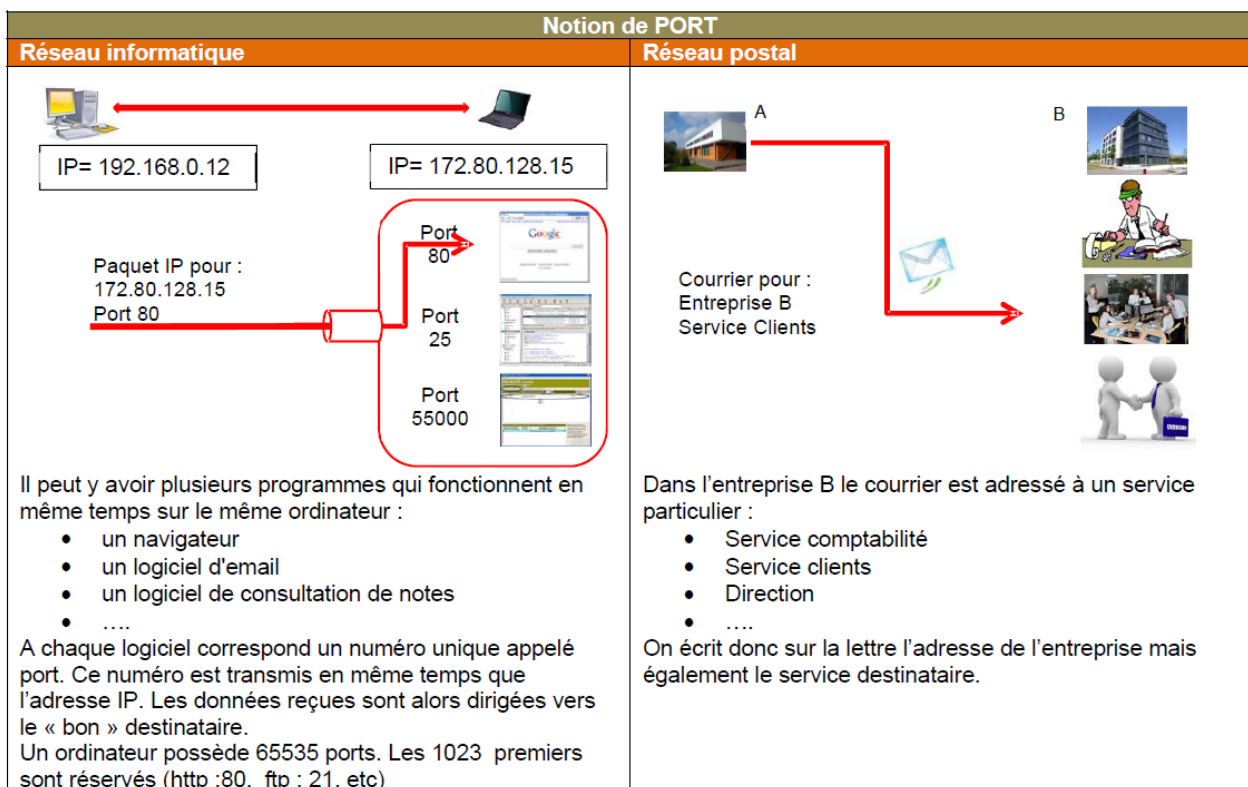
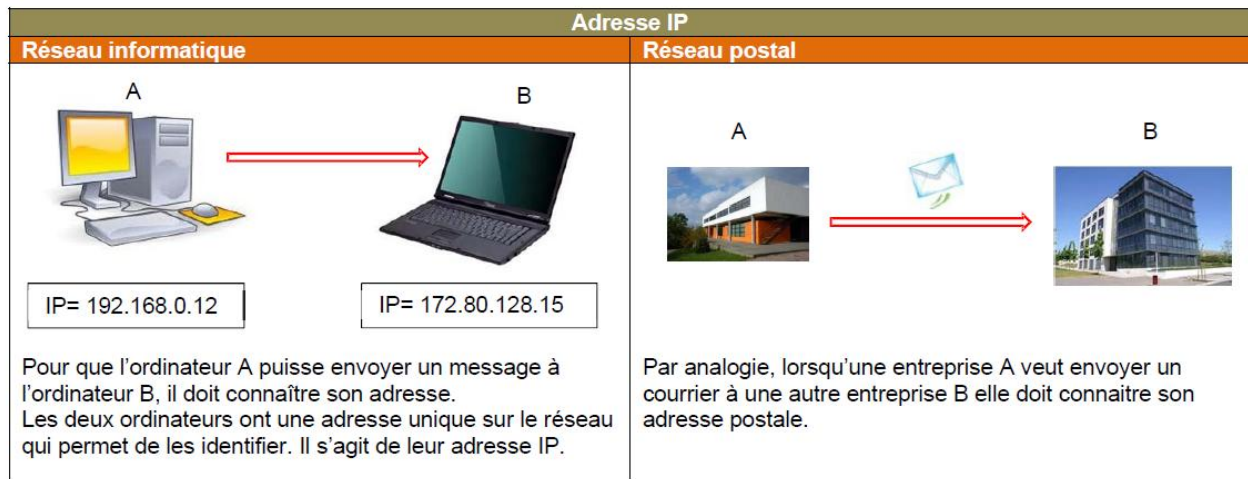

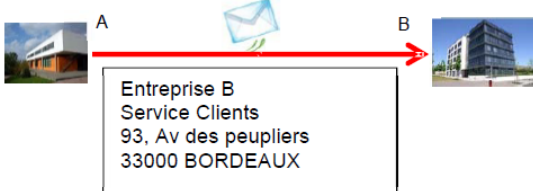


# Modèle en couche réseau

## La communication : un problème d'adresse.

La communication entre deux ordinateurs peut être comparée à l'envoi d'un courrier postal entre un expéditeur et un destinataire.



Notion de PROTOCOLE	
Réseau informatique	Réseau postal
 <p>Les deux ordinateurs peuvent également échanger des données parce qu'ils utilisent les mêmes <b>protocoles</b> de communication. Cela permet par exemple :</p> <ul style="list-style-type: none"> <li>• L'envoi d'un message avec OUTLOOK et sa lecture avec GMAIL car le codage du message est reconnu par les deux logiciel.</li> <li>• L'envoi des données par une carte réseau d'un ordinateur APPLE et sa réception par une carte réseau d'un PC car la façon d'ordonner les informations est la même pour les deux cartes.</li> <li>• L'envoi d'un message d'un ordinateur à un autre éloigné de plusieurs km parce qu'il respectent tous les deux le protocole IP.</li> </ul> <p>Quelques exemple de protocoles : IP, ARP, DNS, http, HTTPS, FTP, TCP, UDP, ICMP, RIP, OSPF, SMTP, POP, TELNET, SSH, DHCP,....</p>	 <p>Le courrier transmis par l'entreprise (A) arrive bien à l'entreprise (B) par ce que :</p> <ul style="list-style-type: none"> <li>• Le format de l'enveloppe respecte la norme</li> <li>• Les éléments de l'adresse respectent les règles fixées par les sociétés d'acheminement du courrier.(N° de rue, Code postal, ...)</li> </ul> <p>On respecte là aussi des PROTOCOLES de communication.</p>

## A Le Modèle TCP-IP

### A.1 Les 4 couches du modèle TCP/IP

TCP/IP est en fait une **suite de protocoles**. Cette appellation vient des noms des deux principaux protocoles de la suite, à savoir TCP - **Transmission Control Protocol** (littéralement, « protocole de contrôle de transmissions ») et IP – **Internet Protocol** (Protocole Internet).



- **TCP** s'occupe de contrôler que la transmission des données s'effectue sans erreurs.
- **IP** s'occupe de découper l'information à transmettre en paquets, de les adresser, de les transporter indépendamment les uns des autres et de recomposer le message initial à l'arrivée.

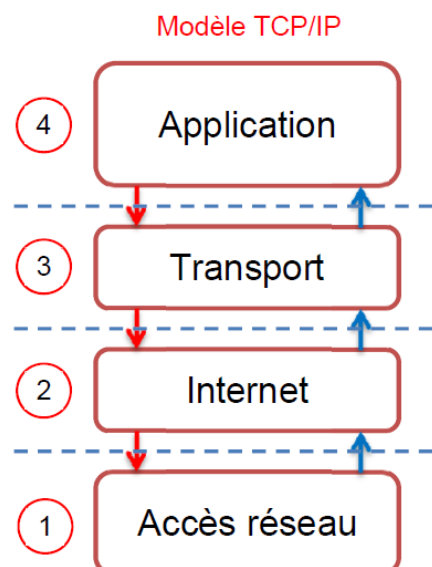
TCP/IP représente l'ensemble des règles de communication sur internet et se base sur le fait que chaque machine possède une adresse IP.  
TCP/IP effectue :

- @ Le fractionnement des données en paquets
- @ L'utilisation d'un système d'adressage (IP)
- @ L'acheminement des données sur le réseau (routage)
- @ La détection et la correction des erreurs de transmission.

Afin de pouvoir appliquer le modèle TCP/IP à n'importe quelles machines, logiciels et matériels, le système de protocoles TCP/IP a été décomposé en plusieurs modules effectuant les uns après les autres une tâche précise. On a donc un système stratifié, c'est la raison pour laquelle on parle de **modèle en couches**.

Chaque couche a une tâche précise. Une fois cette tâche réalisée elle transmet l'information à la couche voisine :

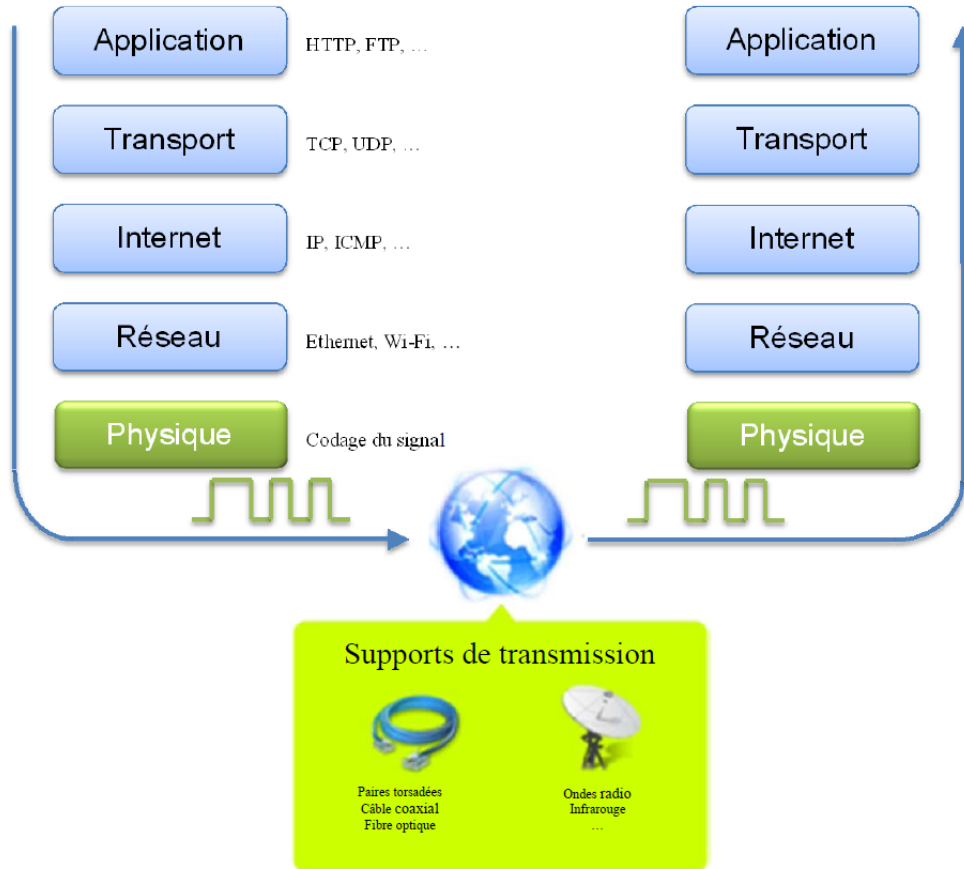
- Au-dessus lors d'une réception 
- Au-dessous lors d'une émission 



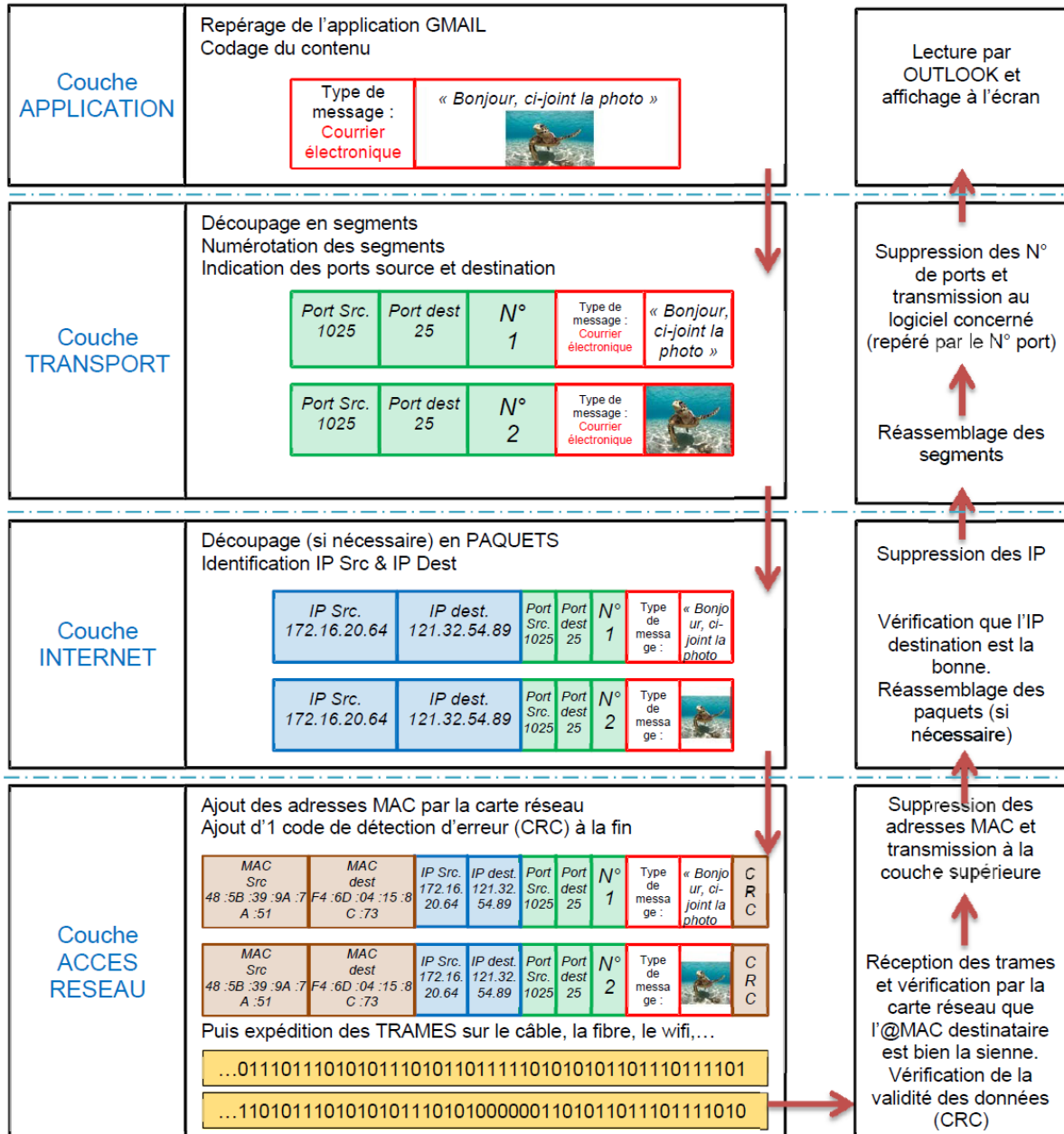
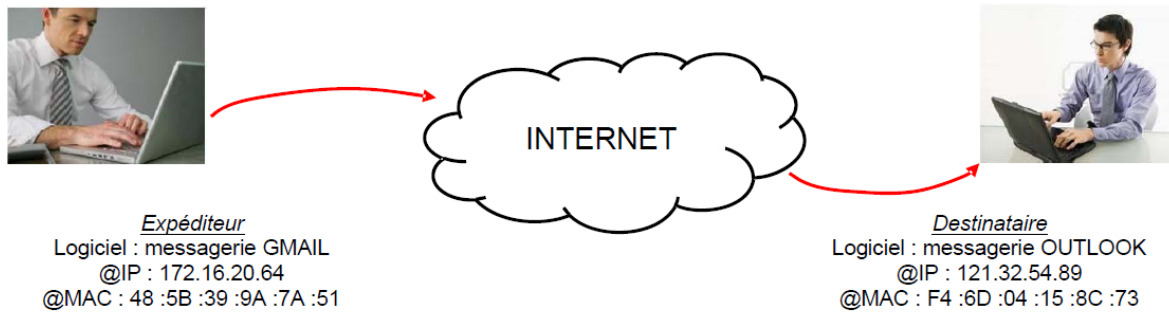
Emetteur



Récepteur



## A.2 Exemple simplifié du fonctionnement par « Couches » du modèle TCP/IP

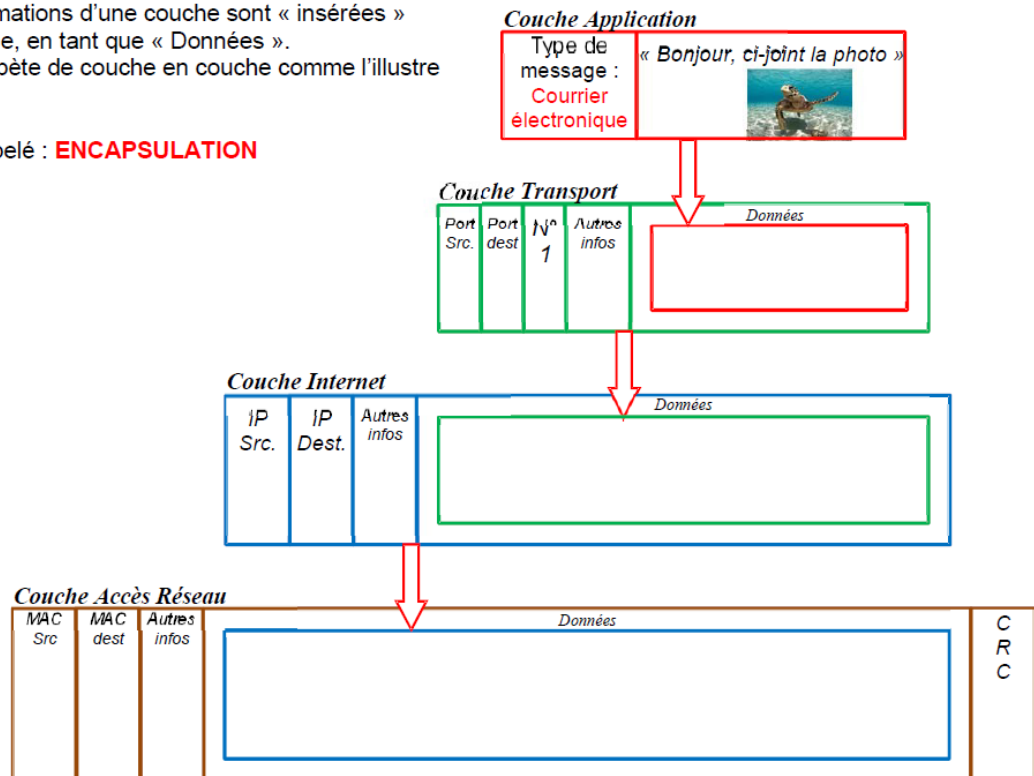


Le schéma précédent montre que chaque couche ajoute des informations à celles fournies par la couche.

### A.3 ENCAPSULATION / DECAPSULATION

A l'émission, les informations d'une couche sont « insérées » dans la couche voisine, en tant que « Données ». Ce phénomène se répète de couche en couche comme l'illustre le schéma ci-contre.

Ce processus est appelé : **ENCAPSULATION**



Remarque : A la réception, le processus inverse se produit : la **DECAPSULATION**

### A.4 Protocoles des différentes couches

<b>Couche 4: APPLICATIONS</b>							Sur cette couche circulent des <b>données</b> encore appelées <b>flot de données</b> ou <b>messages</b>
FTP	SMTP	POP	IMAP	SSH	RPC	etc...	
<b>Couche 3: TRANSPORT</b>							Sur cette couche circulent des <b>segments TCP</b> ou bien des <b>paquets UDP</b>
TCP	UDP						
<b>Couche 2: INTERNET</b>							Sur cette couche circulent des <b>datagrammes IP/ARP/ICMP</b>
IP	ARP	RARP	ICMP	IGMP			
<b>Couche 1: RESEAU</b>							Sur cette couche circulent des <b>trames Ethernet</b> (s'il s'agit d'un réseau Ethernet bien sûr)
ATM	X25	Ethernet	Token ring	FTS	FDDI	etc...	



## B Le modèle OSI

### B.1 Les 7 couches du modèle OSI

Il existe d'autres modèles décrivant la transmission de l'information. Parmi ceux-ci, il en existe un, concurrent du modèle TCP/IP et plus détaillé : c'est le modèle **OSI**.

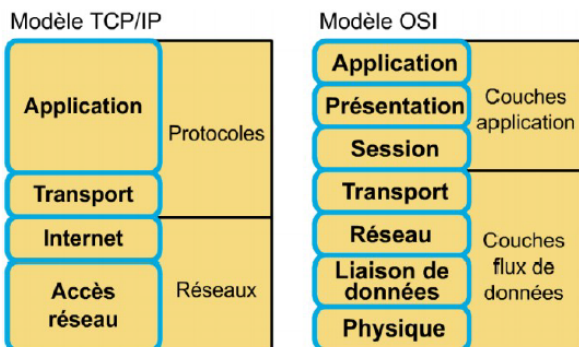
Le principe de fonctionnement est exactement le même que celui observé précédemment (rôle de chaque couche, encapsulation / Décapsulation).

Le modèle OSI possède 7 couches contre 4 pour le modèle TCP/IP

Position dans le modèle OSI	Nom de la couche	Rôle de la couche
7	Application	Point de contact avec les services réseaux.
6	Présentation	Elle s'occupe de tout aspect lié à la présentation des données : format, cryptage, encodage, etc.
5	Session	Responsable de l'initialisation de la session, de sa gestion et de sa fermeture.
4	Transport	Choix du protocole de transmission et préparation de l'envoi des données. Elle spécifie le numéro de port utilisé par l'application émettrice ainsi que le numéro de port de l'application réceptrice. Elle fragmente les données en plusieurs séquences (ou segments).
3	Réseau	Connexion logique entre les hôtes. Elle traite de tout ce qui concerne l'identification et le routage dans le réseau.
2	Liaison de données	Établissement d'une liaison physique entre les hôtes. Fragmente les données en plusieurs trames.
1	Physique	Conversion des trames en bits et transmission physique des données sur le média.

### 3.2 Comparaison OSI – TCP/IP

Les deux modèles fonctionnent sur le même principe de l'Encapsulation / Décapsulation et possèdent beaucoup de points communs.



## TD Modèle OSI Exercices d'applications

### Question :

Quelle est la taille (mini, maxi) d'une trame Ethernet II ?

.....

### EXERCICE N°1

Lors d'un dialogue réseau, la trame suivante a été capturée :

```
0000 08 00 3e 26 75 2f 00 0f 1f 83 f2 26 08 00 45 00 ..>&u/.. ...&...E.
0010 00 2b 06 46 40 00 80 06 c6 0d 96 32 01 14 96 32 .+.F@... ...2...2
0020 01 01 04 5e 00 17 49 9b c3 a5 01 7c dc 05 50 18 ...^...I. ...|..P.
0030 ff fc 2e 97 00 00 ff fd 01 .....
```



Au niveau du paquet IP, aucune option et « padding » ne sont effectués.  
Les adresses IP seront notées en décimale pointée et les numéros de port en base 10.

### Question :

Complétez le tableau ci-dessous.

Adresse MAC source :	Adresse MAC destination :
Protocole de la couche 3 utilisé :	
Trame IP fragmentée ? :	Numéro du fragment IP :
Adresse IP source :	Adresse IP destination :
Protocole porté par le paquet IP :	
Numéro de port source :	Numéro de port destination :

## EXERCICE N°2

Lors d'un dialogue réseau, la trame suivante a été capturée :

```
0000  00 12 17 41 c2 c7 00 1a 73 24 44 89 08 00 45 00
0010  00 3c 27 30 00 00 80 01 8f d6 c0 a8 01 69 c0 a8
0020  01 01 08 00 4d 56 00 01 00 05 61 62 63 64 65 66
0030  67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76
0040  77 61 62 63 64 65 66 67 68 69 . . . . .
```

### Question :

**Complétez le tableau ci-dessous.**

Adresse MAC source :	Adresse MAC destination :
Protocole de la couche 3 utilisé :	
Trame IP fragmentée ? :	Numéro du fragment IP :
Adresse IP source :	Adresse IP destination :
Protocole porté par le paquet IP :	
Numéro de port source :	Numéro de port destination :



## EXERCICE N°2

Lors d'un dialogue réseau, la trame suivante a été capturée :

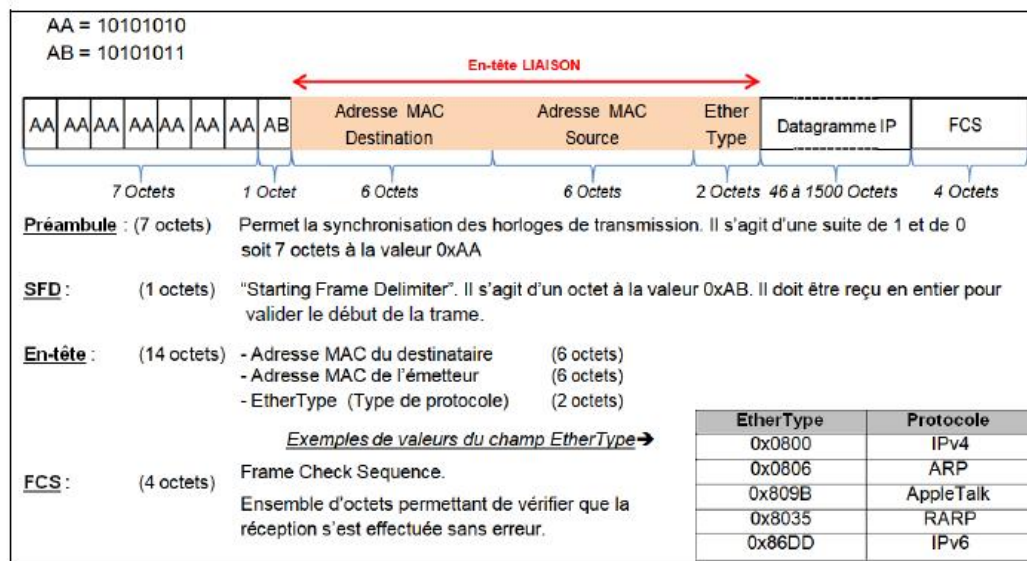
```
0000  00 12 17 41 c2 c7 00 1a 73 24 44 89 08 00 45 00
0010  00 3c 27 30 00 00 80 01 8f d6 c0 a8 01 69 c0 a8
0020  01 01 08 00 4d 56 00 01 00 05 61 62 63 64 65 66
0030  67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76
0040  77 61 62 63 64 65 66 67 68 69 . . . . .
```

### Question :

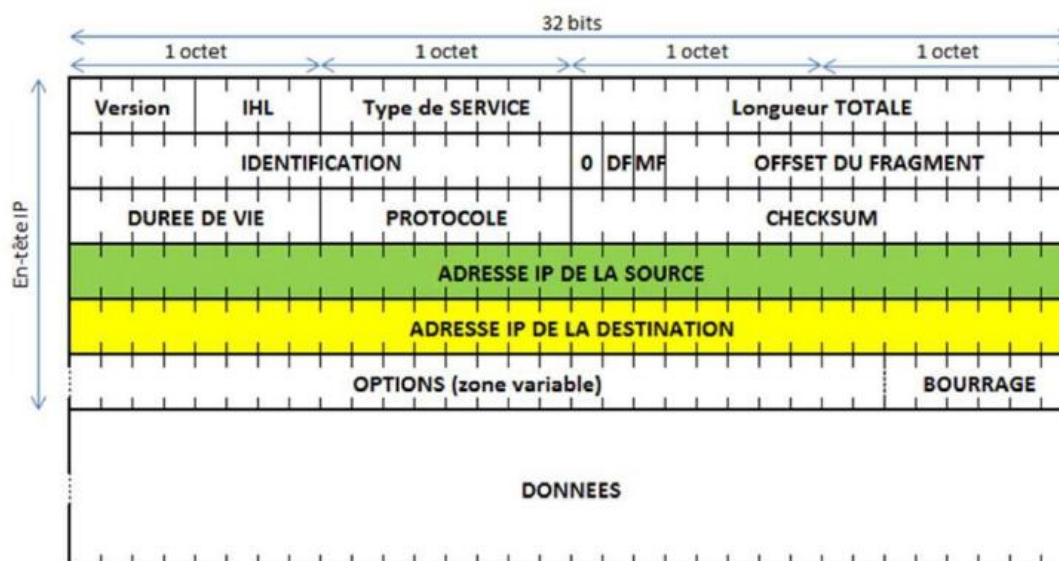
**Complétez le tableau ci-dessous.**

Adresse MAC source :	Adresse MAC destination :
Protocole de la couche 3 utilisé :	
Trame IP fragmentée ? :	Numéro du fragment IP :
Adresse IP source :	Adresse IP destination :
Protocole porté par le paquet IP :	
Numéro de port source :	Numéro de port destination :

## STRUCTURE DE LA TRAME ETHERNET II



## STRUCTURE D'UN PAQUET (DATAGRAMME) IP



- Version :** (4 bits) il indique le numéro de version du protocole IP utilisé (généralement 4).
- IHL :** (4 bits) Internet Header Length (Longueur d'entête). Spécifie la longueur de l'en-tête du Datagramme en nombre de mots de 32 bits. Ce champ ne peut prendre une valeur inférieure à 5.
- Type de service :** (8 bits) Donne une indication sur la qualité de « service » souhaitée pour l'acheminement des données.

0	1	2	3	4	5	6	7
Priorité	D	T	R	C	x		

Bits 0- 2	Priorité	010→Immédiate	001→Normale	000→Basse
Bit 3	D	0 = Retard standard	1 = Retard faible	
Bit 4	T	0 = Débit standard	1 = Haut débit	
Bit 5	R	0 = Taux d'erreur standard	1 = Taux d'erreur faible	
Bit 6	C	0 = Coût standard	1 = Coût faible	
Bit 7	x	Réservé		

<b>Longueur totale :</b>	(16 bits)	Longueur du datagramme entier y compris en-tête et données mesurée en octets.
<b>Identification :</b>	(16 bits)	Valeur assignée par l'émetteur pour identifier les fragments d'un même datagramme.
<b>Flags :</b>	(3 bits)	Commutateurs de contrôle : <ul style="list-style-type: none"> <li>- Bit 0 : Réservé, doit être laissé à 0</li> <li>- Bit 1 : (DF - Don't fragment) 0= Fragmenté 1= Non fragmenté</li> <li>- Bit 2 : (MF - More Fragment) 0= Dernier fragment 1= Fragment</li> </ul>
<b>OFFSET :</b>	(13 bits)	Décalage du premier octet du fragment par rapport au datagramme complet non fragmenté. Cette position est mesurée en blocs de 8 octets (64 bits).
<b>Durée de vie :</b>	(8 bits)	Temps en secondes pendant lequel le datagramme doit rester dans le réseau. Si ce champ vaut 0, le datagramme doit être détruit. Ce temps diminue à chaque passage du datagramme d'une machine à l'autre.
<b>Protocole :</b>	(8 bits)	Protocole porté par le datagramme

Valeur	Protocole
1	ICMP
6	TCP
17	UDP
Etc	etc

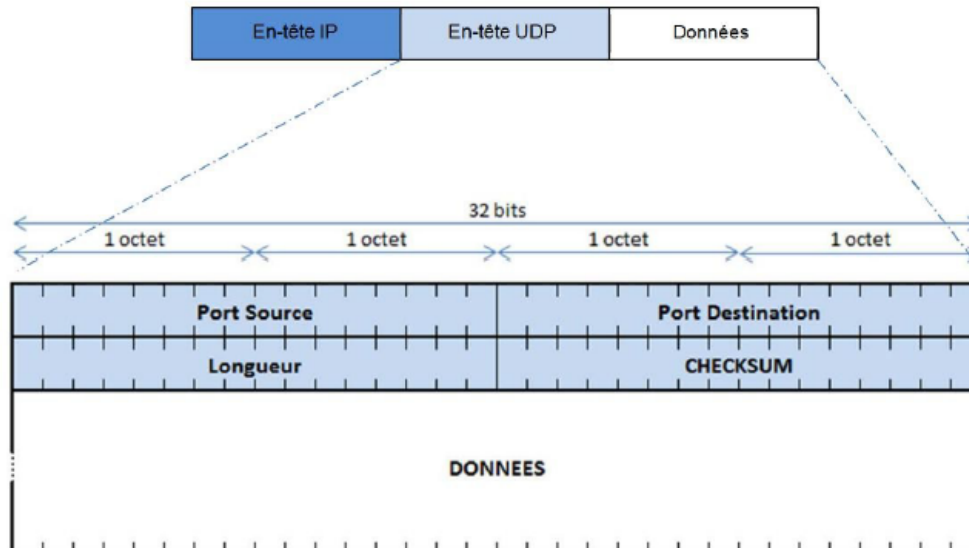
  

<b>Checksum :</b>	(16 bits)	(Somme de contrôle) C'est une valeur qui permet de détecter une éventuelle erreur de transmission avec une très grande probabilité.
<b>IP Source :</b>	(32 bits)	Adresse IP de l'émetteur.
<b>IP Destination :</b>	(32 bits)	Adresse IP du destinataire.
<b>Options :</b>	(Variable)	Le champ est de longueur variable. Un datagramme peut comporter 0 ou plusieurs options.
<b>Bourrage :</b>	(Variable)	Le champ Bourrage n'existe que pour assurer à l'en-tête une taille totale multiple de 4 octets. Le bourrage se fait par des octets à 0.

## STRUCTURE D'UN DATAGRAMME UDP

**UDP** est encapsulé dans un paquet IP. Il comporte un en-tête suivi des données proprement dites à transporter.

L'en-tête d'un datagramme UDP est plus simple que celui de TCP :



Il contient les quatre champs suivants :

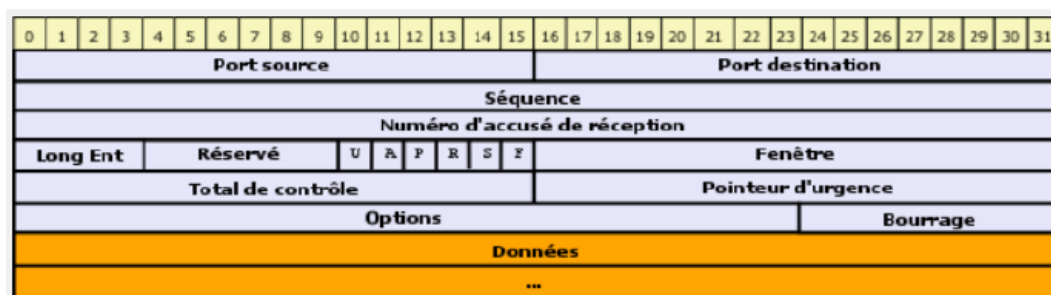
**Port Source :** indique depuis quel port le paquet a été envoyé.

**Port Destination :** indique à quel port le paquet doit être envoyé.

**Longueur :** indique la longueur totale (exprimée en octets, en-tête et données). La longueur minimale est donc de 8 octets (taille de l'en-tête).

**Somme de contrôle (CHECKSUM) :** celle-ci permet de s'assurer de l'intégrité du paquet reçu quand elle est différente de zéro. Elle est calculée sur l'ensemble de l'en-tête UDP et des données, mais aussi sur un pseudo en-tête (extrait de l'en-tête IP).

## STRUCTURE D'UN SEGMENT TCP



**Port source** : c'est le port utilisé pour les données à émettre. C'est un des éléments du quadruplet qui identifie une connexion.

**Port destination** : c'est le port où les données sont envoyées. Il doit être connu pour identifier l'application à laquelle les données sont envoyées. C'est le deuxième élément du quadruplet identificateur.

**Numéro de séquence** : il donne la position du segment dans le flux de l'émetteur. Deux cas sont à considérer :

- le bit SYN est positionné à 1, alors le numéro de séquence a pour valeur *ISN* [*Initial Sequence Number*] + 1.
- le bit SYN est positionné à 0, alors le numéro de séquence a pour valeur le numéro du premier octet de données relativement au début de la transmission.

**Numéro d'accusé de réception** : il indique le numéro du prochain octet attendu par le récepteur.

**Longueur en-tête** : il indique la longueur de l'en-tête d'un segment TCP et est exprimé comme un multiple de 32 bits. Ce champ est rendu indispensable dans la mesure où la longueur du champ option est variable (selon les options choisies).

**Réservé** : comme son nom l'indique, il est réservé à un usage futur. Il est donc positionné à zéro.

**Bits de code TCP** :

bit <i>URG</i>	C'est le pointeur de données urgentes s'il est positionné à 1. Indique que les données doivent être délivrées immédiatement (notification d'événement en temps réel).
bit <i>ACK</i>	Le paquet est un accusé de réception, s'il est positionné à 1. Le segment requiert un <i>push</i> , s'il est positionné à 1.
bit <i>PSH</i>	Ce flag indique au récepteur que les données doivent être remise immédiatement à l'application, sans bufférisation. Utilisé dans les sessions interactive comme <i>OpenSSH</i> . Réinitialiser la connexion, s'il est positionné à 1.
bit <i>RST</i>	Ce flag positionné par une des extrémités indique une condition d'erreur non récupérable. Dans ce cas, les deux extrémités termine la connexion, libère les ressources allouées à la connexion et détruit tous les paquets subséquents en transit. Demande de synchronisation des numéros de séquence, s'il est positionné à 1.
bit <i>SYN</i>	Ce flag est armé dans le premier paquet envoyé par le client et le serveur. Chaque octet de données est séquentiellement numéroté ( <i>ISN</i> [ <i>Initial Sequence Number</i> ] doit être choisi aléatoirement (éviter les prédictions), les échanges subséquent se font en incrémentant cet ISN.
bit <i>FIN</i>	Indique la fin d'une connexion, s'il est positionné à 1. Indique que la transmission est terminée (complète), envoi d'un message <i>SYN+FIN</i> , attente de l'acquiescement de confirmation ; message <i>FIN</i> puis message <i>SYN+FIN</i> , envoi d'un message <i>ACK</i> final.

**Fenêtre** : indique le nombre d'octet que le récepteur peut admettre (à partir de la position contenu dans l'accusé de réception) sans qu'un accusé de réception soit nécessaire.

**Total de contrôle** : ce champ permet de vérifier l'intégrité de l'en-tête TCP et des données. C'est le complément à 1 (sur 16 bits) de la somme des compléments à 1 des octets de l'en-tête et des données (par mots de 32 bits). À noter que le champ de 16 bits le représentant est positionné à 0 lors du calcul.

**Pointeur d'urgence** : ce champ est utilisé si le bit *URG* est positionné (à 1) indique dans la fenêtre la position où les données urgentes s'arrêtent.

**Options** : ce champ contient les différentes options TCP. Par exemple, le *MSS* ([*Maximum Segment Size*], taille maximale des segments), le *window scale option*, le *timestamp option* ...

**Bourrage (taille variable)** : permet de parvenir à un en-tête d'une taille multiple de 32 bits. Il complète par des 0 la fin du champ *options*.

**Données (taille variable)** : il s'agit des données à transmettre.