# Les Réseaux

Principes fondamentaux

Sommaire

1 Protocole de communication ..... 1
1.1 Rappel sur l'adresse IP ..... 2
1.2 Rappel sur la masque de sous réseau ..... 3
1.3 Le routage ..... 3
1.4 Les Ports ..... 3
1.5 Notion d'adresse physique : adresse MAC ..... 5
1.6 Le protocole UDP/IP ..... 6
1.7 Le protocole TCP/IP ..... 7
2 Format des trames et organisation en couche ..... 9
2.1 Trame Ethernet II ..... 10
2.2 Datagramme IP ..... 11
2.3 Segment ICMP, commande « ping » ..... 12
2.4 Datagramme Arp ..... 13
3 Dialogue entre 2 équipements ..... 14

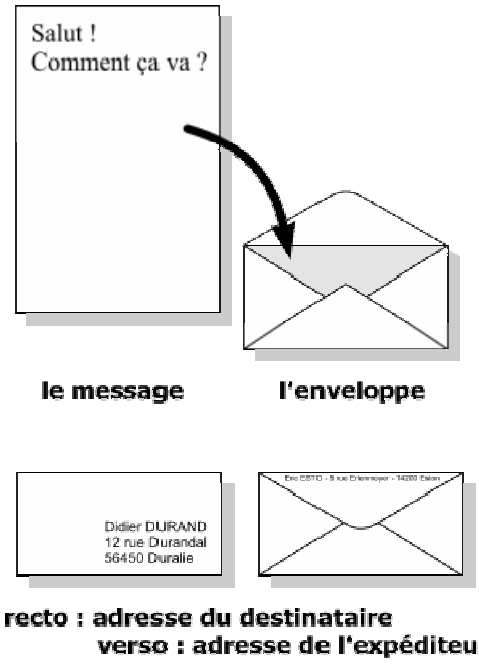
## 1 Protocole de communication

Un protocole est une méthode standard qui permet la communication entre des processus (s'exécutant éventuellement sur différentes machines), c'est-à-dire un ensemble de règles et de procédures à respecter pour émettre et recevoir des données sur un réseau. Il en existe plusieurs selon ce que l'on attend de la communication. Certains protocoles seront par exemple spécialisés dans l'échange de fichiers (le FTP), d'autres pourront servir à gérer simplement l'état de la transmission et des erreurs (c'est le cas du protocole ICMP), ...
Si vous vous baladez sur Internet, vous avez dû, à un moment ou à un autre, entendre parler de TCP/IP : Que signifie t-il et comment cela fonctionne ?

TCP/IP est un nom générique qui regroupe en fait un ensemble de **protocoles**, c'est à dire des **règles de communication**.

Principe d'une transmission sur le réseau

→ Analogie avec la transmission d'une lettre par la poste :



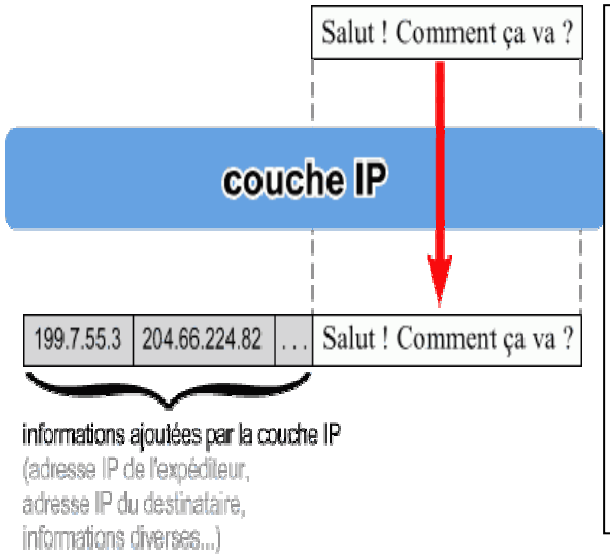

Quand vous voulez envoyer une lettre par la poste:

- Vous placez votre lettre dans une enveloppe,

- sur le recto vous inscrivez l'adresse du destinataire,

- au dos, vous écrivez l'adresse de l'expéditeur (la vôtre).

Ce sont des règles utilisées par tout le monde. C'est un **protocole**.




Sur Internet, c'est à peu près la même chose: chaque message (chaque petit paquet de données) est enveloppé par IP qui y ajoute différentes informations:

- l'adresse de l'expéditeur (votre adresse IP),
- l'adresse IP du destinataire,
- différentes données supplémentaires (qui permettent de bien contrôler l'acheminement du message).

## 1.1 Rappel sur l'adresse IP

**L'adresse IP** est une adresse unique attribuée à chaque ordinateur sur Internet (c'est-à-dire qu'il n'existe pas sur Internet deux ordinateurs ayant la même adresse IP).
De même, l'adresse postale (nom, prénom, rue, numéro, code postal et ville) permet d'identifier de manière unique un destinataire.

Tout comme avec l'adresse postale, il faut connaître au préalable l'adresse IP de l'ordinateur avec lequel vous voulez communiquer.

Une **adresse IP** est une adresse 32 bits, généralement notée sous forme de 4 nombres entiers séparés par des points. Par exemple: 204.35.129.3. On distingue en fait deux parties dans l'adresse IP :

- une partie des nombres à gauche désigne le réseau est appelé **ID de réseau** (en anglais *netID*),

- Les nombres de droite désignent les ordinateurs de ce réseau est appelé **ID d'hôte** (en anglais *host-ID*).

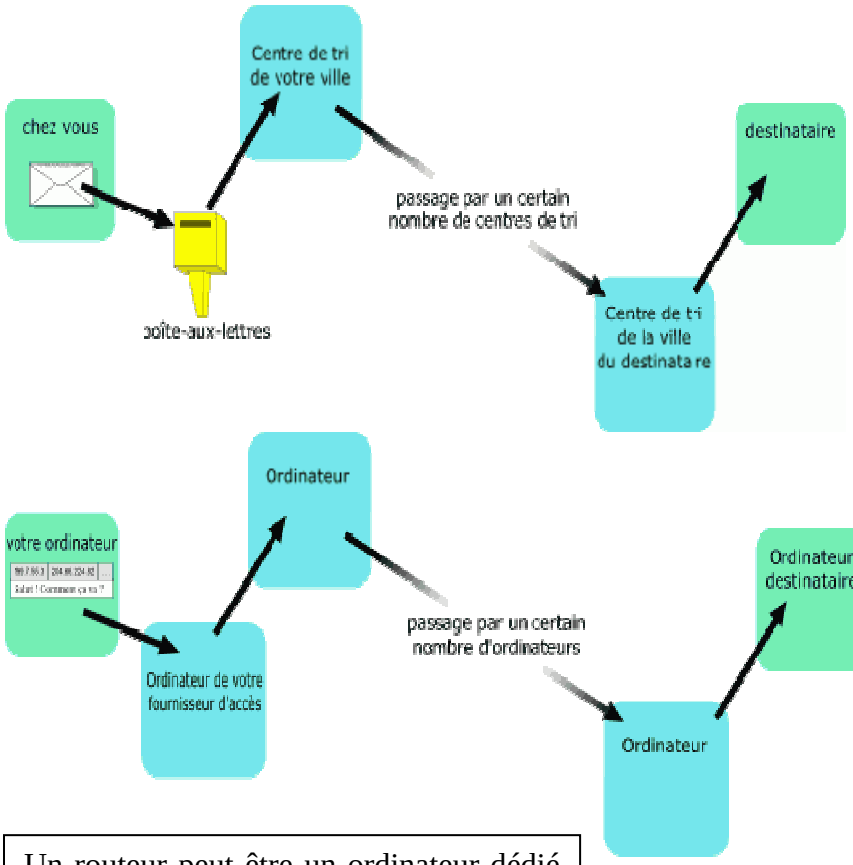
Une adresse IP est fixée soit manuellement par l'utilisateur soit attribuée par un serveur DHCP.

## 1.2 Rappel sur la masque de sous réseau

Pour identifier le réseau auquel appartient un ordinateur, on utilise un « masque ».
Par exemple, si un ordinateur a la config suivante :
IP : 134. 3 . 9 .7
Masque : 255.255. 0 .0
En faisant un « ET » logique entre l'IP et le masque, on obtient 134 . 3 . 0 .0

Lorsqu'on configure un réseau, on parle souvent de masque de sous réseau qui permet à un ensemble d'ordinateurs de communiquer : ils se « verront » ou non.

## 1.3 Le routage



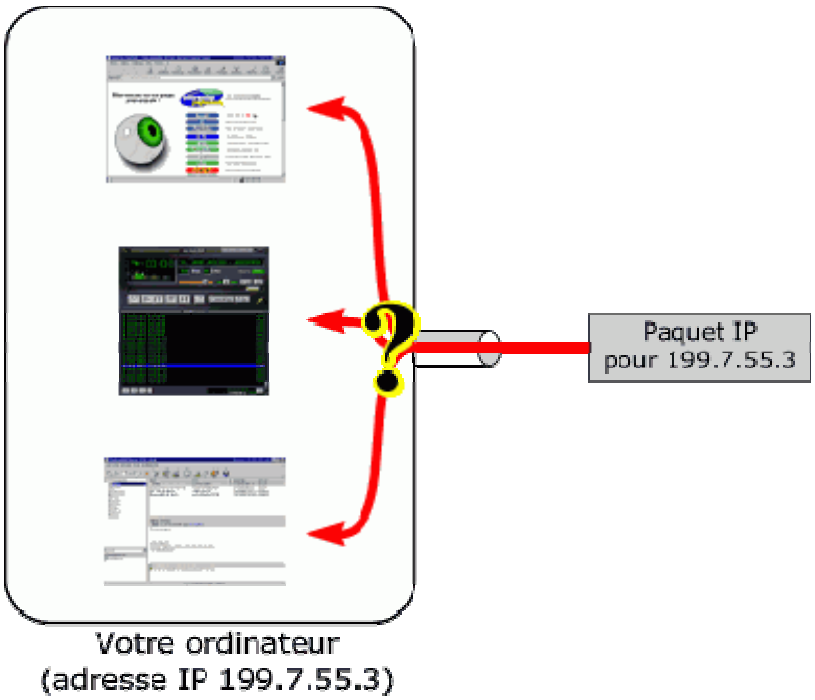

Pour envoyer votre lettre, vous la postez dans la boîte aux lettres la plus proche. Ce courrier est relevé, envoyé au centre de tri de votre ville, puis transmis à d'autres centres de tri jusqu'à atteindre le destinataire

C'est un peu la même chose sur Internet !

Vous déposez le paquet IP sur l'ordinateur le plus proche (celui de votre fournisseur d'accès en général). Le paquet IP va transiter d'ordinateur en ordinateur jusqu'à atteindre le destinataire.

Un routeur peut être un ordinateur dédié ou un boîtier chargé de transmettre tous les paquets d'un réseau vers un autre réseau ou internet.

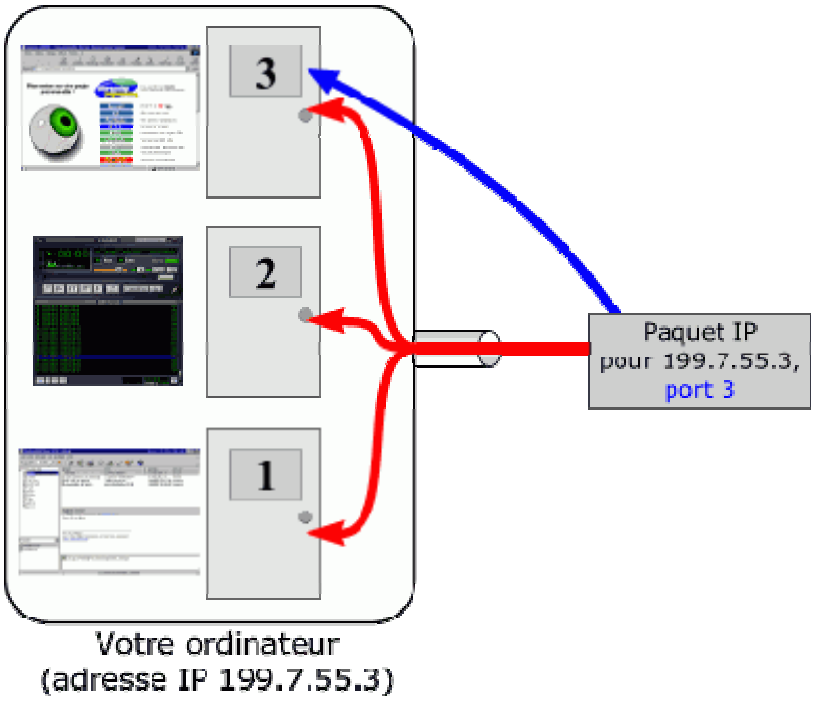
Avec IP, nous avons de quoi envoyer et recevoir des paquets de données d'un ordinateur à l'autre.




Imaginons maintenant que nous ayons plusieurs programmes qui fonctionnent en même temps sur le même ordinateur: un navigateur, un logiciel d'email et un logiciel pour écouter la radio sur Internet.

Si l'ordinateur reçoit un paquet IP, comment savoir à quel logiciel donner ce paquet IP ?

**Comment savoir à quel logiciel est destiné ce paquet IP ? Le navigateur, le logiciel de radio ou le logiciel d'email ?**




Pour cela, on attribut numéro unique à chaque logiciel dans l'ordinateur appelé **numéro de PORT**

Ainsi, l'adresse IP permet de s'adresser à un ordinateur donné, et le numéro de port permet de s'adresser à un logiciel particulier sur cet ordinateur.

## 1.5 Notion d'adresse physique : adresse MAC

Deux cartes réseaux qui communiquent s'échangent des messages (suite d'octets) appelés **trames** (« frame » en anglais).

L'adresse MAC est constituée de 6 octets. Elle est toujours affiché sur la base hexadécimale (contrairement aux quatre octets de l'adresse IP qui sont affichées en décimal). Il existe une adresse particulière, dite adresse de diffusion, qui permet d'interroger toutes les interfaces connectées sur le réseau.

Tous les postes connectés au même câble reçoivent le message, mais **seul** celui à qui il est destiné le lit.

*Comment sait-il que cette trame lui est adressée ?*

Il reconnaît l'adresse de destination, contenue dans la trame comme étant la sienne.

*Comment sait-il qui lui a envoyé la trame ?*

La trame contient aussi l'adresse de l'émetteur.

L'adresse correspond à l'adresse de la carte réseau. On parle d'adresse physique, d'adresse MAC (Media Access Control)

L'adresse d'une carte réseau correspond à l'adresse d'un poste et d'un seul. Or les postes sont généralement regroupés en réseau. Comment identifier le réseau auquel appartient le poste ?

Il faut une adresse logique qui soit indépendante de l'adresse physique. C'est ce que propose le protocole IP.

***Pourquoi identifier le réseau ?***

Pour permettre à 2 postes qui ne sont pas connectés au même réseau de communiquer.

Cela est impossible avec une adresse MAC, il faut une adresse de niveau supérieur, comme nous le verrons un peu plus loin et surtout avec le **routage IP.**

Le message véhiculé par la trame va contenir une autre adresse destinataire dont un des objectifs sera de définir le réseau destinataire du message. On appelle le message contenu dans une trame un **paquet.**

Ce qu'il nous faut savoir à ce stade, c'est qu'une machine sait que le paquet n'est pas destiné au réseau si l'adresse réseau de destination est différente de la sienne, dans ce cas elle envoie le paquet à une machine spéciale dont le rôle est d'acheminer les paquets qui sortent du réseau.

Cette machine s'appelle une **passerelle** (gateway) dans la **terminologie IP** ou un **routeur**

Pour pouvoir être correctement transmis, le paquet va être mis dans une trame avec une adresse MAC de destination et une adresse MAC d'émission.

On dit qu'un paquet IP est **encapsulé** dans une trame.

***Pourquoi s'encombrer de l'adresse MAC ?***

Le décodage d'une information est 1000 fois plus rapide avec une adresse matérielle qu'une adresse logicielle. La carte réseau ou le routeur décode donc très rapidement une trame avec une adresse MAC et peut savoir très rapidement si le paquet lui est adressé ou non (gain de temps énorme dans le traitement).

Mais pour que tout cela fonctionne, il faut un mécanisme qui permettra de passer d'une adresse logique à une adresse physique, et réciproquement.

**Résolution d'adresses logiques en adresses physiques**

Toute machine sur un réseau IP et Ethernet a donc 2 adresses, une adresse MAC et une adresse IP.

Les processus de niveaux supérieurs utilisent toujours l'adresse IP et donc lorsqu'un processus communique avec un autre processus, il lui envoie un message dont l'adresse destinataire est une adresse IP, mais pour pouvoir atteindre la carte réseau du destinataire, il faut connaître son adresse MAC. Comment faire ?

C'est le rôle du protocole **ARP** (Adress Resolution Protocol)

## 1.6 Le protocole UDP/IP

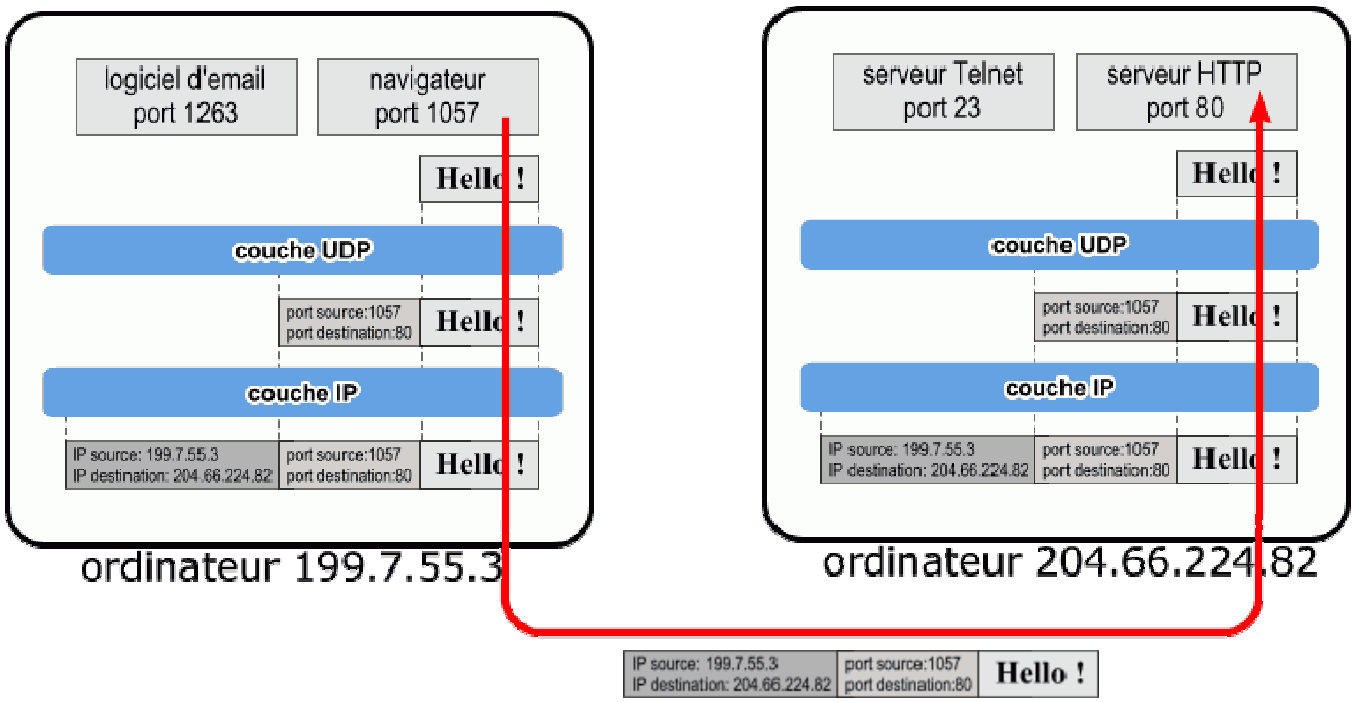
**UDP/IP** est un protocole qui permet justement d'utiliser des numéros de **ports** en plus des **adresses IP** (On l'appelle UDP/IP car il fonctionne au dessus d'IP).

IP s'occupe des adresses IP et UDP s'occupe des ports.

Avec le protocole **IP** on pouvait envoyer des données d'un ordinateur A à un ordinateur B.


Avec **UDP/IP**, on peut être plus précis: on envoie des données d'une **application x** sur l'**ordinateur A** vers une **application y** sur l'**ordinateur B**.

Par exemple, votre navigateur peut envoyer un message à un serveur HTTP (un serveur Web):




- Chaque couche (UDP et IP) va ajouter ses informations.
  Les informations de **IP** vont permettre d'acheminer le paquet à destination du bon **ordinateur**. Une fois arrivé à l'ordinateur en question, la couche **UDP** va délivrer le paquet au bon **logiciel** (ici: au serveur HTTP).

- Les deux logiciels se contentent d'émettre et de recevoir des données ("**Hello !**"). Les couches UDP et IP en dessous s'occupent de tout.

Ce couple (199.7.55.3:1057, 204.66.224.82:80) est appelé un **socket**. Un socket identifie de façon unique une communication entre deux logiciels.

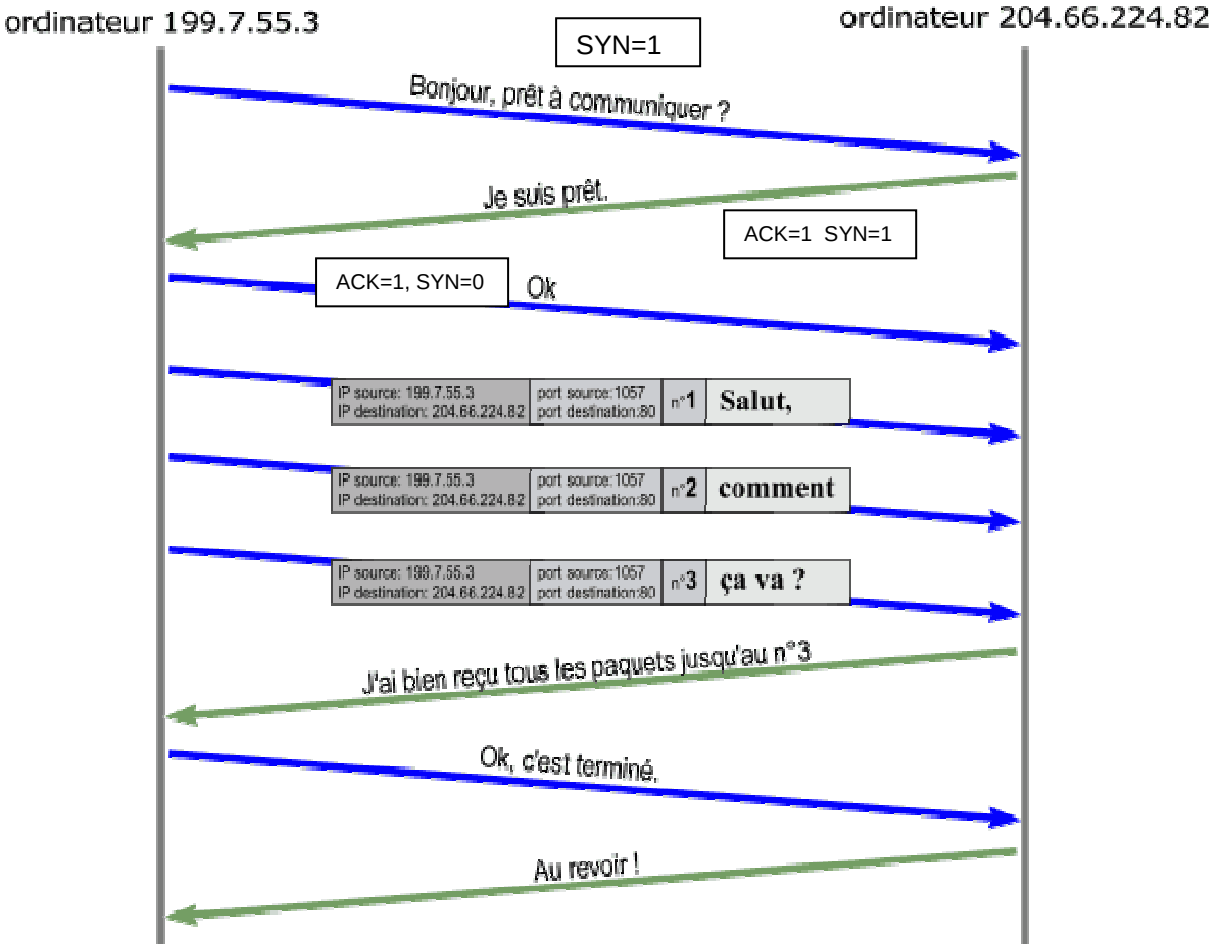
## 1.7 Le protocole TCP/IP

On peut donc maintenant faire communiquer 2 logiciels situés sur des ordinateurs différents.

Mais il y a encore des petits problèmes:

- Quand vous envoyez un paquet IP sur Internet, il passe par des dizaines d'ordinateurs (ou routeurs). Et il arrive que des paquets IP se **perdent** ou arrivent en **double exemplaires**.
  Ça peut être gênant : imaginez un ordre de débit sur votre compte bancaire arrivant deux fois ou un ordre de crédit perdu !
- Même si le paquet arrive à destination, rien ne vous permet de savoir si le paquet est bien arrivé (aucun accusé de réception).
- **La taille des paquets IP est limitée** (environ 1500 octets). Un paquet de 1500 octets est aussi appelé « datagramme ». Comment faire pour envoyer la photo JPEG qui fait 115000 octets ?

**Voici le protocole de communication initié par TCP :** Par exemple, pour envoyer le message "**Salut, comment ça va ?**", (Chaque flèche représente 1 paquet IP):




A l'arrivée, sur l'ordinateur 204.66.224.82, la couche TCP reconstitue le message "**Salut, comment ça va ?**" à partir des 3 paquets IP reçus et le donne au logiciel qui est sur le port 80.

C'est pour cela qu'a été conçu TCP.

TCP est capable:

- de faire tout ce que UDP sait faire (ports).
- de vérifier que le destinataire est prêt à recevoir les données.
- de **découper** les gros paquets de données en paquets plus petits pour que l'IP les accepte
- de **numéroter** les paquets, et à la réception de **vérifier** qu'ils sont tous bien arrivés, de **redemander** les paquets manquants et de les **réassembler** avant de les donner aux logiciels. Des accusés de réception sont envoyés pour prévenir l'expéditeur que les données sont bien arrivées.

**TCP/IP : Conclusion**

Avec TCP/IP, on peut maintenant **communiquer de façon fiable** entre logiciels situés sur des ordinateurs différents.

TCP/IP est utilisé par de nombreux logiciels et protocoles :

- Dans votre navigateur, le protocole **HTTP** utilise le protocole TCP/IP pour envoyer et recevoir des pages HTML, des images GIF, JPG et toutes sortes d'autres données.

- **FTP** est un protocole qui permet d'envoyer et recevoir des fichiers. Il utilise également TCP/IP.

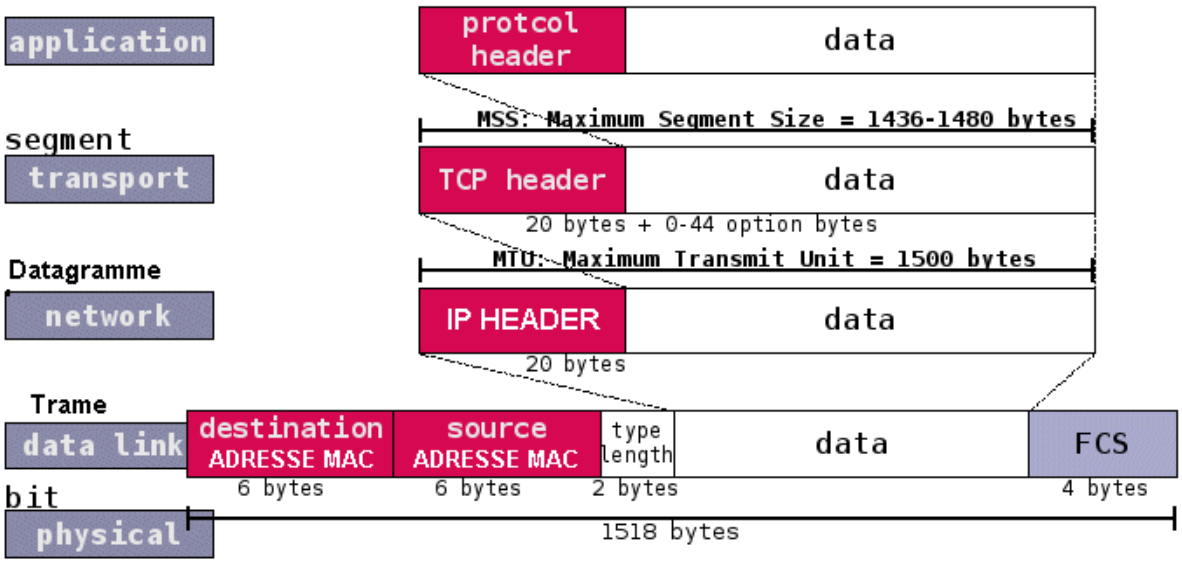
- Votre logiciel de courrier électronique utilise les protocoles **SMTP** et **POP3** pour envoyer et recevoir des emails. SMTP et POP3 utilisent eux aussi TCP/IP.

- Votre navigateur IE (ou autre Mozilla …) utilise le protocole **DNS** pour trouver l'adresse IP d'un ordinateur à partir de son nom (par exemple, de trouver 216.32.74.52 à partir de 'www.yahoo.com'). Le protocole DNS utilise UDP/IP et TCP/IP en fonction de ses besoins.

Il existe ainsi des centaines de protocoles différents qui utilisent TCP/IP ou UDP/IP.

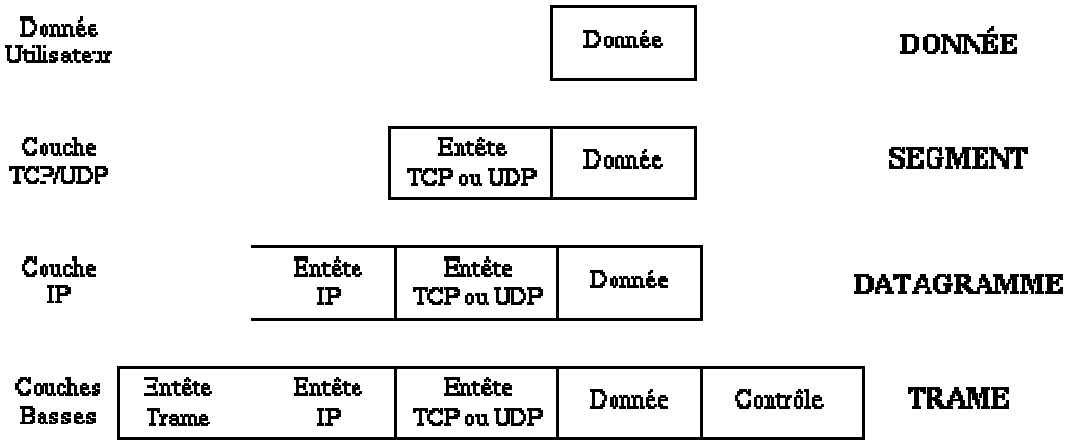
L'avantage de TCP sur UDP est que TCP permet des communications fiables. L'inconvénient est qu'il nécessite une négociation ("*Bonjour, prêt à communiquer ?*" etc.), ce qui prend du temps.

## 2 Format des trames et organisation en couche




Dans la trame est empaqueté le datagramme qui empaquette le segment qui empaquette la requête ou donnée de l'application. C'est la trame qui circule sur le réseau physique

Autre représentation :




La donnée, c'est le texte ou l'image que vous allez lire ou transmettre qui occupe un certain nombre de bits

Le segment est constitué de la DONNEE + d'un entête TCP ou UDP qui est un mot sur 20 ou 24 octets permettant le contrôle de la transmission (bits SYN, ACK, RST …) et d'assurer l'arrivée à bon « port » de la donnée. TCP fragmente le segment pour qu'il soit compatible avec la longueur max d'un datagramme (d'où la notion de « paquets » dans une transmission IP). Il les réassemble à la réception des paquets (un paquet –datagramme- fait environ 1500 Octets (même pas 2ko))

Le Datagramme est constitué du SEGMENT + un entête IP sur 20 ou 24 octets qui contient l'adresse IP du destinataire et de l'émetteur, ainsi que les informations pour la gestion de la fragmentation du datagramme.

La trame est constituée du DATAGRAMME + entête trame sur 18 octets qui contient des informations d'adressage physique (adresse MAC de la carte réseau), de synchronisation, format, contrôle d'erreur (CRC). C'est la trame qui va circuler sur les lignes physique du réseau (câbles réseau).

Note : L'adresse MAC (dans l'entête de la trame) correspond à l'adresse physique de la carte réseau du destinataire ou du routeur, à ne pas confondre avec l'adresse IP qui est une adresse « logique » et permet la structuration en réseau des ordinateurs que nous allons maintenant voir.

## 2.1 Trame Ethernet II

En octets

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 ---------- 1513 | 1514 | 1515 | 1516 | 1517 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Adresse MAC destination | | | | | | Adresse MAC source | | | | | | Type de protocole | | Données | FCS/CRC | | | |

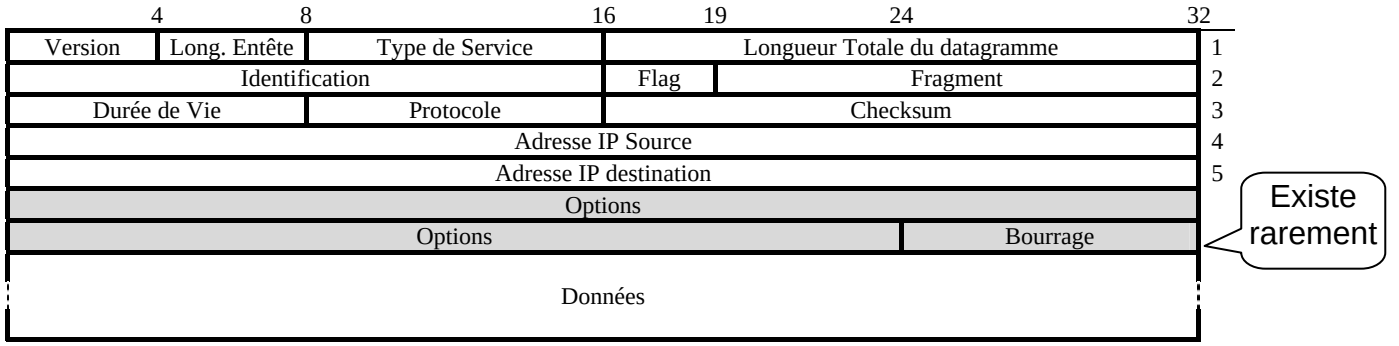
Attention : il existe d'autres types de trames Ethernet qui possèdent d'autres particularités. D'autre part, le champ FCS (Frame Check Sequence)/CRC n'est pas toujours visible dans les logiciels de capture

Le champ *Type de protocole* peut prendre les valeurs suivantes :
0x0800 : IPv4
0x86DD : IPv6
0x0806 : ARP
0x8035 : RARP

## 2.2 Datagramme IP

La taille maximale d'un datagramme IP (entête + données) est de 65535 octets. Comme il se peut qu'un réseau traversé ne puisse pas acheminer des blocs de données aussi grands, le protocole IP est capable de gérer la **fragmentation des datagrammes**, ce qui permet de découper un grand datagramme en datagrammes plus petits.

La structure générale d'un datagramme IP est représentée sur la figure suivante :




Voici la signification des différents champs :

- **Version** (4 bits) : il s'agit de la version du protocole IP que l'on utilise (actuellement on utilise la version 4 *IPv4*) afin de vérifier la validité du datagramme. Elle est codée sur 4 bits.
- **Longueur d'en-tête**, ou *IHL* pour *Internet Header Length* (4 bits) : il s'agit du nombre de mots de 32 bits constituant l'en-tête (nota : la valeur minimale est 5). Ce champ est codé sur 4 bits.
- **Type de service** (8 bits) : il indique la façon selon laquelle le datagramme doit être traité (priorité).
- **Longueur totale** (16 bits): il indique la taille totale du datagramme en octets. La taille de ce champ étant de 2 octets, la taille totale du datagramme ne peut dépasser 65536 octets. Utilisé conjointement avec la taille de l'en-tête, ce champ permet de déterminer où sont situées les données.
- **Identification, drapeaux (flags) et déplacement de fragment** sont des champs qui permettent de gérer la fragmentation des datagrammes. En effet, la taille d'un datagramme maximale est de 65536 octets. Toutefois cette valeur n'est jamais atteinte car les réseaux n'ont pas une capacité suffisante pour envoyer de si gros paquets.
- **Durée de vie** appelée aussi **TTL**, pour *Time To Live* (8 bits) : ce champ indique le nombre maximal de routeurs à travers lesquels le datagramme peut passer. Ainsi ce champ est décrémenté à chaque passage dans un routeur, lorsque celui-ci atteint la valeur critique de 0, le routeur détruit le datagramme. Cela évite l'encombrement du réseau par les datagrammes perdus.
- **Protocole** (8 bits) : ce champ, en notation décimale, permet de savoir de quel protocole est issu le datagramme
  - ICMP : 1
  - IGMP : 2
  - TCP : 6
  - UDP : 17
- **Somme de contrôle de l'en-tête, ou en anglais *header checksum* (16 bits)** : ce champ contient une valeur codée sur 16 bits qui permet de contrôler l'intégrité de l'en-tête afin de déterminer si celui-ci n'a pas été altéré pendant la transmission. La somme de contrôle est le complément à un de tous les mots de 16 bits de l'en-tête (champ *somme de contrôle* exclu). Celle-ci est en fait telle que lorsque l'on fait la somme des champs de l'en-tête (somme de contrôle incluse), on obtient un nombre avec tous les bits positionnés à 1
- **Adresse IP source** (32 bits) : Ce champ représente l'adresse IP de la machine émettrice, il permet au destinataire de répondre
- **Adresse IP destination** (32 bits) : adresse IP du destinataire du message
- **Options IP *:*** Il s'agit d'un champ de longueur variable utilisé principalement pour la mise au point ou pour des expérimentations. Les options ne sont pas obligatoires. Elles sont toutes codées les unes à la suite des autres, sans séparateur particulier.
- **Bourrage :** A cause de la structure du champ "longueur d'entête", la longueur de l'entête doit être un multiple de 32 bits. Si des options IP sont utilisées, il se peut que ce ne soit plus le cas. On remplit alors le champ "Bourrage" de zéros jusqu'à ce que la longueur de l'entête soit à nouveau multiple de 32 bits.
- **Données :** Ce sont les données du datagramme proprement dites

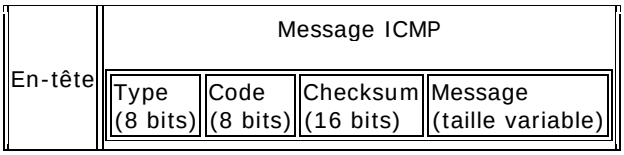
## 2.3 Segment ICMP, commande « ping »

Comme nous l'avons vu auparavant, le datagramme IP permet le transport de données de la couche supérieur, selon le protocole TCP ou UDP. Mais le réseau, bien que créé pour transporter des données à besoin d'outils qui permettent de contrôler et des diagnostiquer des erreurs de transmission. C'est la fonction remplie par le protocole ICMP (**Internet Control Message Protocol**).

Nous allons en voir la commande « ping » qui fonctionne sur le protocole ICMP. **Ping** utilise ainsi deux types de messages du protocole ICMP (sur les 18 proposés par ICMP) :

- Le type 0 correspondant à une commande "echo request", émis par la machine source ;
- Le type 8 correspondant à une commande "echo reply", émis

A intervalles réguliers (par défaut chaque seconde), la machine source (celle sur laquelle la commande ping est exécutée) envoie une commande "echo request" à la machine cible. Dès réception du paquet "echo reply", la machine source affiche une ligne contenant un certain nombre d'informations. En cas de non réception de la réponse, une ligne indiquant "délai dépassé" s'affichera.




**Type 8 (echo)**
Type : 8
Code : 0
Message : il contient l'IDENTIFIER et le SEQUENCE NUMBER qui sont utilisés par l'émetteur pour contrôler les réponses aux requêtes, (CODE = 0). Le reste des données en souvent du bourrage qui sert à donner une longueur plus ou moins grande

Signification : demande de renvoi d'informations, avec la commande ping par exemple

**Type 0 (réponse echo)**
Type : 0
Code : 0
Message : il contient le contenu du message envoyé lors de la requête ; l'IDENTIFIER, le SEQUENCE NUMBER et le message de bourrage sont donc inchangés.

Signification : réponse au message de type 8

```
C:\WINNT\system32>ping 192.186.11.1

Pinging 192.186.11.1 avec 32 octets de données :

Réponse de 192.186.11.1 : octets=32 temps=111ms TTL=255
Réponse de 192.186.11.1 : octets=32 temps<10ms TTL=255
Réponse de 192.186.11.1 : octets=32 temps<10ms TTL=255
Réponse de 192.186.11.1 : octets=32 temps<10ms TTL=255
```

Les options à la commande sont nombreuses. Entre autres :
ping –n 10 ... : pour envoyer 10 paquets
ping –l 50 ... : pour envoyer des paquets de tailles 50 octets

ping : sans argument, pour avoir la liste des options (Windows).
man ping : pour avoir la liste des options (Unix).

## 2.4 Datagramme Arp

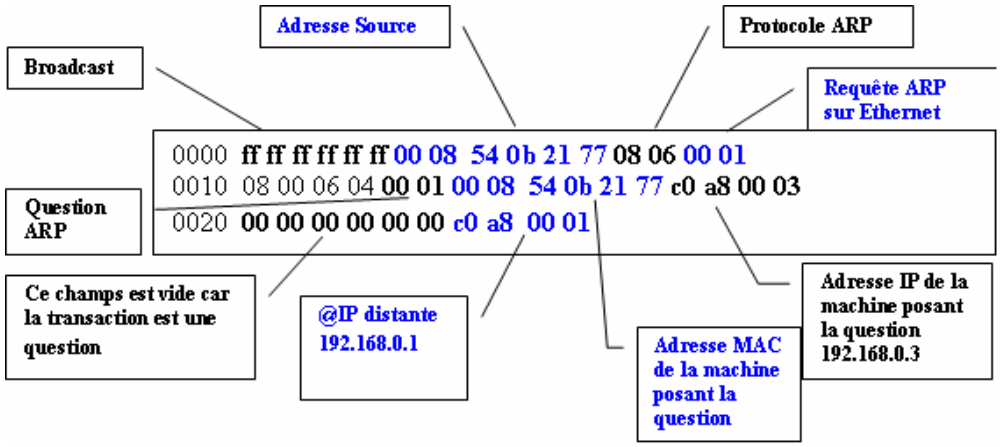
Le protocole ARP permet de connaître l'adresse physique d'une carte réseau correspondant à une adresse IP, c'est pour cela qu'il s'appelle Protocole de résolution d'adresse (en anglais ARP signifie Address Resolution Protocol). Le protocole ARP émet une requête sur le réseau. L'ensemble des machines du réseau vont comparer cette adresse logique à la leur. Si l'une d'entre-elles s'identifie à cette adresse, la machine va répondre à la requête ARP qui va stocker le couple d'adresses dans la table de correspondance et la communication va alors pouvoir avoir lieu... Pour visualiser cette table, il existe la commande « arp » sous windows.

### Trame Arp / Rarp

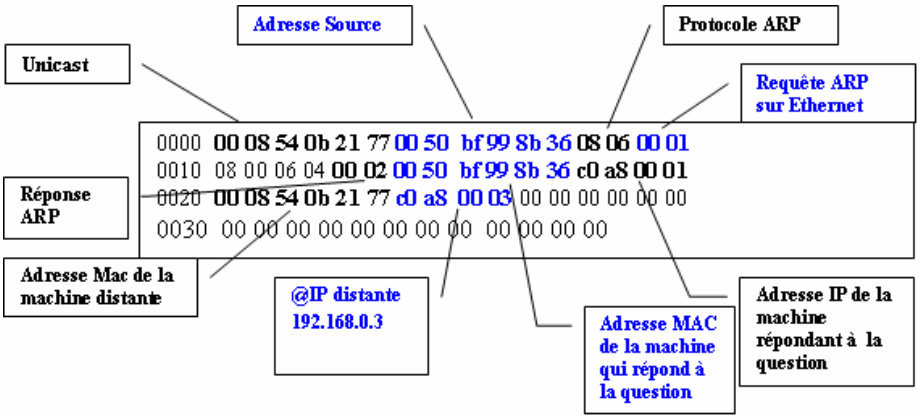
| 0 | 8 | 16 | 31 |
|---|---|---|---|
| Type de matériel (réseau) | | Type de protocole | |
| Lg. Adr. MAC | Lg. Adr. Prot. | Opération | |
| Adresse MAC Emetteur (0-3) | | | |
| Adresse MAC Emetteur (4-5) | | Adresse IP Emetteur (0-1) | |
| Adresse IP Emetteur (2-3) | | Adresse MAC Récepteur (0-1) | |
| Adresse MAC Récepteur (2-5) | | | |
| Adresse IP Récepteur (0-3) | | | |

- *Type de matériel = 1 pour Ethernet*
- *Type de protocole = 0x0800 pour IP*
- *Lg. Adr. MAC = 6 octets pour 'adresse MAC*
- *Lg. Adr. Prot. = 4 octets pour IP*
- *Opération = 1 (requête ARP), 2 (réponse ARP), 3 (requête RARP), 4 (réponse RARP)*
- *Adresses Physiques et IP: adresses MAC sur 48 bits et IP sur 32 bits de l'émetteur et du récepteur remplies selon l'opération en cours.*

### Arp Request



### Arp Reply




**Le cache** **arp –a**

```
C:\Documents and Settings>arp -a

Interface : 192.168.0.74 --- 0x2
  Adresse Internet      Adresse physique      Type
  192.168.0.1           02-30-8d-00-00-67     dynamique
```

Les options à la commande sont nombreuses. Il est par exemple possible de créer manuellement une correspondance ou bien en supprimer. Par exemple :
**ARP** –d *
Va supprimer toute la table
Exemples :

arp -s 157.55.85.212 00-aa-00-62-c6-09 .... Ajoute une entrée statique.
arp -a .... Affiche la table ARP.

## 3 Dialogue entre 2 équipements

Une station A veut envoyer un paquet IP à une station B du réseau local. La station A connaît l'adresse IP de la station B mais ne connaît pas son adresse MAC.

**Étape 1**
La station A va donc fabriquer un paquet de requête ARP. Ce paquet contient (entre autre) les éléments suivants :

- Le type de paquet ARP (demande, requête : opération=1)
- @ MAC de A (adresse MAC source)
- @ IP de A (adresse IP source)
- @ MAC à 0 (indiquant @MAC destination inconnue)
- @ IP de B (Adresse IP destination)

**Étape 2**
Ce paquet est ensuite encapsulé dans une trame qui va être diffusée à toutes les stations du réseau local (il est nécessaire de la diffuser puisque l'adresse MAC destinataire n'est pas connue). Les éléments de la trame seront donc :

- @ MAC de A (adresse MAC de la source)
- @ MAC de diffusion (FF FF FF FF FF FF)
- Type de protocole au niveau 3 (0806 indique que le champ « info » de la trame Ethernet II contient un paquet ARP)

**Étape 3**
Toutes les stations du réseau local vont donc devoir prendre en compte cette trame et analyser le paquet qui s'y trouve encapsulé. Les stations qui ne reconnaissent pas leur propre adresse IP dans le champ Adresse IP destination ne font rien de plus.

**Étape 4**
Par contre la station B qui reconnaît son adresse IP doit fabriquer un paquet de réponse ARP. Ce paquet contient entre autre les éléments suivants :

- Le type de paquet ARP (Réponse : opération=2)
- @ MAC de B (adresse MAC source. C'était l'adresse recherchée)
- @ IP de B (adresse IP source = *@IP destination dans la Requête*)
- @ MAC de A (adresse MAC destination = *@MAC source dans la Requête* )
- @ IP de A (Adresse IP destination = *@IP source dans la Requête*)

**Étape 5**
Ce paquet est ensuite encapsulé dans une trame qui va être envoyée à la station A. Les éléments de la trame seront donc :

- @ MAC de B (adresse MAC de la source. C'était l'adresse recherchée)
- @ MAC de A (adresse MAC destination)

**Étape 6**
La station A qui reçoit directement la trame (point à point) va pouvoir envoyer ses datagrammes IP (contenant de l'UDP, du TCP de l'IGMP (ping), etc…) dans une trame ayant la bonne adresse MAC de destination. La station A va mettre aussi à jour sa table ARP. Ceci lui permettra, la prochaine fois, de ne pas refaire une requête ARP. Toutefois, étant donné qu'un réseau peut évoluer et que les cartes réseau peuvent être changées, il est nécessaire qu'une station vérifie régulièrement sa table ARP en refaisant des requêtes ARP.

Sources utilisés :

http://sebsauvage.net/comprendre/tcpip/