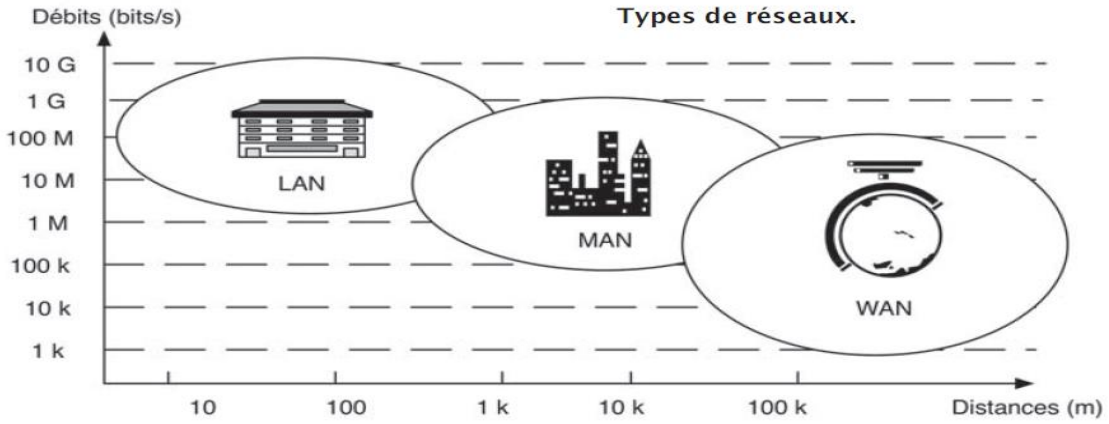
CHAP2 : Notions sur les réseaux informatiques :

I-Introduction générale : Un réseau (Network) est un ensemble d'ordinateurs et d'équipements interconnectés pour échanger des informations ou des données numériques selon des règles bien définies. La connexion entre les différents éléments constitutifs d'un réseau, peut s'effectuer à l'aide de liens permanents comme des câbles, mais aussi au travers des réseaux de télécommunications publics, comme le réseau téléphonique. Ces éléments communiquent entre eux à partir de règles appelées **protocoles**.

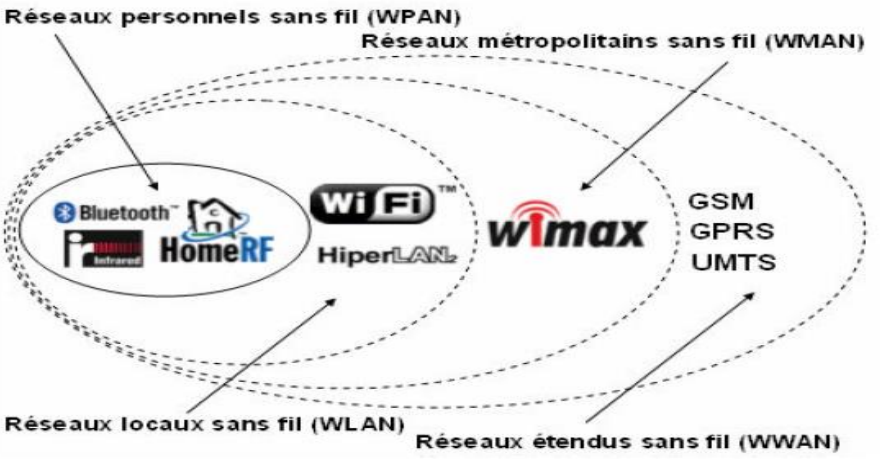
II-Caractéristiques : il n'existe pas de classification universelle des réseaux, mais deux critères importants les caractérisent : La technologie de transmission utilisée ou La taille du réseau :

- **Selon la technologie de transmission** : Diffusion (canal partagé par toutes les machines) ou Point à point (connexion entre machines 2 à2 :P2P)
- **Selon la taille :** WPAN (WirelessPersonnalArea Network), LAN (Local Area etwork) : distance<10km, debit 10Mbps-10Gbps, version 'W', MAN (MetropolitanArea Network):Distance <100km, 56kbps-1Gbps, version 'W'style UMTS : WBWA ou WAN (WideArea Network)

Figure 1:



- **Les réseaux sans fil :** Un réseau sans fil (en anglais wireless network) est, comme son nom l'indique, un réseau dans lequel au moins deux terminaux peuvent communiquer sans liaison filaire. Grâce aux réseaux sans fil, un utilisateur a la possibilité de rester connecté tout en se déplaçant dans un périmètre géographique plus ou moins étendu, c'est la raison pour laquelle on entend parfois parler de "mobilité".



III- **Protocoles réseaux :**

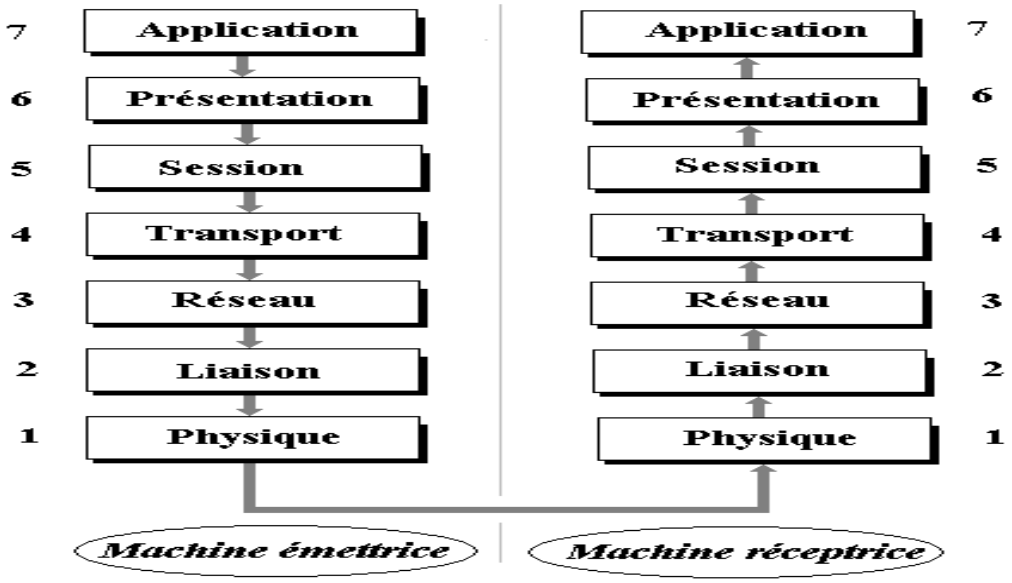
3.1 : Notion d'un protocole : Un **protocole de communication** est un ensemble de règles permettant à plusieurs périphériques, éventuellement sur des réseaux physiques différents et utilisant des OS différents, de dialoguer entre eux.

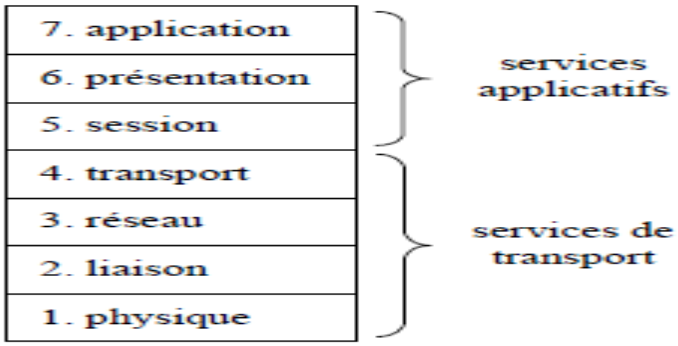
Les suites de protocoles réseau décrivent des processus tels que :

- le format ou la structure du message ;
- la méthode selon laquelle des périphériques réseau partagent des informations sur des chemins avec d'autres réseaux ;
- comment et à quel moment des messages d'erreur et système sont transférés entre des périphériques ;
- la configuration et l'arrêt des sessions de transfert de données.

3.2 : Modèle de référence à sept couches (Modèle ou Norme OSI) : Le modèle OSI (**Open Systems Interconnection**) constitue le modèle de référence inter réseau le plus répandu. Il est utilisé pour la conception de réseaux de données, pour les spécifications de fonctionnement et pour le dépannage (Figure 2)

Figure 2 : Modèle OSI-7 Couches





Chaque couche rend un service décrit dans la documentation de l'ISO et géré par un protocole permettant de réaliser ce service lorsque la couche est abstraite. Lorsque la couche est matérielle la documentation décrit comment le service est rendu par le composant matériel.

Chaque couche de niveau **n** communique avec la couche immédiatement supérieure **n+1** (lorsqu'elle existe) et la couche immédiatement inférieure **n-1** (lorsqu'elle existe).

La couche physique la plus basse est la plus concrète elle est numérotée 1, la couche application la plus haute est la plus abstraite, elle est numérotée 7.

Cette organisation en couche d'abstractions descendantes va se retrouver aussi dans la notion de programmation structurée par abstractions descendantes, il s'agit donc d'un fonctionnement constant de l'esprit des informaticiens.

Nous décrivons brièvement chacune des 7 couches du modèle OSI :

Nom et numéro de la couche	Description du service rendu par la couche
7 - Application	Transfert des fichiers des applications s'exécutant sur l'ordinateur.
6 - Présentation	Codage des données selon un mode approprié.
5 - Session	Gestion des connexions entre les ordinateurs.
4 - Transport	Gestion du transfert des données vers le destinataire.
3 - Réseau	Schéma général d'interconnexion (adressage) afin d'assurer le repérage physique du destinataire.
2 - Liaison	Règles permettant d'effectuer le réassemblage et l'acheminement des données vers le matériel physique de la couche 1.
1 - Physique	Description physique du transport des données à travers des câbles, des hubs…

3.3 : Le modèle TCP/IP : Principe du modèle (Figure 3 et 4): Le premier modèle de protocole en couches pour les communications inter réseau fut créé au début des années 70 et est appelé modèle Internet. Il définit quatre catégories de fonctions qui doivent s'exécuter pour que les communications réussissent. L'architecture de la suite de protocoles TCP/IP suit la structure de ce modèle. Pour cette raison, le modèle Internet est généralement appelé modèle TCP/IP.

Le réseau qui utilise TCP/IP est un réseau à **commutation de paquets (Figure 3)**. Ce type de réseau transmet des informations sous forme de petits groupes d'octets appelés **Paquets**. Si un fichier doit être transmis, il est d'abord fragmenté en paquets à l'émission puis, le fichier est réassemblé en regroupant les paquets à la réception.

Figure 3 : Fonction des couches du modèle TCP/IP.

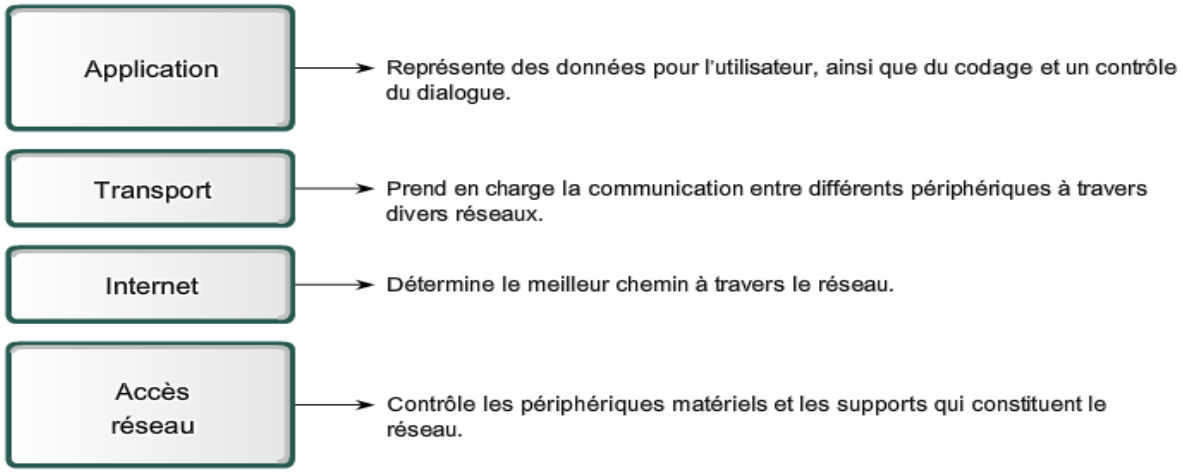


Figure 4 : Comparaison modèle OSI et modèle TCP/IP

IV- Terminologie liée au modèle OSI et TCP/IP

4. 1 : Processus de communication : Le modèle TCP/IP décrit la fonctionnalité des protocoles qui constituent la suite de protocoles TCP/IP. Ces protocoles, qui sont implémentés sur les hôtes émetteurs et récepteurs, interagissent pour fournir une livraison de bout en bout d'applications sur un réseau.

Un processus de communication complet comprend ces étapes :

a. Création de données sur la couche application du périphérique final d'origine
b. Segmentation et encapsulation des données lorsqu'elles descendent la pile de protocoles dans le périphérique final source
c. Génération des données sur les supports au niveau de la couche d'accès au réseau de la pile
d. Transport des données via l'inter réseau, qui est constitué de supports et de n'importe quels périphériques intermédiaires
e. Réception des données au niveau de la couche d'accès au réseau du périphérique final de destination
f. Décapsulation et assemblage des données lorsqu'elles remontent la pile dans le périphérique de destination
g. Transmission de ces données à l'application de destination, au niveau de la couche application du périphérique final de destination

Lorsque les données d'application descendent la pile de protocoles en vue de leur transmission sur le support réseau, différents protocoles ajoutent des informations à chaque niveau. Il s'agit du processus d'encapsulation.

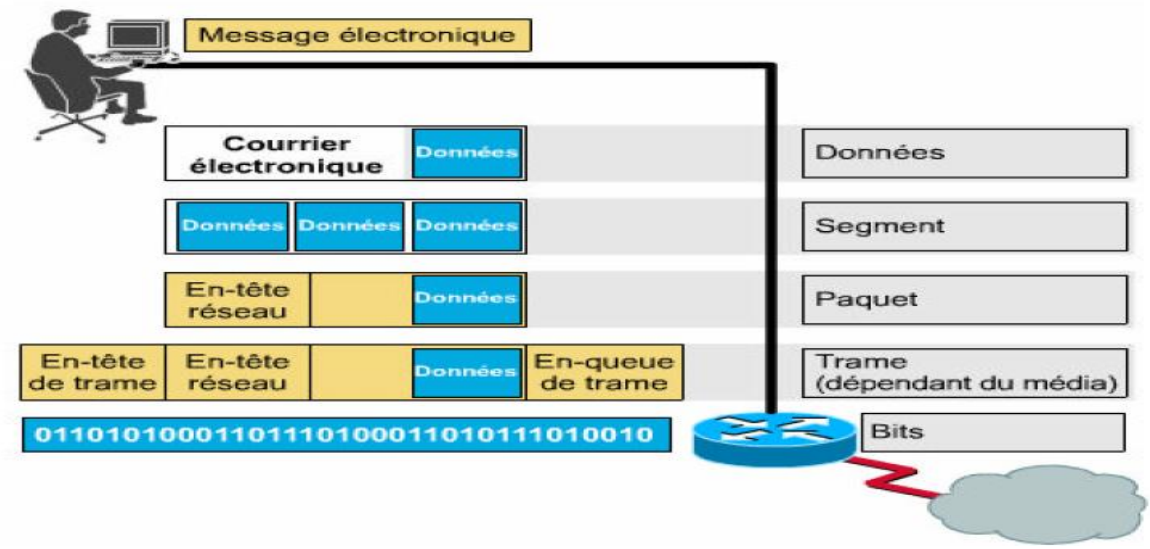
4.2 : Encapsulation (Figure 4) : La forme qu'emprunte une donnée sur n'importe quelle couche est appelée unité de données de protocole. Au cours de l'encapsulation, chaque couche suivante encapsule l'unité de données de protocole qu'elle reçoit de la couche supérieure en respectant le protocole en cours d'utilisation. À chaque étape du processus, une unité de données de protocole possède un nom différent qui reflète sa nouvelle apparence.

Chaque couche ajoute des informations nommées **en-têtes**, destinées à communiquer avec la couche homologue située dans l'ordinateur de l'autre extrémité. Chaque nouveau paquet ainsi formé est inséré dans un paquet de la couche inférieure. Cette opération s'appelle **encapsulation**.

- Les données de l'**application**, avec leur en-tête sont passées à la couche **TCP** qui rajoute le sien. L'ensemble est appelé **segment TCP**.
- L'ensemble des données qu'envoie IP à la couche Ethernet est appelé **Paquet IP**.
- L'ensemble de bits structuré envoyé sur le réseau est une **trame Ethernet**.

L'ensemble des données inclus dans IP aurait pu être un **datagramme UDP**, si l'application utilisait ce type de protocole plutôt que TCP. **Figure 4**

Exemple d'encapsulation des données



Fonctionnement de la pile des protocoles TCP/IP

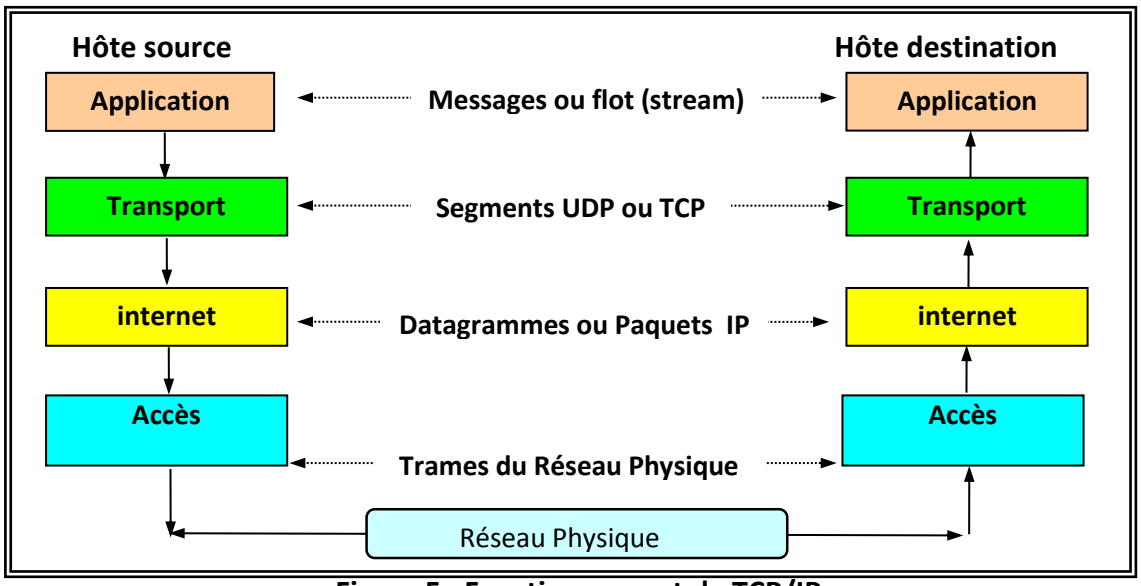


Figure 5 : Fonctionnement de TCP/IP

Message : c'est un regroupement logique de données au niveau de la couche 7 (application), souvent composé d'un certain nombre de groupes logiques de couches inférieures, par exemple des paquets.

Segment : c'est un terme utilisé pour décrire une unité d'information de la couche de transport.

Paquet : c'est un regroupement logique d'informations comportant un en-tête qui contient les données de contrôle et (habituellement) les données utilisateur. Le terme paquets est le plus souvent utilisé pour désigner les unités de données au niveau de la couche réseau

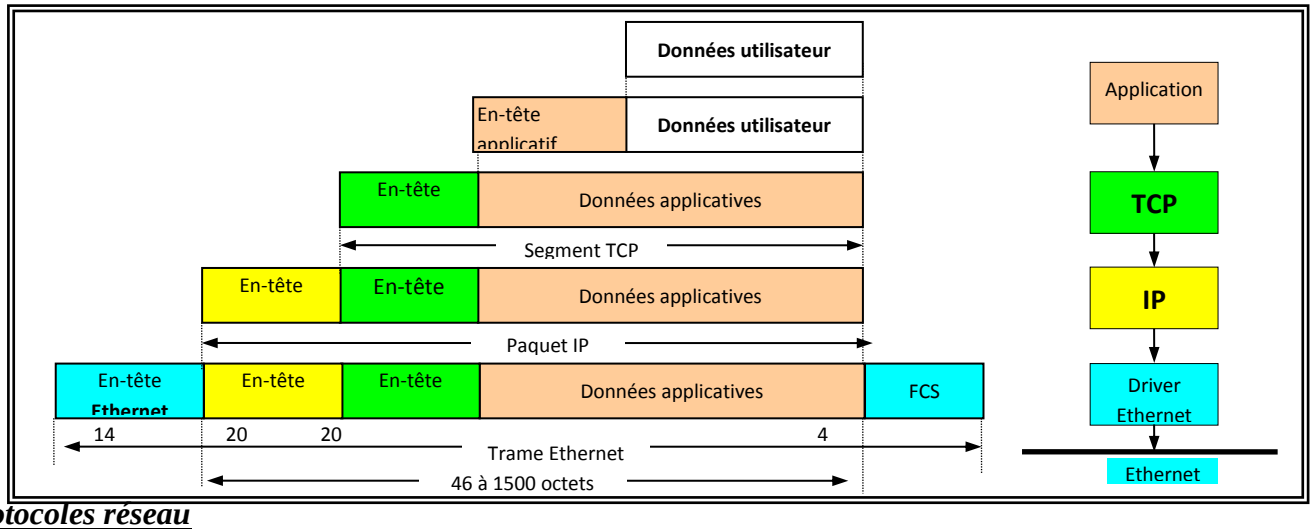
Trame : c'est un regroupement logique de données envoyé comme unité de couche liaison de données par un média de transmission.

Datagramme : c'est un regroupement logique de données envoyé comme unité de couche réseau par un média de transmission, sans établissement préalable d'un circuit virtuel. Les datagrammes IP sont les principales unités d'information sur Internet.

Bits : unité de données de protocole utilisée lors de la transmission physique de données à travers le support

La suite des protocoles appelée aussi **pile de protocoles IP** ne correspond pas au **modèle OSI** de l'**ISO**, celui-ci a été normalisé en 1979, il est donc postérieur à la création de TCP/IP.

La pile de protocoles IP correspond au **modèle DoD** (Department of Defence). Le dessin suivant montre l'équivalence entre ces couches et les différents protocoles de la pile.



Protocoles réseau

IP	Internet Protocol	Fournit les services de communication d'inter-réseau aux clients de la couche 4.
ARP	Address Resolution Protocol	Protocole permettant de faire correspondre une adresse IP à une adresse Physique.
RARP	Reverse ARP	Protocole inverse faisant correspondre une adresse Physique à une adresse IP.
ICMP	Internet Control Message Protocol	Contrôle la transmission des messages d'erreur et des messages entre hôtes, passerelles ou routeurs.
IGMP	Internet Group Management Protocol	Permet d'envoyer des datagrammes à un groupe de machines grâce à un adressage multicast.

Protocoles transport

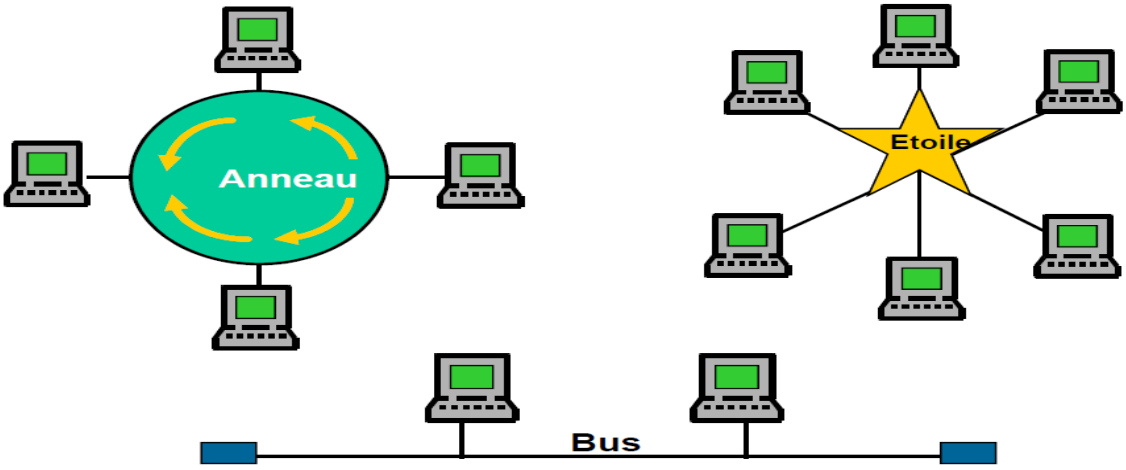
TCP	Transmission Control Protocol	Protocole orienté connexion, fiable et à flot de données.
UDP	User Datagram Protocol	Protocole sans connexion, orienté transaction sans accusé de réception, parallèle à TCP.

On limite l'étude aux protocoles suivants :

- **la couche réseau (Norme Ethernet)**
- **Protocoles de la couche transport UDP et TCP**
- **Les supports de transmission**
- **Codage d'information.**

V-Topologies Réseaux Ethernet(LAN) :

5.1 : Topologie : Pour assurer la communication entre équipements, les entreprises installent des réseaux locaux, souvent désignés par les abréviations RLE ou RLI permettant d'interconnecter de manière relativement simple les différents équipements (micro-ordinateurs, imprimantes, API, capteur…). Il existe une grande variété de réseaux locaux qui se distinguent par leurs structures, leurs protocoles d'accès, leurs supports de transmission et leurs performances.



5.2 : Modes de fonctionnement d'un réseau :

5.2. A : Mode avec connexion : toute communication entre 2 équipements suit le processus suivant:

1. L'émetteur demande l'établissement d'une connexion par
2. l'envoie d'un bloc de données spéciales.
3. Si le récepteur refuse cette connexion la communication
4. n'a pas lieu.
5. Si la connexion est acceptée, elle est établie par mise en
6. place d'un circuit virtuel dans le réseau reliant l'émetteur
7. au récepteur.
8. Les données sont ensuite transférées d'un point à l'autre.
9. La connexion est libérée.

5.2. B : Mode sans connexion : les blocs de données, appelés datagramme, sont émis sans vérifier à l'avance si l'équipement à atteindre, ainsi que les nœuds intérimaires éventuels, sont bien actifs. C'est alors aux équipements gérant le réseau d'acheminer le message étape par étape et en assurant éventuellement sa temporisation jusqu'à ce que le destinataire soit actif. Ce service est celui du courrier postal classique et suit les principes généraux suivants:

1. Le client poste une lettre dans une boite aux lettres
2. Chaque client à une @ propre et une boite aux lettres
3. Le contenu de l'information reste inconnu
4. Les supports du transport sont inconnus de l'utilisateur du service.

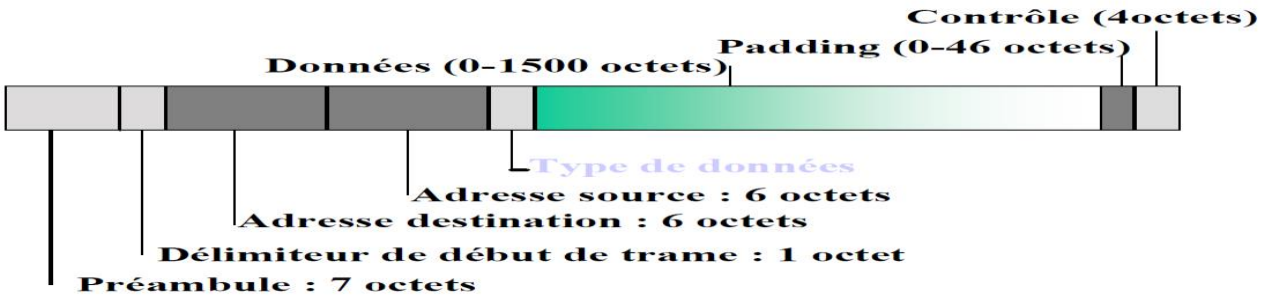
VI-Présentation du protocole Ethernet : Ethernet est un protocole **de réseau local** standardisé sous le nom **IEEE 802.3**. C'est maintenant une norme internationale. La Norme Ethernet définit toutes les règles du réseau : Mécaniques, Electriques et fonctionnelles. Désignation: *XType-Y avec X :* le débit en *Mbps et :* le type de transmission (Base = bande de base) et *Y :* la nature du support (avec la longueur max du brin)

On distingue différentes variantes:

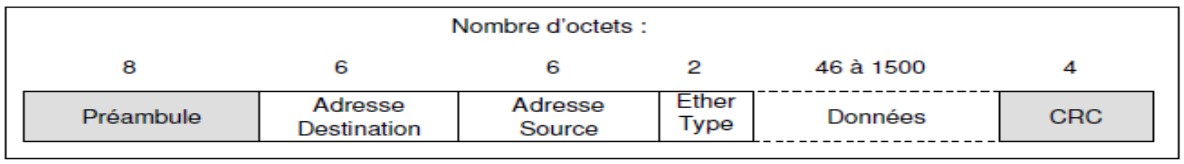
Sigle	Dénomination	Câble	Débit	Portée
10Base-T	Ethernet standard	Paire torsadée (catégorie 3)	10 Mb/s	100m
100Base-TX	Fast Ethernet	Double paire torsadée (catégorie 5)	100 Mb/s	100m
1000Base-T	Ethernet Gigabit	Double paire torsadée (catégorie 5e)	1000 Mb/s	100m
1000Base-LX	Ethernet Gigabit	Fibre optique mono ou multimode	1000 Mb/s	550m
10GBase-SR	Ethernet 10Gigabit	Fibre optique multimode	10 Gbit/s	500m

6.1 : Format de donnée Ethernet ou trame Ethernet : Le format utilisé pour transmettre les données est présenté par la figure suivante :

* **Norme : IEEE 802.2, ISO 8802.2**



Format de la trame Ethernet V2



Description des champs de la trame Ethernet :

- ✓ **Préambule** : (8 octets) : Annonce le début de la trame et permet aux récepteurs de se synchroniser. Il contient 8 octets dont la valeur est 10101010 (on alterne des 1 et des 0), sauf pour le dernier octet dont les 2 derniers bits sont à 1 :56 bits = 7 X (1010101010permet la 'synchronisation bit' et **Délimiteur de début de trame**(StartFrameDelimiter) : 8 bits = 10101011; permet la 'synchronisation trame/caractère'.
- ✓ **Adresse Destination** : (6 octets) : Adresse MAC de l'interface (carte d'accès) Ethernet destinataire de la trame. On représente une adresse Ethernet comme ses 6 octets en hexadécimal séparés par des ':' Exemple **08:00:07:5c:10:0a**

Une seule trame peut avoir plusieurs destinataires. En effet, le format des adresses MAC permet de coder 3 types de destinations :

- **Unicast** : (monodiffusion) un destinataire unique (celui qui possède cette adresse MAC).
- **multicast** : (multidiffusion) un groupe d'interfaces est destinataire. Ce sont des adresses dont le bit de poids faible du premier octet est 1 (exemple : 01:00:5E:00:00:01). Il faut paramétrer la couche Ethernet d'un équipement pour qu'il se reconnaisse dans une adresse multicast (c'est à dire pour faire partie d'un groupe). Un équipement peut faire partie de plusieurs groupes multicast.
- **broadcast** : (diffusion générale) c'est l'adresse ff:ff:ff:ff:ff:ff. Elle correspond à toutes les interfaces Ethernet actives sur un réseau Ethernet (tous les équipements se reconnaissent dans cette adresse)

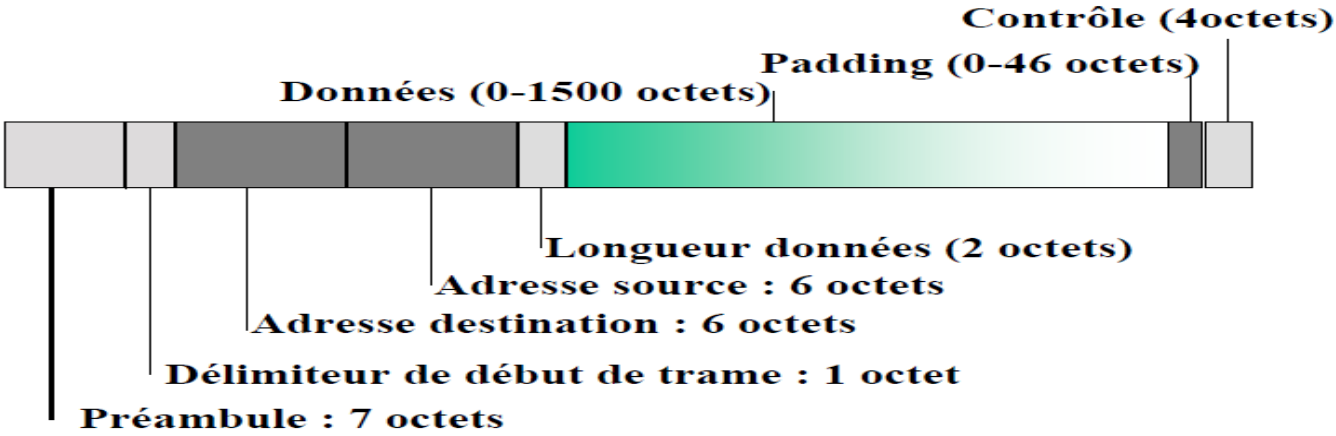
✓ **Adresse Source : (6 octets)** : Adresse MAC de la carte Ethernet émettrice de la trame. C'est forcément une adresse unicast (monodiffusion).

✓ **EtherType : ou type de trame (2 octets)** : Indique quel protocole est concerné par le message. La carte réalise un démultiplexage en fournissant les données au protocole concerné. Quelques types courants (en hexadécimal) :**0x0600 : Xerox Network Systems * 0x0800 : IP (Internet Protocol) 0x0806 : ARP (Address Resolution Protocol) 0x8035 : RARP (Reverse ARP)**

Données : (46 à 1500 octets) : Les données véhiculées par la trame. Sur la station destinataire de la trame, ces octets seront communiqués à l'entité (protocole) indiquée par le champ EtherType. Notons que la taille minimale des données est 46 octets. Des octets de bourrage (padding) à 0, sont utilisés pour compléter des données dont la taille est inférieure à 46 octets. **Padding**: contenu sans signification complétant une trame dont la longueur des données est inférieure à46 octets. (Ces octets de bourrage ne sont pas présents dans les captures de trame)

CRC(Ou FCS) 4 octets : (Frame Check Sequence) : Champ de contrôle de la redondance cyclique. Permet de s'assurer que la trame a **été correctement transmise et** que les données peuvent donc être délivrées au protocole destinataire. Ce champ est recalculé à la réception S'il y a une différence donc erreur transmission Les bits de ce champ sont calculés à partir d'un code cyclique basé sur le polynôme générateur : $\mathbf{G(x)=\sum_{n=0}^{32} x^n}$ **(Voir Annexe 1) (Il ne figure pas dans les captures)**

- **Norme : IEEE 802.3, ISO 8802.3**



Par rapport à la trame Ethernet V2, seul change le champ EtherType qui est remplacé par un champ Longueur qui indique la longueur de la trame : valeur comprise entre 1 et 1500, indique le nombre d'octets des données (compatibilité avec Ethernet…).

VII-Protocoles de transport du modèle TCP/IP : Deux protocoles permettant la communication entre applications utilisatrices :

- **TCP *(Transmission Control Protocol)* en mode orienté connexion.**
- **UDP *(User Datagram Protocol)* en mode sans connexion.**

TCP/IP utilise des adresses IP de 32 bits, écrites sous la forme de 4 séries de 8 bits chacune de 0 à 255 séparées par des points : xxx.xxx.xxx.xxx

7.1 : Adressage IP : Dans un réseau IPv4, les hôtes peuvent communiquer de trois façons : **Monodiffusion :** processus consistant à envoyer un paquet d'un hôte à un autre. **Diffusion** : processus consistant à envoyer un paquet d'un hôte à tous les hôtes du réseau ou **Multidiffusion :** processus consistant à envoyer un paquet d'un hôte à un groupe d'hôtes en particulier.
Ces trois types de transmission sont utilisés différemment dans les réseaux de données. Dans les trois cas, l'adresse IPv4 de l'hôte émetteur est placée dans l'en-tête du paquet comme adresse source.

À l'origine, la spécification RFC1700 regroupait les plages monodiffusion selon certaines tailles appelées des adresses de classe A, B et C. Elle a également établi des adresses de classe D (multidiffusion) et de classe E (expérimentales),
Les classes d'adresse monodiffusion A, B et C définissaient des réseaux d'une certaine taille, ainsi que des blocs d'adresses particuliers pour ces réseaux. Une entreprise ou une administration se voyait attribuer un bloc d'adresses entier de classe A, B ou C. L'utilisation de l'espace d'adressage s'appelait adressage par classe.

Blocs d'adresses A : Un bloc d'adresses de classe A, a été créé pour prendre en charge les réseaux de très grande taille.

Blocs de classe B : L'espace d'adressage de classe B a été créé pour répondre aux besoins des rése aux de taille moyenne ou de grande taille.

Blocs de classe C : L'espace d'adressage de la classe C était le plus disponible des anciennes classes d'adresses. Cet espace d'adressage était réservé aux réseaux de petite taille, comportant 254 hôtes au maximum.

Adressage sans classe

Le système que nous utilisons aujourd'hui s'appelle adressage sans classe. Avec ce type d'adressage, des blocs d'adresses correspondant au nombre d'hôtes sont attribués aux entreprises ou aux administrations, quelle que soit la classe multidiffusion.

Classe d'adresse	Plage du premier octet (décimale)	Bits du premier octet (les bits verts ne changent pas)	Parties réseau(N) et hôte (H) de l'adresse	Masque de sous-réseau par défaut (décimal et binaire)	Nombre de réseaux et d'hôtes possibles par réseau
A	1-127**	00000000-01111111	N.H.H.H	255.0.0.0	128 réseaux (2^7) 16 777 214 hôtes par réseau (2^24-2)
B	128-191	10000000-10111111	N.N.H.H	255.255.0.0	16 384 réseaux (2^14) 65 534 hôtes par réseau (2^16-2)
C	192-223	11000000-11011111	N.N.N.H	255.255.255.0	2 097 150 réseaux (2^21) 254 hôtes par réseau (2^8-2)
D	224-239	11100000-11101111	S.O. (multidiffusion)		
E	240-255	11110000-11111111	S.O. (expérimental)		

** Les adresses d'hôtes contenant uniquement des zéros (0) et des uns (1) ne sont pas valides.

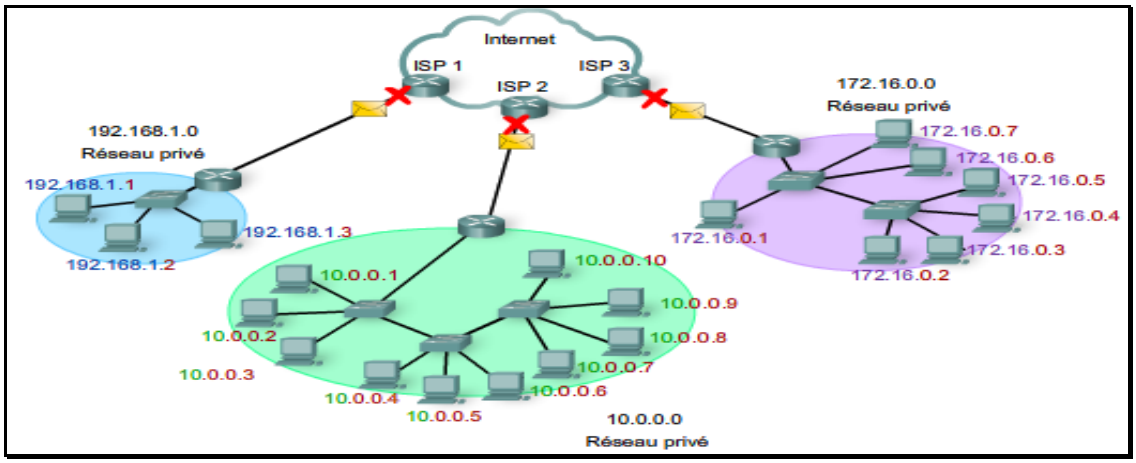
Figure : Classes d'adresses IP

- ***Adresses privées et publiques :*** Bien que la majorité des adresses d'hôte IPv4 soient des adresses publiques utilisées dans les réseaux accessibles sur Internet, d'autres blocs d'adresses sont

attribués à des réseaux qui ne nécessitent pas d'accès à Internet, ou uniquement un accès limité. Ces adresses sont appelées des adresses privées. ***Voici ces plages d'adresses privées :***

- ***de 10.0.0.0 à 10.255.255.255 (10.0.0.0 /8),***
- ***de 172.16.0.0 à 172.31.255.255 (172.16.0.0 /12),***
- ***de 192.168.0.0 à 192.168.255.255 (192.168.0.0 /16).***

Les blocs d'adresses de l'espace privé, sont réservés aux réseaux privés. L'utilisation de ces adresses ne doit pas forcément être unique entre des réseaux externes. En règle générale, les hôtes qui ne nécessitent pas d'accès à Internet peuvent utiliser les adresses privées sans limitation. Toutefois, les réseaux internes doivent configurer des schémas d'adressage réseau pour garantir que les hôtes des réseaux privés utilisent des adresses IP qui sont uniques au sein de leur environnement de réseau.

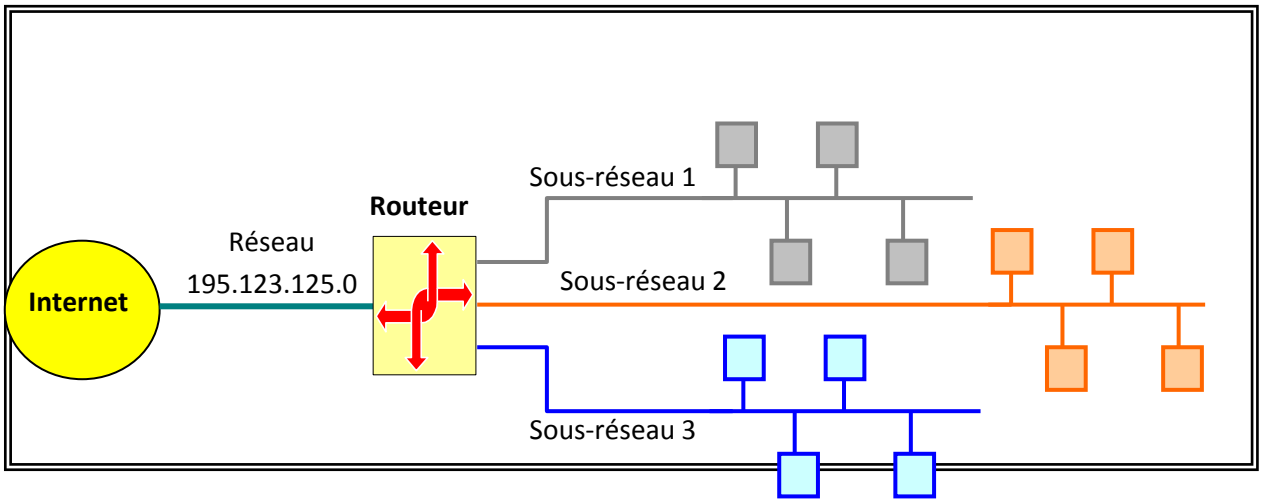


Réseaux et sous réseaux : Un réseau peut être divisé en sous-réseaux afin de pouvoir :

- éviter le gaspillage des adresses nœuds d'un réseau
- utiliser des supports physiques différents.
- réduire le trafic sur le réseau.
- isoler une partie du réseau en cas de défaillance d'un composant du réseau.
- augmenter la sécurité.

Chaque sous-réseau est relié à un autre par un routeur.

Exemple :



Dans la figure ci-dessus, le routeur est connecté à Internet par un réseau de classe **C** 195.123.125.0. Il est donc possible d'utiliser 256 (- 2) adresses pour les nœuds. Cependant si tous les nœuds sont sur le même réseau, celui-ci risque d'être chargé. On répartit les nœuds sur 3 réseaux que l'on connecte à un routeur. Chacun de ces réseaux devant avoir une adresse distincte, on crée des adresses de sous-réseaux pour chacun d'eux.

Masque sous Réseau : la notion de sous-réseaux était inexistante au début de IP. Elle est apparue avec la RFC 950 vers 1985. L'adressage de sous-réseaux va se faire avec des bits normalement réservés à l'adressage des nœuds.

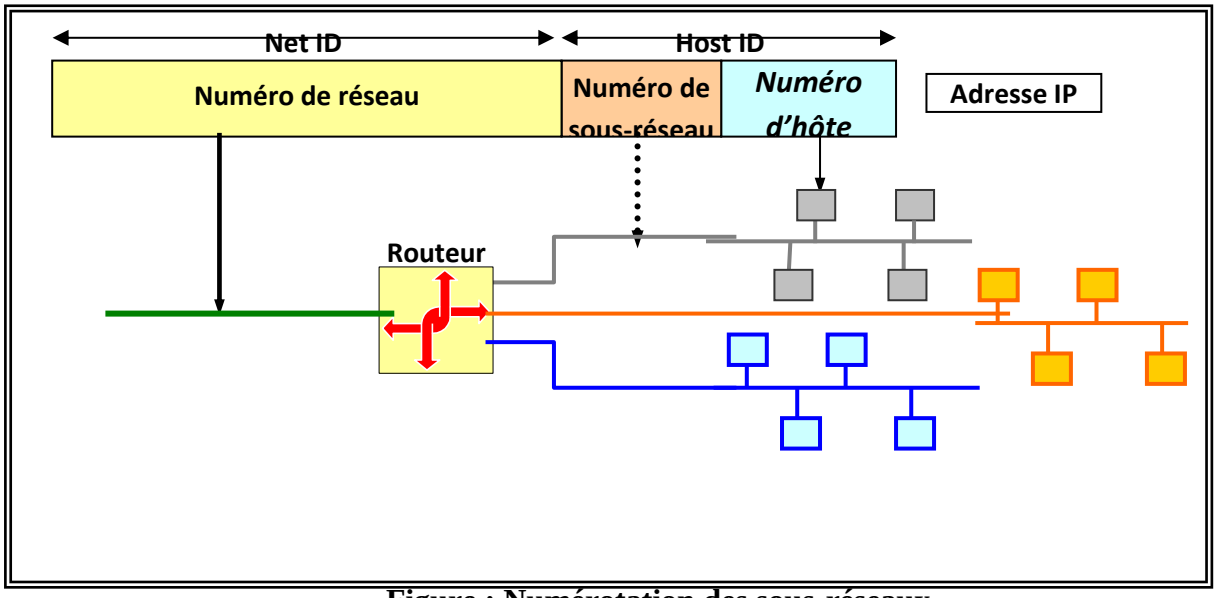


Figure : Numérotation des sous-réseaux.

Pour indiquer le nombre de bits pris sur la partie HostID comme numéro de sous-réseau, on va utiliser un masque de sous-réseaux. Ce masque indique par des **bits à 1** le nombre de bits de l'adresse IP qui correspondent à l'adresse réseau et à l'adresse sous-réseaux. Les **bits à 0** du masque indiquent les bits de l'adresse IP qui correspondent à l'HostID.

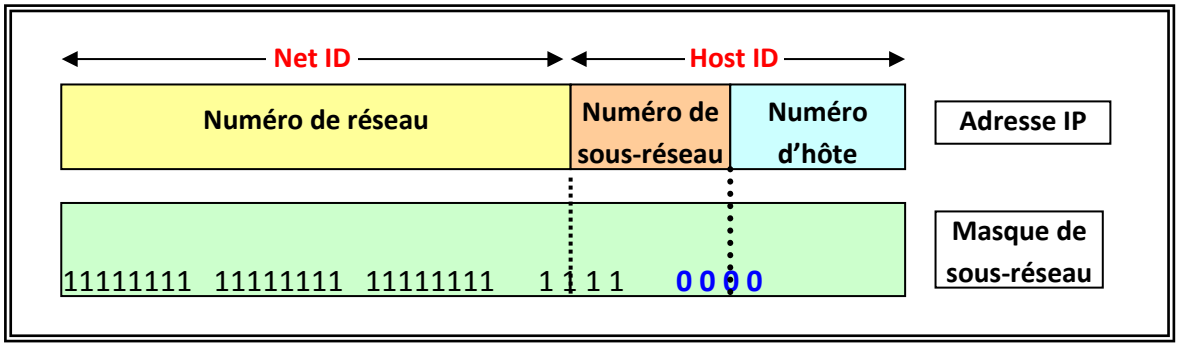


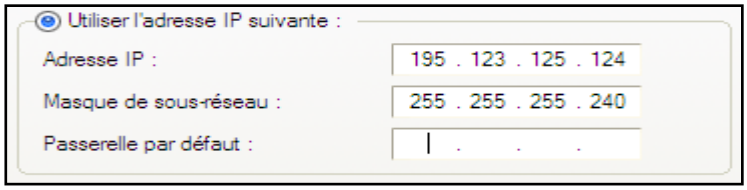
Figure : Masque de sous-réseau.

Dans l'exemple ci-dessus, l'adresse IP est une adresse de classe C. On désire créer 16 sous-réseaux. Il est donc nécessaire d'utiliser 4 bits de la partie HostID pour indiquer le numéro de sous-réseau.

Le masque comporte **28** bits à **1**, c'est à dire :

- **24** bits correspondant à la partie NetID de l'adresse et **4** bits pour indiquer les bits de l'adresse IP qui doivent être interprétés comme étant l'adresse de sous-réseaux.
- **4** bits à **0**, indiquent les bits de l'adresse IP qui doivent être interprétés comme des adresses de nœuds.

Les masques de sous réseaux sont à entrer dans chaque ordinateur travaillant en IP. Les valeurs des masques se rentrent la plupart du temps en notation décimale pointée. Pour illustrer l'exemple ci-dessus, voici comment il conviendrait d'indiquer à une station Windows, son adresse IP et son masque de sous-réseau.



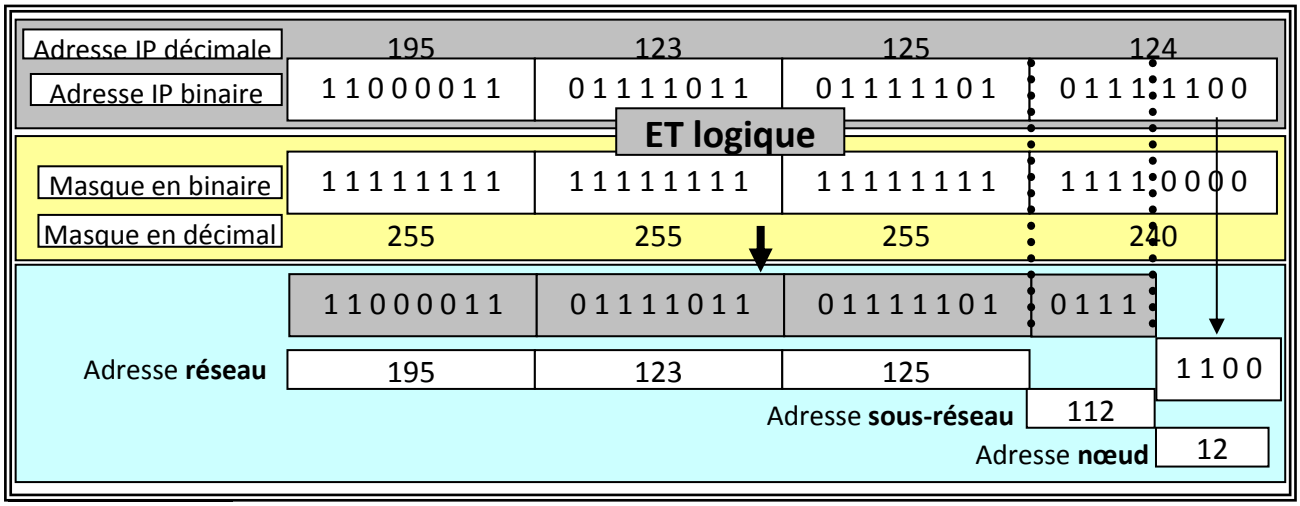


Figure : Calcul de l'adresse de sous-réseau et de l'adresse nœud.

Dans cet exemple, le masque de sous-réseau comporte **28 bits**. L'adresse IP 195.123.125.124 est une adresse de classe C. Les **24 premiers bits** du masque correspondent au **NetID**. Les **4 bits** suivants à **1** dans le masque indiquent qu'il faut interpréter les 4 premiers bits du dernier octet comme une **adresse de sous-réseau** et non comme une adresse HostID. Les 4 bits à 0 du masque indiquent qu'il faut interpréter les 4 derniers bits du dernier octet de l'adresse IP comme une adresse nœud.

On calcule l'adresse du sous-réseau en tenant compte du poids binaire de chaque bit. Ici, (128 x 0) + (1 x 64) + (1 x 32) + (1 x 16)= 112. L'adresse nœud correspond aux 4 derniers bits de l'adresse IP (12).

Quelques exemples d'adresses avec une signification particulière :

0.0.0.0 Hôte inconnu, sur ce réseau , 0.0.0.1 L'hôte 1 de ce réseau , 255.255.255.255 Tous les hôtes

138.195.52.1 L'hôte 52.1 du réseau 138.195.0.0 ; 138.195.0.0 Cet hôte sur le 138.195.0.0

193.104.1.255 Tous les hôtes du 193.104.1.0 ; 127.0.0.1 Cet hôte (boucle locale).

7.2 : Protocole IP : Le protocole IP permet d'avoir une vue globale de l'inter-réseau sans tenir compte des différents types de réseaux physiques qui le constituent. Les protocoles TCP et UDP ne dialoguent qu'avec IP sans voir les réseaux physiques. Les datagrammes IP peuvent être encapsulés dans des trames Ethernet, Token-Ring ou des paquets X25 ou même des liaisons séries asynchrones en utilisant un protocole de transports sur ce type de liaison comme PPP (Point To Point Protocol).

IP est aussi utilisable sur des liaisons spécialisées ou des réseaux de type Frame Relay ou ATM (Asynchronous Transfert Mode).

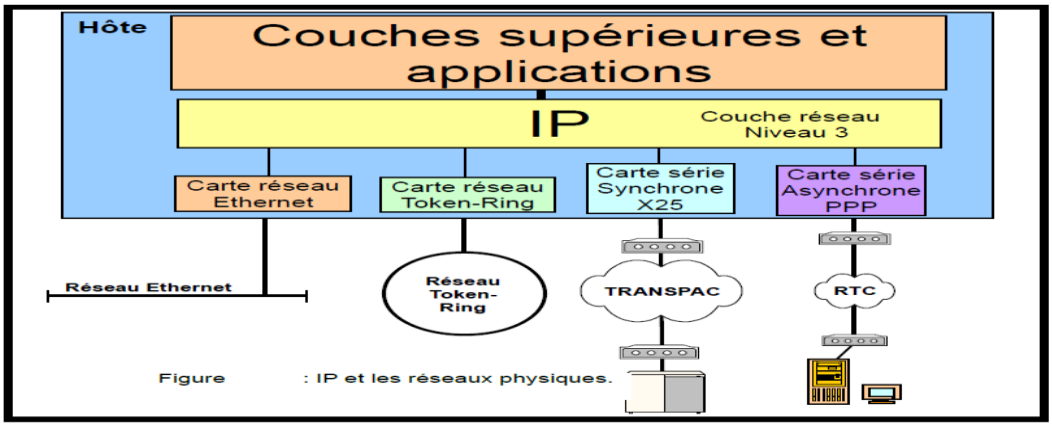


Figure : IP et les réseaux physiques.

7.2. A : Notion de Fragmentation : IP génère des paquets de taille variable en fonction du nombre d'octets de données à transporter et de la **MTU (**Maximum Transmission Unit. Taille maximum d'une trame sur un réseau physique du réseau physique). La MTU d'un réseau Ethernet est de 1500 octets, alors que celle d'un réseau Token-Ring à 16 Mbps est de 17940 octets. Il se peut donc qu'un paquet IP encapsulé dans une trame Token-Ring soit **fragmenté** en plusieurs paquets IP pour être véhiculé ensuite dans une trame Ethernet ou des paquets X25.

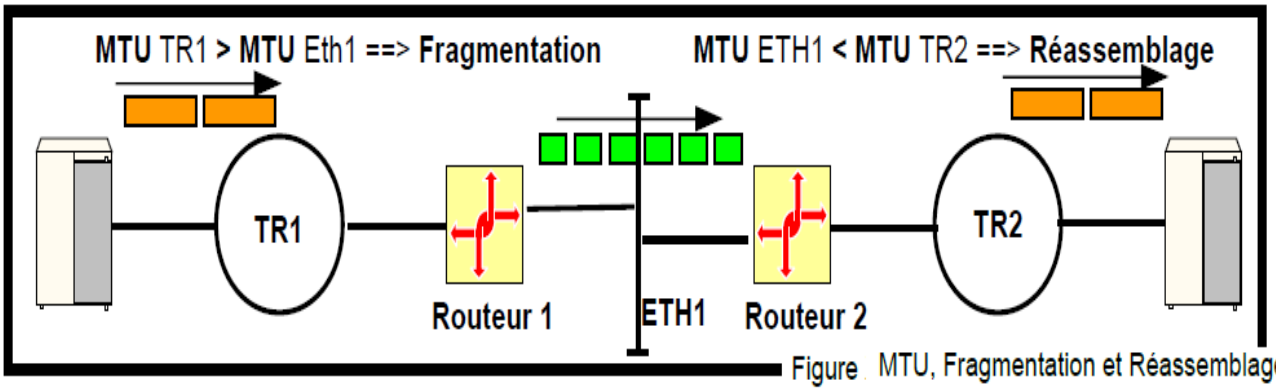
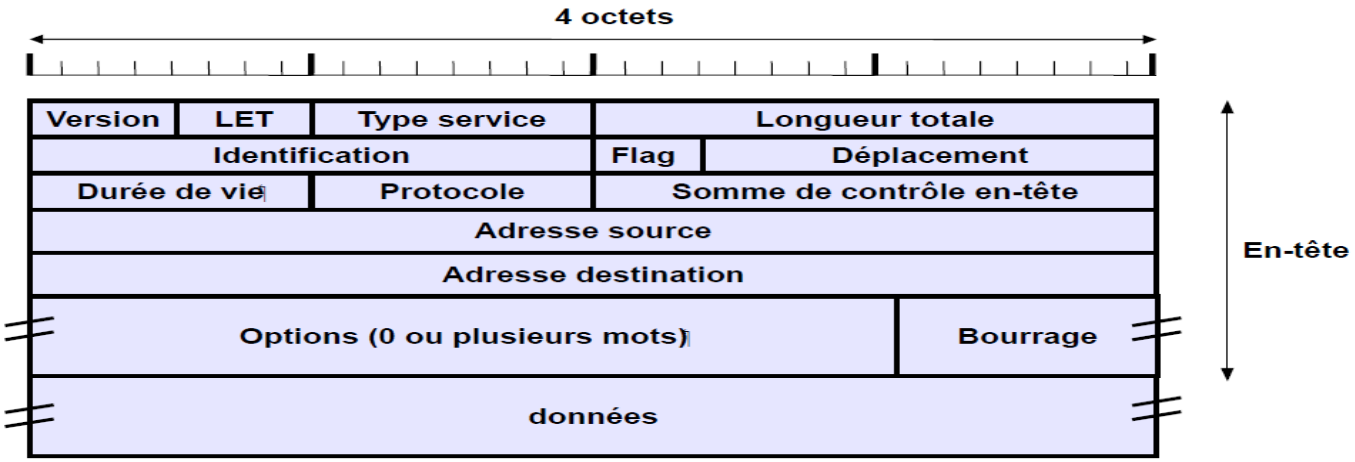


Figure MTU, Fragmentation et Réassemblage.

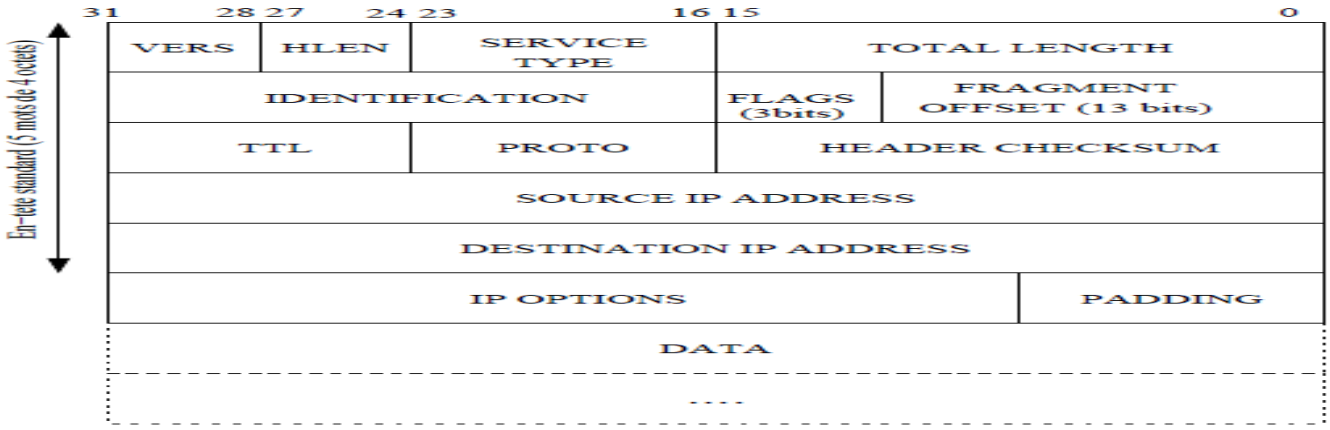
Par exemple, un bloc de 1481 octets sur Ethernet sera décomposé en un datagramme de 1480 (1480 + 20 soit finalement 1500 octets) et un datagramme de 1 octet ! Il existe une exception à cette opération, due à la présence active du bit " Don't Fragment bit " du champ FLAGS de l'en-tête IP. La présence à 1 de ce bit interdit la fragmentation du dit datagramme par la couche IP qui en aurait besoin. C'est une situation de blocage, la couche émettrice est tenue au courant par un message ICMP " *Fragmentation needed but don't fragment bit set* " et bien sûr le datagramme n'est pas transmis plus loin**. La fragmentation Se fait au niveau des routeurs.**

7.2-B : Datagramme IP : IP travaille en mode datagramme sans connexion, c'est-à-dire que chaque paquet IP est véhiculé dans un internet de manière indépendante. Il se peut donc que plusieurs paquets successifs empruntent des chemins différents. L'ordre d'arrivée des datagrammes peut être différent de celui de départ. On dit que le protocole IP n'assure pas le **séquencement**, ni d'ailleurs la fiabilité de la transmission (pas **d'accusé de réception**, ni de **checksum** sur les données transportées). Ces fonctions, si elles sont nécessaires pour les applications qui utilisent IP doivent être assurées par le protocole de la couche supérieure à IP. **TCP** assure le séquencement des paquets, les accusés de réception et la checksum.

Format d'un datagramme IP : Les octets issus de la couche de transport et encapsulés à l'aide d'un en-tête IP avant d'ˆêtre propagés vers la couche réseau (Ethernet par exemple), sont collectivement nommés par le mot **" datagramme IP ",** datagramme Internet ou datagramme tout court. Ces datagrammes ont une taille maximale liée aux caractéristiques de propagation du support physique, c'est le " Maximum Transfer Unit " ou MTU.



Notation en Anglais :

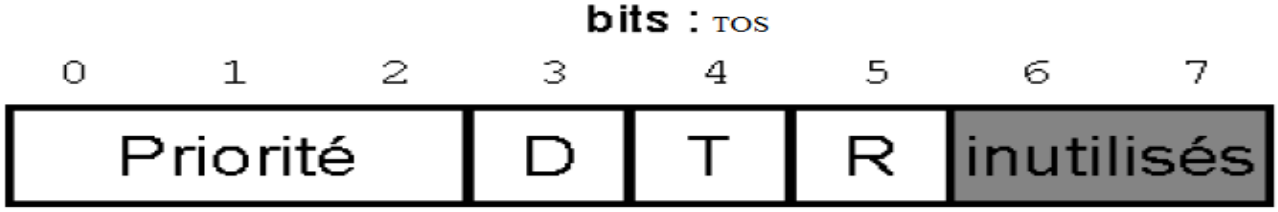


Description de l'en-tête du datagramme IP:

Version (4 bits): la version d'IP utilisée : Il indique la version IP utilisée. La valeur actuelle est 4 (**O1OO** en binaire). Elle passera à 6 lorsque la version IPV6 (Pas très utilisée à nos jours).

LET (4 bits) : Longueur d'En-Tête en nombre de mots de 32 bits OU Internet Header Length (IHL) = Longueur de l'en-tête IP. Par exemple si la valeur est 0101 (5), cela veut dire que l'en-tête mesure 5 fois 32 bits, soit 20 octets. Ce champ est indispensable, car la longueur de l'en-tête varie s'il y a des Options à la fin de l'en-tête standard.

Type de service (8 bits) : TOS= Type of Service. Ce champ informe les routeurs des réseaux de la qualité de service désirée. Il indique la qualité du service demandé pour ce datagramme (ou le flot de datagrammes dans lequel il s'inscrit) où les 8 bits sont décomposés comme suit (les deux derniers devant être à 0) Il est divisé en 6 parties.



Priorité : (3 bits)

Indique la priorité voulue pour le datagramme. La priorité augmente avec la valeur de ce champ. Les valeurs possibles sont les suivantes :

- 000 : Routine ;
- 001 : Priority ;
- 010 : Immediate ;
- 011 : Flash ;
- 100 : Flash Override ;
- 101 : Critic ;
- 110 : InterNetwork Control
- 111 : Network Control

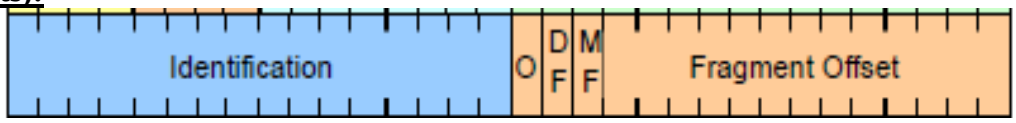
Sa représentation courante est l'intitulé correspondant à sa valeur.

Bit D (Delay) : à 1, indique que l'acheminement du datagramme doit privilégier le délai (il doit arriver le plus rapidement possible).

Bit T(hroughput) : à 1, indique que le datagramme fait partie d'une communication ayant besoin d'un gros débit

Bit R(eliability) : à 1, indique qu'il faut privilégier la fiabilité : un effort particulier doit être fait pour acheminer correctement ce datagramme
Bits inutilisés : doivent être à 0.
La représentation courante de ces bits est l'indication des traitements souhaités pour le datagramme.
Longueur Totale (16 bits) : taille (entête + données) en octet (Total Length) : Donne la taille totale en octets du datagramme (ou du fragment). Ainsi, un datagramme ne peut pas excéder 65535 octets (2^16-1). La norme impose à toute implémentation de pouvoir traiter des datagrammes d'au moins 576 octets. Si un datagramme devant traverser un réseau est de taille supérieure à ce que le réseau peut transmettre (càd au Maximum Transfer Unit ou MTU du réseau), il doit être fragmenté par le routeur ou la station l'injectant dans le réseau. Fragmenter veut dire que le datagramme sera découpé en datagrammes plus petits (des fragments) qui pourront être transmis. Ces fragments auront pour Longueur Totale la taille des données qu'ils transportent plus la longueur de l'en-tête. Le datagramme d'origine sera reconstitué par le destinataire.

Identification (16bits) : id des fragments d'un même paquet : Ce champ indique le numéro du paquet émis par la couche réseau d'un nœud. Le compteur compte de 0 à 65535, puis repasse à 0.
Flags (3 bits):



Bit 0: réservé, doit être à zéro ; (non utilisé)
Bit 1: (AF ou DF) bit Don't Fragment : si positionné à 1, indique que ce datagramme ne doit pas être Fragmenté sinon il est à 0

Bit 2: Le bit MF (More Frags) : si positionné à 1, indique que le datagramme n'est qu'une partie (fragment) du datagramme d'origine et que ce n'est pas le dernier fragment. Si à 0, indique que le datagramme est le dernier fragment du datagramme d'origine. On reconnaît un datagramme non fragmenté lorsque le bit More est à 0 et que le Déplacement est aussi à 0.

Déplacement- ou Offset : Déplacement (13 bits) : Position du fragment par rapport au paquet de départ, en nombre de mots de 8 octets : La valeur de ce champ est à multiplier par 8 pour connaître l'emplacement du premier octet de données par rapport au datagramme d'origine. Le Déplacement est différent de 0 uniquement si le datagramme a été fragmenté.

Durée de vie (TTL) 8 bits: pour que le paquet puisse rester dans le réseau : Time To Live : Ce champ représente en secondes la durée de vie d'un datagramme IP. La valeur de départ est de 60. A chaque passage dans un routeur la valeur est décrémentée d'une seconde (pour simplifier le travail).
Lorsque la valeur atteint 0, le routeur qui reçoit le paquet le détruit et envoie un paquet ICMP sur le réseau. Ce mécanisme a pour but d'éviter que des datagrammes dont l'adresse est erronée tournent sans fin dans l'internet. 255 seconds maximums de temps de vie pour un datagramme sur le net.
Protocole (8 bits) : type du protocole de niveau supérieur : Il permet à IP d'adresser les données extraites à l'une ou l'autre des couches de transport. Exemples : ICMP(1), IGMP(2), IP-ENCAP(4), **TCP(6), UDP(17),** ESP(50), AH(51), OSPF(89). **IP(0**)
Checksum d'en-tête (16 bits) : code de contrôle d'erreur pour l'entête pour s'assurer de l'intégrité de l'en-tête. Lors du calcul de ce " checksum " ce champ est à 0. A la réception de chaque paquet, la couche calcule cette valeur, si elle ne correspond pas à celle trouvée dans l'en-tête le datagramme est oublié (" discard ") sans message d'erreur.
Adresse source (32 bits) : IP de la machine source
Adresse destination (32 bits) : IP de la machine destination

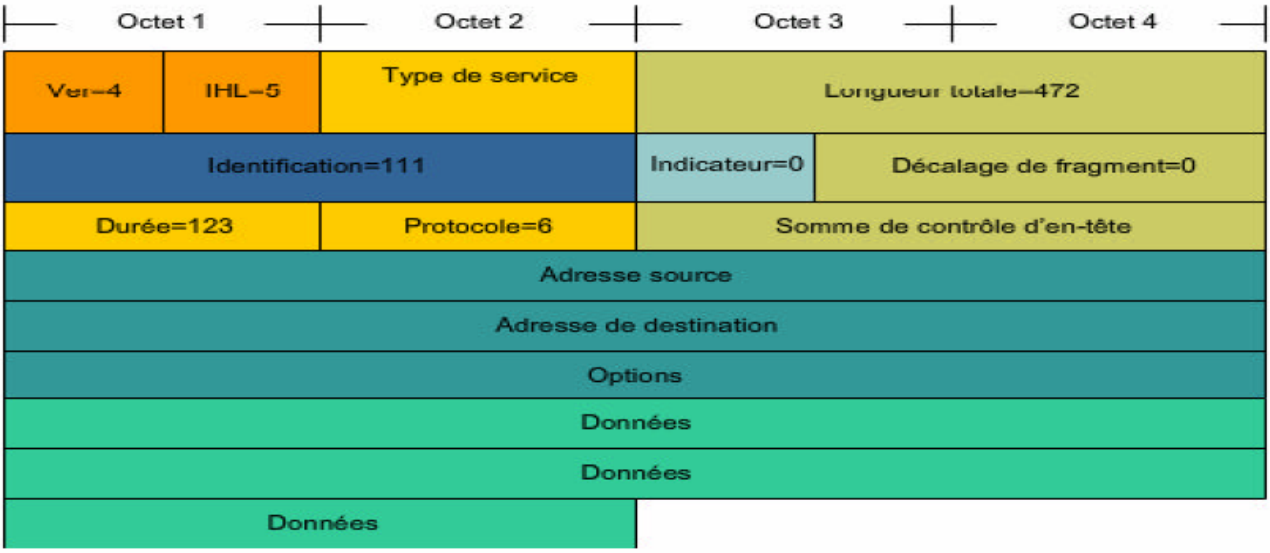
IP OPTIONS 24 bits : pour préciser des options de comportement des couches IP traversées et destinatrices. Les options les plus courantes concernent Des problèmes de sécurité, Des enregistrements de routes, Des enregistrements d'heure et Des spécifications de route à suivre.

PADDING (Bourrage) Remplissage pour aligner sur 32 bits : N'est présent que pour compléter la taille des options jusqu'à un multiple de 4 octets. Ceci parce que la taille de l'en-tête est HLEN × 4 octets.

En conclusion partielle que peut-on dire du travail de la couche IP ?

- Il consiste à router les datagrammes en les acheminant " au mieux ", C'est son travail principal.
- Il peut avoir à fragmenter les données de taille supérieure au MTU du support physique à employer.

Exemple de paquet IP : La figure suivante représente un paquet IP complet avec des valeurs de champ d'en-tête types.



Ver = 4 : version IP. , **IHL** = 5 : taille d'en-tête en mots de 32 bits (4 octets). Cet en-tête est de 5*4 = 20 octets, la taille minimale valide, **Longueur totale** = 472 : la taille de paquet (en-tête et données) est de 472 octets**. Identification** = 111 : identifiant de paquet initial (requis s'il est fragmenté par la suite). **Indicateur(Flag)** = 0 : stipule que le paquet peut être fragmenté si nécessaire. **Décalage du fragment** = 0 : indique que ce paquet n'est pas fragmenté actuellement (absence de décalage). **Durée de vie (TTL)=** 123 : indique le temps de traitement de la couche 3 en secondes avant abandon du paquet (décrémenté d'au moins 1 chaque fois qu'un périphérique traite l'en-tête de paquet). **Protocole** = 6 : indique que les données transportées par ce paquet constituent un segment TCP.

Remarque :

L'en-tête IP minimale fait 5 mots de 4 octets, soit 20 octets. S'il y a des options la taille maximale peut atteindre 60 octets.

7.2. C : Notions sur le Multiplexage et démultiplexage :

Le Multiplexage/démultiplexage est : Fondé sur le port de réception, le port d'émission et les adresses IP

Les ports source et destination sont répétés dans chaque segment, Certaines applications utilisent des ports spécifiques

Multiplexage : Le champ "Type" dans une trame Ethernet permet d'indiquer le code des différents types de protocoles (IP, ARP et RARP). De même au niveau IP, le champ "Type" de l'en-tête IP, permet de transporter TCP ou UDP. Enfin au niveau transport, les numéros de ports indiquent les applications concernées. Cette propriété de mélanger les protocoles est appelée multiplexage. **Il permet d'Encapsuler les données de plusieurs applications dans un même segment avec un en-tête qui permettra le démultiplexage.**

Démultiplexage : A l'inverse lorsqu'une machine reçoit une trame Ethernet, les données applicatives doivent remonter jusqu'aux couches supérieures en traversant les couches basses. **Il s'agit donc de distribuer chaque segment à l'application à laquelle il est destiné.**

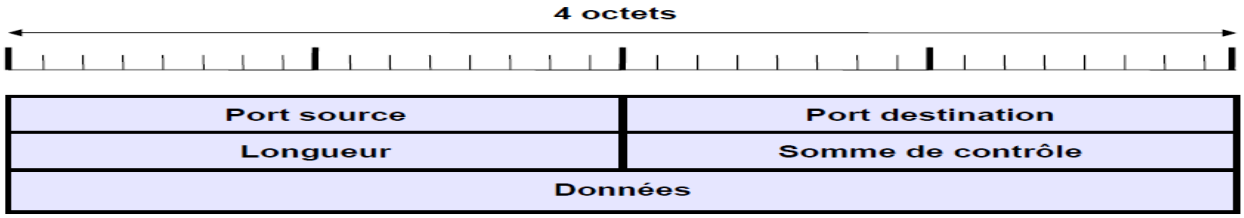
Numéros des ports : Les **numéros de ports** contenus dans les en-têtes **TCP** et **UDP** permettent de connaître l'**application** à laquelle il faut restituer les données.

Chaque application côté *serveur* utilise un numéro de port "**bien connu**" (well-know). Ainsi, l'application **Telnet** serveur utilise en principe le port **TCP (port =23)** et **FTP** le port **TCP 21**, alors que **TFTP** serveur utilise le port **UDP (port 69)**. Les numéros de port côté serveur sont compris entre **1** et **1023**.

Les applications côté *client* utilisent des "**ports éphémères**" dont les numéros sont compris entre **1024** et **5000**. La gestion de ces numéros de ports est complètement transparente pour les utilisateurs. La liste des **Ports TCP et UDP** est contenue dans le fichier **Services** des ordinateurs travaillant sous IP.

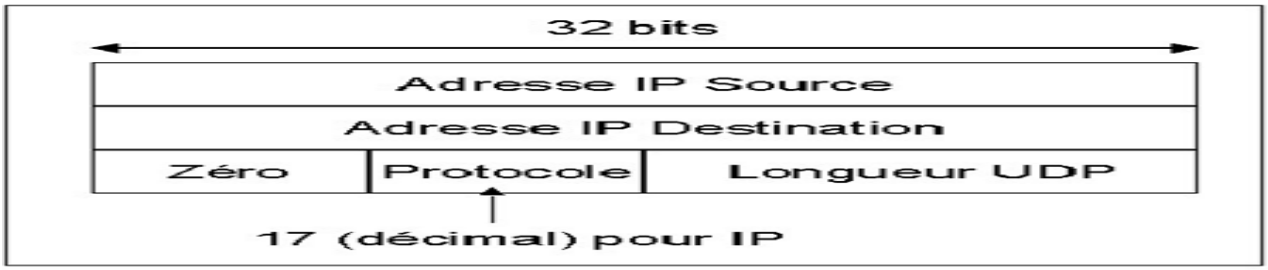
7.2. D : Le Protocole UDP : C'est un protocole *(User Datagram Protocol)* en mode sans connexion.

- **Caractéristiques :** Protocole simple, Service "au mieux", les segments UDP peuvent être Perdus ou délivrés dans le désordre, Aucun contrôle de flux ou de récupération d'erreurs en Mode sans connexion, Sans handshaking entre l'émetteur et le récepteur , Chaque segment UDP est traité indépendamment des autres Sans contrôle de congestion et il peut émettre aussi rapidement qu'il le souhaite.
- **Le segment UDP :**



Port source : numéro de port de l'application émettrice du segment, **Port destination :** numéro de port de l'application destinataire, **Longueur :** longueur de l'en-tête + données : La longueur minimale est 8 et La longueur maximale est 65 535 et **Somme de contrôle :** code de contrôle d'erreurs.

Le pseudo en-tête utilisé par UDP est le suivant (informations provenant d'IP) :

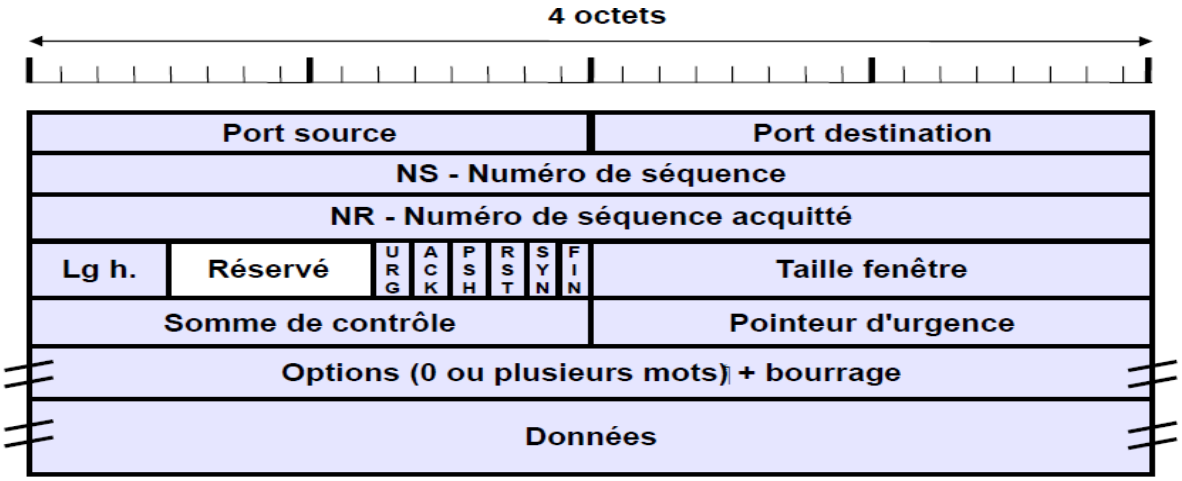


7.2. E : Protocole TCP (Transport Control Protocol (*Protocole de Contrôle de Transmission*)

Caractéristiques du protocole :

- **Arrivée garantie des données :** Récupération des erreurs par réémission et Réassemblage des données dans le bon ordre
- Vérification du flot de données afin d'éviter une saturation du réseau :
- Multiplexage/démultiplexage des données.
- Initialisation et fin d'une communication
- **Communication en mode connecté :** ✔ Ouverture d'un canal, ✔ Communication Full-Duplex ✔ Fermeture du canal.

Segment TCP : Les messages transmis par TCP (Transmission Control Protocol), normalement via IP, sont appelés des segments.



Port Source TCP (2 oct) : numéro de port de l'application émettrice : Entier non signé servant à identifier l'application émettrice du datagramme. Cette application correspond à un service (serveur) bien connu si le port est un port réservé.

Port Destination (2 oct) : numéro de port de l'application distante

Lg h. (4 bits) : longueur de l'en-tête en multiple de 4 octets : (ou Data Offset) Longueur de l'entête du segment TCP (4 bits).La valeur de ce champ est à multiplier par 4 pour connaître le nombre d'octets constituant l'en-tête. Ceci à cause de la présence d'éventuelles options (comme pour IP). Sa valeur a été multipliée par 4, et on indique par la même occasion s'il y a des options.

NS (4 oct) : Numéro de séquence : Numéro indiquant la place du 1er octet des données, dans le flot d'octets transmis par la source depuis le début de la connexion. Le numéro de séquence est choisi aléatoirement à l'établissement de la connexion puis croît avec les (nouveaux) octets transmis.

NR (4 oct) : numéro du prochain octet attendu.

Réservé (6 bits) : champ inutilisé actuellement : ces 6 bits doivent être à 0.

CODE (6 bits) ou Flags : URG: si 1 le paquet doit être traité de façon urgente ,**ACK**: si 1 le paquet est un accusé de réception ; **PSH** (PUSH): si 1 les données collectées doivent être transmises à l'application sans attendre les données qui suivent ,**RST**: si 1 la connexion est réinitialisée ,**SYN**: si 1 indique une demande d'établissement de connexion ,**FIN**: si 1 la connexion s'interrompt.

Taille fenêtre (2 oct) : Indique le nombre d'octets de données pouvant encore être reçus (pour l'instant) par l'émetteur du segment. Si 0, le récepteur du segment ne doit plus envoyer de données, jusqu'à ce qu'une "réouverture" de la fenêtre ait lieu (envoi d'un segment avec fenêtre **non nulle).**

Somme de contrôle (2 oct) ou Checksum : (16 bits) : permet de vérifier l'intégrité de l'en-tête.Total de contrôle portant sur la totalité du segment plus le pseudo en-tête TCP, permettant de vérifier la validité de l'ensemble du segment (données comprises). La méthode de calcul est la même que celle du checksum IP et UDP. Celle-ci utilise des mots de 16 bits : si les données comportent un nombre impair d'octets, il faut rajouter un octet fictif à 0 à la fin des données pour calculer le checksum TCP.

Pointeur d'urgence (2 oct): numéro d'ordre à partir duquel l'information devient urgente

Options (Taille variable): diverses options

Bourrage (Taille variable) : bits à zéro pour avoir une longueur en-tête multiple de 32 bits

(Exercice d'application didactique VOIR CNC 2016)

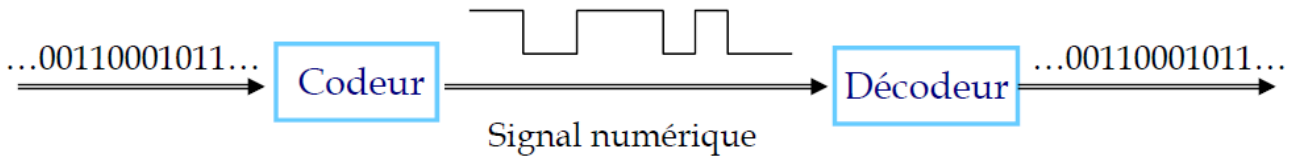
VIII-la couche physique :(1ere couche du modèle OSI)

8.1 : Principe :

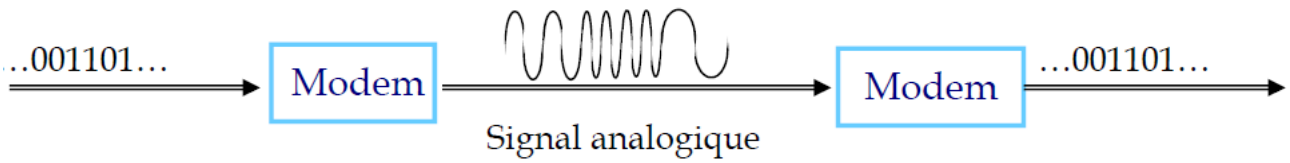
On définit :

- **Données continues** : Résultent de la variation continue d'un phénomène physique (voix, température, image, lumière, …) elles Possèdent une infinité de valeurs dans un intervalle borné.
- **Données discrètes :** Suite discontinue de valeurs dénombrables : Exemple: un texte est une association de mots eux-mêmes composés de lettres (symboles élémentaires).

Transmission en bande de base : Les bits sont directement codés par des valeurs de tensions (code NRZ, code Manchester…) ce qui permet la simplicité du codage mais distances limitées et le signal Occupe toute la bande passante (pas de multiplexage) dans ce cas le Signal transmis est discontinue (numérique).



Transmission en large bande (par transposition de fréquence) : Transposition dans un domaine de fréquences adapté au support (protection contre le bruit et **Adaptée aux longues distances** dans ce cas il y _ Possibilité de multiplexage/démultiplexage (Voir chapitre 1) le type du signal dans le support et : Signal continue (analogique) :



8.2 : Les supports de transmissions : Les caractéristiques d'un support de transmission sont données au chapitre 1.les principaux supports utilisés sont :

Paires torsadées et câble coaxial : Généralement en cuivre, Forte atténuation, Sensibles aux perturbations électromagnétiques

Fibre optique : Bande passante de l'ordre de 1 GHz/1 km _Haut débit Très faible atténuation et Robuste face à la température et aux perturbations électromagnétiques, Encombrement minimum

Air et Faisceaux hertziens

Paires torsadées :Caractéristiques

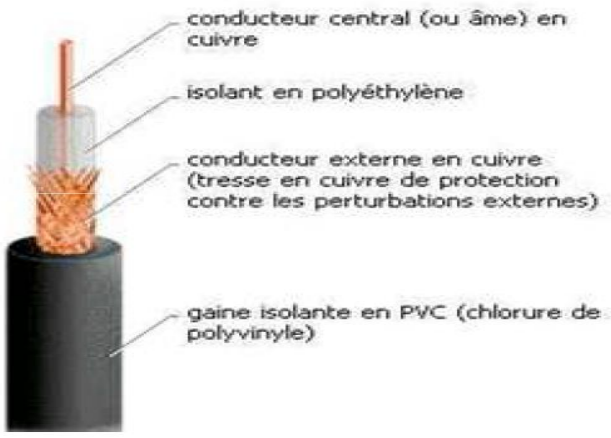
Distance maximale	100m (sinon ajouter un répéteur)
Capacité	10 – 100 Mbits/s
Raccordement	Connecteur RJ-45
Impédance	100 Ohms
Coût	Faible
Liaison	point à point ou multipoint
Transmission	analogique ou numérique
Utilisation	répandu

Câble coaxial


Terminaisons de câbles coaxiaux


Câble coaxial


Connecteur BNC en T (thin)

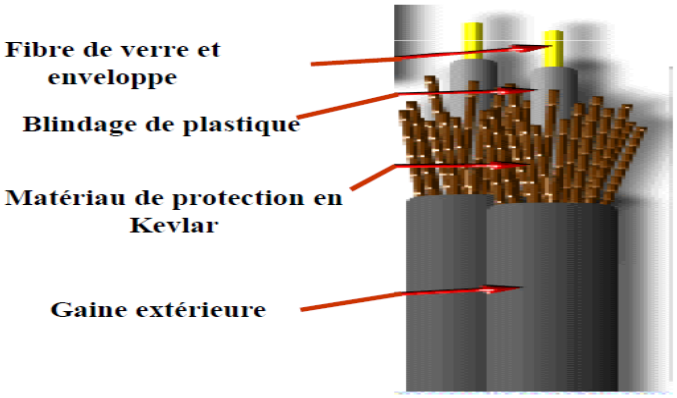


Caractéristiques d'un câble coaxial :

Bande passante	400 Mhz
Capacité	10 – 100 Mbits/s
Raccordement	Connecteur BNC
Impédance	150 Ohms
Coût	Peu cher
Liaison	point à point ou multipoint
Transmission	analogique ou numérique
Utilisation	En baisse

Fibre optique :

- Constitué par:
  - Un noyau: guide cylindrique en verre (caractérisé par un fort indice de réfraction) dans lequel se propagent des faisceaux lumineux (ondes optiques)
  - Un ou plusieurs enveloppes de protection



Caractéristiques :

Distance maximale	jusqu'à 3Km
Capacité	jusqu'à 1 Gbits/s
Coût	Cher
Liaison	point à point (multipoint difficile)
Transmission	analogique ou numérique
Utilisation	moyenne

8.3 : Le Codage : (Voir Réseaux Informatiques chapitre 1partie 3-5 : Codage des signaux numériques)

Annexe 1 : Méthode de calcul de CRC : les codes cycliques :

Méthode de calcul du CRC : Calcul d'un checksum basé sur l'arithmétique polynomiale modulo 2

– On considère le mot binaire suivant de taille n bits : b= (bn-1, bn-2,..., b1, b0), Ce mot s'exprime sous la forme polynomiale suivante (polynôme de degrés n-1, à coefficients binaire):

B(X)=bn-1.Xn-1 + bn-2.Xn-2 + ... + b1.X + b0

La clé C(X) associée à un tel mot est définie comme étant le reste de la division de B(X).Xk par un polynôme générateur G(X) de degré k.

– **Le mot à transmettre est alors M(X) = B(X).Xk + C(X).**

Exemple d'utilisation des CRCs

- **CRC-1 (bit de parité) : G(X) = X + 1**
- **CRC-8 (ATM) : G(X) = X8 + X2 + X + 1**
- **CRC-16 (USB, PPP, Bluetooth, ...)**
- **CRC-32 (Ethernet) : G(X) = X32 + X26 + X23 + X22 + X16 + X12 + X11 + X10 + X8 + X7 + X5 + X4**
- **Xi=X^i**

Question : Quelle est la clé associée au mot 110111

Avec CRC-2 : G(X) = le polynôme générateur est :$G(x) = X^2 + X + 1$

– Mot = 110111 soit : B(X) =$X^5 + X^4 + X^2 + X^1 + 1$

- $B(X) * X^2 = X^7 + X^6 + X^4 + X^3 + X^2$

- Calcul de la division : $B(X) * X^2$ *divisée par* $G(X)$ donne :

$B(X).X^2 = X^7 + X^6 + X^4 + X^3 + X^2$	$G(X) = X^2 + X + 1$
$-(X^7 + X^6 + X^5)$	$P(X) = X^5 + X^3 + 1$
$X^5 + X^4 + X^3 + X^2$	
$-(X^5 + X^4 + X^3)$	
X^2	
$-(X^2 + X + 1)$	
$C(X) = X + 1$	

En algèbre binaire (modulo 2), on a : 1+1 = 0 ou encore 1 = -1, par conséquent ajouter est identique à soustraire !

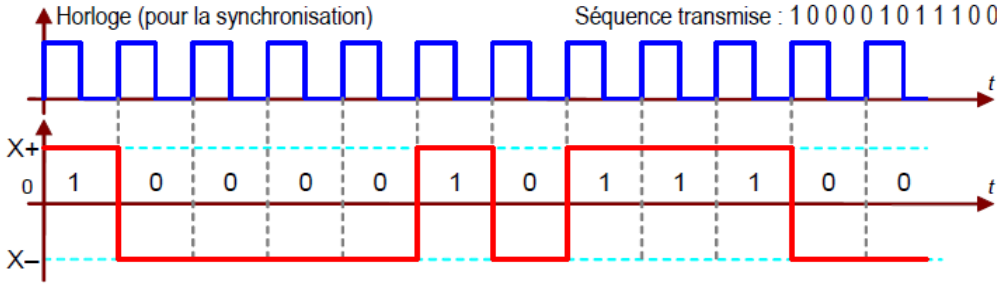
– Le reste est C(X) = X+1, Donc la clé est 11 (coefficients de C(X)) et Le mot à envoyer sera 110111 **11 (les deux bits représente le reste donc la clé)**

Comment peut-on détecter une erreur ? M(X) est le polynôme correspondant au mot transmis... donc M(X) doit être divisible par G(X). On peut le vérifier en effectuant la division de M(X) par G(X) ; le reste R(X) doit être nul. Si ce n'est pas le cas, une erreur est détectée !

Annexe 2 : Le codage de l'information :

Principe du Non Return to Zero (NRZ)

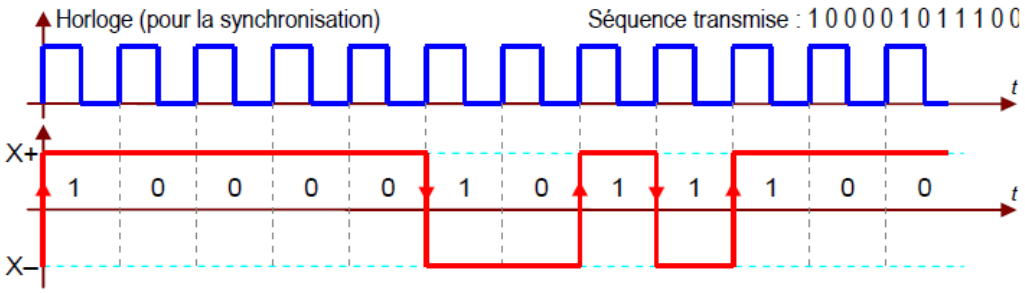
- **Règles de codage : codage à deux niveaux**
  - Les « 1 » logiques sont associés à une grandeur +X et les « 0 » logiques à une grandeur –X



- **Permet de transmettre un signal de moyenne statistique nulle**
- **Avantage**
  - Mise en œuvre très facile
  - Bonne immunité au bruit (grand écart entre les « 0 » et les « 1 »)
- **Inconvénient**
  - Synchronisation difficile, voire impossible, pour les très longues séquences de « 1 » ou « 0 » en raison du manque de transitions

Principe du Non Return to Zero Inverted (NRZI)

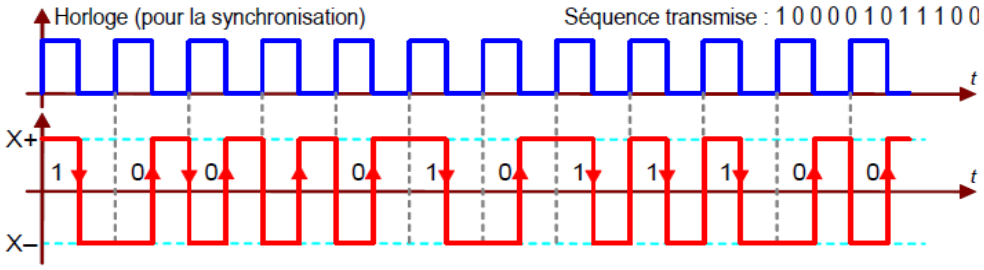
- **Élimine l'inconvénient du manque de transitions du codage NRZ**
- **Règles de codage : codage à deux niveaux (comme NRZ)**
  - Une transition du signal est produite pour chaque « 1 » logique, mais pas pour les « 0 »



- **Avantage**
  - L'immunité au bruit est préservée
- **Inconvénient**
  - Signal sans transition pour les longues séquences de « 0 »

Principe du codage biphase ou Manchester

- **Règles de codage : un autre codage à deux niveaux**
  - Une transition du signal est provoquée pour chaque bit transmis : $+V \rightarrow -V$ pour « 1 » et $-V \rightarrow +V$ pour « 0 »



- **Avantages**
  - La bonne immunité au bruit est maintenue
  - La synchronisation des échanges est toujours assurée
- **Inconvénients**
  - Débit sur le canal de transmission deux fois plus élevé : 1 bit/Hz
  - Difficilement envisageable pour des débits élevés

Spectre des signaux transmis : Le choix d'un code (NRZ….) se fait selon son spectre, pour les codes cités avant (Code binaire).On veut transmettre une suite de données binaire (0 ou 1), auxquelles on va affecter deux symboles S0(t) pour le '0' et S1(t) pour le '1' soit S0(f) et S1(t) les transformées de Fourier de S0(t) et S1(t) :

Ces deux symboles apparaissent avec une probabilité p0 pour le '0' et p1 pour le '1' la densité spectrale est donnée **par la formule de Bennett** :

$$\gamma(f) = \frac{p0p1}{Tb} * |S0(f) - S1(f)|^2 + \frac{1}{(Tb)^2} |p0.S0(f) + p1 * S1(f)|^2 * \sum_n \delta\left(f - \frac{n}{Tb}\right)$$

Avec Tb=Durée d'un bit en seconde et δ(t) Impulsion de Dirac.

Exemple : NRZ :

Le système équiprobable : p0=p1=1/2 et S0(t)= - V, S1(t)=+V.

$$S1(f) = -S0(f) = V.Tb.\left(\frac{\sin(\boldsymbol{\pi.f.Tb})}{\boldsymbol{\pi.f.Tb}}\right) * \boldsymbol{exp}(-\boldsymbol{j\pi.f.Tb})$$

Donc : $|\boldsymbol{S0(f)}| = |\boldsymbol{S1(f)}| = \boldsymbol{V.Tb.}\left|\left(\frac{\sin(\boldsymbol{\pi.f.Tb})}{\boldsymbol{\pi.f.Tb}}\right)\right| = \boldsymbol{V.Tb.}|\boldsymbol{sinc}(\boldsymbol{\pi.f.Tb})|$

Avec la formule de Bennett on trouve :

$$\gamma(f) = V^2.Tb.\left(\frac{sin(\boldsymbol{\pi.f.Tb})}{\boldsymbol{\pi.f.Tb}}\right)^2$$

BONNE CHANCE AUX CONCOURS !