Lycée polyvalent DIDEROT	Département IRIS
61, rue David d'Angers	
75019 PARIS	
http://www.diderot.org	

COURS RESEAUX

Chap4 –TCP/IP

Description courte		
Auteur	**Version - Date**	**Nom du fichier**
G.VALET	Version 1.6 - Juin 2009	chap4-tcpip.docx

Sommaire

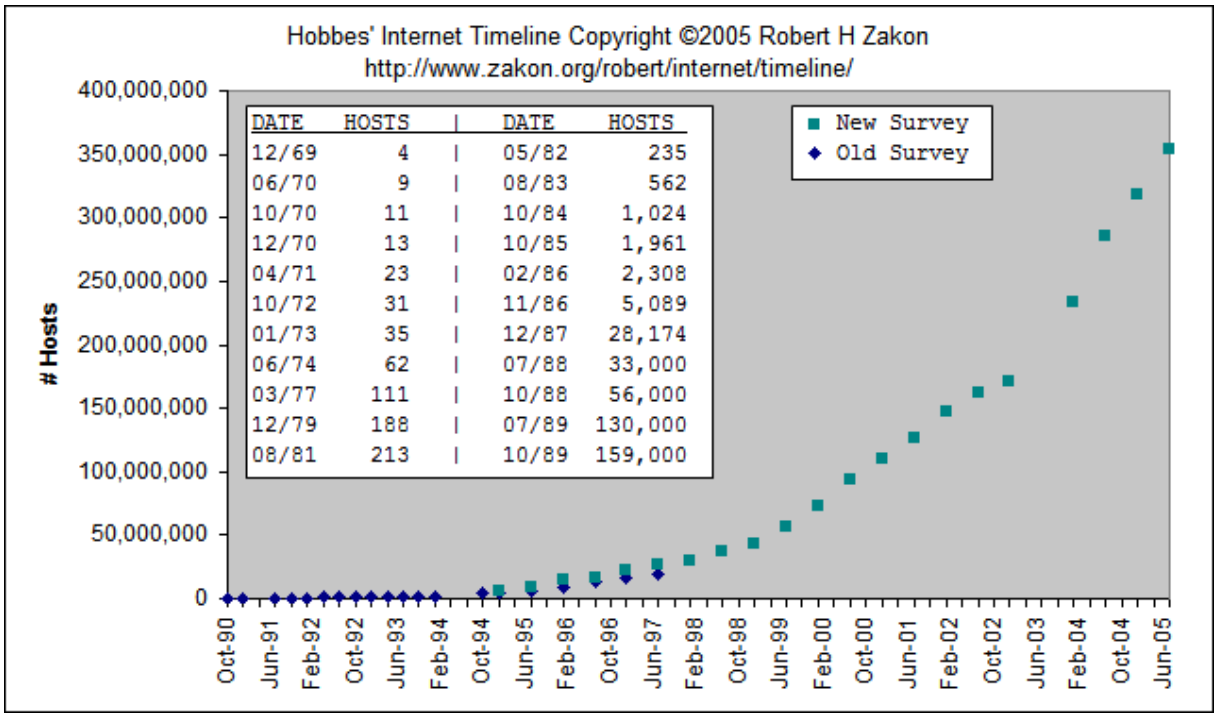
A. Introduction

Appelé également « Modèle Dod » ou « Darpa », le **modèle TCP/IP** a été initialement développé par l'agence ARPA (*Advanced Research Projects Agency*) sous le nom « Arpanet ». Destiné à une utilisation militaire, TCP/IP est devenu un standard aussi bien au niveau des réseaux locaux que des réseaux étendus comme l'Internet. De ce fait, il est très important de comprendre les mécanismes et les protocoles qui interagissent avec les différentes couches du modèle OSI (Voir **Erreur ! Source du renvoi introuvable.** - **Erreur ! Source du renvoi introuvable.**).

Le modèle TCP/IP correspond à une simplification du modèle OSI plus pragmatique et représentatif des technologies existantes.

Indépendamment des types de réseaux (ATM, Ethernet ou FDDI), on parle de réseau TCP/IP lorsque la famille de protocoles TCP/IP est utilisée.

Notons par exemple l'existence de « technologies passerelles » permettant { des réseaux non IP d'assurer la transmission de données IP (IP sur ATM par exemple)



DATE	HOSTS	DATE	HOSTS
12/69	4	05/82	235
06/70	9	08/83	562
10/70	11	10/84	1,024
12/70	13	10/85	1,961
04/71	23	02/86	2,308
10/72	31	11/86	5,089
01/73	35	12/87	28,174
06/74	62	07/88	33,000
03/77	111	10/88	56,000
12/79	188	07/89	130,000
08/81	213	10/89	159,000

Nombre d'hôtes connectés à L'internet

B. Le modèle TCP/IP

Il existe une différence essentielle entre le modèle et son implémentation. TCP/IP est en fait les deux à la fois. Il fait référence à 2 notions bien distinctes :

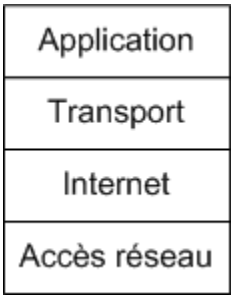
- La notion de modèle basé sur des couches (comme le modèle OSI)
- La notion d'implémentation : TCP/IP est une appellation souvent étendue aux logiciels basés sur les protocoles TCP/IP. Néanmoins, les applications TCP/IP sont en fait des logiciels implémentant le modèle TCP/IP

Le modèle TCP/IP s'inspire beaucoup du modèle OSI. Ceci est du au fait qu'ils ont été mis en œuvre à peu près au même moment.

Le nom TCP/IP provient des deux protocoles principaux de ce modèle : TCP (*Transmission Control Protocol*) et IP (*Internet Protocol*)
Ce modèle est celui adopté par le réseau mondial Internet.

B.1. Les 4 couches

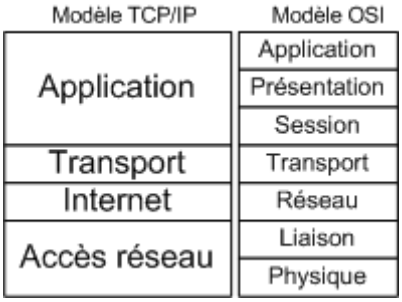
Ce modèle est basé sur 4 couches :



Les 4 couches du modèle TCP/IP

B.2. Correspondance entre OSI – TCP/IP

Il existe une correspondance entre le modèle OSI et TCP/IP



Correspondance OSI, TCP/IP

Les 3 dernières couches OSI sont confondues dans la couche *Application*. Il en va de même pour les 2 premières couches avec la couche *Accès réseau*.

B.3. L'encapsulation

Le principe de l'encapsulation vu dans le chapitre **Erreur ! Source du renvoi introuvable.** - **Erreur ! Source du renvoi introuvable.** - **Erreur ! Source du renvoi introuvable.** s'applique également au modèle TCP/IP

B.4. Les rôles des différentes couches

Nous allons voir comment les tâches sont réparties entre les couches.

a. La couche Accès réseau

La couche accès réseau est la première couche de la pile TCP/IP, elle offre les capacités à accéder à un réseau physique quel qu'il soit, c'est-à-dire les moyens à mettre en oeuvre afin de transmettre des données via un réseau.

Ainsi, la couche accès réseau contient toutes les spécifications concernant **la transmission de données** sur un réseau physique, qu'il s'agisse de réseau local (Anneau à jeton - *token ring*, *ethernet*, *FDDI*), de connexion à une ligne téléphonique ou n'importe quel type de liaison à un réseau. Elle prend en charge les notions suivantes :

- Acheminement des données sur la liaison
- Coordination de la transmission de données (synchronisation)
- Format des données
- Conversion des signaux (analogique/numérique)
- Contrôle des erreurs à l'arrivée
- ...

Heureusement toutes ces spécifications sont transparentes aux yeux de l'utilisateur, car l'ensemble de ces tâches est en fait réalisé par le système d'exploitation, ainsi que les pilotes du matériel et le matériel lui-même, permettant la connexion au réseau (ex : driver de carte réseau).

b. La couche Internet

La couche Internet est la couche "la plus importante" car c'est elle qui définit les datagrammes, et qui gère les notions d'adressage IP.
Son rôle est de permettre l'injection de paquets dans n'importe quel réseau et l'acheminement des ces paquets indépendamment les uns des autres jusqu'à destination. Les paquets sont alors rassemblés par cette couche.

La couche Internet contient 5 protocoles :

- Le protocole IP (*Internet Protocol*)
- Le protocole ARP (*Address Resolution Protocol*)

- Le protocole ICMP (*Internet Control Message Protocol*)
- Le protocole RARP (*Reverse Address Resolution Protocol*)
- Le protocole IGMP (*Internet Group Management Protocol*)

c. La couche Transport

Son rôle est le même que celui de la couche transport du modèle OSI : **permettre à des entités paires de soutenir une conversation**.

Officiellement, cette couche n'a que deux implémentations : le protocole *TCP* (Transmission Control Protocol) et le protocole *UDP* (User Datagram Protocol).

Officiellement, cette couche n'a que deux implémentations :

- TCP, un protocole orienté connexion qui assure le contrôle des erreurs
- UDP, un protocole non orienté connexion dont le contrôle d'erreur est peu fiable

d. La couche Application

Contrairement au modèle OSI, c'est la couche immédiatement supérieure à la couche transport, tout simplement parce que les couches présentation et session sont apparues inutiles.

On s'est en effet aperçu avec l'usage que les logiciels réseau n'utilisent que très rarement ces 2 couches, et finalement, le modèle OSI dépouillé de ces 2 couches ressemble fortement au modèle TCP/IP.

Cette couche contient un nombre très important de protocoles de haut niveau dont le rôle est de fournir des services réseaux évolués (Comme *Netbios* de *Microsoft*).

Voici quelques exemples de protocoles très utilisés :

- SMTP (*Simple Mail Transfer Protocol*)
- Telnet
- HTTP (*HyperText Transfer Protocol*)
- FTP (*File Transfer Protocol*)

C. Le niveau Trame

Le niveau trame fait référence à la couche 2 du modèle OSI. Pour l'instant, nous ne parlons pas directement de TCP/IP qui n'intervient qu'à partir de la couche 3.

A ce niveau, les données sont transmises sous la forme de trames. Ces trames sont envoyées et reçues grâce à la présence d'une adresse physique. Cette adresse physique est utilisée pour mener à bien la transmission jusqu'au destinataire final.

Cette trame de compose de la manière suivante :

Exemple de trame

L'unité d'information de la couche liaison est **la trame**

C.1. L'adresse physique

L'adresse physique, à ne pas confondre avec l'adresse logique (Adresse IP) permet de gérer les communications **au niveau local** (Même segment réseau). Lors du passage d'un réseau à un autre, nous verrons que le niveau paquet utilise des adresses logiques et que les adresses physiques changent.

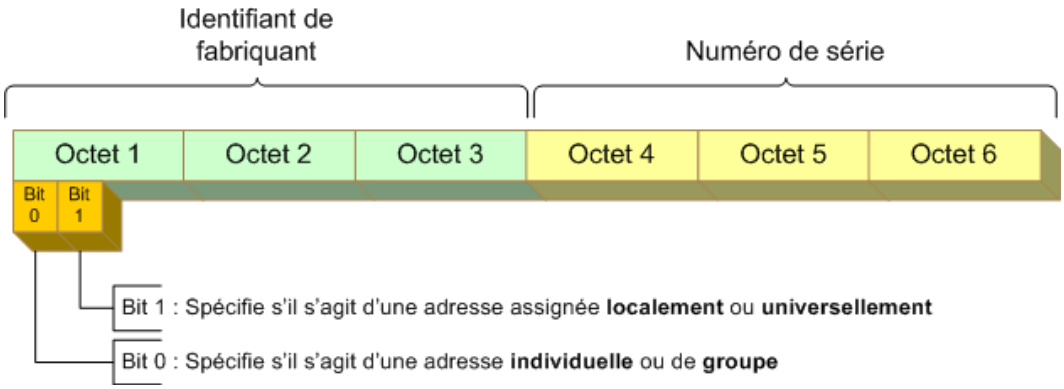
Ce qui signifie que **le niveau trame assure des communications poste à poste localement** (sur le même segment de réseau) et que ce sera **le niveau paquet** (Voir plus loin) qui permettra le dialogue avec des machines situées sur d'autres réseaux.

Chaque norme de réseau (Ethernet, ATM, FDDI, Token Ring) utilise son propre adressage physique. Nous allons donc voir comment ces différentes normes définissent l'adresse physique.

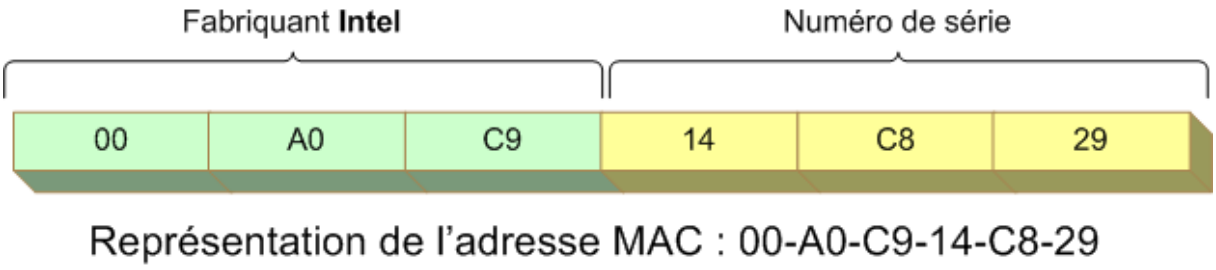
a. L'adresse MAC Ethernet

L'adresse MAC (*Media Access Control*) est historiquement l'une des premières adresses physiques apparues avec les réseaux locaux.

En théorie, chaque interface réseau matérielle (carte réseau) possède une adresse MAC unique. Cette adresse est composée de 6 octets (48 bits) répartit de la manière suivante :



Chaque octet est représenté par un nombre hexadécimal variant de 00 à FF. Voici un exemple d'adresse MAC :



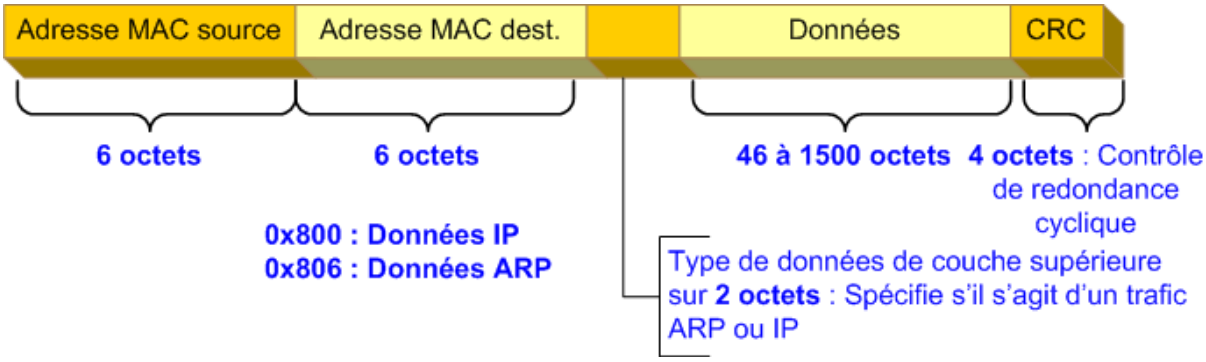
Exemple d'une adresse MAC de chez Intel

Les bits d'une adresse MAC sont transmis ***de gauche à droite*** *sur le support.*

C.2. La trame Ethernet

Il existe plusieurs types de trames Ethernet. Ces types dépendent du standard choisi. La société Xerox, à l'origine de la norme Ethernet a mis au point une trame de type Ethernet I. La 2ème version fut appelée Ethernet II. L'IEEE a normalisé la trame sous le nom 802.3. Pour résoudre les problèmes de compatibilité ascendante, l'IEEE a crée la trame 802.2 SNAP.

a. La trame Ethernet II



La trame Ethernet II

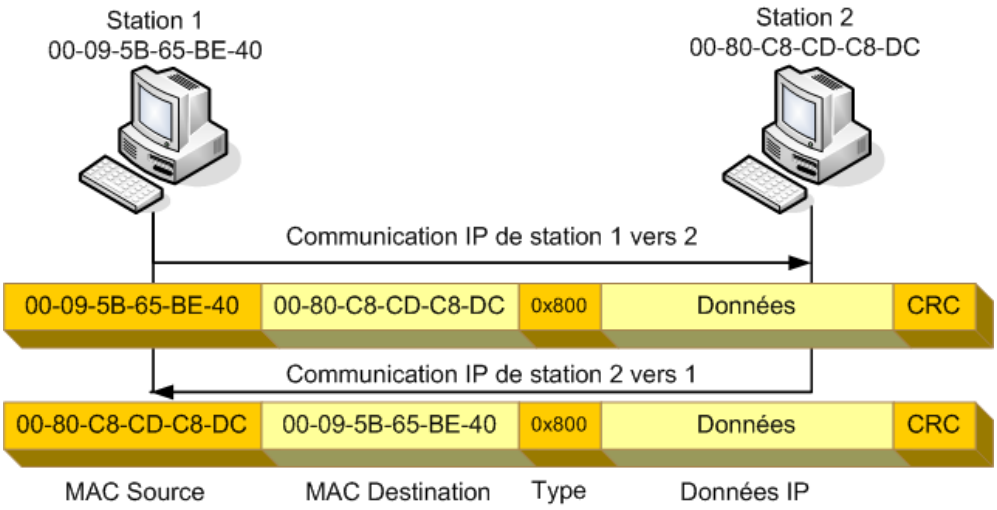
Le « *type de trame* » précise si les données qui suivent correspondent à un trafic de type IP ou de type ARP (*Address Resolution Protocol*) que nous verrons plus loin

C.3. Communication au niveau trame

Dans l'exemple ci-dessous, la station 1 communique avec la station 2 :

- La station 1 envoie donc une trame avec l'adresse destination de la station 2
- Lorsque la station 2 reçoit la trame, elle lit les données afin de les transmettre à la couche suivante (niveau paquet).

- La station 2 connaissant l'adresse de l'expéditeur peut donc devenir émetteur puisqu'elle connaît l'adresse de la station 1



Exemple de communication de niveau trame entre deux stations

D. Le niveau paquet

Au dessus du « niveau trame », figure le « niveau paquet ». Les données vont être fragmentées en petits éléments appelés des paquets. **Ces paquets doivent être transportés sur d'autres réseaux**, vers **le bon point de destination** et dans **les meilleures conditions possibles**.

La couche réseau du modèle OSI correspond au niveau paquet.

Le niveau paquet introduit la notion **d'adressage logique**. En effet, nous avons vu que les adresses physiques étaient utilisées uniquement sur un même réseau.

Sur les réseaux TCP/IP, l'adresse logique est nommée « **adresse IP** ». Elle sert d'identificateur et est indépendante de la couche matérielle.

Il existe actuellement 2 versions d'IP : IPv4 et IPv6. IPv4 est une ancienne version d'IP qui est utilisée sur la majorité des réseaux locaux. IPv6 a été conçu pour pallier au manque d'adresses disponibles sur le réseau Internet.

L'unité d'information la couche réseau est **le paquet**

D.1. L'adressage IP (IPv4)

a. Pourquoi une adresse IP alors qu'il existe déjà l'adresse MAC ?

Il est nécessaire de différencier les adresses de couche 2 et de couche 3, car l'adressage au niveau de chacune de ces couches n'a pas le même rôle :

- L'adressage MAC en couche 2 permet d'identifier les machines SUR UN MEME RESEAU.
- L'adressage IP en couche 3 permet d'adresser les machines SUR DES RESEAUX DISTINCTS.

Peut-on alors utiliser pour ces deux couches une seule de ces deux adresses ? La réponse est malheureusement non. Les adresses de couche 2 sont en rapport avec le matériel réseau utilisé (le protocole de couche 2 est géré au niveau de la carte connectée au réseau et non pas par le système d'exploitation comme les couches supérieures) il est donc difficile de modifier les adresses MAC qui sont censées être codées directement sur la carte réseau. Cela est notamment du au fait que chaque adresse MAC doit être unique sous peine de conflit matériel, et que cette adresse doit être accessible très tôt lors du boot d'une machine.

Les adresses de couche 3 quant à elles demandent une certaine souplesse d'utilisation car on ne connaît pas à priori l'adresse du réseau sur lequel une machine va se trouver. Il y a donc une incompatibilité d'utilisation d'une adresse de couche 2 pour une adresse de couche 3, et vice versa.

Enfin, les protocoles réseau évoluant au fil du temps, il est nécessaire que chaque couche soit indépendante des autres.

b. Constitution d'une adresse IP

Constituée de 4 octets, elle est découpée en 2 parties :

- Le numéro de réseau (netid)
- Le numéro de l'hôte sur ce réseau (hostid)

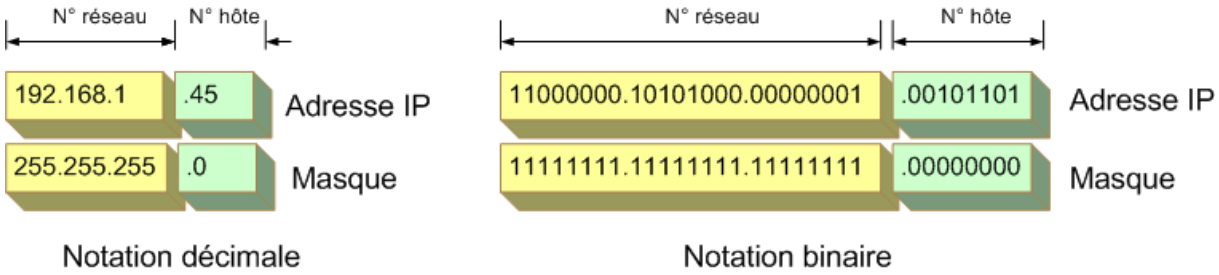
La taille du *netid* dépend de **la classe d'adresse IP** utilisée. Il existe plusieurs classes d'adresses IP dédiées à des usages différents. Plus le numéro de réseau est grand et plus le nombre d'hôtes sur ce réseau sera petit.

Principe d'adressage IP

c. Et le masque ?

Le masque agit comme un séparateur entre le n° de réseau et le n° d'hôte. Le masque est également constitué de 4 octets et est souvent associé à l'adresse IP.

Tous les bits à 1 du masque permettent de définir chaque bit correspondant de l'adresse IP comme un bit faisant partie du n° de réseau. Par opposition, tous les bits à 0 du masque permettent de définir chaque bit correspondant de l'adresse IP comme un bit faisant partie du n° d'hôte.



Exemple d'association entre adresse et masque

Le masque servant à faire la séparation en deux parties sur une adresse IP, il est donc indissociable de celle-ci. Une adresse seule ne voudra rien dire puisqu'on ne saura pas quelle est la partie réseau et quelle est la partie machine. De la même façon, un masque seul n'aura pas de valeur puisqu'on n'aura pas d'adresse sur laquelle l'appliquer. L'adresse IP et le masque sont donc liés l'un à l'autre, même si l'on peut choisir l'un indépendamment de l'autre.

d. Représentation du masque avec un / (/24)

Il existe une autre manière de noter le masque. Il s'agit de compter le nombre de bits à 1 du masque et de noter ce chiffre à la fin de l'adresse :

Par exemple, le couple adresse et masque suivant :

- 192.168.1.13
- 255.255.255.0

S'écrira dans cette nouvelle notation :

- 192.168.1.13 / 24

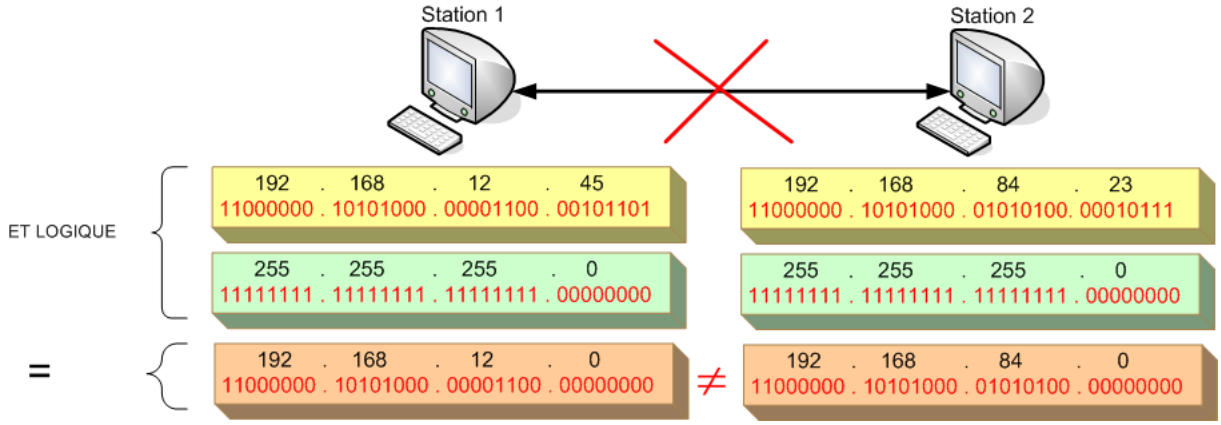
Le chiffre 24 indique que 24 bits du masque sont à 1.

> *Cette notation est aujourd'hui très utilisée dans tous les équipements réseaux récents afin de simplifier l'écriture et rendre la configuration plus aisée*

e. Application du masque

Pour que 2 stations puissent communiquer, la règle est la suivante :

- Un ET logique entre l'adresse IP et le masque de sous réseaux doit donner le même résultat



Exemple d'application du masque

f. Les classes d'adresses IP

Nous avons vu que l'adresse IP est indissociable de son masque. Néanmoins le choix des adresses et des masques ne doit pas être faite au hasard. C'est là qu'interviennent les classes A, B, C, D et E.

Les classes ont été – car elles ne le sont plus – un moyen efficace de segmenter l'espace d'adressage d'Internet. Nous verrons que si les classes sont encore très utilisées dans les réseaux locaux, d'autres techniques comme le « surnetting » ou l'adressage « CIDR » sont monnaies courantes sur Internet.

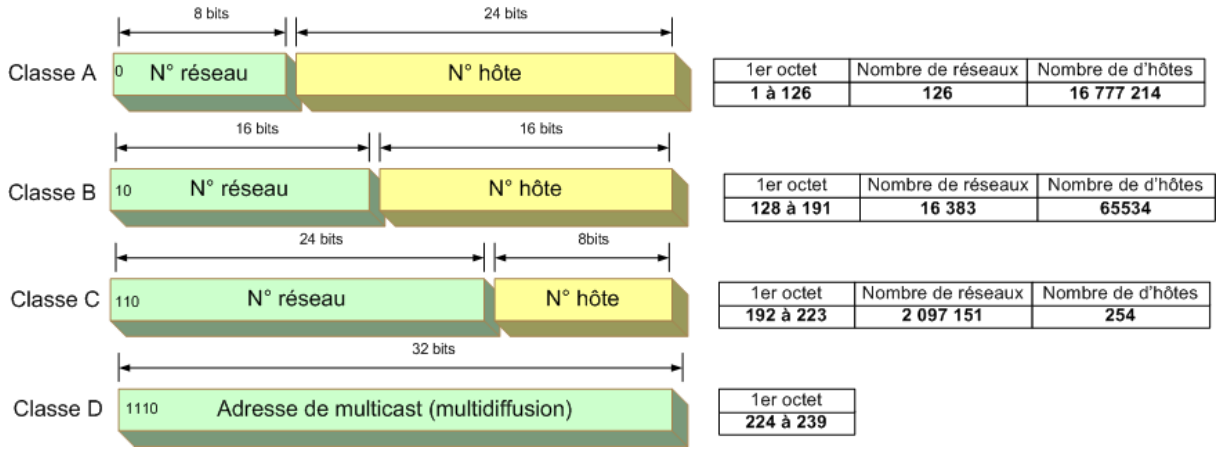
Pour savoir à quelle classe appartient une adresse IP, il faut se pencher sur sa représentation binaire. En effet, la valeur des 4 premiers bits de l'adresse détermine la classe.

Voici un tableau récapitulatif des différentes classes d'adresses IP en fonction des 1 er s bits du 1er octet :

Bits de poids fort	Classe
0	A
10	B
110	C
1110	D
1111	E

Tableau des classes d'IP en fonction des bits de poids fort

Ceci nous amène à faire quelques constatations sur le potentiel d'adresses et de réseaux détenu par chaque classe d'adresse :



Les classes d'adresses IP

g. Exemples

- 83.206.23.134 : Adresse de classe A , netid = 83 , hostid = 206.23.134
- 190.12.24.56 : Adresse de classe B , netid = 190.12 , hostid = 24.56
- 192.168.1.5 : Adresse de classe C, netid=192.168.1 , hostid=5

h. Les adresses IP publiques et privées

Les adresses publiques sont celles qui sont utilisées sur Internet. Pour des raisons que nous verrons lorsque nous parlerons du routage, un réseau local **ne doit pas utiliser d'adresses publiques**. Il utilise des adresses privées. La liste des adresses privées autorisées sont :

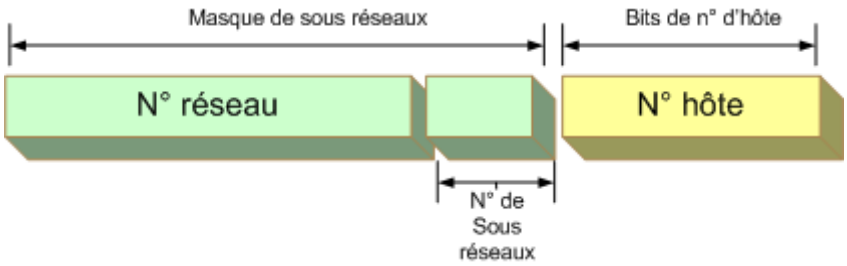
- Classe A : de 10.10.0.1 à 10.255.255.254
- Classe B : de 172.16.0.1 à 172.31.255.254
- Classe C : de 192.168.0.1 à 192.168.255.254

i. Technique du « subnetting »

L'objectif du « subnetting » est de scinder une classe d'adresses en sous réseaux indépendants. Chaque sous-réseau dispose d'un nombre limité d'adresses.

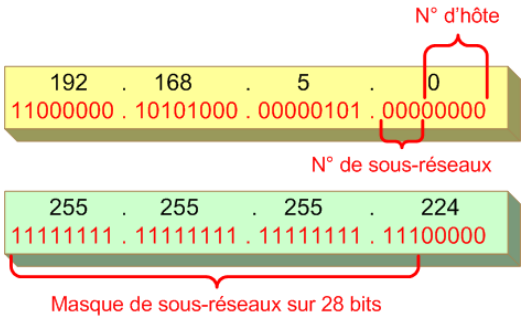
Cette technique vient combler certaines lacunes des classes d'adresses. En effet, les classes A,B et C sont mal adaptées aux besoins croissants des réseaux. La classe B, par exemple, offre un nombre total de 65534 hôtes alors que la classe C n'en offre que 254. Il apparaît que si les besoins sont légèrement supérieurs aux 254 hôtes qu'offre un classe C, il en résulte un gâchis d'adresses si on choisit un classe B.

L'idée est de réserver quelques bits du numéro de réseau pour créer des sous réseaux. A chaque n° de réseau, on associe un certain nombre de sous réseaux qui eux-mêmes contiennent un nombre déterminé d'hôtes :



Principe du masque de sous réseau

Dans l'exemple suivant basé sur un classe C, le masque de sous réseaux est de 28 bits, le numéro de sous réseaux sur 3 bits :



Exemple d'utilisation de la technique de « subnetting »

j. Technique du « supernetting »

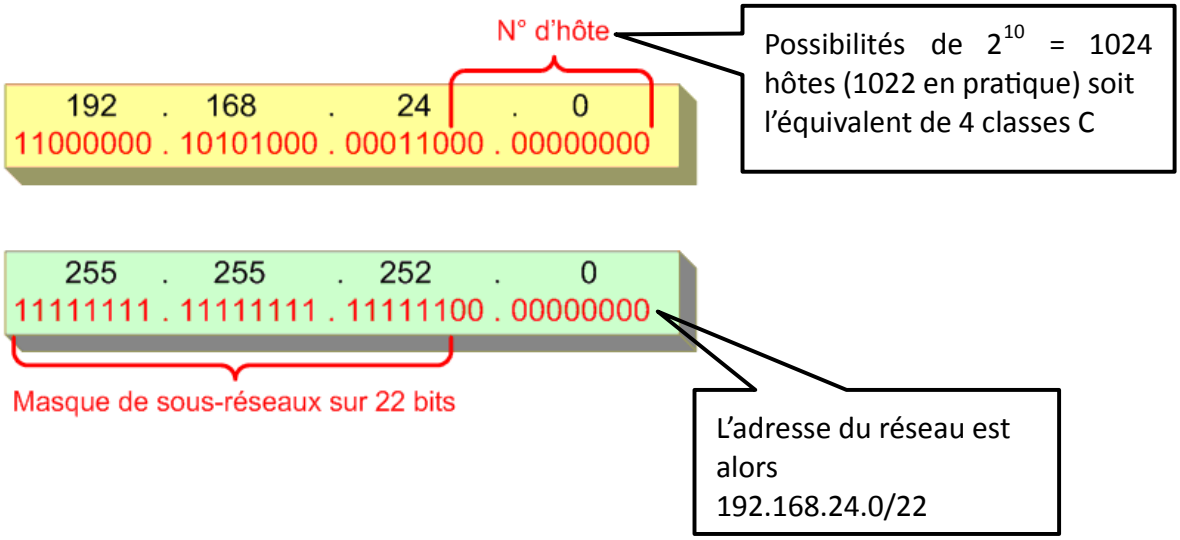
La technique du « supernetting » consiste en une agrégation de plusieurs classes (La classe C) pour former des réseaux de taille permettant d'éviter le gâchis d'adresses.

Nous avons vu que la technique du « subnetting » consistait à utiliser une partie du « netid » pour créer des sous réseaux. La technique du « supernetting » consiste à utiliser une partie du masque de sous-réseau.

En utilisant quelques bits du masque de sous-réseaux, cela revient à obtenir plusieurs réseaux d'une même classe d'adresse.

Voici un exemple d'un réseau utilisant le « supernetting » en agrégeant 4 classes C :

Ici, le numéro d'hôte (hostid) est sur 10 bits, soit 2 de plus qu'un classe C alors que le numéro de réseau est sur 22 bits.



Exemple d'agrégation de 4 classes C

k. Adressage *CIDR (Classless InterDomain Routing*)

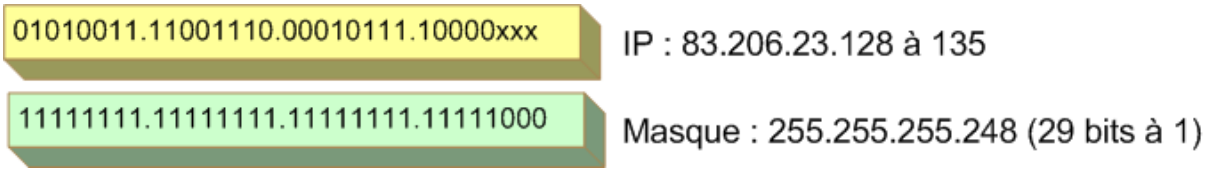
A partir de 1994, le nombre d'adresses IP n'était plus suffisant pour satisfaire toutes les demandes d'accès à Internet. De plus, les *tables de routage* des routeurs devenaient tellement énormes que les performances chutaient.

Si rien n'avait été fait, nous n'entendrions certainement plus parler d'Internet aujourd'hui.

Heureusement, l'adressage basé sur les classes A, B et C a été remplacé par l'adressage CIDR. Il s'agit d'une technique permettant de **répartir plus efficacement les adresses IP** entre les demandeurs.

Au lieu d'avoir une taille fixe pour le numéro de réseau (8,16 ou 24), l'adressage CIDR utilise une taille variable pouvant aller de 13 à 29.

Par exemple pour le numéro de réseau 83.206.23.128 et un masque de 255.255.255.248 on obtient **un réseau de 8 postes connectés directement à Internet** (« pool d'IP fixes »)



Exemple de « pool de 8 IP fixes »

Une adresse prise dans le pool s'écrit : **83.206.23.131/29**

Voici une liste d'adressage possible avec leur équivalent en adressage classique :

Masque	Nombre d'adresses	Equivalent classe C
/28	16	1/16ème de classe C
/27	32	1/8ème de classe C
/26	64	1/4ème de classe C
/25	128	1/2 classe C
/24	256	1 classe C
/23	512	2 classes C
/22	1 024	4 classes C
/21	2 048	8 classes C
/20	4 096	16 classes C
/19	8 192	32 classes C
/18	16 384	64 classes C
/17	32 768	128 classes C
/16	65 536	256 classes C (1 classe B)
/15	131 072	512 classes C
/14	262 144	1024 classes C
/13	524 288	2048 classes C

Listes des différents adressages CIDR possibles

D.2. Les adresses IP spéciales

Certaines adresses IP sont spéciales et ne peuvent être utilisées par un hôte sur le réseau.

Comme nous l'avons vu, une adresse IP est composée d'un numéro de réseau complété par un numéro d'hôte : {**<numéro de réseau>** , **<numéro d'hôte>**} = {**<netid>** , **<hostid>**}.

Certaines valeurs sont donc interdites (0 signifie *tous les bits à 0* et -1 signifie *tous les bits à 1* :

- {<netid> , 0}
- {<netid> , -1}
- {-1 , -1}
- {0 , 0}

- {0 , <hostid>}
- {127 , <quelconque>}
- Adresses de classe D

a. {<netid> , 0} : Numéro de réseau logique

Cette adresse indique un réseau logique, et donc l'ensemble de ses nœuds. Elle ne peut donc être utilisée ni comme source ni comme destination.

Exemple : 192.168.34.0 ne peut pas être affectée à un hôte

b. {<netid> , -1} : Diffusion dirigée (Broadcast)

Cette adresse est utilisée pour contacter **tous les hôtes d'un même réseau logique**. Dans ce cas, un paquet IP qui a comme destination cette adresse sera envoyé à tous les hôtes appartenant au même réseau logique (Et sous réseaux)

Exemple : 192.168.34.255

c. {-1 , -1} : Diffusion (broadcast)

Cette adresse est utilisée pour contacter **tous les hôtes quels que soient leur réseau logique**. Dans ce cas, un paquet IP qui a comme destination cette adresse sera envoyé à tous les hôtes du réseau physique.

Exemple : 255.255.255.255

Bien évidemment, la plupart des routeurs sont programmés pour ne pas laisser passer ce genre de diffusion (Imaginez un instant que l'on puisse envoyer un message de ce type sur Internet !!!!)

d. {0 , 0} : DHCP et BOOTP

Les protocoles DHCP (*Dynamic Host Configuration Protocol*) et BOOTP sont prévus pour permettre à des postes qui n'ont pas d'adresse IP d'en obtenir une automatiquement sans intervention particulière sur la station.

Cette adresse IP est utilisée pour effectuer une demande d'adresse IP auprès d'un serveur DHCP.

Exemple : 0.0.0.0

e. {0 , <hostid>} : Hôte dans tous les réseaux logiques

Un paquet IP envoyé avec cette adresse de destination sera envoyé à tous les hôtes ayant le même numéro d'hôte mais situés dans des réseaux logiques différents.

Exemple : 0.0.0.12 concernera 192.168.0.12 et 192.168.23.12 , etc, etc …

f. {127 , <quelconque>} : Adresse de boucle locale

Un paquet IP envoyé avec cette adresse de destination sera reçu…. Par l'envoyeur lui-même. Le paquet ne sort pas sur la carte réseau, il reste local.

g. Adresses de classe D

Il s'agit d'adresses spéciales permettant d'envoyer des paquets IP à un groupe de stations. En effet, une station peut se mettre à l'écoute d'une de ces adresses de multidiffusion. Dans ce cas, elle recevra les données.

L'appartenance à un groupe est dynamique. C'est-à-dire qu'une station peut très bien décider d'entrer ou de quitter un groupe de multidiffusion

Parmi les adresses de multidiffusion, il en existe ayant une signification spéciale. La liste suivante n'est pas exhaustive. (La liste complète peut-être obtenue en consultant l'adresse http://www.iana.org/assignments/multicast-addresses) :

- **224.0.0.1** : Tous les hôtes d'un même sous réseau
- **224.0.0.2** : Tous les routeurs d'un même sous réseau
- **224.0.0.9** : Routeurs utilisant le protocole RIP2
- …

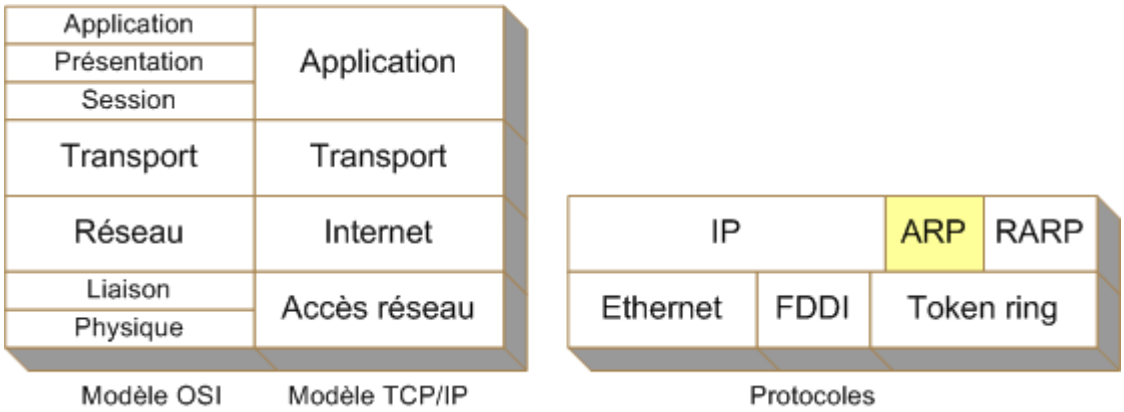
D.3. L'adressage IP (IPv6)

A venir

D.4. Le protocole ARP (Address Resolution Protocol)

Lorsqu'une station veut dialoguer avec une autre sur un même segment de réseau, elle doit connaître l'adresse physique du destinataire (MAC). Une station ne peut pas connaître à l'avance toutes les adresses physiques des stations faisant partie de son segment de réseau.

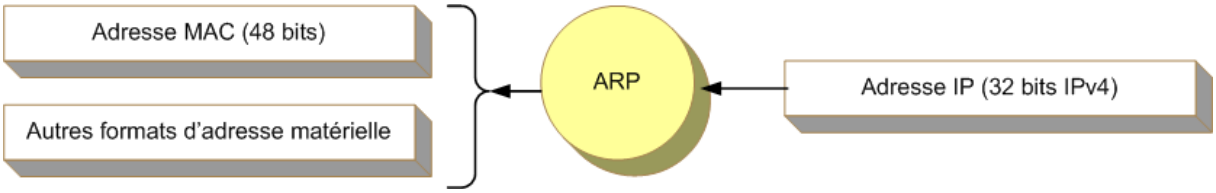
Le protocole ARP se situe sur la couche 3 (Réseau) du modèle OSI



Correspondance OSI – TCP/IP - Protocoles

a. Le principe

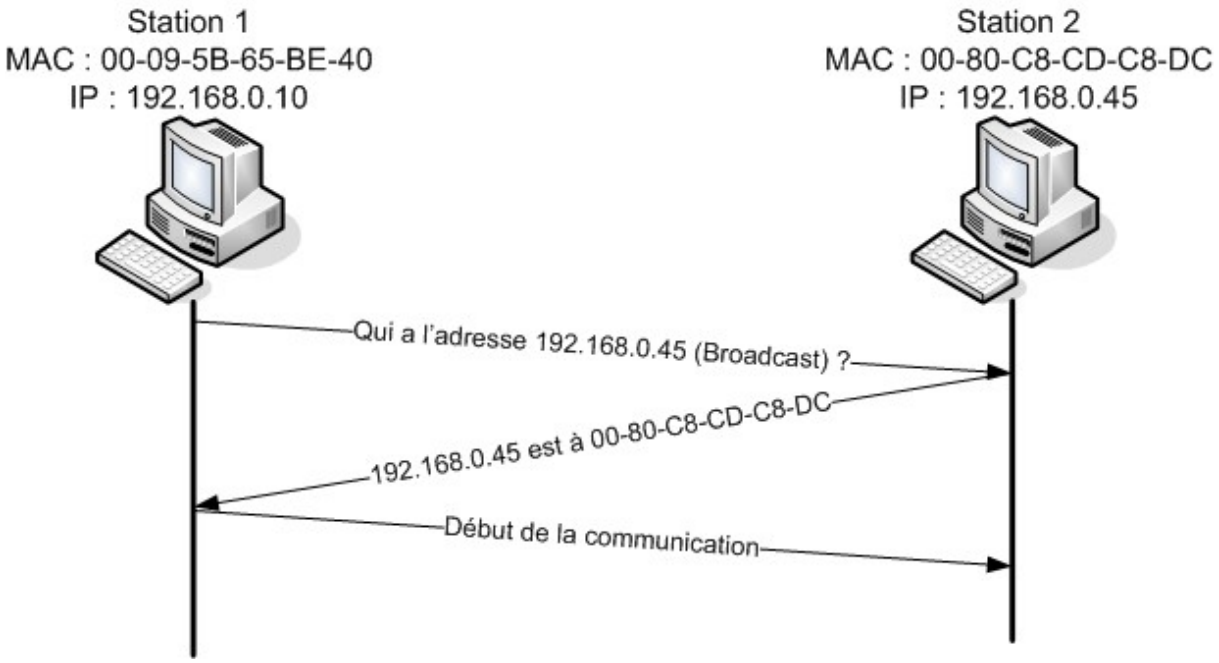
Il existe donc un protocole (ARP) permettant de connaître l'adresse MAC à partir d'une adresse IP :



Résolution ARP

Lorsque qu'une station veut dialoguer avec une autre, elle diffuse sur le réseau un message indiquant qu'elle souhaite connaître l'adresse physique du destinataire. Pour diffuser ce message, elle utilise le protocole ARP.

Voici comment 2 stations peuvent communiquer sur un réseau local :



Exemple de requête ARP

Pour des raisons évidentes de performances, chaque station conserve ***un cache de toutes les adresses MAC*** *connues dans les requêtes précédentes, ce qui évite de « polluer » le réseau de nouvelles demandes inutiles. (`arp -a` sous windows)*

b. Format des messages ARP

L'entête de la trame ARP respecte la structure suivante :

0 … 8	8 … 15	16 … 31
Type matériel		Type de protocole
HLEN	PLEN	Opération
Adresse MAC du demandeur		
Adresse MAC demandeur (suite)		Adresse IP demandeur
Adresse IP demandeur (suite)		Adresse MAC destination
Adresse MAC destination (suite)		
Adresse IP destination		

Format d'entête ARP

Voici la signification des champs de la trame :

- **Type matériel** : Fait référence au format de la trame ARP qui dépend fortement des couches inférieures. (01 : Ethernet, 16 : ATM)
- **Type de protocole** : Nombre sur 2 octets représentant le type de protocole de couche 3 utilisant ARP (0x0800 pour IP)
- **HLEN** : Taille en octet de l'adresse matérielle (6 sur Ethernet)
- **PLEN** : Taille en octet de l'adresse du protocole (4 pour IP)
- **Opération** : 2 types d'opération possible (ARP request , ARP reply)
- **Adresse MAC du demandeur**
- **Adresse IP du demandeur**
- **Adresse MAC du destinataire** (00 :00 :00 :00 :00 :00 lorsque l'adresse MAC n'est pas encore connue)
- **Adresse IP du destinataire**

c. Exemple de trame

Voici une trame ARP capturée en utilisant un analyseur de protocole (Ethereal) :

```
Ethernet II, Src: 192.168.1.2 (00:13:ce:0e:1c:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  Destination: Broadcast (ff:ff:ff:ff:ff:ff)   Adresse de broadcast comme destinataire
  Source: 192.168.1.2 (00:13:ce:0e:1c:00)   Adresse MAC de l'envoyeur
  Type: ARP (0x0806)   Que contient la trame ?
Address Resolution Protocol (request)
  Hardware type: Ethernet (0x0001)
  Protocol type: IP (0x0800)   Type de protocole de couche utilisant ARP
  Hardware size: 6   Taille d'une adresse MAC
  Protocol size: 4   Taille d'une adresse IP
  Opcode: request (0x0001)   Il s'agit d'une requête
  Sender MAC address: 192.168.1.2 (00:13:ce:0e:1c:00)   Adresse MAC du demandeur
  Sender IP address: 192.168.1.2 (192.168.1.2)
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)   Adresse MAC pas encore connue
  Target IP address: 192.168.1.3 (192.168.1.3)   Adresse IP de la requête
```

Exemple de trame ARP, couches 2 et 3

d. Exemple de séquence ARP

Voici le cas typique d'une communication ARP basée sur le principe de requête-réponse :

La requête d'abord :

```
Ethernet II, Src: 192.168.1.2 (00:13:ce:0e:1c:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  Source: 192.168.1.2 (00:13:ce:0e:1c:00)
  Type: ARP (0x0806)
Address Resolution Protocol (request)
  Hardware type: Ethernet (0x0001)
  Protocol type: IP (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (0x0001)   Il s'agit d'une requête
  Sender MAC address: 192.168.1.2 (00:13:ce:0e:1c:00)   Adresse MAC du demandeur
  Sender IP address: 192.168.1.2 (192.168.1.2)
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)   Adresse MAC pas encore connue
  Target IP address: 192.168.1.3 (192.168.1.3)   Adresse IP objet de la requête
```

La réponse ensuite :

```
⊟ Ethernet II, Src: 192.168.1.3 (00:15:f2:05:68:de), Dst: 192.168.1.2 (00:13:ce:0e:1c:00)
    Destination: 192.168.1.2 (00:13:ce:0e:1c:00)
    Source: 192.168.1.3 (00:15:f2:05:68:de)
    Type: ARP (0x0806)
    Trailer: 000000000000000000000000000000000000
⊟ Address Resolution Protocol (reply)
    Hardware type: Ethernet (0x0001)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (0x0002)        Il s'agit d'une réponse
    Sender MAC address: 192.168.1.3 (00:15:f2:05:68:de)    Adresse MAC demandée
    Sender IP address: 192.168.1.3 (192.168.1.3)
    Target MAC address: 192.168.1.2 (00:13:ce:0e:1c:00)    Adresses du demandeur
    Target IP address: 192.168.1.2 (192.168.1.2)
```

e. La table ARP

Afin de ne pas saturer le réseau de requêtes ARP inutiles, la totalité des stations et même les éléments réseau tels que les commutateurs et routeurs gardent un tableau de correspondance entre adresse IP et MAC dans un cache ARP (station) ou table des adresses MAC (commutateur).

Ce cache doit être rafraichi périodiquement pour éviter la présence d'entrées erronées dans la table. La fréquence de rafraîchissement dépend des types d'équipements et peut dans la plupart des cas être modifiée.

Ces tables peuvent être statiques ou dynamiques. Seule la table dynamique doit être rafraîchie.

```
C:\>arp -a

Interface : 192.168.1.2 --- 0x20003
  Adresse Internet      Adresse physique      Type
  192.168.1.1           00-03-c9-9f-4f-ee     dynamique
  192.168.1.3           00-15-f2-05-68-de     dynamique
```

Table ARP d'une station sous Windows

```
routeur:~# arp -i eth1
Address                  HWtype  HWaddress           Flags Mask            I
web.diderot.org          ether   00:C0:9F:33:6B:E1   C                     e
mail.diderot.org         ether   00:C0:9F:33:6A:F9   C                     e
```

Table ARP d'une station sous Linux

D.5. Le protocole RARP (*Reverse ARP*)

Utilisé par des stations n'ayant pas d'adresse IP (Souvent des terminaux) pour obtenir l'adresse IP qui leur correspond. Dans ce cas, **un serveur RARP** (Souvent un routeur) **stocke dans une table ARP la correspondance entre IP et MAC.**

La table stockée est statique et suppose de connaître à l'avance toutes les adresses MAC d'un réseau (Ce qui est souvent très fastidieux surtout lors d'ajout de machine).

Dans la plupart des réseaux Ethernet, on utilise plutôt **des solutions DHCP** (*Dynamic Host Configuration Protocol*) ou **BOOTP**

D.6. Le protocole IP

« *Du point de vue de l'utilisateur, le réseau Internet est un réseau virtuel unique qui interconnecte toutes les machines et au travers duquel il peut communiquer* » (Pujolle – Les réseaux)

Ce protocole est chargé **de délivrer des paquets** vers une destination. Le chemin parcouru par ces paquets n'est pas connu à l'avance. Ce qui signifie que certains paquets peuvent prendre **des routes différentes et donc arriver dans le désordre**. **L'émetteur doit ordonnancer ces paquets** et **le récepteur doit ré assembler les paquets** pour reconstituer le message original.

a. Principe

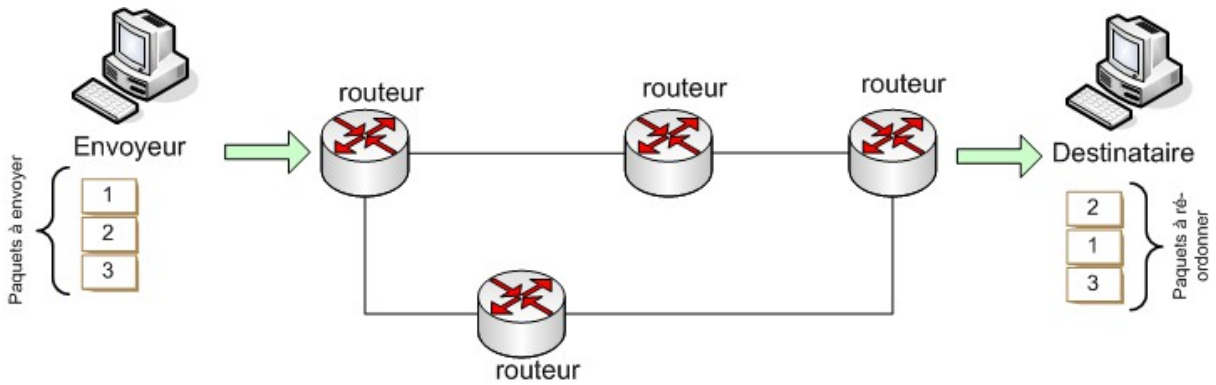
Ce message original est appelé « **datagramme IP** ». Ce datagramme qui peut être de taille variable et sera fragmenté pour des raisons de performances. En effet, un routeur chargé d'orienter un paquet vers un autre réseau doit pouvoir traiter un nombre important de paquets venant d'expéditeurs différents.

Il faut trouver un compromis pour :

- Alléger le travail des routeurs pour qu'ils puissent traiter un maximum de demandes
- Minimiser le travail de réassemblage du destinataire

> *C'est pourquoi nous verrons que pour le routage, chaque réseau dispose d'un MTU (*Maximum Transfer Unit*) qui précise la taille maximum d'un paquet traversant ce réseau*

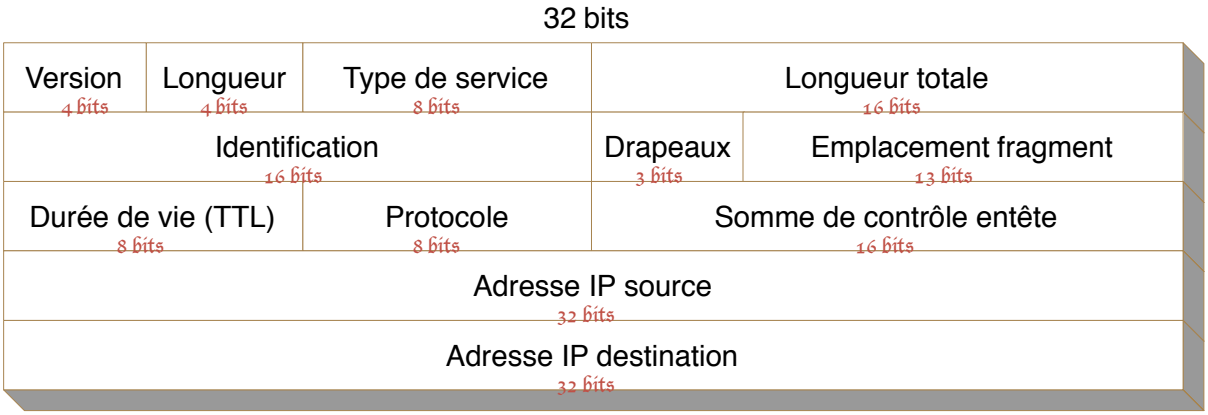
Fragmenter le datagramme IP revient donc à le diviser en plusieurs paquets. Chaque paquet a le même format que le datagramme d'origine. L'entête de chaque fragment reprend la plupart des informations de l'entête d'origine.



Exemple de réordonnancement de paquets

b. Le datagramme IP (IPv4)

Chaque paquet a le format suivant :



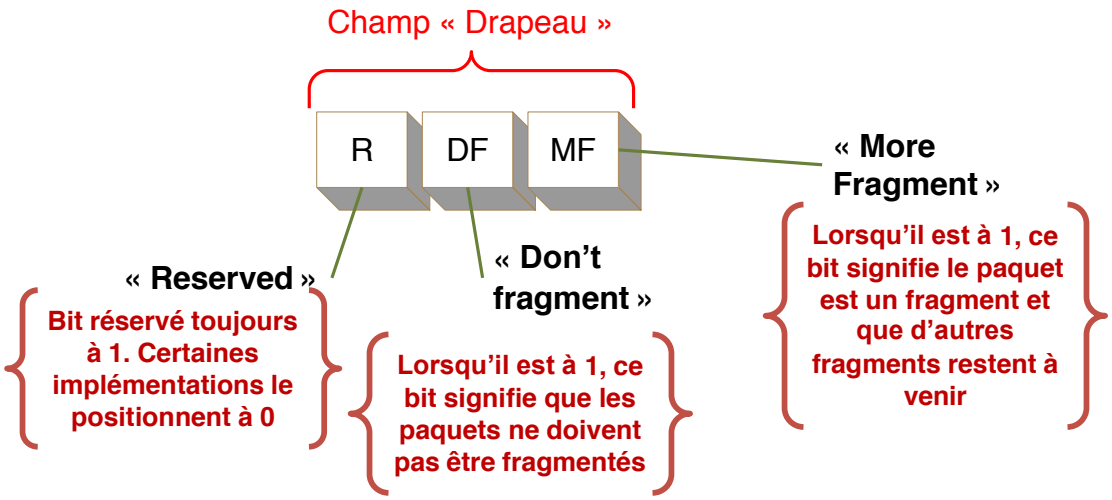
Paquet IPv4

Voici la signification des champs :

- **Version (4 bits)** : 4 pour IPv4 ou 6 pour IPv6
- **Longueur d'entête (4 bits)** : Permet de connaître rapidement l'endroit où se situe le début des données du paquet.
- **Type de service ou ToS (8 bits)** : Précise le type de service fourni par les données. Peu utilisé jusqu'à l'arrivée de protocoles gérant la qualité de service (QoS). La plupart des équipements de « backbone » n'utilise pas ce champ et certains le réinitialise à 0.
- **Longueur totale (16 bits)** : Longueur totale du paquet (en-tête comprise)
- **Identification (16 bits)** : Identificateur de datagramme. Chaque paquet possédant le même n° d'identification fait partie du même datagramme.
- **Drapeaux (3 bits)** : Voir page suivante
- **Emplacement fragment (13 bits)** : Si le datagramme a été fragmenté, ce champ indique l'emplacement du fragment courant dans l'ensemble des fragments
- **Temps de vie (8 bits)** : Durée de vie du paquet. Lorsqu'il est à 0, le paquet est détruit. Habituellement, ce champ est décrémenté à chaque passage par un routeur
- **Numéro de protocole (8 bits)** : Indique le type de protocole encapsulé dans les données (0x06 pour TCP). La liste complète peut-être obtenue à l'adresse http://www.daemon.org/ip.html#protolist
- **Somme de contrôle d'erreur (16 bits)** : Code de contrôle d'erreur de l'en-tête (Comme le TTL varie à chaque passage par un routeur, ce code doit être recalculé à chaque fois)
- **Adresse source et destination (32 bits chacune)**
- **Données (<65535 octets) :** Données du paquet

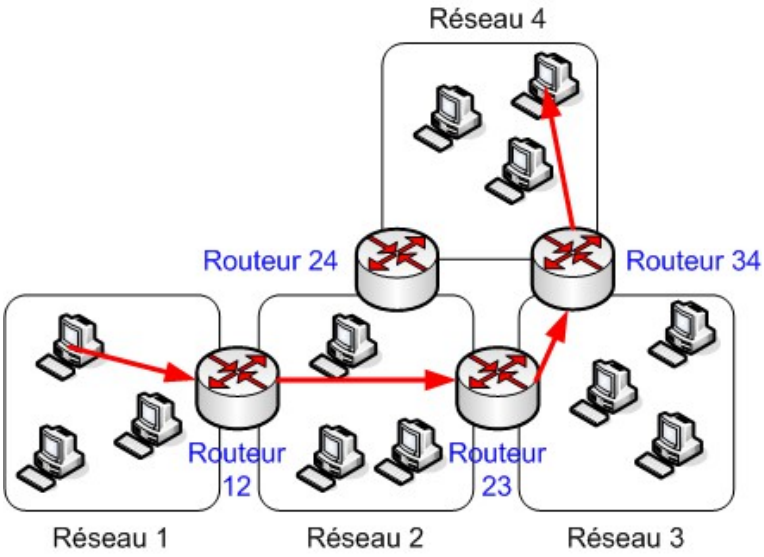
c. Les drapeaux

Sur 3 bits, le champ « drapeau » donne des indications sur l'état de fragmentation du paquet :


Champ « Drapeau »
R
DF
MF
« More Fragment »
Lorsqu'il est à 1, ce bit signifie le paquet est un fragment et que d'autres fragments restent à venir
« Reserved »
Bit réservé toujours à 1. Certaines implémentations le positionnent à 0
« Don't fragment »
Lorsqu'il est à 1, ce bit signifie que les paquets ne doivent pas être fragmentés

D.7. Le routage IP

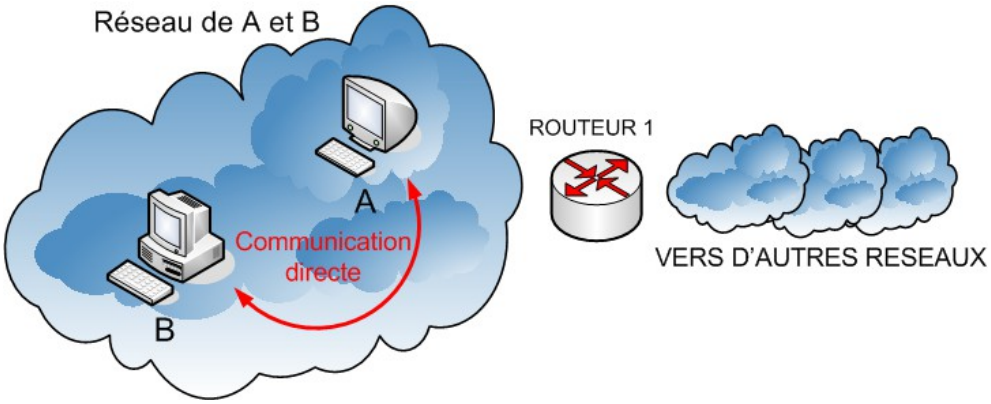
Il s'agit sans là de la pierre angulaire des réseaux TCP/IP. Le routage **est une orientation des paquets vers un réseau de destination**. Chaque routeur est connecté à plusieurs réseaux. Ils maintiennent à jour **une table de routage** leur permettant de prendre des décisions en fonction des caractéristiques du paquet à orienter.



Exemple de routage

a. Le routage direct

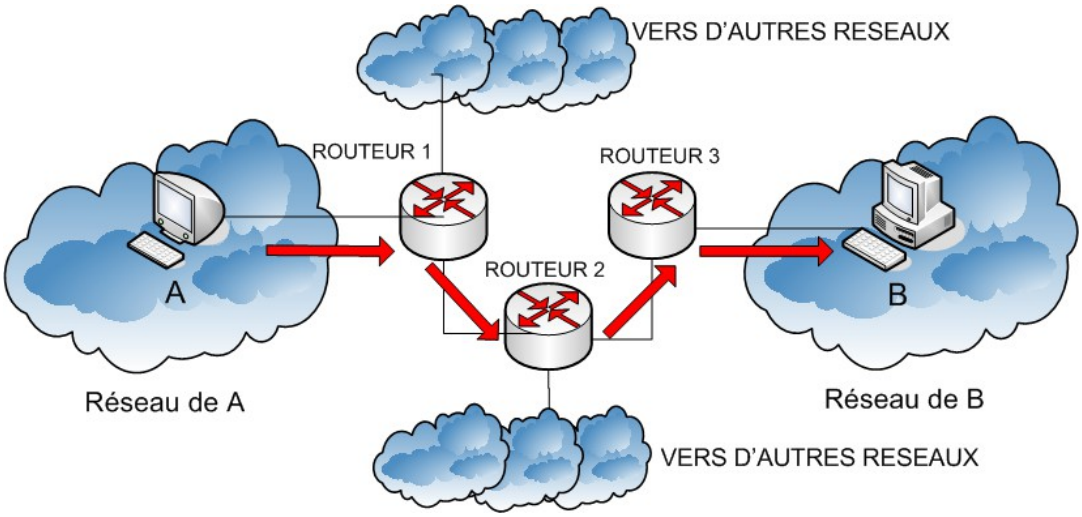
C'est le cas de deux stations situées dans le même segment de réseau. Il peut s'agir aussi d'une communication entre une station et un routeur. Dans ce cas, la table de routage interne de la station est utilisée pour envoyer le paquet directement au destinataire dont l'adresse physique aura été connue (via le protocole ARP)



Cas du routage direct

b. Le routage indirect

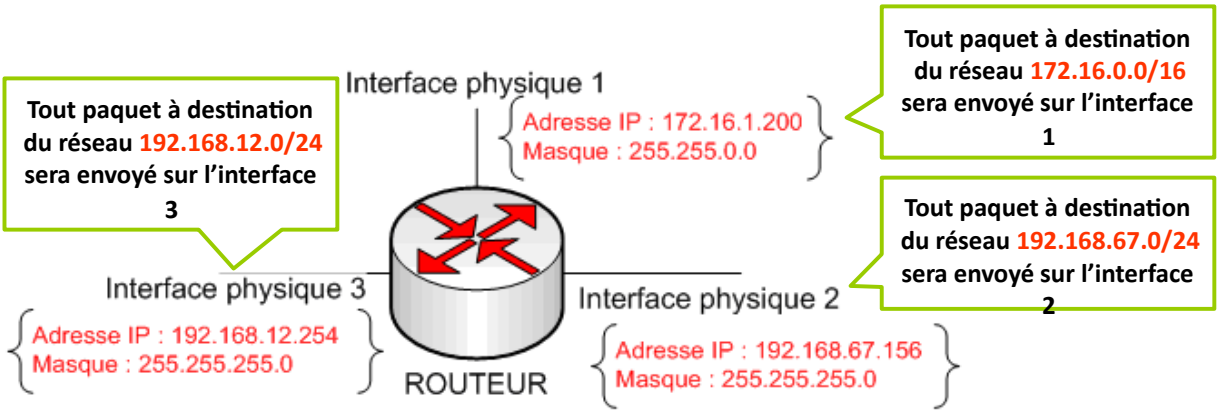
Dans ce cas, le routage est plus complexe car il faut déterminer le routeur qui va se charger d'orienter le paquet vers le bon réseau. C'est généralement le rôle de **la passerelle par défaut** que tout client peut utiliser lorsque l'adresse IP du destinataire n'est pas une adresse IP faisant partie du sous réseau logique.



Cas du routage indirect

c. La table de routage

Le routage est effectué à partir **du numéro de réseau de l'adresse IP de l'hôte de destination**. La table contient, pour chaque numéro de réseau à atteindre, **l'adresse IP du routeur auquel il faut envoyer le paquet**.



Exemple de constitution d'une table de routage

Ce qui donnerait la table de routage suivante :

Réseau destination	Masque	Interface
192.168.12.0	255.255.255.0	ethernet3
192.168.67.0	255.255.255.0	ethernet2
172.16.0.0	255.255.0.0	ethernet1
0.0.0.0	0.0.0.0	ethernet1

d. Exemple de table de routage

```
routeur:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
83.206.23.128   0.0.0.0         255.255.255.248 U     0      0        0 eth3
172.16.101.0    172.16.1.101    255.255.255.0   UG    2      0        0 eth2
172.16.100.0    172.16.1.100    255.255.255.0   UG    2      0        0 eth2
172.16.103.0    172.16.1.103    255.255.255.0   UG    2      0        0 eth2
172.16.102.0    172.16.1.102    255.255.255.0   UG    2      0        0 eth2
172.16.115.0    172.16.1.115    255.255.255.0   UG    2      0        0 eth2
192.168.0.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
172.16.1.0      0.0.0.0         255.255.255.0   U     0      0        0 eth2
172.16.114.0    172.16.1.114    255.255.255.0   UG    2      0        0 eth2
172.16.109.0    172.16.1.109    255.255.255.0   UG    2      0        0 eth2
172.16.200.0    172.16.1.200    255.255.255.0   UG    2      0        0 eth2
172.16.108.0    172.16.1.108    255.255.255.0   UG    2      0        0 eth2
172.16.111.0    172.16.1.111    255.255.255.0   UG    2      0        0 eth2
172.16.110.0    172.16.1.110    255.255.255.0   UG    2      0        0 eth2
172.16.105.0    172.16.1.105    255.255.255.0   UG    2      0        0 eth2
172.16.104.0    172.16.1.104    255.255.255.0   UG    2      0        0 eth2
172.16.107.0    172.16.1.107    255.255.255.0   UG    2      0        0 eth2
172.16.106.0    172.16.1.106    255.255.255.0   UG    2      0        0 eth2
10.0.0.0        0.0.0.0         255.0.0.0       U     0      0        0 eth0
0.0.0.0         83.206.23.134   0.0.0.0         UG    0      0        0 eth3
routeur:~#
```

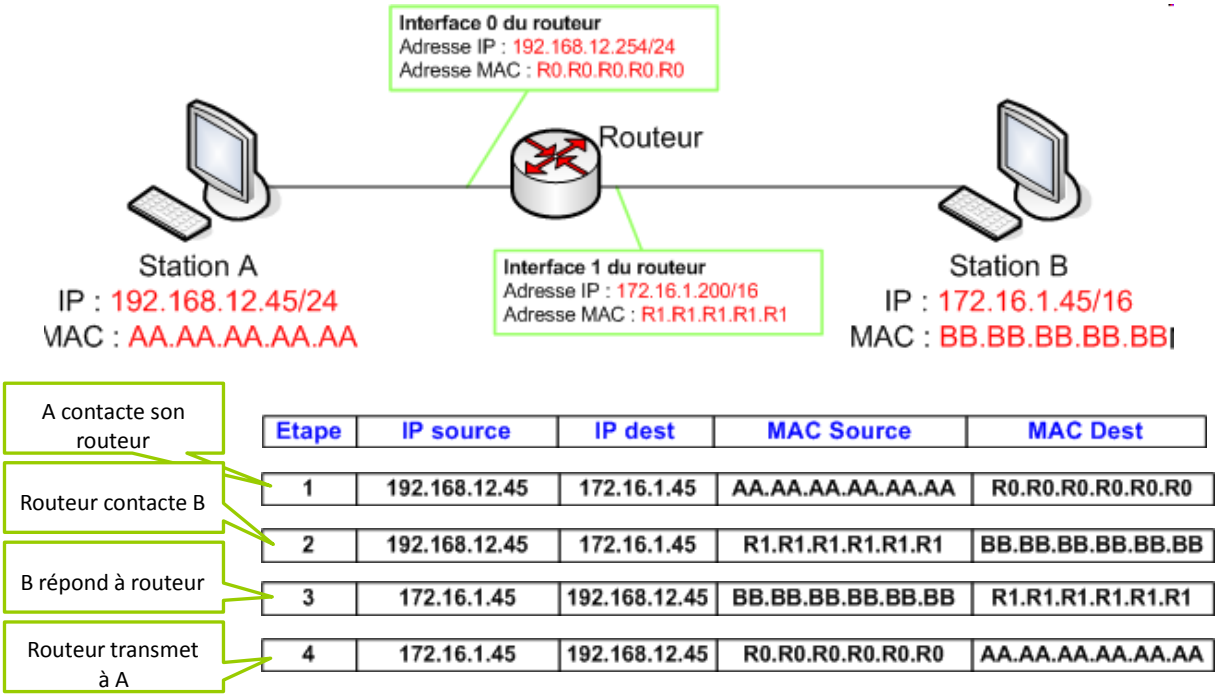
Table de routage d'un routeur sous LINUX

Voici la signification des colonnes intéressante de cet exemple :

- **Destination** : Il s'agit de l'adresse de l'hôte ou du réseau de destination. Par exemple 192.168.0.0 avec un masque de 255.255.255.0 désigne le numéro de réseau logique 192.168.0
- **Gateway** : La passerelle (routeur) à utiliser pour joindre ce réseau (0.0.0.0 signifie qu'il s'agit du routeur sur lequel est stocké cette table)
- **Genmask** : Masque de sous réseau associé à l'adresse de destination
- **Flags** : Précise l'état courant de la route
- **Metric** : Le nombre de sauts max (routeurs) pour parvenir jusqu'au réseau de destination
- **Ref** : Nombre de références à cette route
- **Use** : Nombre de recherches effectuées sur cette route
- **IFace** : Interface matérielle ou logicielle utilisée pour envoyer les paquets vers cette route

e. Le routage et la couche 2 (MAC)

Le routage intervient au niveau de la couche 3 du modèle OSI. Nous allons voir comment la couche 2 peut utiliser les informations de la couche 3 pour mener à bien une communication entre 2 stations situées dans des réseaux différents.



Etape	IP source	IP dest	MAC Source	MAC Dest
1	192.168.12.45	172.16.1.45	AA.AA.AA.AA.AA.AA	R0.R0.R0.R0.R0.R0
2	192.168.12.45	172.16.1.45	R1.R1.R1.R1.R1.R1	BB.BB.BB.BB.BB.BB
3	172.16.1.45	192.168.12.45	BB.BB.BB.BB.BB.BB	R1.R1.R1.R1.R1.R1
4	172.16.1.45	192.168.12.45	R0.R0.R0.R0.R0.R0	AA.AA.AA.AA.AA.AA

Exemple de routage indirect avec les informations de couche 2 (MAC)

Nous avons vu précédemment que les adresses matérielles (MAC) des stations d'un réseau physique ne sont pas connues par les stations d'autres réseaux physiques.

Une station A souhaitant communiquer avec une station B, va donc utiliser l'adresse MAC de sa passerelle par défaut. De l'autre côté, la station B va répondre en utilisant l'adresse MAC de l'interface 1 du routeur. A aucun moment, la station B ne connaît l'adresse MAC de la station A.

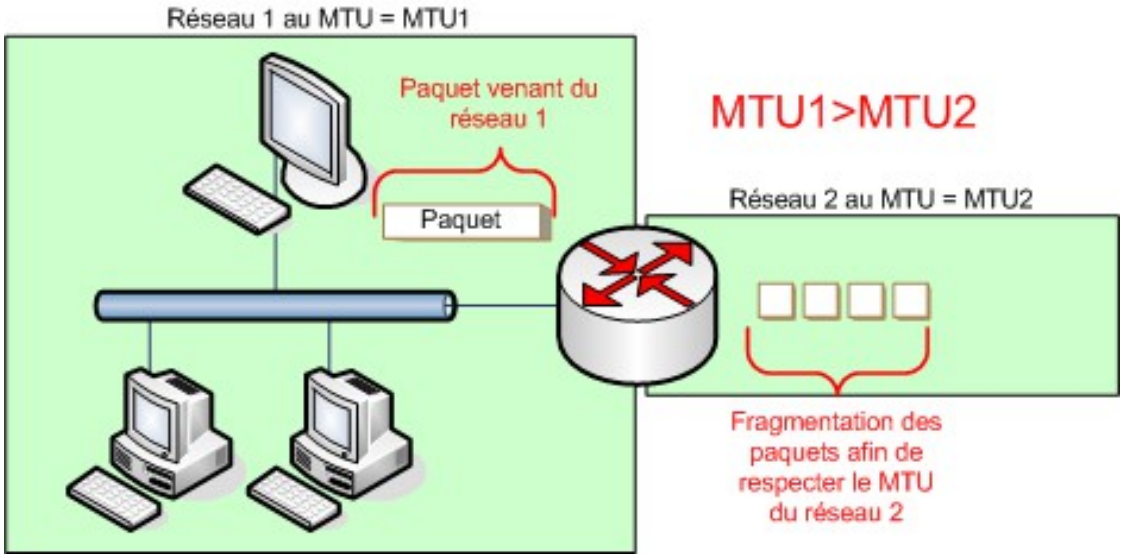
f. MTU (Maximum Transmission Unit)

Dans le cas de transfert de données dans un réseau routé, la capacité des routeurs à traiter le maximum de paquets en provenance de sources différentes dans un minimum de temps est déterminante.

En effet, un utilisateur donné ne doit monopoliser la bande passante des routeurs en transmettant de gros fichiers par exemple.

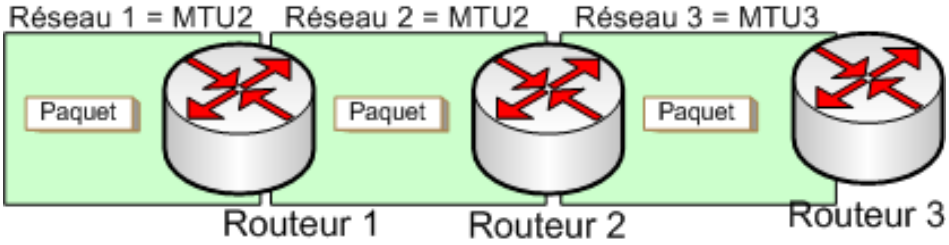
De cette remarque est née l'idée de la fragmentation de données. Cette fragmentation va permettre à chaque routeur de transférer des paquets d'une taille convenable qui permettra à ces équipements de répartir équitablement la bande passante.

Le MTU correspond justement à la taille maximum d'un paquet qui peut être transféré sur un réseau. Au-delà de cette taille, le paquet devra être fragmenté.



Exemple d'illustration de la notion de MTU

L'idéal consiste à avoir un MTU identique sur l'ensemble du réseau.



MTU1=MTU2=MTU3

Cas d'un MTU identique sur l'ensemble des réseaux

g. Le TTL (Time To Live)

Imaginons un instant que l'on envoie un paquet sur un réseau IP en utilisant une adresse IP qui n'existe pas. Dans la pratique, le paquet peut tourner indéfiniment de routeur en routeur sans jamais trouver son destinataire. Si rien n'est fait, un réseau comme Internet serait très vite saturé.

C'est pourquoi, la notion de temps de vie est définie pour chaque paquet transmis. Le fameux TTL représente le nombre routeurs que le paquet peut traverser jusqu'à ce qu'il soit détruit.

En fait, chaque fois que le paquet traverse un routeur, son champ TTL est diminué de 1. Le routeur qui constate qu'un paquet a un TTL de 0 le détruit sans autre forme de procès.

Lorsque qu'un routeur élimine un paquet, il envoie { la source un message ICMP l'informant que le paquet a été éliminé pendant son transit. Le message ICMP envoyé est de type 11, code 0 (Voir chap ICMP).

D.8. Les protocoles de routage

a. Introduction

Il existe deux types de protocole de routage :

- Les IGP (*Interior Gateway Protocol*)
- Les EGP (*Exterior Gateway Protocol*)

Les IGP permettent d'assurer le routage pour un « domaine de routage ». Un domaine de routage est un réseau géré par la même entité administrative et technique. Dans ce cas, les administrateurs des ces réseaux peuvent choisir leur propre organisation de routage.

Par opposition les EGP assurent l'interconnexion avec les différents domaines de routage. Ils sont souvent associés à des réseaux publics ou gérés par des prestataires différents ou des opérateurs de télécommunication.

b. RIP (Routing Information Protocol)

RIP est un protocole de type « IGP »

Ce protocole permet à des routeurs de dialoguer entre eux pour **déterminer le chemin le plus court**. Il permet également à un routeur d'**obtenir des informations sur l'état des autres routeurs**, offrant ainsi **une meilleure adaptation lorsqu'un routeur ou une liaison est défaillante** .

RIP est bien adapté aux réseaux locaux où le nombre de sauts pour passer d'une extrémité du réseau à l'autre ne dépasse pas 15. Au-delà, le routeur considère que le réseau est inaccessible (valeur 16).

La version RIP1 a été améliorée par RIP2 sur plusieurs points :

- Routage par sous réseau
- Authentification des messages entre routeurs
- Transmission multipoint
- ….

c. *OSPF (Open Shortest Path First*)

OSPF est un protocole de type IGP

Il s'agit de la 2 ème génération de protocole de routage. Il est plus complexe et plus adapté aux grands réseaux. Il maintient une véritable base de données distribuée sur l'état des différents routeurs et des liaisons.

d. Les autres protocoles de routage

A noter quelques autres protocoles de routage :

- **IS-IS** (IGP) : Développé par 'ISO , il décompose le réseau en domaines
- **IGRP** (IGP) : Développé par CISCO pour ses routeurs. Il s'agit d'une version améliorée de RIP
- **EIGRP** (IGP) : Amélioration d'IGRP pour lutter contre les bouclages
- **BGP** *(Border Gateway Protocol)* : Amélioration du protocole EGP permettant de gérer les bouclages involontaires.
- **IDRP** (*InterDomain Routing Protocol*) : Protocole de type EGP basé sur BGP

D.9. Gestion et contrôle

Nous savons que le protocole IP n'offre aucune garantie qu'un paquet a été transmis correctement. IP travaille en mode déconnecté. Afin de gérer le contrôle des flux de paquets et les éventuelles erreurs de routage et les problèmes de congestion du réseau, on fait appel à divers protocoles :

- ICMP
- IGMP
- RSVP
- RTP

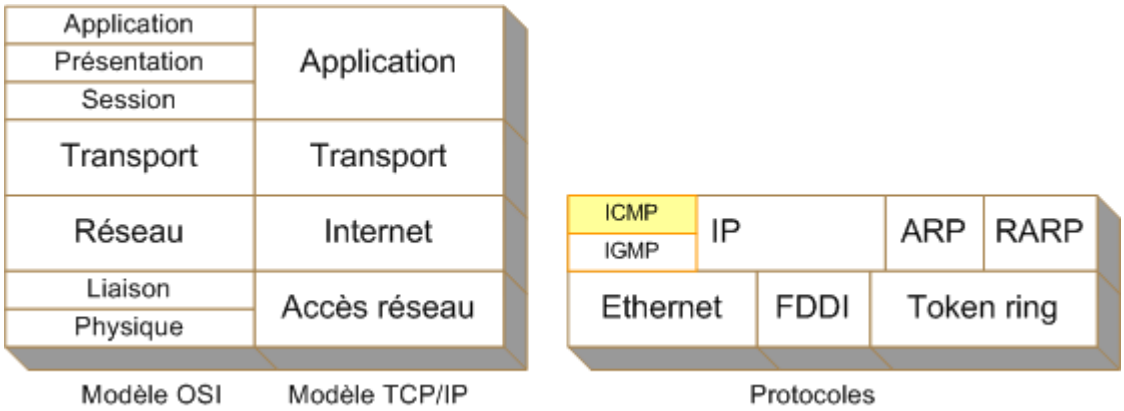
a. Le protocole ICMP (Internet Control Message Protocol)

Qui n'a pas entendu parler du fameux « ping » qui permet de savoir si un hôte distant est présent sur le réseau.

La commande « ping », basée sur le protocole ICMP, envoie un message particulier vers une adresse IP distante. Ce message véhiculé par l'intermédiaire d'un paquet IP, peut être reçu par tout équipement doté d'une interface réseau (switch manageable, ordinateur, imprimante, ….).

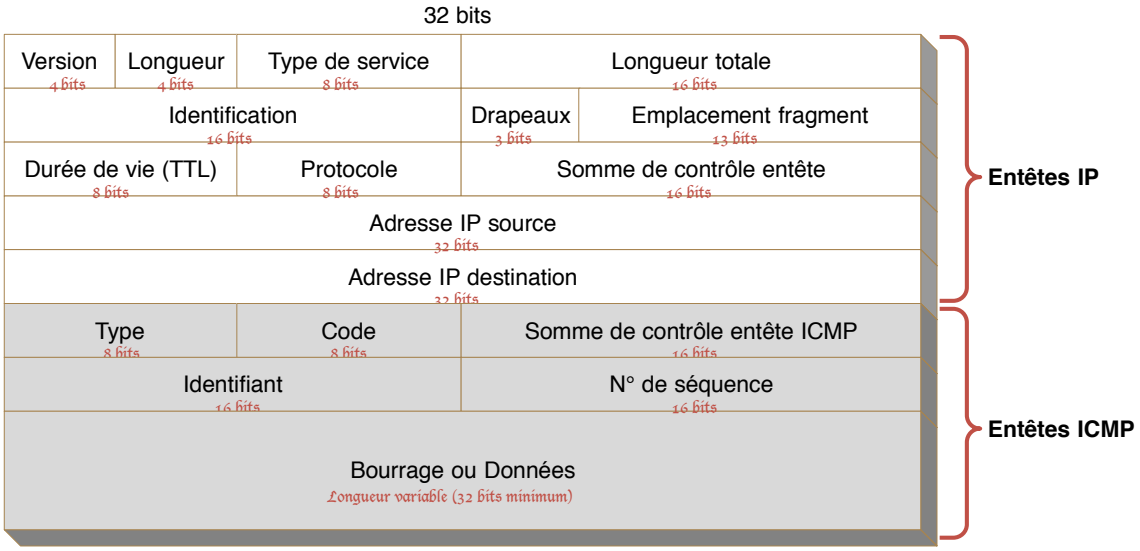
Ce protocole permet aux différents éléments réseau d'obtenir des informations sur l'état du réseau, d'un périphérique ou encore sur l'état d'une requête envoyée sur le réseau.

Bien qu'il soit « encapsulé » dans un datagramme IP, ICMP fait partie de la couche réseau.



Correspondance OSI – TCP/IP - Protocoles

Voici le contenu d'une trame ICMP qui indique les différents champs permettant d'accéder aux fonctionnalités de ce protocole :



Datagramme IP + ICMP

Signification des champs :

- Voir (*Rubrique D.6 Le protocole IP* pour les champs IP en blanc)
- **Type** : Correspond à une famille d'incidents (voir tableau plus loin)
- **Code** : Correspond à un incident précis parmi une famille
- **Contrôle d'erreur** : Permet de vérifier l'intégrité du paquet ICMP
- **Identifiant** : Permet de connaître l'identifiant d'un émetteur
- **N° de séquence** : Lorsqu'un qu'un datagramme icmp est fragmenté, le n° de séquence permet de savoir si tous les paquets ont bien été reçus. Mais comme ce champ n'est pas normalisé, il n'existe aucune garantie que le récepteur va utiliser ce champ. L'identifiant et le n° de séquence sont applicables uniquement aux messages de type ECHO (ping). Ils servent à associer une requête avec sa réponse.

Voici un tableau de tous les types de messages ICMP :

Type		Code	Nom	Description
	0	0	Echo-reply	Réponse à une demande d'écho
	8	0	Echo request	Demande d'écho
	4	0	Source quench	Volume de données trop important. L'équipement ne parvient pas à traiter toutes les trames
Destination non valide	3	0	Net unreachable	Réseau inaccessible
		1	Host unreachable	Hôte inaccessible
		2	Protocole unreachable	Protocole inaccessible
		3	Port unreachable	Port inaccessible
		4		Fragmentation nécessaire mais le champ DF du datagramme IP est à 1
		5	Source Route Failed	Echec de routage par la source
		6	Destination Network Unknown	Réseau de destination inconnu
		7	Destination Host Unknown	Hôte de destination inconnue
		8	Source Host Isolated	Machine source isolée
		9	Communication with Destination *Network* is Administratively Prohibited	Réseau de destination interdit administrativement
		10	Communication with Destination *Host* is Administratively Prohibited	Hôte de destination interdite administrativement
		11	Destination Network Unreachable for Type of Service	Réseau inaccessible pour ce type de service
		12	Destination Host Unreachable for Type of Service	Hôte inaccessible pour ce type de service
		13	Communication Administratively Prohibited	Communication interdite par un filtre
		14	Host Precedence Violation	
		15	Precedence cutoff in effect	
Redirections	5	0	Redirect Datagram for the Network (or subnet)	Redirection pour un réseau
		1	Redirect Datagram for the Host	Redirection pour un hôte
		2	Redirect Datagram for the Type of Service and Network	Redirection pour un réseau et pour un service donné
		3	Redirect Datagram for the Type of Service and Host	Redirection pour un hôte et pour un service donné
	9	0		Avertissement routeur : annonces de route par les routeurs
	10	0		Sollicitation routeur : L'hôte veut obtenir l'adresse d'un routeur
Temps dépassé	11	0	Time To Live exceeded in Transit	Durée de vie expirée pendant le transit
		1	Fragment Reassembly Time Exceeded	Durée de réassemblage des fragments expirée
Pb de paramètres	12	0	Pointer indicates the error	En-tête IP invalide
		1	Missing a Required Option	Manque d'une option obligatoire

		2	Bad Length	Mauvaise longueur d'entête
Demande d'informations	15	0	Address Request	Demande d'adresse réseau
	16	0	Address Reply	Réponse d'adresse réseau
	17	0	Mask request	Demande de masque de sous réseau
	18	0	Mask reply	Réponse de masque de sous réseau

Messages et code ICMP avec leur signification

Le fameux « ping » utilise le type 8 , code 0 pour l'envoi et le type 0 ,code 0.

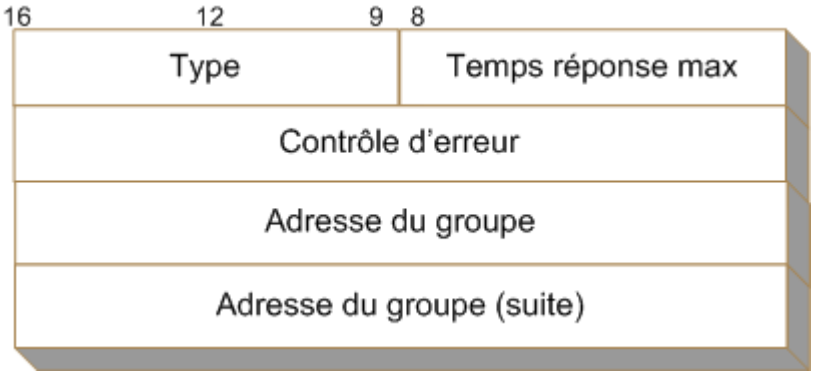
b. Le protocole IGMP (Internet Group Message Protocol)

Ce protocole permet l'utilisation des adresses de multidiffusion. Rappelons que la multidiffusion consiste à envoyer un datagramme à plusieurs destinations. En plus de l'adresse IP dite « unicast » , un hôte peut utiliser une adresse IP de classe D dite « multicast ».

IGMP utilise la notion de **groupe de diffusion**. Tout hôte peut décider de rejoindre ou quitter un groupe. La capacité de multidiffusion n'est pas limitée à un sous réseau physique et les routeurs ou passerelles peuvent propager les informations d'appartenance à un groupe.

Donc, les stations communiquent avec les passerelles par l'intermédiaire du protocole IGMP pour préciser leur appartenance au groupe.

Voici le format de la trame IGMP :



Champs de la trame IGMP

Les champs de la trame IGMP signifient :

- **Type (8bits)** : Type de message IGMP (Voir plus loin)
- **Temps de réponse max (8 bits)** : Ce champ n'est utilisé que pour les messages de type 11. Il indique le temps d'attente maximum pour un client avant l'émission du rapport d'appartenance. L'unité utilisée est le 1/10 de seconde. Pour les autres types, ce champ est marqué à 0
- **Contrôle d'erreur (16 bits)**

- **Adresse du groupe (32 bits)** : Adresse IP de classe D

Il existe plusieurs versions de IGMP : IGMPv1, IGMPv2

E. Le niveau Message (Transport)

L'envoi de données par paquets est assuré par la couche 3 (Réseau). La couche transport va permettre de fournir des services supplémentaires permettant d'utiliser **le mode connecté**. Ce mode assure aux deux extrémités d'une connexion des garanties sur l'envoi et la réception des messages. En effet, rien jusqu'à présent ne nous permettait de savoir si des informations envoyées avaient réellement été reçues.

La couche Transport et notamment **le protocole TCP (*Transport Control Protocol*)** vont largement contribuer à l'augmentation de fiabilité des communications. TCP est fortement orienté connexion.

Le protocole UDP (*User Datagram Protocol*) qui lui **n'est pas orienté connexion**, va permettre de faire circuler des messages rapidement sans garantie fiable au niveau de la réception de ces messages.

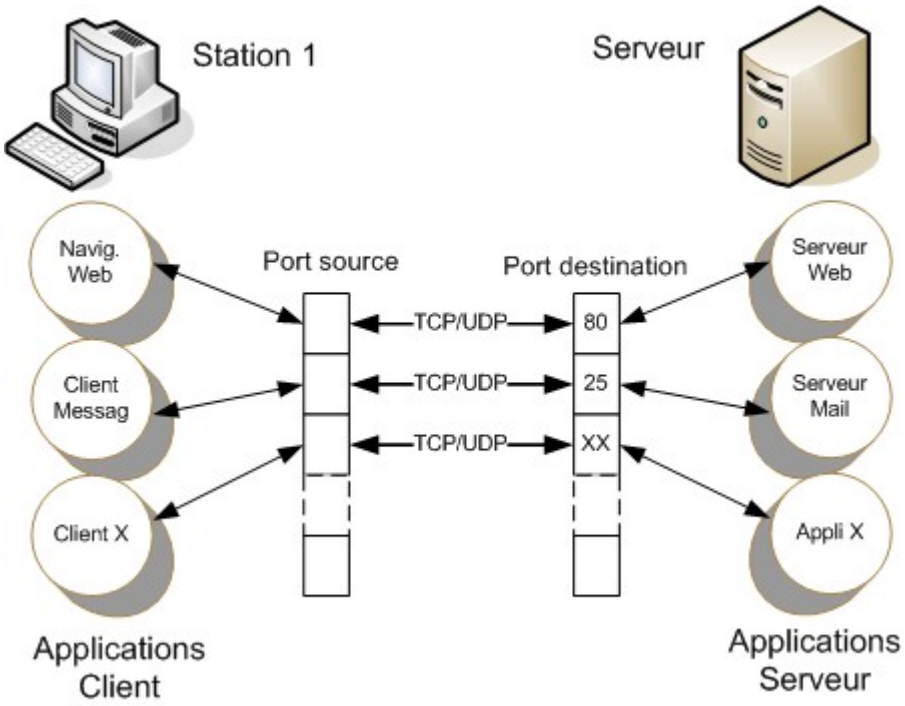
L'unité d'information de la couche transport est **le message**

E.1. La notion de port

TCP et UDP introduisent également **la notion de « port »**. Un port correspond à une application. En effet, le seul moyen dont dispose une station pour orienter les données vers la bonne application de destination est le port utilisé pour le transport. Cette orientation est effectuée au niveau du système d'exploitation dans les couches supérieures du modèle OSI (Session, présentation et application).

Dans la relation client-serveur, **le port source d'un client est choisi par le système d'exploitation** (Le premier disponible) alors que **le port de destination du serveur est un port identifié avec une application spécifique** (53 pour DNS , 80 pour http, 21 pour FTP, etc).

Voici comment un client et un serveur communiquent par le biais des ports source et destination :



Utilisation des ports dans la relation « client-serveur »

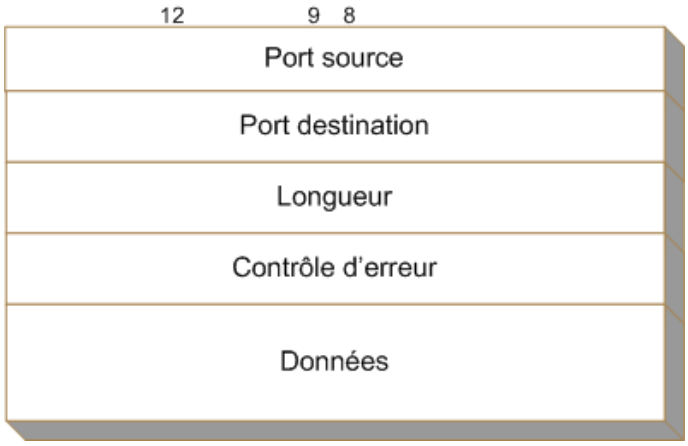
E.2. UDP (User Datagram Protocol)

Le protocole UDP est utilisé pour transmettre **une faible quantité de données** dans des situations où le coût de la création de connexions et du maintient de transmissions fiables s'avèrent supérieur aux données à émettre.

UDP peut également être utilisé pour les applications satisfaisant à **un modèle de type "interrogation réponse"**. La réponse étant utilisée comme **un accusé de réception à l'interrogation**. Citons par exemple les protocoles *DNS, Netbios* ou *SNMP* qui illustrent bien cet état de fait.

UDP peut aussi être utilisé dans **des applications « temps réel »** où la retransmission d'une trame perdue est parfois inutile (Vidéo par exemple)

Voici le détail d'un message UDP :



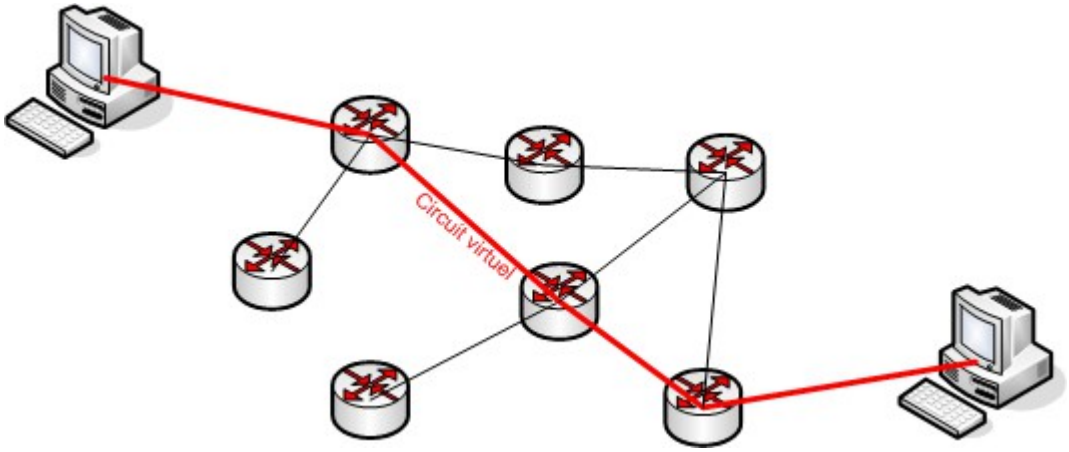
Champs d'un message UDP

Voici la signification des champs :

- **Port source (16 bits)** : Correspond au port utilisé par l'application de la station émettrice (source)
- **Port destination (16 bits)** : Correspond au port utilisé par l'application de la station réceptrice (destination). La liste officielle des ports et leur correspondance applicative sont donnés à l'adresse suivante http://www.frameip.com/liste-des-ports-tcp-udp/
- **Longueur (16 bits)** : Le champ Longueur est codé sur 16 bits et il représente la taille de l'entête et des données. Son unité est l'octet et sa valeur maximale est 64 Koctets (2^{16}).
- **Le contrôle d'erreur (16 bits)** : Codé sur 16 bits, il représente la validité du paquet de la couche 4 UDP. Il est calculé sur tous les octets du message UDP + les 12 octets précédents (IP)
- **Données (taille variable)**

E.3. TCP (Transport Control Protocol)

TCP est un service de transport fiable. Il offre **un mode connecté**. Aucune transmission de données n'est possible tant que les deux parties n'ont pas établies une connexion. On parle alors de circuit virtuel :



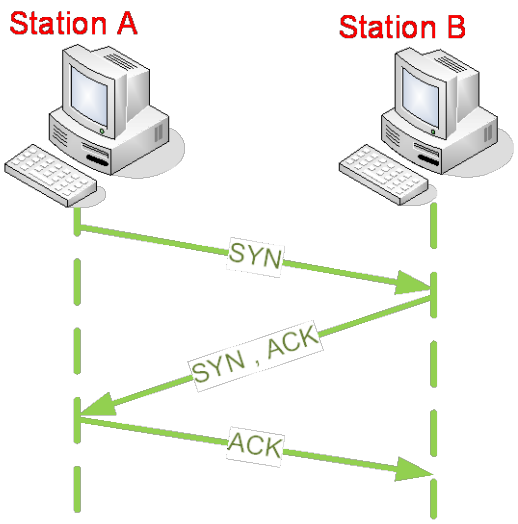
Un circuit virtuel grâce à TCP

a. Messages et segments

Un message est l'élément que doit transporter TCP. Pour des raisons de performances, TCP peut être amené à **décomposer le message en segments**. Chaque en-tête de segment contient **un numéro d'ordre**. Ce numéro d'ordre permet aux fonctions de la couche transport, au niveau de l'hôte de destination, de réassembler les segments dans l'ordre de leur envoi. L'application de destination peut ainsi disposer des données sous la forme exacte voulue par l'expéditeur.

b. Etablissement et fermeture d'une connexion

Avant tout transfert de données, une connexion est établie selon le principe de la « **poignée de main à 3 temps** ». Chaque segment est émis avec un drapeau précisant sa signification



1ère étape

La station 1 émet un segment avec le drapeau SYN (Demande de connexion).

2ème étape

La station 2 répond en émettant un segment avec le drapeau SYN/ACK (SYN : La station B envoie elle aussi une demande d'ouverture de connexion. ACK : Accusé de réception de la demande de connexion précédente).

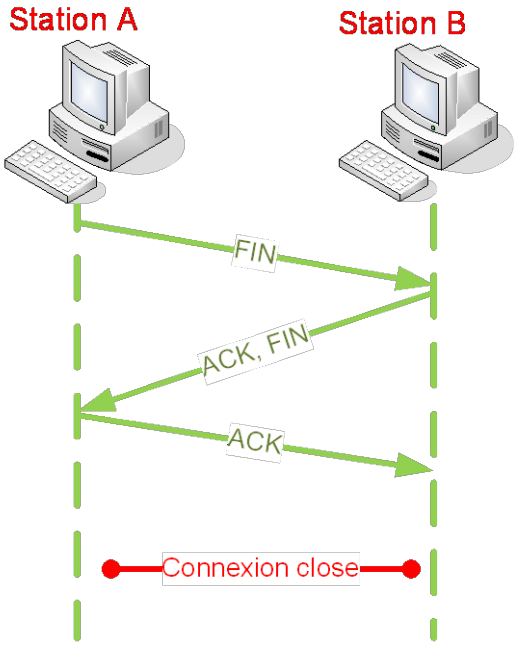
3ème étape

La station 1 répond par un segment avec le drapeau ACK précisant qu'elle a reçue la confirmation de la demande de connexion

Voici un exemple de 3 segments TCP capturés. On distingue bien l'état des drapeaux TCP :

3	0.044811	192.168.16.5	209.85.227.104	TCP	netobjects1 > http [SYN] Seq=0 Win=65535 Len=0 MS
4	0.093567	209.85.227.104	192.168.16.5	TCP	http > netobjects1 [SYN, ACK] Seq=0 Ack=1 Win=572
5	0.093605	192.168.16.5	209.85.227.104	TCP	netobjects1 > http [ACK] Seq=1 Ack=1 Win=65535 Le

Pour la fermeture de la connexion, le drapeau FIN est utilisé :



1ère étape
La station qui souhaite mettre fin à la connexion envoie le drapeau FIN.

2ème étape
La station qui reçoit le drapeau FIN acquitte avec le drapeau ACK puis envoie aussi le drapeau FIN. Selon les cas, l'envoi de ACK et FIN se fait dans 2 segments différents

3ème étape
La station A acquitte la demande de FIN

c. Le numéro de séquence

Il faut préciser que lors de l'échange en 3 étapes, les 2 hôtes transmettent un **numéro de séquence initial** (ISN : Initial Sequence Number). Ce numéro de séquence est incrémenté tout au long de la communication afin de situer le segment dans la séquence transmise. Lorsque n segments sont transmis, le numéro de séquence du prochain segment sera égal à n+1 :

L'ISN est un nombre pseudo-aléatoire généré par l'émetteur. Ce nombre est codé sur 32 bits. Voici une capture de trame correspondant à l'initialisation du numéro de séquence :

No.	Time	Source	Destination	Protocol	Info
803	2.309703	192.168.16.5	128.107.229.50	TCP	csdmbase > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460
855	2.516646	128.107.229.50	192.168.16.5	TCP	http > csdmbase [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
856	2.516663	192.168.16.5	128.107.229.50	TCP	csdmbase > http [ACK] Seq=1 Ack=1 Win=65535 Len=0

```
Transmission Control Protocol, Src Port: csdmbase (1471), Dst Port: http (80), Seq: 0, Len: 0
    Source port: csdmbase (1471)
    Destination port: http (80)
    Sequence number: 0    (relative sequence number)
    Header length: 28 bytes
    Flags: 0x02 (SYN)

0010  00 30 85 18 40 00 80 06  3f 64 c0 a8 10 05 80 0b   .0..@... ?d......
0020  e5 32 05 bf 00 50 d0 03  ef a3 00 00 00 00 70 02   .2...P.. ......p.
0030  ff ff 87 1d 00 00 02 04  05 b4 01 01 04 02         ........ ......
```

Le logiciel de capture mentionne un numéro de séquence relatif à l'ISN
0 signifie en réalité ISN

ISN choisi est :
D003EFA3 en hexa
3489918883 en décimal

> *Dans la suite du cours, il sera plus facile de travailler avec un numéro de séquence relatif { l'ISN. Par exemple, pour un numéro de séquence relatif de 152, il s'agira dans la réalité de ISN + 152*

Voici un exemple de 2 segments envoyés successivement (l'un après l'autre) :

1er segment (N° séquence relatif = 1 , 403 octets transmis) :

```
⊟ Transmission Control Protocol, Src Port: http (80), Dst Port: gwen-sonya (2778), Seq: 1, Ack: 714, Len: 403
    Source port: http (80)
    Destination port: gwen-sonya (2778)
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 404    (relative sequence number)]
    Acknowledgement number: 714    (relative ack number)
    Header length: 20 bytes
  ⊞ Flags: 0x18 (PSH, ACK)
    Window size: 7130
  ⊞ Checksum: 0xebf2 [correct]
    TCP segment data (403 bytes)
```

2ème segment (N° séquence relatif = 404) :

```
⊟ Transmission Control Protocol, Src Port: http (80), Dst Port: gwen-sonya (2778), Seq: 404, Ack: 714, Len: 319
    Source port: http (80)
    Destination port: gwen-sonya (2778)
    Sequence number: 404    (relative sequence number)
    [Next sequence number: 723    (relative sequence nu
    Acknowledgement number: 714    (relative ack number
    Header length: 20 bytes
  ⊞ Flags: 0x18 (PSH, ACK)
    Window size: 7130
```

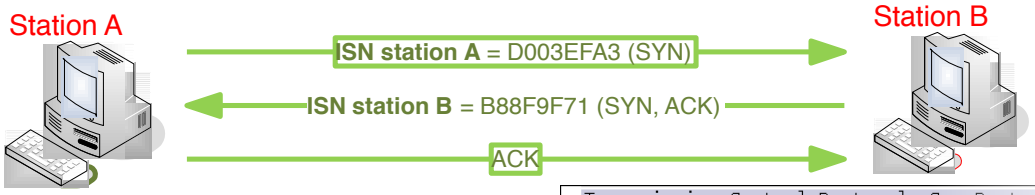
N° Seq = 1 + 403 octets transmis, donc
Le prochain N° Seq = 404

d. Deux flux unidirectionnels

Une communication entre 2 hôtes est en réalité un ensemble de **2 communications unidirectionnelles**. En effet, les 2 hôtes sont tour à tour, émetteur et récepteur.

Il y aura donc 2 numéros de séquence (Un pour chaque sens de communication). Il en sera de même pour les accusés de réception. Voici un exemple d'ouverture de connexion puis d'initialisation des ISN

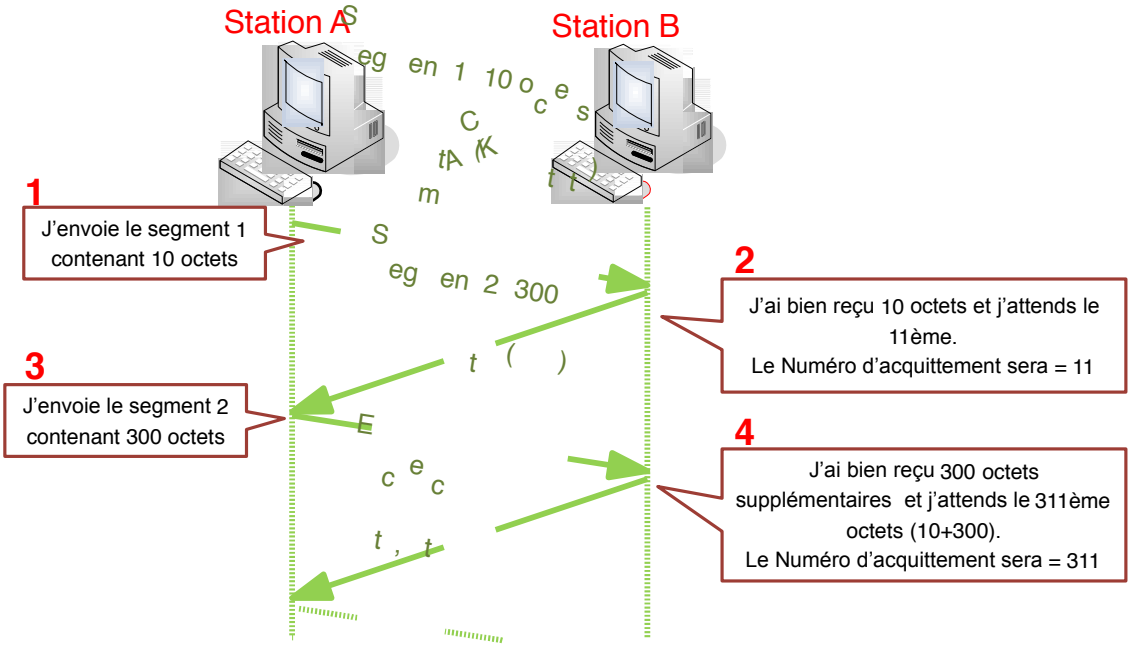
```
⊟ Transmission Control Protocol, Src Port: csdmbase (147
    Source port: csdmbase (1471)
    Destination port: http (80)
    Sequence number: 0    (relative sequence number)
    Header length: 28 bytes
0000  00 07 cb 3a 03 5b 00 1d  09 ce 09 18 08 00 45 00
0010  00 30 85 18 40 00 80 06  3f 64 c0 a8 10 05 80 6b
0020  e5 32 05 bf 00 50 d0 03  ef a3 00 00 00 00 70 02
0030  ff ff 87 1d 00 00 02 04  05 b4 01 01 04 02
```




```
⊟ Transmission Control Protocol, Src Port: http (80),
    Source port: http (80)
    Destination port: csdmbase (1471)
    Sequence number: 0    (relative sequence number)
    Acknowledgement number: 1    (relative ack number)
0000  00 1d 09 ce 09 18 00 07  cb 3a 03 5b 08 00 45 00
0010  00 30 00 00 40 00 32 06  12 7d 80 6b e5 32 c0 a8
0020  10 05 00 50 05 bf b8 8f  9f 71 d0 03 ef a4 70 12
0030  16 d0 18 3b 00 00 02 04  05 b4 01 01 04 02
```

e. Accusé de réception

En mode connecté les deux stations vont adopter **le principe de l'accusé de réception**. Les segments envoyés par l'émetteur feront l'objet d'un accusé de réception de la part du récepteur. Cet accusé de réception se matérialise sous la forme d'un segment dont **le drapeau ACK est levé** ainsi qu'un numéro de séquence indiquant le **prochain octet attendu**.



Principe d'acquittement des données

Exemple d'accusé de réception :
128.107.229.50 envoie un segment de données à 192.168.16.5

```
26731 38.198438   192.168.16.5    128.107.229.50  TCP   csdmbase > http [ACK] Seq=2707 Ack=9106539 Win=65535 Len=0
26744 38.207487   128.107.229.50  192.168.16.5    TCP   [TCP segment of a reassembled PDU]

Transmission Control Protocol, Src Port: http (80), Dst Port: csdmbase (1471), Seq: 1979, Ack: 2707, Len: 1460
  Source port: http (80)
  Destination port: csdmbase (1471)
  Sequence number: 1979    (relative sequence number)
  [Next sequence number: 3439    (relative sequence number)]
  Acknowledgement number: 2707    (relative ack number)
  Header length: 20 bytes
```

En retour, 192.168.16.5 envoie un accusé de réception à 128.107.229.50

```
2410 6.437092   128.107.229.50  192.168.16.5    TCP   [TCP segment of a reassembled PDU]
2411 6.437227   192.168.16.5    128.107.229.50  TCP   csdmbase > http [ACK] Seq=2707 Ack=3439 Win=65535 Len=0
2412 6.438523   128.107.229.50  192.168.16.5    TCP   [TCP segment of a reassembled PDU]

Transmission Control Protocol, Src Port: csdmbase (1471), Dst Port: http (80), Seq: 2707, Ack: 3439, Len: 0
  Source port: csdmbase (1471)
  Destination port: http (80)
  Sequence number: 2707    (relative sequence number)
  Acknowledgement number: 3439    (relative ack number)
  Header length: 20 bytes
  Flags: 0x10 (ACK)
  Window size: 65535
```

f. Contrôle de flux

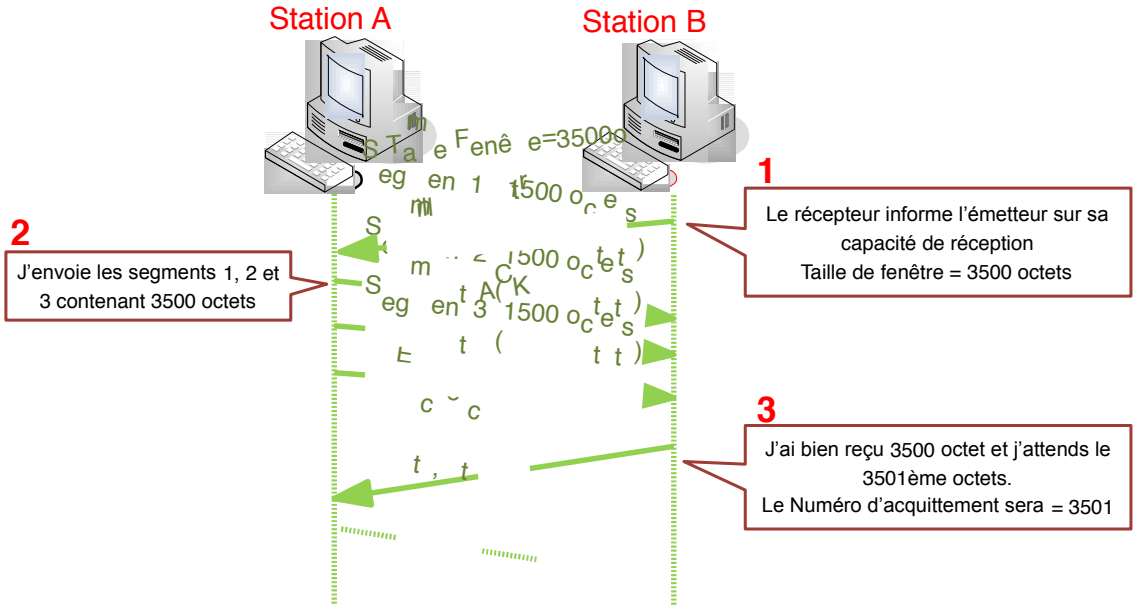
Dans une communication TCP, le nombre d'accusés de réception peut devenir très important. En effet, pour chaque segment transmis, il faudrait un accusé de réception.

Il faut un mécanisme permettant à la fois de limiter le nombre d'accusés de réception, tout en contrôlant le flux de données. Si le récepteur des données n'envoie pas les accusés de réception, l'émetteur doit comprendre qu'il faut retransmettre les segments perdus mais aussi ralentir le débit des données.

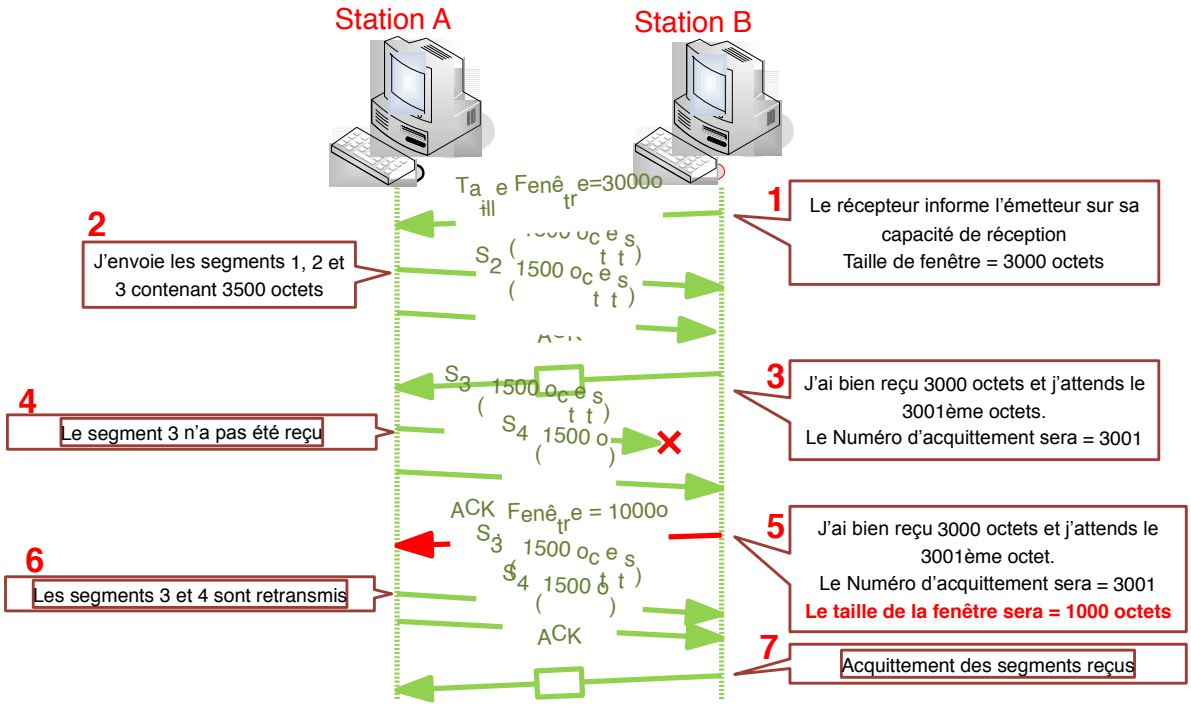
Ce mécanisme, appelé « Fenêtre glissante » (Window TCP), a pour principe de fixer un nombre maximum d'octets à envoyer (Taille de la fenêtre) sans qu'il soit nécessaire d'émettre un accusé

Une station pourra donc émettre plusieurs segments sans attendre un accusé de réception. C'est le récepteur qui informe l'émetteur de sa capacité de réception.

Exemple d'une communication sans perte de segments :



Maintenant supposons qu'un segment n'est pas reçu à cause d'un problème de transmission :



On remarque alors que le récepteur ajuste la taille de la fenêtre après avoir constaté un problème de réception.

g. Acquittement sélectif (SACK)

En cas de perte de segment (Comme dans le cas évoqué précédemment), on s'aperçoit que le segment S3 est transmis 2 fois alors qu'il a été reçu correctement la 1ère fois. Cette situation génère un trafic supplémentaire dans un contexte déjà perturbé. Dans ces conditions le remède peut être pire que le mal surtout si la taille de la fenêtre est grande.

La plupart des implémentations TCP actuelles utilisent la technique de « **l'acquittement sélectif** ». Cette technique autorise l'envoi d'acquittements pour des segments non contigus, ce qui permet à l'émetteur de retransmettre uniquement les segments perdus et pas ceux qui ont été correctement transmis.

Pour plus d'information, voir la RFC 2018 (http://www.faqs.org/rfcs/rfc2018.html)

h. Les drapeaux (Flags)

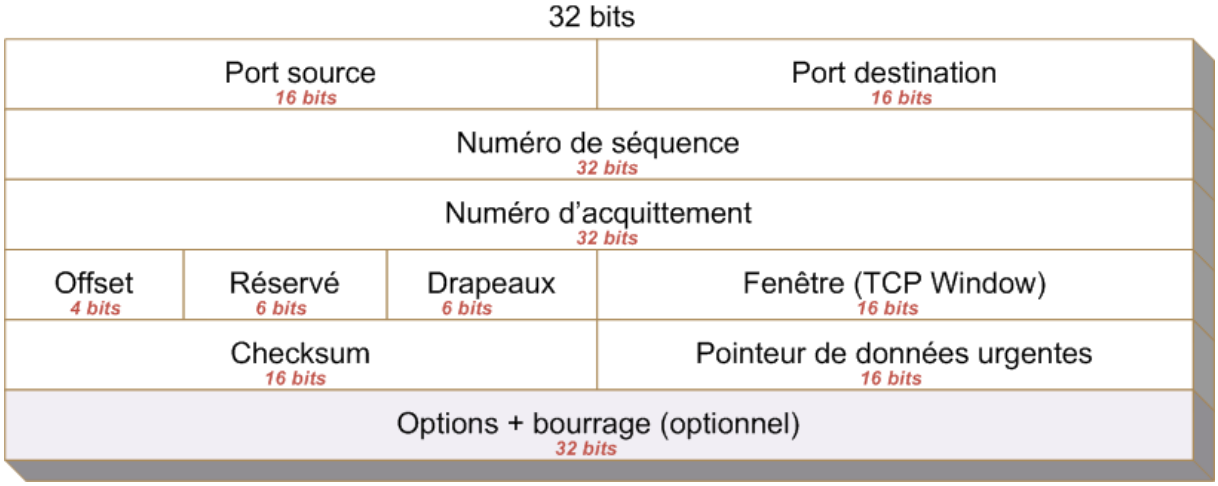
Il s'agit d'information de type binaire (un bit à 0 ou 1) permettant d'exprimer un certain état pour un segment. Il existe plusieurs drapeaux dans TCP. En voici la signification :

- **URG** : Indique que le champ « Pointeur de donnée urgente » est utilisé.
- **ACK** : Indique que le numéro de séquence pour les accusés de réceptions est valide.
- **PSH** : Indique au récepteur de délivrer les données à l'application et de ne pas attendre le remplissage des tampons.
- **RST** : Demande la réinitialisation de la connexion.
- **SYN** : Indique la synchronisation des numéros de séquence.
- **FIN** : Indique fin de transmission.

Nous allons maintenant voir dans quels cas ces drapeaux sont utilisés.

i. Entêtes du protocole TCP

Voici un récapitulatif des différentes entêtes incluses dans un segment TCP.



Entêtes du segment TCP

Le tableau suivant précise la signification de ces entêtes :

Nom du champ	Taille	Description
Port source	16 bits	Port utilisé par l'expéditeur du segment
Port destination	16 bits	Port du destinataire
N° Séquence	32 bits	Numéro de séquence du 1er octet de ce segment. Chaque octet est numéroté à partir d'un numéro initial (souvent 1). Il est donc possible de connaître la position d'un octet par rapport à une séquence complète
N° Accusé	32 bits	Numéro du prochain octet à recevoir. L'émetteur comprend que tous les octets précédant ce numéro ont bien été reçus
Offset	4 bits	Nombre de mots de 32 bits de l'entête (segment sans les données). Indique où commencent les données utiles.
Réservé	6 bits	Inutilisé la plupart du temps
Drapeaux	6 bits	Voir
Fenêtre	16 bits	Nombre d'octets à partir de la position marquée dans l'accusé de réception que le récepteur est capable de recevoir sans accusé
Checksum	16 bits	Code d'erreur calculée sur l'ensemble du segment + 12 octets précédents (pseudo-entête)
Pointeur de données urgentes	16 bits	Communique la position d'une donnée urgente en donnant son décalage par rapport au numéro de séquence. Le pointeur doit pointer sur l'octet suivant la donnée urgente. Ce champ n'est interprété que lorsque le Flag URG est marqué à 1. Dès que cet octet est reçu, la pile TCP doit envoyer les données à l'application sans attendre.
Options	32 bits avec bourrage	Permettent d'ajouter des fonctionnalités non prises en charge dans la version native du protocole. Par exemple, l'option SACK (Selective Acknowledgment option) qui permet de gérer de manière plus rigoureuse les accusés de réception. La liste des options se trouve à l'adresse : http://www.iana.org/assignments/tcp-parameters
bourrage		Bits dont la taille est variable. Le bourrage sert à compléter les bits des options pour faire un multiple de 32 bits

Signification des champs du segment TCP