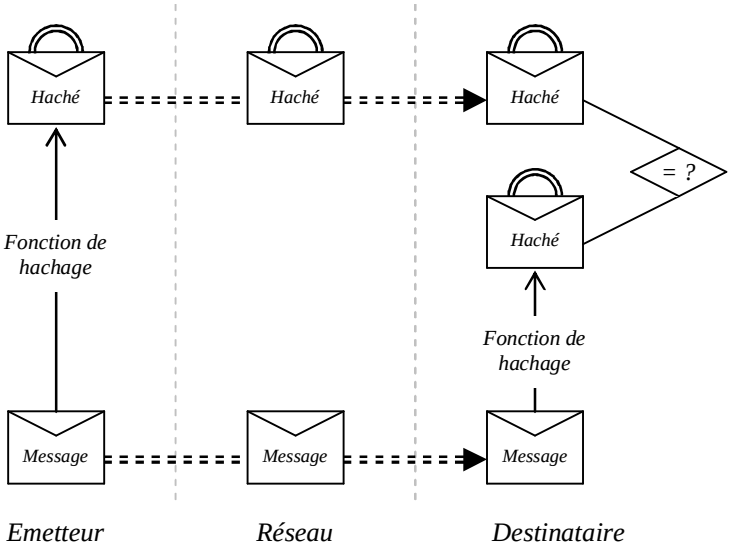


Figure Utilisation de haché pour garantir l'intégrité de message

b) Utilisation d'empreinte pour l'authentification

L'utilisation d'une fonction de hachage permet d'assurer de l'intégrité du message, mais rien ne prouve que le message a bien été envoyé par celui que l'on croit être l'expéditeur.

Ainsi, pour garantir l'authenticité du message, il suffit à l'expéditeur de chiffrer (c'est-à-dire de *signer*) le haché à l'aide de sa clé privée (on obtient un *sceau*) et d'envoyer le sceau au destinataire. A réception du message, il suffit au destinataire de déchiffrer le sceau avec la clé publique de l'expéditeur, puis de comparer le haché obtenu avec la fonction de hachage au haché reçu en pièce jointe.

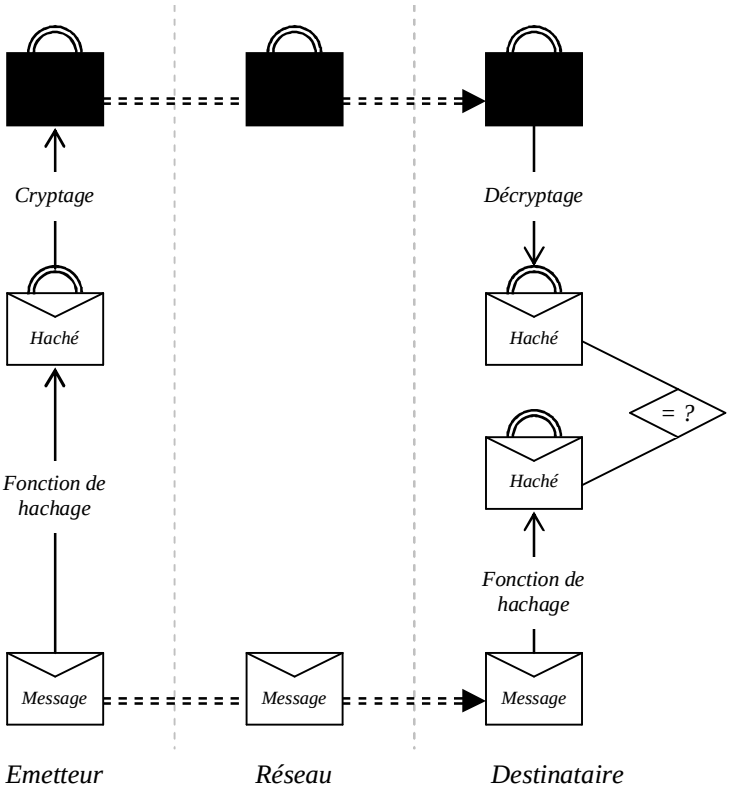


Figure Utilisation de haché pour garantir l'authenticité de message

IV.2.3 Présentation du MD5

a) Principe de base

MD5 (Message Digest 5), est le protocole qui a succédé au MD4. MD5, comme MD4, a été proposé par Ron Rivest, l'un des trois co-auteurs de RSA, et est devenu ensuite un standard dans Internet [Riv92]. Le MD5 permet de produire des hachés (empreintes) de message de 128 bits pour des messages de n'importe quelle taille.

Le MD5 est conçu de telle sorte qu'il est impossible[2] (ou plus exactement très peu probable) de :

1) trouver deux messages différents qui ont la même empreinte (il faut environ 2^{64} opérations pour trouver deux messages ayant la même empreinte).

2) pouvoir retrouver un message à partir de son empreinte (dans certains cas limités, il faut environ 2^{128} opérations pour retrouver un message si on connaît son empreinte).

Le MD5 est destiné à être utilisé en particulier dans les applications qui ont besoin de signatures pour s'assurer de l'authenticité des messages.

b) Fonctionnement de l'algorithme MD5

Les messages manipulés par MD5 sont constitués de mots de 32 bits chacun. Un mot contient quatre octets.

Soit M un message à b bits pour lequel on souhaite calculer une empreinte. M est représenté sous la forme d'une chaîne de bits $m_0 m_1 \ldots m_{b-1}$
La fonction de hachage fonctionne en cinq étapes (résumées par la figure suivante) :

[2] *Avec les moyens de calcul actuels, MD5 n'est plus considéré comme sûr, on lui préfère l'algorithme SHA-2.*

Etape 1 : Extension de message
Le message *M* est étendu de telle sorte que sa longueur (en bits) soit congruente à 448 modulo 512. L'extension est réalisée de la manière suivante : un seul bit *1* est rajouté au message initial *M* et ensuite une suite de bits *0* est rajoutée jusqu'à ce que la taille du message étendu soit congruente à 448 modulo 512. Dans tous les cas, on rajoute au moins un bit et au plus 512 bits (l'extension est toujours appliquée même si la valeur initiale de *M* satisfait la contrainte de congruence 448 modulo 512).

Etape 2 : adjonction de la taille de message
La valeur de *b* (qui représente la taille initiale du message *M*), sur 64 bits, est rajoutée (concaténée) au résultat de l'étape 1. Ainsi, le nouveau message *M'* a une taille qui est un multiple de 512. Il est représenté sous forme de *N* mots de 32 bits chacun : *M' = M'[0 .. N-1]*. MD5 travaille itérativement sur des blocs de 16 mots consécutifs (c'est-à-dire 512 bits consécutifs).

Etape : Initialisation du tampon MD
Un tampon à quatre mots, notés *A*, *B*, *C* et *D*, est utilisé pour calculer l'empreinte. Ces mots sont initialisés par les valeurs hexadécimales suivantes :

```
A ← 01 23 45 67
B ← 89 ab cd ef
C ← fe dc ba 98
D ← 76 54 32 10
```

Etape 4 : Traitement du message étendu bloc par bloc
Quatre fonctions non linéaires, qui opèrent sur des mots de 32 bits et qui ont comme résultat un mot de 32 bits, sont définies de la manière suivante[3] :

F(X, Y, Z) = (X and Y) or (not(X) and Z)
G(X, Y, Z) = (X and Z) or (Y and not(Z))
H(X, Y, Z) = X xor Y xor Z
I(X, Y ,Z) = Y xor (X or not(Z))

On définit une table *T[i]* de 64 éléments numérotés de *1* à *64*, à partir de la fonction *sinus* :

$$T[i] = \lfloor 4294967296 \times abs(sin(i)) \rfloor$$

abs(a) désigne la valeur absolue de *a*, *sin(α)*, la fonction *sinus(α)* où l'angle *α* est exprimé en radians et $\lfloor z \rfloor$ la partie entière de *z*.
Dans l'algorithme suivant, *X <<< s* désigne la rotation à gauche de *X* de *s* bits.

Algorithme de traitement de MD5 :

```
/* Traiter chacun des blocs du message étendu. Chaque bloc a 16 mots */

Pour i = 0 à N/16 - 1 répéter
     /* Copier le bloc i dans la variable X */
     Pour j = 0 à 15 répéter
             X[j] := M[i*16+j]
     Fin

     /* Sauvegarder les valeurs de A, B, C et D dans AA, BB, CC et DD */
     AA := A
     BB := B
     CC := C
```

[3] *Il est intéressant de noter que ces quatre fonctions ont des propriétés mathématiques (que nous ne développerons pas ici) qui garantissent la robustesse des empreintes générées.*

```
        DD := D

        /************* Round #1 ***************/
        /* On note par [abcd k s i] l'opération :
          a = b + ((a + F(b,c,d) + X[k] + T[i]) <<< s). */
        /* Effectuer les 16 opérations suivantes */
        [ABCD  0  7  1]  [DABC  1 12  2]  [CDAB  2 17  3]  [BCDA  3 22  4]
        [ABCD  4  7  5]  [DABC  5 12  6]  [CDAB  6 17  7]  [BCDA  7 22  8]
        [ABCD  8  7  9]  [DABC  9 12 10]  [CDAB 10 17 11]  [BCDA 11 22 12]
        [ABCD 12  7 13]  [DABC 13 12 14]  [CDAB 14 17 15]  [BCDA 15 22 16]

        /************* Round #2 ***************/
        /* On note par [abcd k s i] l'opération :
           a = b + ((a + G(b,c,d) + X[k] + T[i]) <<< s). */
        /* Effectuer les 16 opérations suivantes */
        [ABCD  1  5 17]  [DABC  6  9 18]  [CDAB 11 14 19]  [BCDA  0 20 20]
        [ABCD  5  5 21]  [DABC 10  9 22]  [CDAB 15 14 23]  [BCDA  4 20 24]
        [ABCD  9  5 25]  [DABC 14  9 26]  [CDAB  3 14 27]  [BCDA  8 20 28]
        [ABCD 13  5 29]  [DABC  2  9 30]  [CDAB  7 14 31]  [BCDA 12 20 32]

        /************* Round #3 ***************/
        /* On note par [abcd k s t] l'opération :
          a = b + ((a + H(b,c,d) + X[k] + T[i]) <<< s). */
        /* Effectuer les 16 opérations suivantes */
        [ABCD  5  4 33]  [DABC  8 11 34]  [CDAB 11 16 35]  [BCDA 14 23 36]
        [ABCD  1  4 37]  [DABC  4 11 38]  [CDAB  7 16 39]  [BCDA 10 23 40]
        [ABCD 13  4 41]  [DABC  0 11 42]  [CDAB  3 16 43]  [BCDA  6 23 44]
        [ABCD  9  4 45]  [DABC 12 11 46]  [CDAB 15 16 47]  [BCDA  2 23 48]

        /************* Round #4 ***************/
        /* On note par [abcd k s t] l'opération :
          a = b + ((a + I(b,c,d) + X[k] + T[i]) <<< s). */
        /* Effectuer les 16 opérations suivantes */
        [ABCD  0  6 49]  [DABC  7 10 50]  [CDAB 14 15 51]  [BCDA  5 21 52]
        [ABCD 12  6 53]  [DABC  3 10 54]  [CDAB 10 15 55]  [BCDA  1 21 56]
        [ABCD  8  6 57]  [DABC 15 10 58]  [CDAB  6 15 59]  [BCDA 13 21 60]
        [ABCD  4  6 61]  [DABC 11 10 62]  [CDAB  2 15 63]  [BCDA  9 21 64]

        /* Effectuer les opérations d'addition (modulo 2^32) suivantes : */
        A = A + AA
        B = B + BB
        C = C + CC
        D = D + DD
    Fin
```

Etape 5 : résultat d'empreinte

L'empreinte calculée est fournie en sortie sous la forme *A, B, C, D*. On commence par l'octet de poids faible du mot *A* et on termine par l'octet de poids fort du mot *D*.

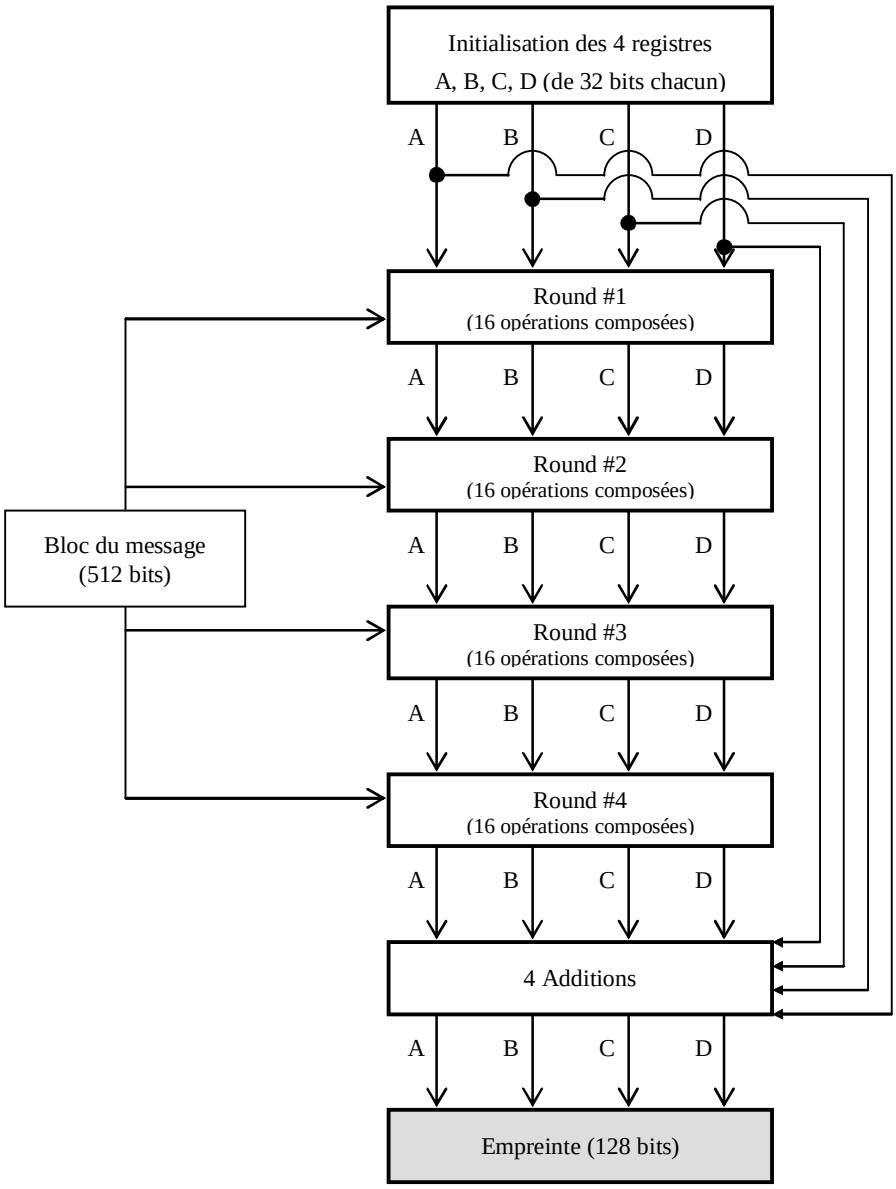


Figure. Principe du MD5 appliqué à un bloc de 512 bits

IV.3 Signature électronique

Pour avoir confiance vis-à-vis de l'information reçue, on peut demander une preuve qu'elle provient bien de celui qui prétend l'avoir envoyée : cette preuve est généralement fournie sous forme de signature.

Une signature peut être retenue, comme preuve, pour attribuer (sans erreur) une information à une source si :

- Elle est attribuée à un seul signataire.
- Elle ne peut pas être imitée.
- Elle doit pouvoir être vérifiable.
- Elle ne doit pas être copiable (on ne peut pas récupérer une signature d'un message valide et l'apposer sur un autre message).
- Elle ne doit pas être répudiable (contestée) par le signataire.

De telles propriétés de signature sont requises dans le domaine de la justice par exemple pour la signature manuscrite. La signature électronique est contrainte par des règles équivalentes mais les mécanismes pour vérifier les propriétés précédemment citées sont différents.

Le paradigme de signature électronique (appelé aussi *signature numérique*) est un procédé permettant de garantir l'authenticité de l'expéditeur (fonction d'*authentification*), ainsi que de vérifier l'intégrité du message reçu.
La signature électronique assure également une fonction de non-répudiation, c'est-à-dire qu'elle permet d'assurer que l'expéditeur a bien envoyé le message (autrement dit elle empêche l'expéditeur de nier avoir expédié le message).

IV.4. Génération de signature

Nous allons présenter deux des techniques pour générer des signatures électroniques.

a) Signature à l'aide de carte à puce
Chaque utilisateur dispose d'une carte à puce qui lui génère la signature. En supposant qu'au niveau matériel, on ne peut ni dupliquer les cartes de signature, ni modifier une carte pour qu'elle change de signature. Cette solution repose sur la confiance que l'on a du fabricant de cartes à puce pour la signature digitale.

b) Signature avec les algorithmes de chiffrement asymétriques
On peut, par exemple, utiliser un algorithme de type RSA avec clé publique et clé privée. La source signe avec sa clé privée (jugée secrète et impossible à deviner par les autres) et tous les autres vérifient la signature avec la clé publique diffusée par cette source. Attention, avec cette solution il faut que la clé publique diffusée par une source ne soit pas interceptée par une personne malicieuse qui la remplacera par la sienne et ensuite elle pourra envoyer des messages qu'elle aura modifiés et signés. Par ailleurs, une source peut renier sa signature. Pour que la signature ne puisse être contestée, il faut réunir toutes les personnes concernées pour qu'elles s'échangent les clés, devant un notaire. Mais une telle procédure est lourde en informatique (qui remplacera le notaire). On peut alléger cette procédure, on fait appel à de tierces parties. Selon l'adage 'les amis de mes amis sont mes amis' on peut faire circuler les clés publiques en passant par des 'amis' (des noeuds en qui on a confiance).

IV.5 Certification de clés, Tiers de confiance

Pour faire confiance aux signatures basées sur les clés publiques, il faut une autorité (incontestable) qui signe les clés publiques avant leur diffusion aux autres. Cette opération est ce que l'on appelle *certification* de clés et son résultat est un *certificat*.

Dans tout système de cryptographie, une des principales difficultés de mise en oeuvre réside dans la génération, distribution et mise à jour des clés. La complexité de cette gestion croit avec le nombre de personnes qui partagent les informations. Pour ce faire, on peut faire appel à une société (un serveur) qui gère les clés. Cette société doit être en mesure :

- de générer une clé de certification pour certifier les clés des utilisateurs et d'assurer la confidentialité de cette clé,
- de recevoir et de contrôler les clés publiques par un canal sûr,
- de vérifier avec rigueur l'identité des utilisateurs et d'indiquer le niveau de contrôle dans les certificats,
- de générer les certificats des clés publiques des utilisateurs,
- gérer les dates de début et de fin de validité de clés,
- gérer les numéros de version de clé pour chaque utilisateur,

- archiver les clés des utilisateurs et pouvoir les restaurer,
- disposer de mécanismes d'alerte pour signaler les clés révoquées,
- disposer d'un mécanisme d'horodatage des transactions.

Une société, qui répond aux critères précédents, constitue ce que l'on appelle une *Tierce partie de confiance* (TTP : Trusted Third Party), tout comme un notaire. Il existe (ou ont existé) de nombreuses sociétés TTP : Véridial (filiale de France télécom) utilisée par les banques, Cerdial permettant la transmission de messages signés dans Transpac, Clipper (USA)...

IV.6 Aspects légaux

Pour que la signature électronique soit retenue comme preuve dans le cadre du commerce électronique ou d'autres opérations par Internet (vote à distance par exemple), il faut un cadre juridique national voire international.

Il faut noter que les aspects juridiques liés à la signature et plus généralement à l'informatique sont complexes et évoluent sans cesse (mais attention : nul n'est censé ignorer). Il faut être constamment à l'écoute des politiques (parlements national et européen) pour être au courant des dernières lois applicables à l'informatique.

Directive européenne sur la signature électronique

En novembre 1999, la commission européenne a adopté une directive définissant un cadre légal pour la reconnaissance de la signature électronique à travers l'union européenne. Cette directive contient les extraits suivants :

Reconnaissance légale de la signature : « ... une signature électronique ne peut être écartée légalement pour la seule raison de sa forme électronique. Si le certificat et le fournisseur de service, de même que la signature utilisée, rencontrent un ensemble de spécifications, la signature sera automatiquement réputée de même valeur qu'une signature manuscrite. De plus, les signatures électroniques acquerront force de preuve dans des procédures judiciaires ».

Libre circulation des biens et services : « ... tous produits et services liés aux signatures électroniques pourront circuler librement et seront soumis seulement à la législation et au contrôle du pays d'origine. Les états-membres ne pourront pas soumettre la prestation de services liés aux signatures électroniques à un régime d'autorisation obligatoire ».

Responsabilité : « ... la législation prévoit un minimum de règles de responsabilité incombant aux fournisseurs de service, en particulier s'agissant de la validité du contenu du certificat. Cette approche assure la libre circulation des certificats et des services de certification au sein du marché intérieur, renforce le sentiment de confiance des consommateurs et encourage les opérateurs à développer des systèmes sûrs, sans qu'il ait de législation par trop restrictive ou contraignante ».

Neutralité technologique : « ... compte tenu de la rapidité des innovations technologiques, la directive prévoit la reconnaissance des signatures électroniques quelque soit la technologie utilisée, signatures digitales ou biométriques... ».

Champ d'application : « ... la législation couvre la délivrance de certificats au public visant à identifier l'expéditeur d'un message électronique ... la législation autorise toutefois le fonctionnement de systèmes régis par des contrats de droit privé ... où une relation de confiance existe déjà ... ».

Séquestre des clés

Jusqu'en 1999, la loi française permet aux utilisateurs d'utiliser des mécanismes de chiffrement fort sous réserve que les conventions secrètes soient stockées (séquestrées) chez une tierce partie de confiance dite tiers de séquestre qui peut être n'importe quelle société qui a les compétences en matière de séquestre de clés. Pour des raisons de sûreté nationale, toutes les clés peuvent être remises aux services d'écoute de l'état, mais sous le contrôle d'un juge. En 1999, la loi a été assouplie et permet à n'importe quel utilisateur d'utiliser librement des clés à condition qu'elles n'aient pas plus de 128 bits.

V. Protocoles et environnements de sécurité

La sécurité est un problème crucial pour tout système informatique. Cependant, tous les utilisateurs n'ont pas besoin des mêmes solutions de sécurité (toute solution de sécurité a un coût). De nombreux travaux ont été entrepris à la fois pour élaborer des standards (ISO ou IETF) et pour proposer des environnements (des produits) sur le marché. Les informations concernant ces aspects sont très nombreuses et nous nous limitons ici à donner quelques informations sur les sigles les plus répandus dans le domaine de la sécurité des réseaux.

V.1. Protocoles de sécurité dans Internet

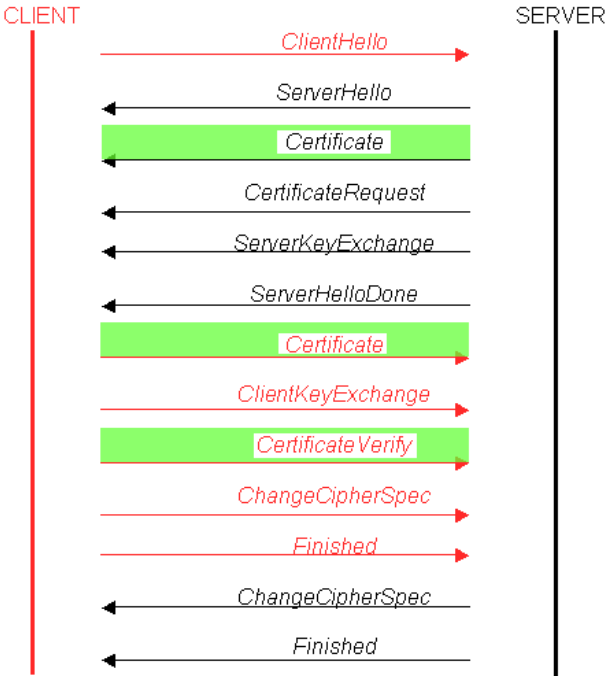
IPsec (IP security) : solution (en termes de protocoles) pour implanter la sécurité au niveau IP (couche 3). IPsec définit en particulier les formats des entêtes IP en présence de mécanismes de sécurité.

SSL (Secure Socket Layer) : c'est un protocole de sécurisation conçu par Netscape qui se situe entre la couche transport (TCP) et les protocoles de la couche application. Il assure les services de sécurit**é** suivantes : confidentialité, intégrité et authentification du serveur et du client. Il repose sur un procédé de cryptographie à clé publique. SSL agit telle une couche supplémentaire, permettant d'assurer la sécurité des données, située entre la couche application et la couche transport (TCP). De cette manière, SSL est transparent pour l'utilisateur (entendez par là qu'il peut ignorer qu'il utilise SSL). Par exemple, un utilisateur utilisant un navigateur Internet pour se connecter à un site de commerce électronique sécurisé par SSL enverra des données chiffrées sans avoir à s'en préoccuper. Presque tous les navigateurs supportent désormais le protocole SSL. *Netscape Navigator* affiche par exemple un cadenas verrouillé pour indiquer la connexion à un site sécurisé par SSL et un cadenas ouvert dans le cas contraire, tandis que *Microsoft Internet Explorer* affiche un cadenas uniquement lors de la connexion à un site sécurisée par SSL.

Les échanges définis par le protocole SSL se déroulent en deux phases:

1) Première phase (authentification du serveur) : suite à la requête d'un client, le serveur envoie son certificat au client et lui liste les algorithmes cryptographiques, qu'il souhaite négocier. Le client vérifie la validité du certificat à l'aide de la clé publique du CA (Certificate Authority) contenue dans le navigateur. Si le certificat est valide, le client génère un pré-master secret (PMS) de 48 octets qui servira à dériver le master secret (MS) de même taille, 48 octets, ce qui représente l'entropie maximale de la clé. Ce PMS est chiffré avec la clé publique du serveur puis transmis à ce dernier. Les données échangées par la suite entre le client et le serveur sont chiffrées et authentifiées à l'aide de clés dérivées de la clé maître.

2) Deuxième phase (authentification (optionnelle) du client) : le serveur (et seulement lui) peut demander au client de s'authentifier en lui demandant tout d'abord son certificat. Le client réplique en envoyant ce certificat puis en signant un message avec sa clé privée (ce message contient des informations sur la session et le contenu de tous les échanges précédents).



Déroulement habituel d'un handshake SSL avec authentification mutuelle

EAP (Extensible Authentication Protocol) : c'est une extension de PPP (Point-to-point protocol) dédiée à l'authentification. PPP commence par établir une liaison logique (de niveau 2). Ensuite, cette liaison logique est utilisée pour réaliser l'authentification. EAP est un protocole d'authentification général (générique) pour supporter d'autres protocoles tels que Kerberos, TLS, SIM... Quatre types de messages sont définis par EAP :

- EAP request : pour demander l'authentification,
- EAP response : pour répondre à une requête d'authentification,
- EAP success : pour indiquer le succès de l'authentification,
- EAP failure : pour indiquer l'échec d'authentification.

EAP définit un protocole minimal, qui fonctionne au-dessus de la couche liaison de données, et il est généralement utilisé par d'autres protocoles, comme TLS ou SIM, pour l'authentification dans les réseaux mobiles. EAP, en lui-même, ne réalise pas l'authentification.

EAP-SIM (EAP- Subscriber Identity Module) : protocole d'authentification pour réseaux mobiles dont les utilisateurs sont équipés de carte SIM (essentiellement celles utilisés par les réseaux GSM).

EAP-TLS (EAP - Transport Layer Security): c'est le protocole qui s'impose de plus en plus pour l'authentification dans les réseaux de mobiles.

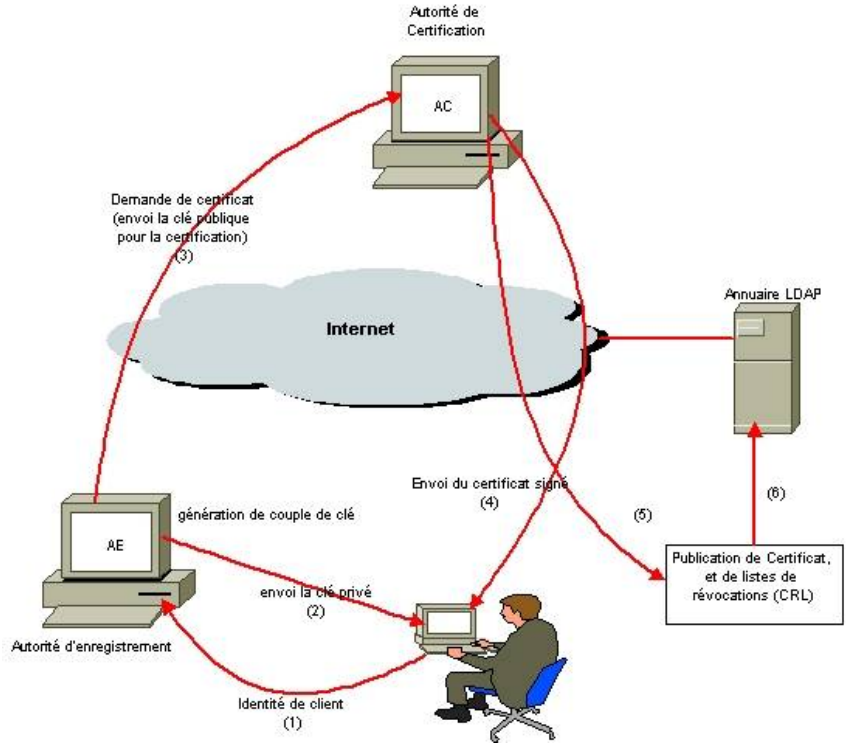
RADIUS (Remote Authentication Dial-In User Server) : c'est le protocole utilisé par les fournisseurs d'accès à Internet pour transporter les paquets EAP, authentifier les utilisateurs, transporter les données de facturation...

V.2. Quelques environnements

PGP (Pretty Good Privacy) : c'est un algorithme qui permet de sécuriser la messagerie électronique en lui apportant authentification et confidentialité. Il permet le chiffrement et signature à base clé symétrique.

PKI (Public Key Infrastructure) est une infrastructure de gestion des clés publiques et certificats (enregistrement de demandes de certificats, création de certificats et leur diffusion, révocation de certificats...).

PKI est généralement déployée au sein d'une même entreprise. La figure suivante montre l'organisation d'une PKI.



Organisation d'une PKI

LDAP = Lightweight Directory Access Protocol

PKCS (Public-Key Cryptography Standards) : c'est un ensemble de standards pour la mise en place des infrastructures de gestion des clés (les PKI). Ces standards définissent les formats des éléments de cryptographie.

Kerberos : c'est un protocole d'authentification fondé sur une cryptographie. Son utilisation ne permet pas à une personne qui écoute la communication d'un client, à son insu, de se faire passer pour lui plus tard (il protège donc contre l'usurpation d'identité). Le mécanisme d'authentification Kerberos V5 émet des tickets d'accès aux services de réseau. Ces tickets contiennent des données cryptées, dont un mot de passe crypté, qui confirment auprès du service demandé l'identité de l'utilisateur. À l'exception du mot de passe qu'il doit fournir ou de son identification par carte à puce, l'utilisateur ne voit rien du processus d'authentification. Dans sa version actuelle (version V5), le processus d'authentification de Kerberos peut être résumé ainsi :

1. À l'aide d'un mot de passe ou d'une carte à puce, l'utilisateur du système client s'authentifie auprès du KDC (Key Distribution Center).
2. Le KDC émet un ticket spécial pour le client. Avec ce ticket, le système client accède au TGS (Ticket Granting Service) qui fait partie du mécanisme d'authentification Kerberos V5 du contrôleur de domaine.
3. Ensuite, le TGS émet un ticket de service à l'intention du client.
4. Le client présente ce ticket de service au service de réseau demandé. Le ticket de service prouve à la fois l'identité de l'utilisateur au service et l'identité du service à l'utilisateur.

VI. Sécurité à l'aide de pare-feux

Un des principes très anciens de sécurité est le cloisonnement : en s'isolant on est protégé (ce n'est pas toujours vrai et l'histoire nous l'a montré à plusieurs reprises). Une des techniques d'isolement (ou plutôt de semi isolement) est celle utilisant les pare-feux.

L'utilisation de pare-feux (firewalls), dits aussi gardes barrière, permet de limiter considérablement les dégâts. Tous les paquets entrants ou sortants doivent transiter par un même passage : le firewall.

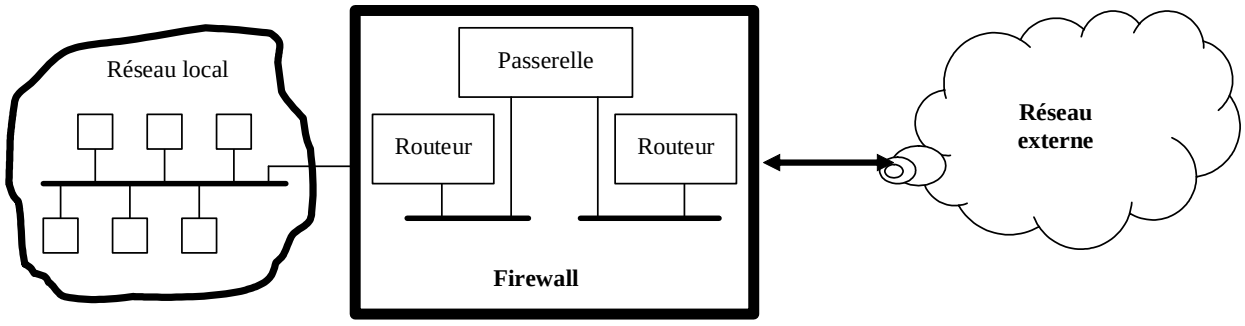
En général, un firewall est composé de :

- un routeur filtrant pour les paquets sortants,
- un routeur filtrant pour les paquets entrants (le filtrage se fait le plus souvent sur les adresses IP seulement),
- une passerelle application (qui analyse plus finement le contenu des messages) pour renforcer la sécurité. Malheureusement, si on fait passer tous les messages par cette passerelle, le système s'en trouverait complètement ralenti (imaginez des gendarmes qui arrêtent, sur une autoroute, tous les véhicules et les fouillent de fond en comble).

L'administrateur de réseau configure :

- les noms et adresses de machines pouvant émettre des paquets vers l'extérieur ou en recevoir,
- les services et fonctions à utiliser par les paquets entrants ou sortants.

Ces informations sont utilisées par le firewall pour assurer la sécurité du système.



Place d'un pare-feu.

Problème : que deviendront les firewalls quand la communication sans fil se généralisera puisque toute station est susceptible de communiquer avec l'extérieur et d'être visitée par autrui ?

Remarque :

Un des domaines très actifs et très complexes de l'informatique est celui de la sécurité. Différents travaux sont en cours pour prendre en compte la sécurité à tous les niveaux : aspects mathématiques et preuve, protocoles, implantation des protocoles et services, politiques de mise en place de sécurité, audit, sensibilisation des personnes à la vigilance, défense du territoire... Il est clair que le problème de la sécurité est aussi vieux que le monde, il y aura toujours des 'méchants' qui chercheront à nuire. Les responsables de systèmes informatiques doivent apprendre les attaques pour les contrecarrer. La sécurité à 100% n'existera jamais, mais il ne faut pas croiser les bras.

Exercices

Exercice 1 (Algorithme RSA)

Prendre p = 3 et q = 5.

Q1. Calculez les clés publique et secrète selon l'algorithme RSA. Ensuite chiffrez deux messages véhiculant les valeurs 3 et 4. Déchiffez les deux messages codés.

Q2. Connaissant *n* et la clé publique, essayez de deviner (retrouver) la clé secrète. Qu'en pensez-vous du temps de recherche de la clé secrète ?

Exercice 2 (Algorithme de partage de secret)

En 1979, Shamir (un des grands spécialistes de la cryptographie) a proposé un algorithme simple permettant à des personnes (ou des ordinateurs) de partager un secret.
On suppose que le secret à partager est une donnée numérique *D*. Le principe de l'algorithme de Shamir est de découper la donnée *D* en *n* morceaux de telle manière que :

- Si une personne connaît (détient) *k* morceaux ou plus de la donnée *D*, elle peut retrouver D.
- Si une personne connaît moins de *k* morceaux de *D*, alors elle ne peut pas retouver *D*.

De manière imagée, c'est comme si on avait une porte qui ferme avec plusieurs clés, et toute personne (ou tout groupe de personnes) qui veut ouvrir la porte doit avoir au moins *k* clés (ce principe est bien connu des militaires).

Pour déterminer les morceaux, on procède de la manière suivante :

- On choisit un polynôme quelconque $q(x)$ de degré $k-1$ ($k \leq n$) tel que :

  $q(x) = D + a_1x + a_2x^2 + ... + a_{k-1}x^{k-1}$ (ce qui veut dire que $a_0 = D$).
- On calcule : $D_1 = q(1), D_2 = q(2) ...., D_n = Q(n)$; où $q(i)$ = la valeur du polynôme q pour $x=i$.
- Les *n* morceaux $D_1, D_2, ..., D_n$ sont distribués (en secret) aux personnes concernées.

Toute personne qui connaît *k* morceaux, avec leurs indices, parmi les morceaux $D_1, D_2, ..., D_n$, peut retrouver *D*.
Dans la pratique, on distribue le nombre de morceaux selon le grade ou l'importance de la personne. Par exemple, pour accéder à une base de données confidentielles, on peut définir un mot de passe ensuite on le traite par l'algorithme de Shamir de manière à obtenir 3 morceaux et on donne un morceau à chaque ingénieur et deux morceaux au directeur. Il faut la présence de trois ingénieurs ou du directeur et d'un ingénieur pour accéder à la base de données.

Q1. Prendre une valeur $D = 17$ ensuite la transformer en 4 morceaux avec $k = 3$.

Q2. En utilisant trois morceaux, montrer sur un exemple que l'on peut retouver la donnée *D*.

Q3. Montrer qu'avec un seul morceau, il est impossible de trouver (mathématiquement) la donnée *D*.

Questions de cours pour tester vos connaissances

1. Rappeler brièvement le modèle OSI (son utilité et les couches qui le composent).
2. Citer les différents supports de transmission utilisés dans les réseaux informatiques.
3. Quels sont les critères de comparaison des supports de transmission ?
4. Quel est le rôle d'un modem ? Un modem est-il toujours obligatoire ? Justifier votre réponse.
5. Quel est le rôle de la fonction de contrôle de flux dans une couche de communication ?
6. Quelles sont les techniques d'acquittement dans les réseaux ?
7. Expliquer le principe de la modulation par impulsions codées.
8. Pourquoi tous les protocoles de liaison de données placent-ils le CRC toujours en fin de trame et jamais au début ?
9. Peut-on assimiler un multiplexeur à un routeur ? Justifiez votre réponse.
10. A quoi sert-elle l'information « durée de vie » dans un paquet IP ? Comment est-elle gérée par IP ?
11. Positionner la structure en couches du modèle TCP/IP par rapport à celle du modèle OSI.
12. A quelles couches du modèle OSI fait référence l'architecture dite TCP/IP ? Quelles sont les principales différences entre la couche réseau du modèle OSI et celle de TCP/IP ?
13. Quelle information est utilisée dans les paquets IP pour leur éviter de boucler indéfiniment dans le réseau ? Comment cette information est-elle gérée ?
14. Comment les techniques de contrôle de flux peuvent-elles contribuer au contrôle de congestion de réseau ?
15. Comparez les principaux avantages et inconvénients du routage fixe et du routage adaptatif.
16. Quels sont les moyens (matériel et logiciel) communément utilisés pour assurer la sécurité dans un réseau ?
17. Expliquez le principe du routage adaptatif.
18. Quel est le rôle de la table de routage dans le protocole IP ?
19. Donnez trois exemples de services de niveau application dans le domaine OSI et quatre dans le domaine Internet.
20. Expliquer le principe d'un pare-feu (ou fire-wall) utilisé dans les réseaux.
21. Donnez un exemple de table de routage dans le cas d'un réseau composé, au moins, de quatre routeurs et dix stations.
22. Expliquez le principe de fragmentation et réassemblage dans IP le long d'un chemin entre une source et une destination de données.
23. Quelles sont les techniques de contrôle de congestion de réseaux ?
24. Quels est le principe du routage dans IP ?
25. Que contient la table de routage d'un routeur dans IP ?
26. Quels sont les champs qui vous paraissent nécessaires dans une trame de données de niveau liaison de données avec un contrôle de flux utilisant une fenêtre d'anticipation de taille n ?

27. Quelles sont les limitations de support non filaires ?
28. Les fragments d'un message peuvent être réassemblés dans le site de destination du message ou dans chaque nœud intermédiaire. Quelle est l'avantage du réassemblage dans la station de destination ?
29. La normalisation et l'ouverture sont deux critères importants pour le choix d'un réseau. Discuter les avantages et les inconvénients d'un réseau ouvert et normalisé.
30. Le contrôle de flux est un mécanisme qui peut être implanté au niveau de toutes les couches du modèle OSI. Dans quels cas a-t-on besoin de contrôle de flux au niveau d'une couche donnée ? Limiter la discussion aux couches 2, 3 et 4.
31. Combien faut-il de temps pour transmettre un fichiers d'un million de caractères ASCII sur un réseau avec un débit de 10 k bits/s ?
32. Dans quel cas la transmission peut s'effectuer sans modulation ? Comment appelle-t-on ce mode de transmission ?
33. Quelles sont les différentes topologies de réseaux ? Discuter leurs avantages et inconvénients ?
34. Quels sont les différents supports de transmission utilisés dans les réseaux informatiques ?
35. Que signifie une classe d'adresse IP ? Quelles sont les classes d'adresse IP ?
36. Citer quelques protocoles utilisés dans Internet.
37. Que veulent dire les sigles HTTP, URL.
38. Que signifie un port TCP.
39. Expliquer le principe du client-serveur.
40. Expliquer le rôle d'un moteur de recherche sur le Web.
41. Expliquer les principales fonctions réalisées par `telnet`.
42. Expliquer le principe de déplacement d'un fichier via `ftp`.
43. Quelles sont les principaux moyens permettant de transférer un fichier d'une machine vers une autre. On suppose que les deux machines appartiennent à la même personne.
44. Peut-on se déplacer dans un répertoire à distance ? Si oui comment ?
45. Quelles différences existe-t-il entre un `ftp` anonyme et un `ftp` normal ?
46. Donner un exemple d'une R-commande et expliquer sa fonction.
47. Quelles sont les principales adresses que l'on spécifie lorsqu'on raccorde un PC à TCP/IP ?
48. Le raccordement d'un PC à un réseau local Ethernet nécessite quoi ? Quels sont les paramètres à fixer ?
49. A quoi sert une adresse IP ? Comment distingue t–on les classes des adresses IP ? Donner un exemple d'adresse de classe A et B.
50. Qu'appelle-t-on serveur de noms ?
51. Que contient le fichier `hosts` dans le système Unix ?
52. A quoi sert la commande `ping` ?
53. A quoi sert la commande `traceroute` ?
54. A quoi sert la commande `tcpdump` ?
55. Quel danger présente la commande `tcpdump` pour les utilisateurs d'un réseau ?
56. Comment simuler les erreurs de transmission si on veut réaliser un protocole avec acquittement et avec anticipation ?

57. Comment simuler la couche physique si on veut implanter un programme émetteur et un programme récepteur sur la même machine.
58. Vous possédez un compte `linfg300` sur `tgv.edu.ups-tlse.fr`. L'un de vos collègues possède le compte `linfg400` sur `corail.cict.fr`. Vous voulez lui faire parvenir le fichier `exo1.c` situé dans votre répertoire (`/home/linfg300/`) sur `tgv`. Votre collègue est au courant et peut éventuellement modifier son fichier `.rhosts`. Parmi tous les outils présentés en Travaux Pratiques, lesquels sont utilisables pour réaliser cette opération ? Lequel vous paraît le plus simple à utiliser ? Expliquez pourquoi.
59. Vous possédez deux comptes `linfg300` sur `tgv` et sur `corail` et vous voulez transférer le fichier `exo2.c` de `tgv` vers `corail`. Parmi tous les outils présentés en Travaux Pratiques, lesquels sont utilisables pour réaliser cette opération ? Lequel vous paraît le plus simple à utiliser ? Expliquez pourquoi.

Quelques ouvrages sur les réseaux

1. Tanenbaum A., *Réseaux : architectures, protocoles et applications*. InterEditions.

2. Pujolle G., Les réseaux. Eyrolles.

3. Comer D., TCP/IP : *architectures, protocoles et applications*. InterEditions.