Pourquoi faire des couches?

- Pour gérer des systèmes complexes
- Permet l'identification explicite des structures et des relations entre entités
- Modèle pour permettre des discussions (organismes de normalisation)
- Modularité simplifie la maintenance et la mise à jour
- Des changements internes dans une couche n'affecte pas les autres couches
- Chaque couche prend les données de la couche supérieure
  - Ajoute un « en-tête » utilisé par l'entité paire
  - Donne les données à la couche inférieure

Modèle TCP/IP (1)

- Modèle à 4 couches.
- **Couche Application (4) :**

  - Cette couche contient tous les protocoles de haut niveau, comme par exemple Telnet, TFTP (trivial File Transfer Protocol), SMTP (Simple Mail Transfer Protocol), HTTP (HyperText Transfer Protocol). Le point important pour cette couche est le choix du protocole de transport à utiliser. Par exemple, TFTP (surtout utilisé sur réseaux locaux) utilisera UDP, car on part du principe que les liaisons physiques sont suffisamment fiables et les temps de transmission suffisamment courts pour qu'il n'y ait pas d'inversion de paquets à l'arrivée. Ce choix rend TFTP plus rapide que le protocole FTP qui utilise TCP. A l'inverse, SMTP utilise TCP, car pour la remise du courrier électronique, on veut que tous les messages parviennent intégralement et sans erreurs.

Modèle TCP/IP (2)

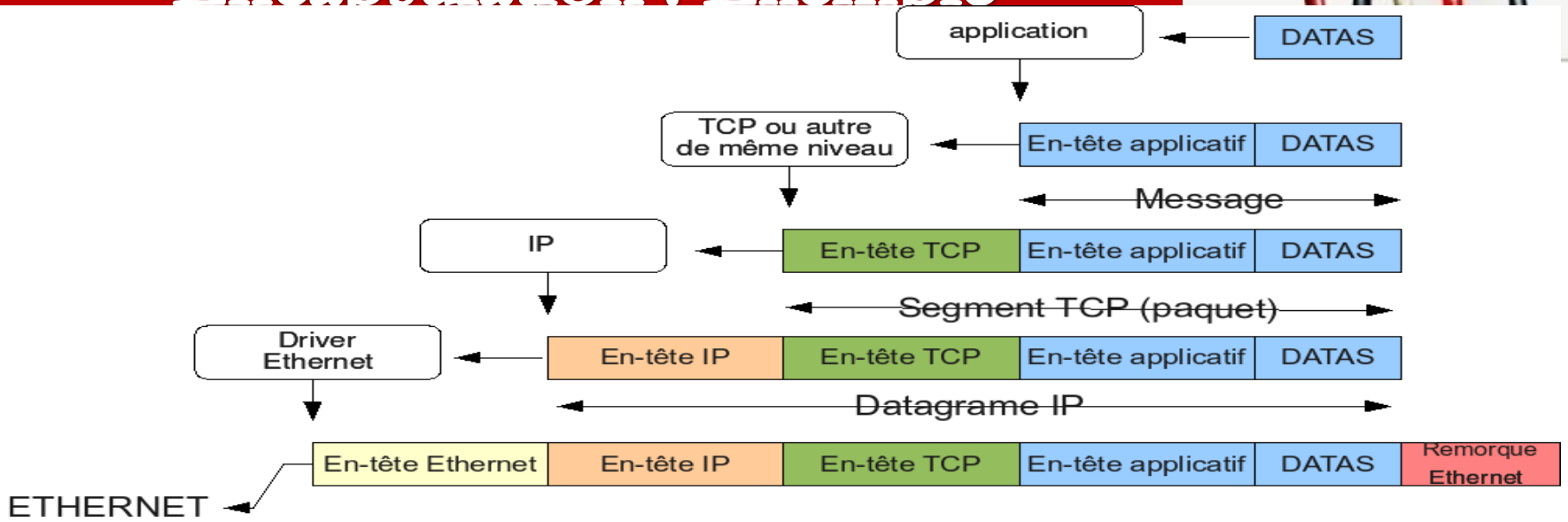
- **Couche Transport :**
  - La principale tâche de la couche de transport est de fournir la communication d'un programme d'application à un autre. Une telle communication est souvent qualifiée de 'point à point'.
  - cette couche n'a que deux implémentations : le protocole TCP (Transmission Control Protocol) et le (User Datagram Protocol). TCP est un protocole fiable, orienté connexion, protocole UDP qui permet l'acheminement sans erreur de paquets issus d'une machine d'un internet à une autre machine du même internet. Son rôle est de fragmenter le message à transmettre de manière à pouvoir le faire passer sur la couche internet. A l'inverse, sur la machine destination, TCP replace dans l'ordre les fragments transmis sur la couche internet pour reconstruire le message initial. TCP s'occupe également du contrôle de flux de la connexion.
    UDP est en revanche un protocole plus simple que TCP : il est non fiable et sans connexion. Son utilisation présuppose que l'on n'a pas besoin ni du contrôle de flux, ni de la conservation de l'ordre de remise des paquets. Par exemple, on l'utilise lorsque la couche application se charge de la remise en ordre des messages;

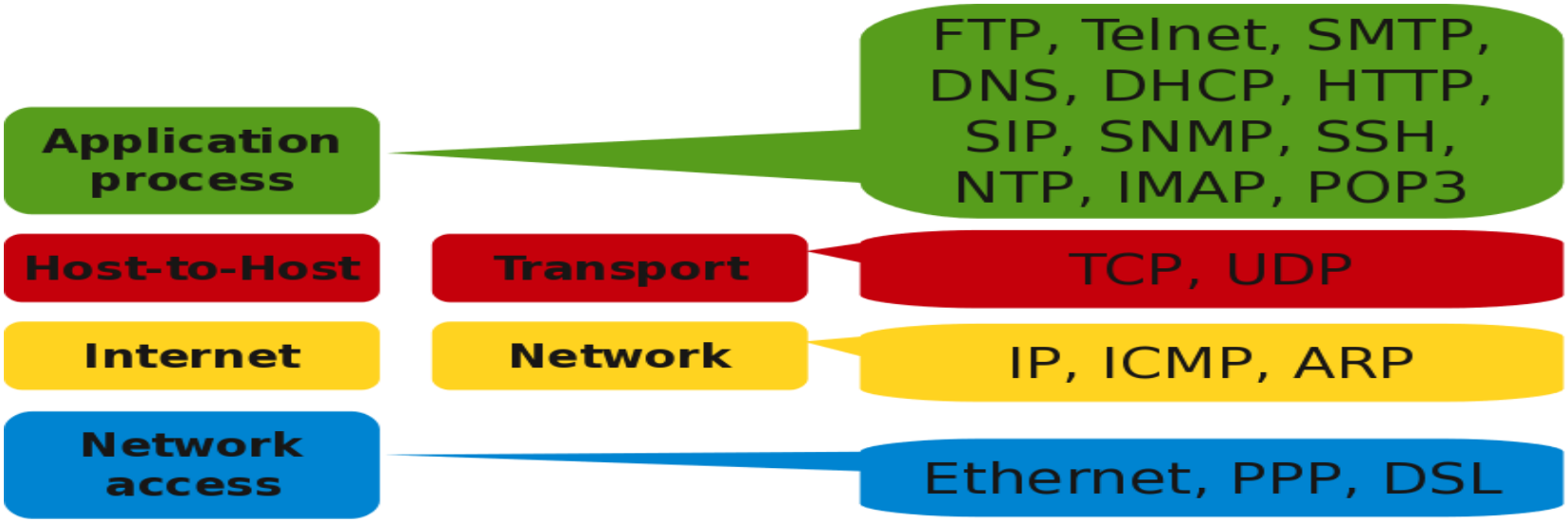
Modèle TCP/IP (3)

- **Couche Internet :**
  - Cette couche doit décapsuler l'en-tête du datagramme pour transmettre les données à la couche de transport et au bon protocole de cette couche (TCP, UDP).
  - Cette couche prend aussi en charge la communication de machine à machine.
  - Elle accepte des requêtes venant de la couche de transport avec une identification de la machine vers laquelle le paquet doit être envoyé.
  - Elle utilise alors l'algorithme de routage pour décider si le paquet doit être envoyé vers une passerelle ou vers une machine directement accessible.

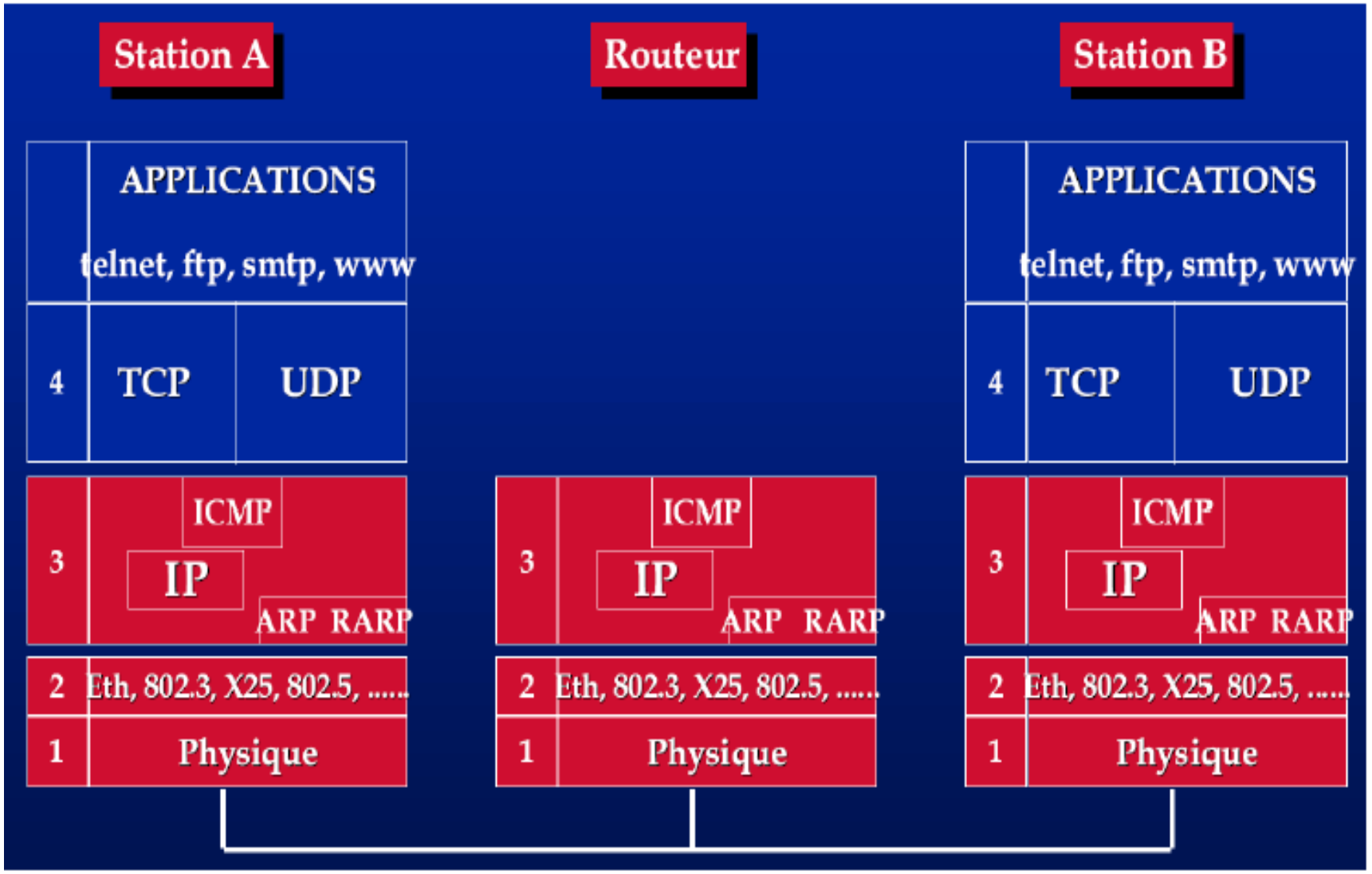
- **Couche Accès au réseau :**
  - Le protocole dans cette couche définit le moyen pour un système de délivrer l'information à un autre système physiquement relié. Il définit comment les datagrammes IP sont transmis. La définition de ceux-ci reste indépendante de la couche réseau, ce qui leur permet de s'adapter à chaque nouvelle technologie au fur et à mesure de leur apparition.

Encapsulation : Exemple





Modèle en couche : place des routeurs



IP : Internet Protocol (1) rfc792

- Il a été conçu en 1980 pour remplacer NCP (Network Control Protocol), le protocole de l'ARPANET.
- Les octets issus de la couche de transport et encapsulés à l'aide d'un en-tête IP avant d'être propagés vers la couche réseau (Ethernet par exemple), sont collectivement nommés « **datagramme IP** », datagramme Internet ou datagramme tout court.
- Ces datagrammes ont une taille maximale liée aux caractéristiques de propagation du support physique, c'est le « **M**aximum **T**ransfer **U**nit » ou MTU.

IP : Internet Protocol (2)

Fonctions assurées :

- **Adressage logique :** chaque machine du réseau possède une adresse unique afin de pouvoir acheminer les données.

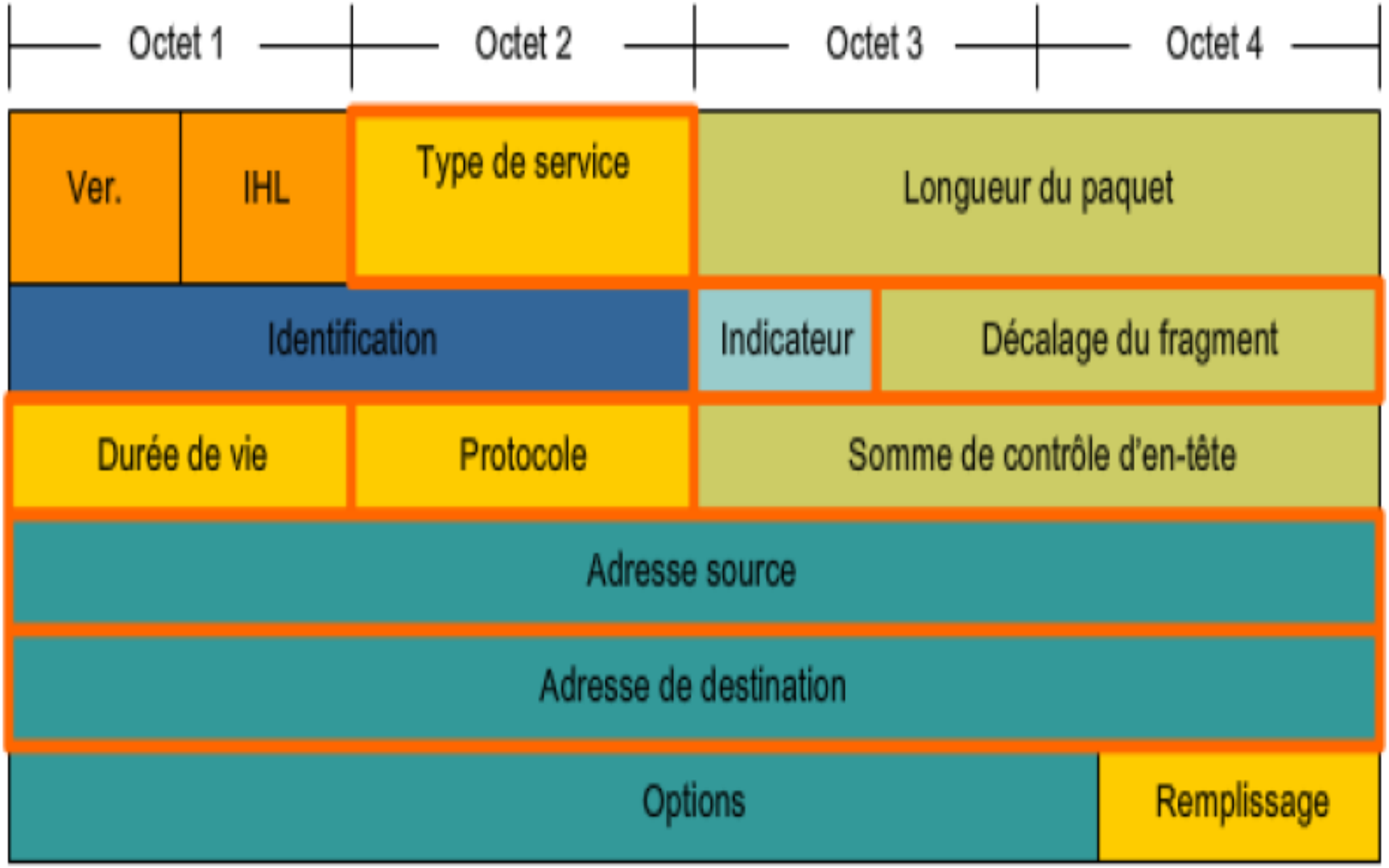
- **Constitution des datagrammes et routage**

  ⇒ sans connexion, ni contrôle d'erreur, ni contrôle de flux, ni remise en ordre des datagrammes… (pas de garantie de remise).

  ⇒ chaque datagramme est traité indépendamment des autres.

→ Ces services non assurés sont laissées *à la charge de la couche supérieure*

Datagramme IP



En-tête IP (1)

- **Version** : Numéro de version du protocole IP. (actuellement 4 : IPv4).
- **IHL : Longueur en-tête** : Longueur de l'entête en mots de 32 bits. Valeur habituelle = 5 : pas d'options.
- **TOS (Type of Service)** :Le champ de type de service contient une valeur binaire de 8 bits utilisée pour définir la priorité de chaque paquet. Cette valeur permet d'appliquer un mécanisme de qualité de service (QS) aux paquets de priorité élevée, tels que ceux transportant des données vocales de téléphonie. Le routeur traitant les paquets peut être configuré pour déterminer le paquet à transmettre en premier en fonction de la valeur de type de service.
- **TL : Longueur totale** : longueur totale des données et de l'entête. Taille maxi: 64ko.

En-tête IP (2)

- **Identification** : permet à l'ordinateur destinataire de déterminer à quel datagramme appartient le fragment reçu.
- **Indicateur:**

Indicateur de fragments supplémentaires :

L'indicateur de fragments supplémentaires **(MF)** est un seul bit du champ Indicateur utilisé avec le décalage du fragment pour la fragmentation et la reconstruction de paquets.

L'indicateur de fragments supplémentaires est défini, indiquant qu'il ne s'agit pas du dernier fragment d'un paquet. Quand un hôte récepteur voit un paquet arriver avec l'indicateur MF = 1, il examine le décalage du fragment pour voir où ce fragment doit être placé dans le paquet reconstruit. Quand un hôte récepteur reçoit une trame avec l'indicateur MF = 0 et une valeur non nulle dans le champ de décalage du fragment, il place ce fragment à la fin du paquet reconstruit. Les informations de fragmentation d'un paquet non fragmenté sont toutes nulles (MF = 0, décalage du fragment = 0).

Indicateur Ne pas fragmenter :

L'indicateur Ne pas fragmenter **(DF)** est un seul bit du champ Indicateur stipulant que la fragmentation du paquet n'est pas autorisée. Si le bit de l'indicateur Ne pas fragmenter est défini, la fragmentation de ce paquet n'est PAS autorisée. Si un routeur doit fragmenter un paquet pour permettre sa transmission descendante à la couche liaison de données mais que le bit DF est défini à 1, le routeur supprime ce paquet.

- **Déplacement Fragment:**
- Comme mentionné précédemment, un routeur peut devoir fragmenter un paquet lors de sa transmission d'un média à un autre de MTU inférieure. Lorsqu'une fragmentation se produit, le paquet IPv4 utilise le champ de décalage du fragment et l'indicateur MF de l'en-tête IP pour reconstruire le paquet à son arrivée sur l'hôte de destination. Le champ de décalage du fragment identifie l'ordre dans lequel placer le fragment de paquet dans la reconstruction.

En-tête IP (3)

- **Durée de vie:** La durée de vie (TTL, Time to live) est une valeur binaire de 8 bits indiquant la durée de vie restante du paquet. La valeur TTL est décrémentée de 1 au moins chaque fois que le paquet est traité par un routeur (c'est-à-dire à chaque saut). Lorsque la valeur devient nulle, le routeur supprime ou abandonne le paquet et il est retiré du flux de données du réseau.
- **Protocole** : protocole de la couche supérieure qui traite le datagramme assemblé (UDP(17) ou TCP (6) ou ICMP(1))
- **Somme Contrôle d'entête :** 16 bits pour s'assurer de l'intégrité de l'en-tête. Lors du calcul de ce 'checksum' ce champ est à 0. A la réception de chaque paquet, la couche calcule cette valeur, si elle ne correspond pas à celle trouvée dans l'en-tête le datagramme est oublié (discard) sans message d'erreur.
- **Adresses sources et destination** : adresses IP (couple numéro réseau : numéro ordinateur sur le réseau)
- **Options** : des champs supplémentaires sont prévus dans l'en-tête IPv4 afin de fournir d'autres services, mais ils sont rarement utilisés.

ICMP : Internet Control Message Protocol rfc792

- Protocole de 'gestion' de réseau
- Implémenté sur tous les équipements IP (stations, routeurs)
- Les ICMP sont les messages d'incident de réseaux.
- Message envoyé par l'équipement destinataire ou un routeur intermédiaire : Quand il s'aperçoit d'un problème dans un datagramme pour avertir l'émetteur afin qu'il modifie son comportement -> mauvais routages, contrôle de flux
- Les commandes ***PING*** et ***TRACEROUTE*** s'appuient sur les ICMP
- Message ICMP contenu dans un datagramme IP

Exemple : demande d'écho : utilisé par `ping`

```
# ping 216.239.59.104
PING 216.239.59.104 (216.239.59.104) 56(84) bytes of data.
64 bytes from 216.239.59.104 : icmp_seq=1 ttl=56 time=36.6 ms
64 bytes from 216.239.59.104 : icmp_seq=2 ttl=56 time=36.3 ms
```

ICMP : Gestion des erreurs

- **Message ICMP encapsulé par IP :**

<table>
<tr><td rowspan="2">En-tête IP</td><td colspan="4">Message ICMP</td></tr>
<tr><td>Type (8 bits)</td><td>Code (8 bits)</td><td>Checksum (16 bits)</td><td>Message (taille variable)</td></tr>
</table>

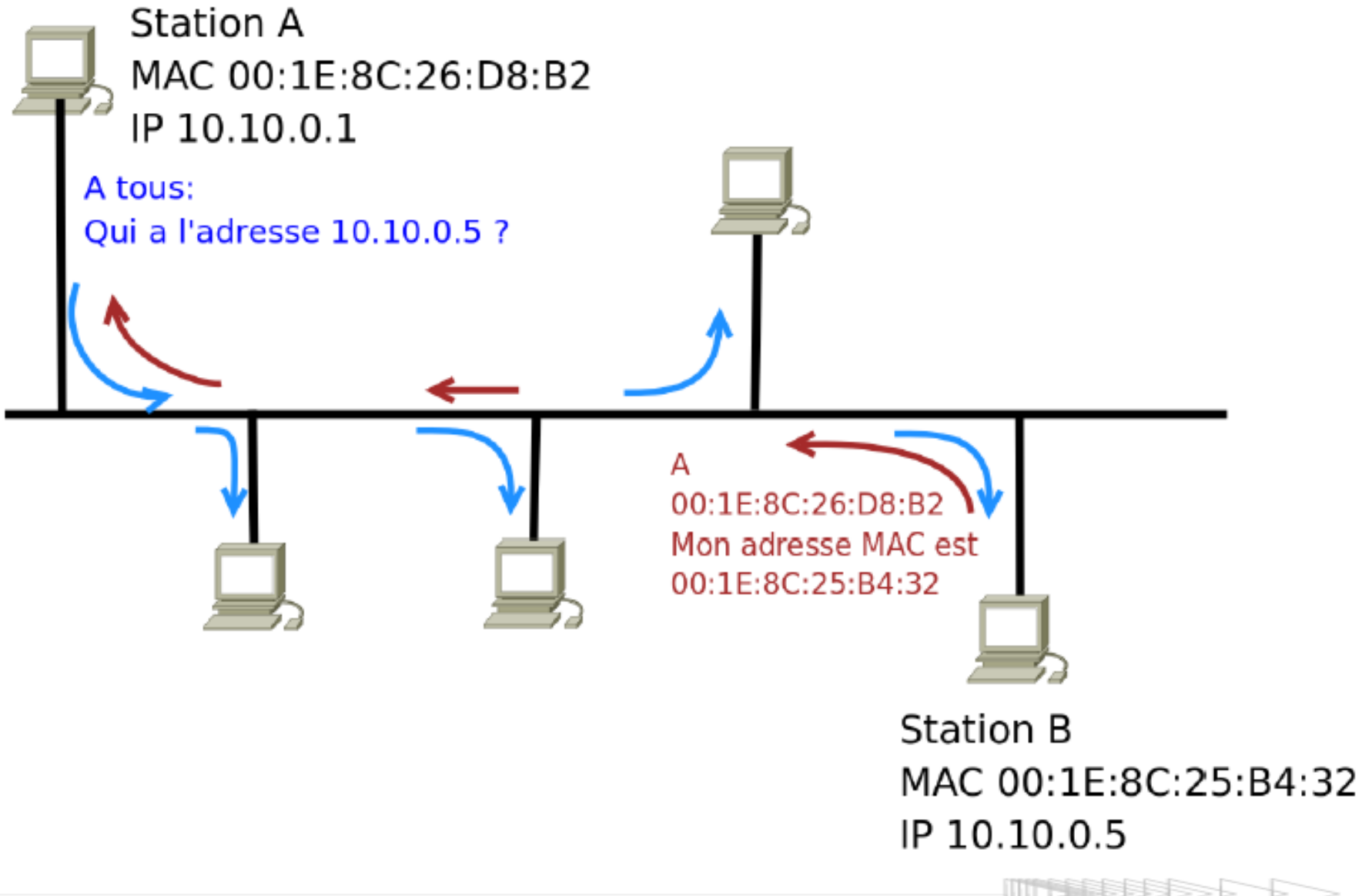
- **Exemples de messages ICMP :**

Type	Code	Message	Signification du message
8	0	Demande d'écho	Utilisé par **ping** pour tester la présence d'une machine
3	0	Destinataire inaccessible	Réseau inaccessible
3	7	Destinataire inaccessible	Machine inconnue, un routeur ne parvient pas à localiser la destination, problème de fragmentation
5	0	Redirection pour un hôte	Route non optimale, indique modification dans la table de routage
11	1	Temps dépassé	Durée de vie d'un datagramme dépassée (TTL a atteint 0), en-tête retourné
12	0	En-tête erroné	Erreur détectée dans l'en-tête, position de l'erreur retournée

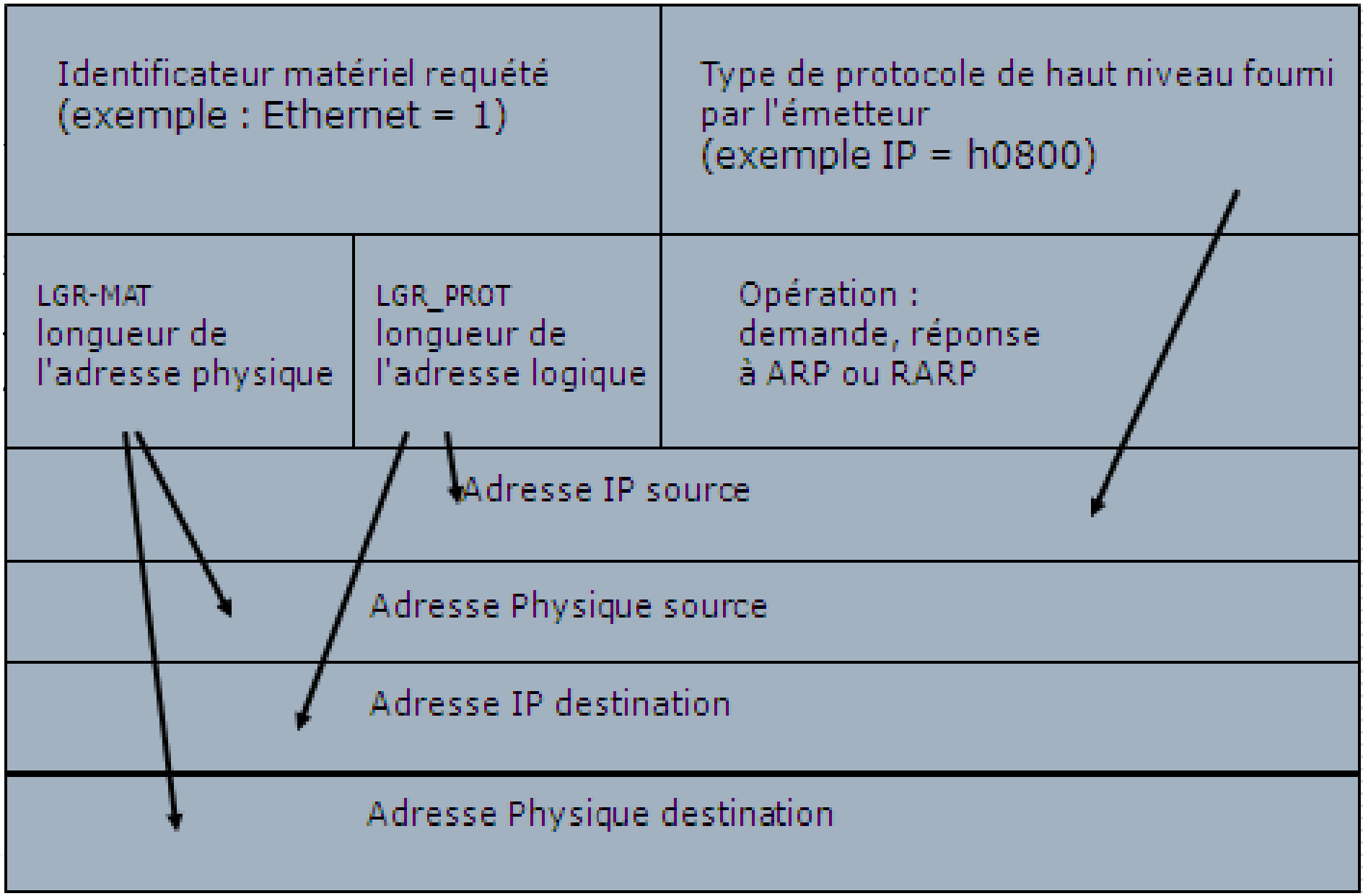
ARP : Address Resolution Protocol *RFC 826*

- Protocole permettant d'obtenir l'adresse Ethernet (adresse physique) d'un équipement à partir de l'adresse IP.
- Effectuer cette correspondance est une résolution d'adresse.
- Cette résolution est dynamique (ne nécessite pas l'intervention d'un opérateur).
- L'adresse Ethernet (adresse physique) est inscrite matériellement sur la carte réseau. Par contre, l'adresse IP est une adresse stockée en mémoire.
- L'adresse IP est totalement indépendante de l'adresse physique (MAC)
- Stockage des adresses MAC dans une table locale : cache ARP

Protocole ARP



ARP : Fonctionnement



ARP : Fonctionnement

```
C:\>arp -a

Interface : 193.54.41.42 on Interface 2
  Adresse Internet      Adresse physique      Type
  193.54.41.33          00-a0-c9-4c-f6-e8     dynamique
  193.54.41.43          00-c0-4f-7d-38-9e     dynamique
  193.54.41.53          00-c0-4f-a4-f1-0a     dynamique
  193.54.41.252         00-60-97-91-68-8f     dynamique
  193.54.41.254         00-d0-d3-33-8f-8c     dynamique
```

- arp –a
  Toutes les entrées du cache
- arp – s 192.14.25.56 00-80-C7-E0-7E-C5
  Entrer une nouvelle adresse IP / MAC
- arp –d 192.14.25.56
  Supprime une adresse IP du cache

RARP : Reverse ARP *RFC 2663*

- Mécanisme permettant à la station d'obtenir son adresse IP depuis le réseau.
- Permet d'obtenir son adresse IP à partir de l'adresse physique qui lui est associée.
- Comme pour ARP, une trame de diffusion Ethernet est émise, contenant une requête RARP.
- Requête : "*Quelle est l'adresse IP correspondant à mon adresse Ethernet ?*".
- On utilise un serveur RARP sur le réseau physique qui fournit les adresses IP associées aux adresses physiques des stations du réseau. Il envoie une réponse en unicast.

Rem 1 : Possible que plusieurs serveurs RARP existent d'où la génération de plusieurs réponses à la requête. La première réponse est considérée.

Rem 2 : Une requête RARP ne peut traverser un routeur. Dans le cas où aucun serveur RARP n'existe sur le réseau physique Ethernet, la requête n'est pas satisfaite.

Couche Transport

Deux protocoles pour la communication entre applications :

- **TCP** : **T**ransmission **C**ontrol **P**rotocol

communication avec connexion, fiable.

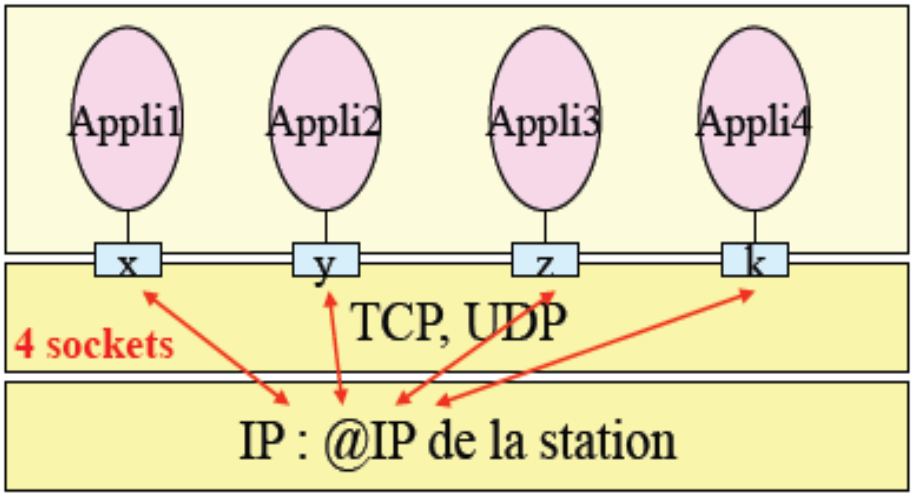
- **UDP** : **U**ser **D**atagram **P**rotocol

communication sans connexion, non fiable.

TCP/UDP : Notion de port

- Permet à plusieurs processus applicatifs de communiquer simultanément sur le réseau, potentiellement avec un même protocole de transport (TCP ou UDP).
- Le port sert d'interface entre le processus applicatif et la couche transport.

N° des ports codés sur 16 bits: 0-> 65535



ports

N°port + @IP = socket

- Pour connaitre les sockets ouvertes et utilisées sur sa machine : commande **netstat**.

TCP & UDP : Notion de port (2)

Exemple :

`216.239.59.104:80` est le serveur web (port 80)
sur la machine `216.239.59.104`

Le couple de deux sockets définit complètement une connexion :

Exemple

`121.89.76.203:1094` ⟷ `216.239.59.104:80`
communication entre un client `121.89.76.203` qui a pris le port numéro 1094 et le serveur sur port 80. Voir commande `netstat -a`

TCP/UDP : Notion de port

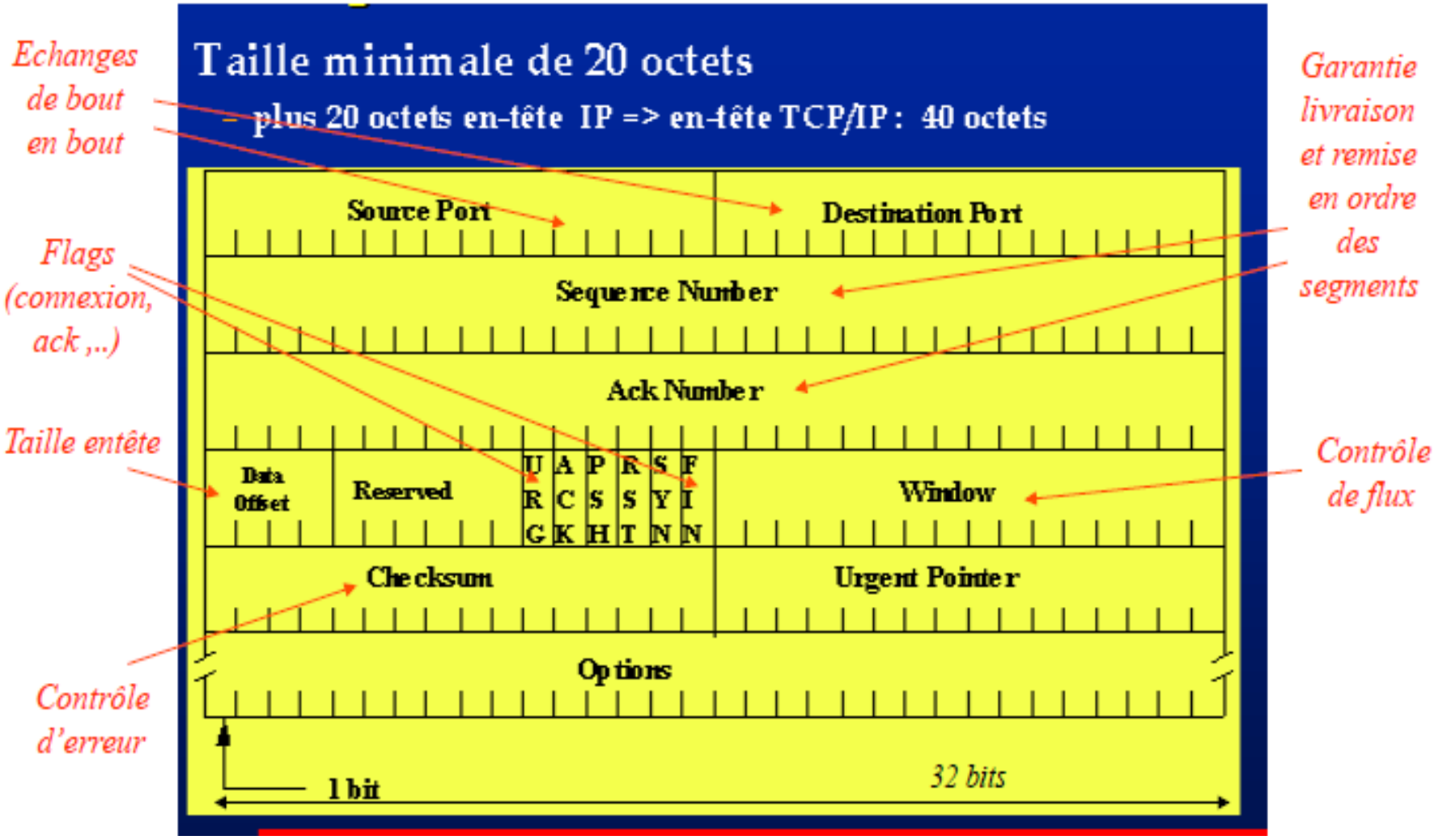
- Attribution des numéros de Port :
  - internationalement « assignés » par l'IANA **< 1023**
  - dynamiquement alloués par l'OS si non spécifiés
  - configurables directement pour certaines applications
- Lors des communications, les infos telles que les n° de ports sont envoyées dans l'entête afin de désigner quelle connexion est concernée.

N° port	Service ou application
21	FTP
23	Telnet
25	SMTP
53	DNS
80	HTTP
110	POP3
443	HTTPS

TCP : Transport Control Protocol *rfc 793*

- TCP ne tourne pas dans les routeurs, uniquement aux extrémités.
- Un protocole de bout en bout entre applications orienté connexion.
  - **Protocole de bout en bout**. Les processus pairs des couches transport de 2 équipements connectés dialoguent l'un avec l'autre sans rien connaître du réseau. C'est au niveau IP que l'on se préoccupe de la fragmentation et du réassemblage des segments TCP.
  - **Protocole orienté connexion.** La fiabilité du transport TCP dépend de l'établissement d'une connexion entre les processus pairs qui veulent dialoguer. L'établissement d'une connexion est réalisé par l'échange d'informations telles que le numéro de port, le numéro de séquence et la taille de fenêtre.
- Services offerts par TCP :
  - en mode connecté (ouverture, fermeture : circuit virtuel)
  - sans erreur : contrôle et retransmission si besoin
  - sans perte : numérotation et retransmission
  - avec contrôle de flux (fenêtre d'émission)
  - full duplex
  - indication du service par le numéro de port

En-tête d'un segment TCP



En-tête d'un segment TCP (2)

- **Source Port (16 bits) :** Numéro du port source.
- **Destination Port (16 bits) :** Numéro du port destination.
- **Sequence Number (32 bits) :**
  - Si **SYN = 0**, le numéro de séquence est celui du premier octet de données de ce segment.
  - Si **SYN = 1**, il s'agit du numéro de séquence initial ISN. le premier octet de donnée est à ISN+1.
- **Acknowledgment Number (32 bits) :**
  - **Si le bit ACK = 1, ce champ contient le numéro de séquence attendu par l'émetteur du segment.**
- **Data Offset (4 bits) :** Taille de l'en-tête TCP en mots de 32 bits.
- **Reserved (6 bits) :** Champ réservé pour une utilisation ultérieure. Les 6 bits doivent être à 0.

En-tête d'un segment TCP(3)

- **Control bits (6 bits)**
  - **URG :** Urgent Pointer field significant
  - **ACK :** Acknowledgment field significant
  - **PSH :** Push Function
  - **RST :** Reset the connection
  - **SYN :** Synchronize sequence numbers
  - **FIN :** No more data from sender
- **Window (16 bits) :** Nombre d'octets de données à partir de celui indiqué par le champ Acknowledgment.
- **Checksum (16 bits) :** Somme de contrôle sur 16 bits de l'en-tête et des données.
- **Urgent Pointer (16 bits) :** Ce champ est interprété uniquement si le bit de contrôle **URG** est à **1**. Le pointeur donne le numéro de séquence de l'octet qui suit les données « urgentes ».
- **Options :** variable il existe 2 formats d'options : un seul octet de catégorie d'option ou un octet de catégorie d'option suivi d'un octet de longueur d'option et de l'octet des données de l'option.

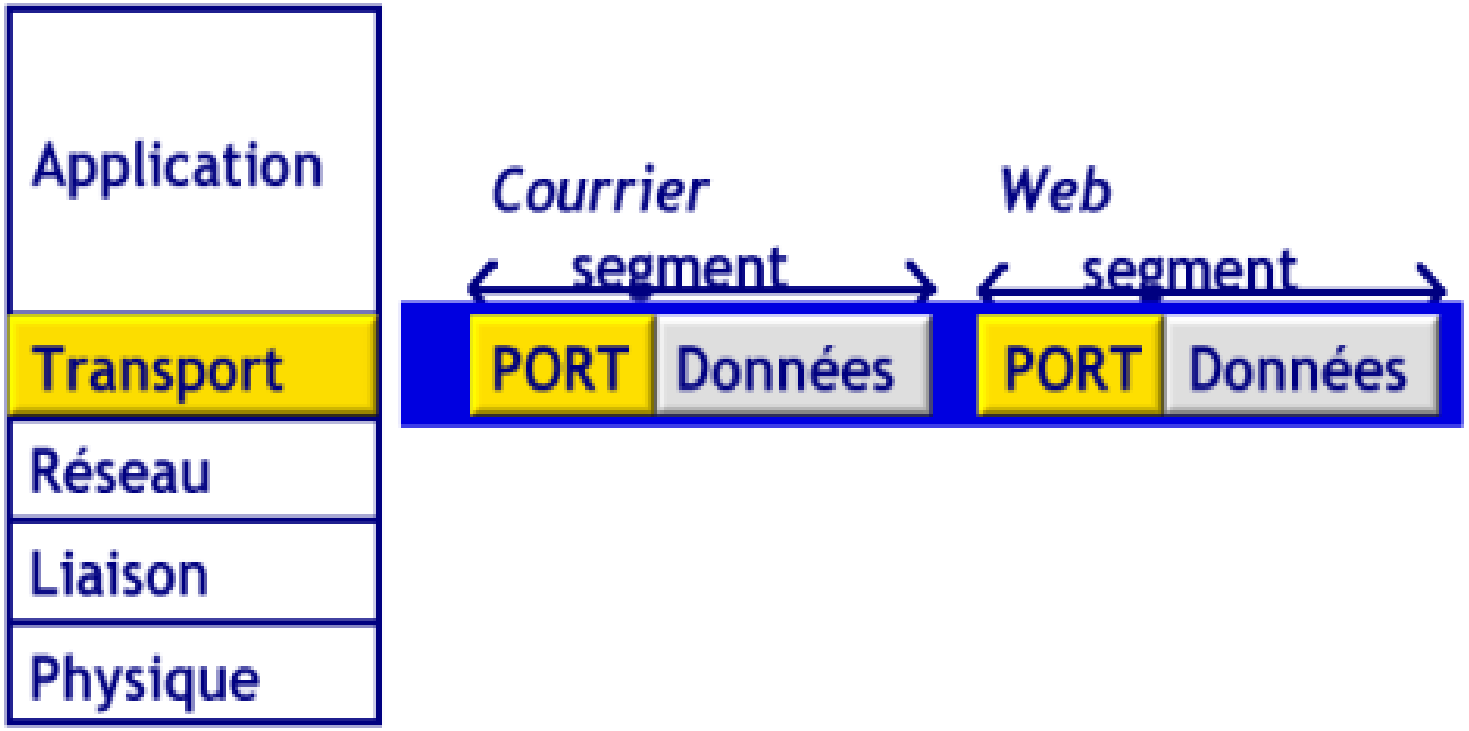
Le fonctionnement du protocole TCP

Le protocole TCP de la couche transport :

1. segmente les données des applications,
2. établit une connexion de bout-en-bout,
3. émet les segments d'un hôte à l'autre,
4. assure la fiabilité du transport des segments entre les hôtes connectés.

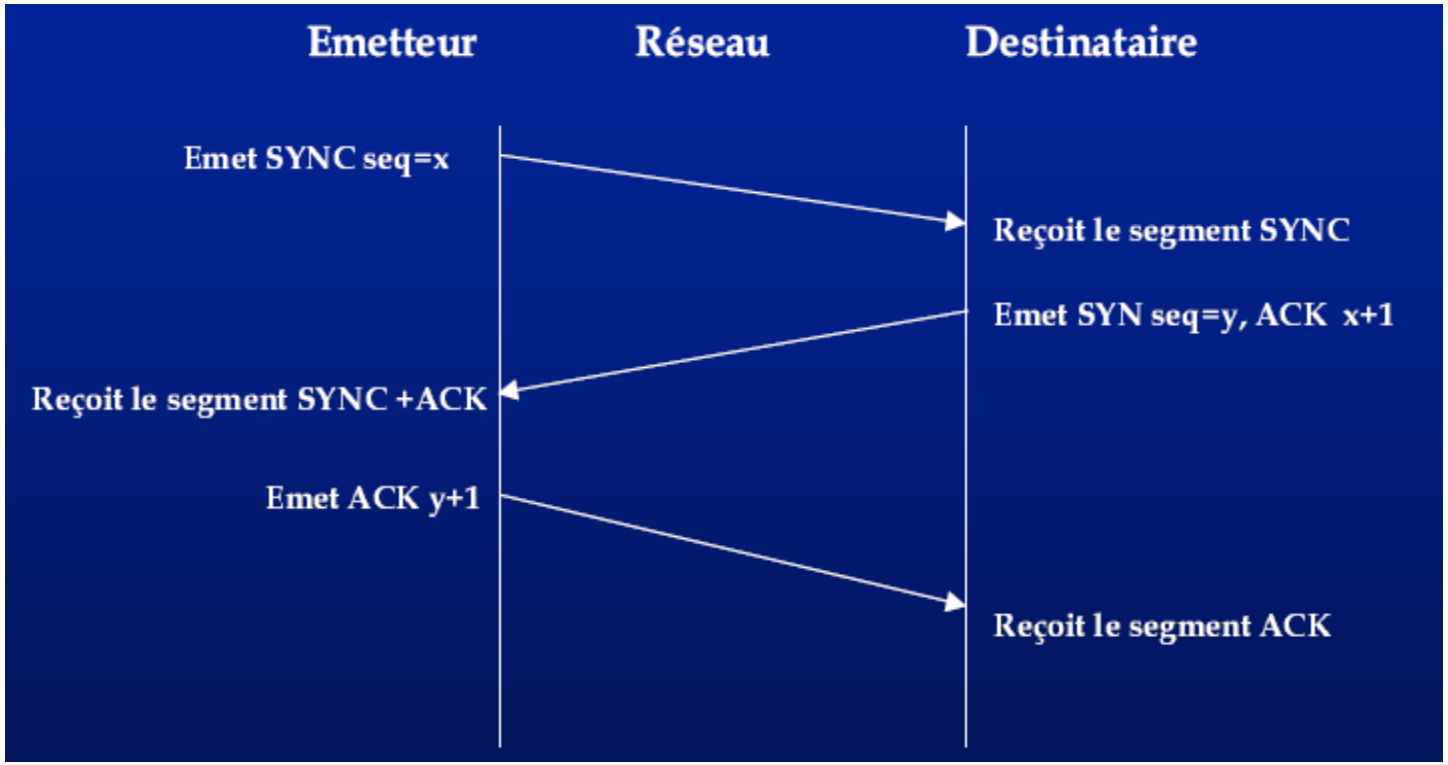
1. La segmentation des données

- La fonction de transport est effectuée segment par segment de façon autonome.
- Plusieurs applications peuvent émettre leurs segments successivement.
- Ces segments peuvent avoir un ou plusieurs destinataires.



2. L'établissement de la connexion

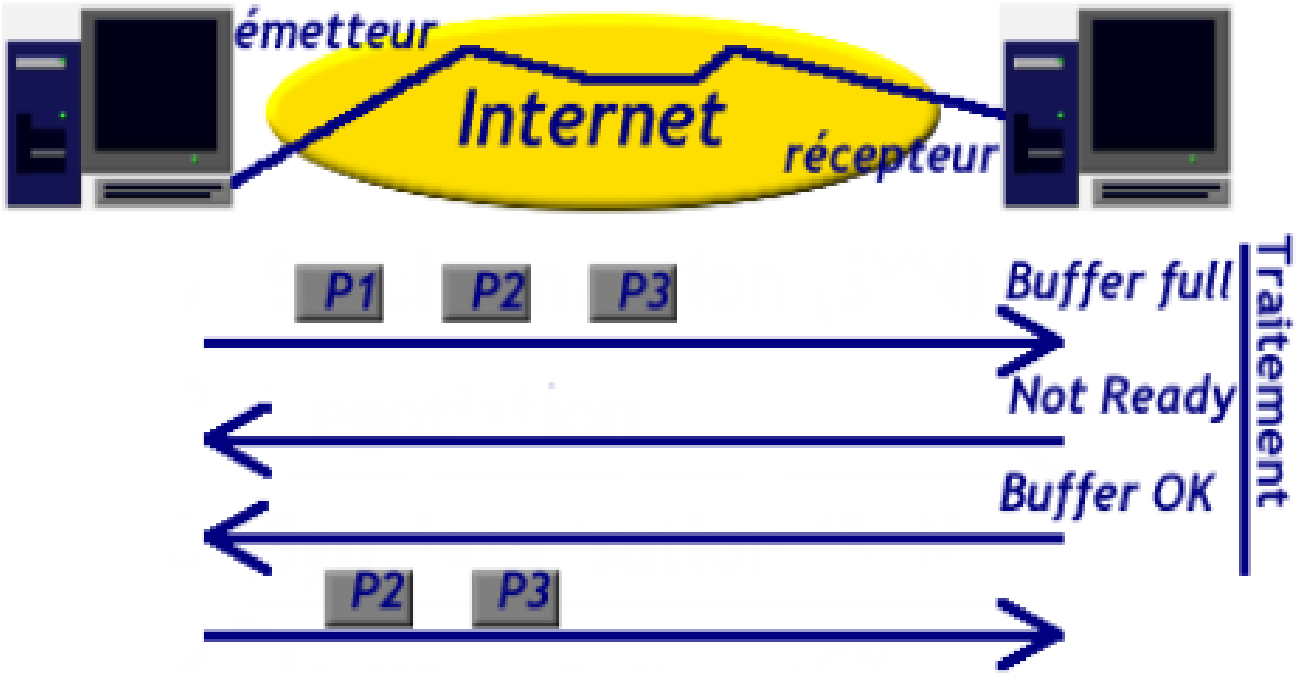
- Les étapes de l'établissement de la connexion sont :



où x et y sont les numéros de séquence initiaux.

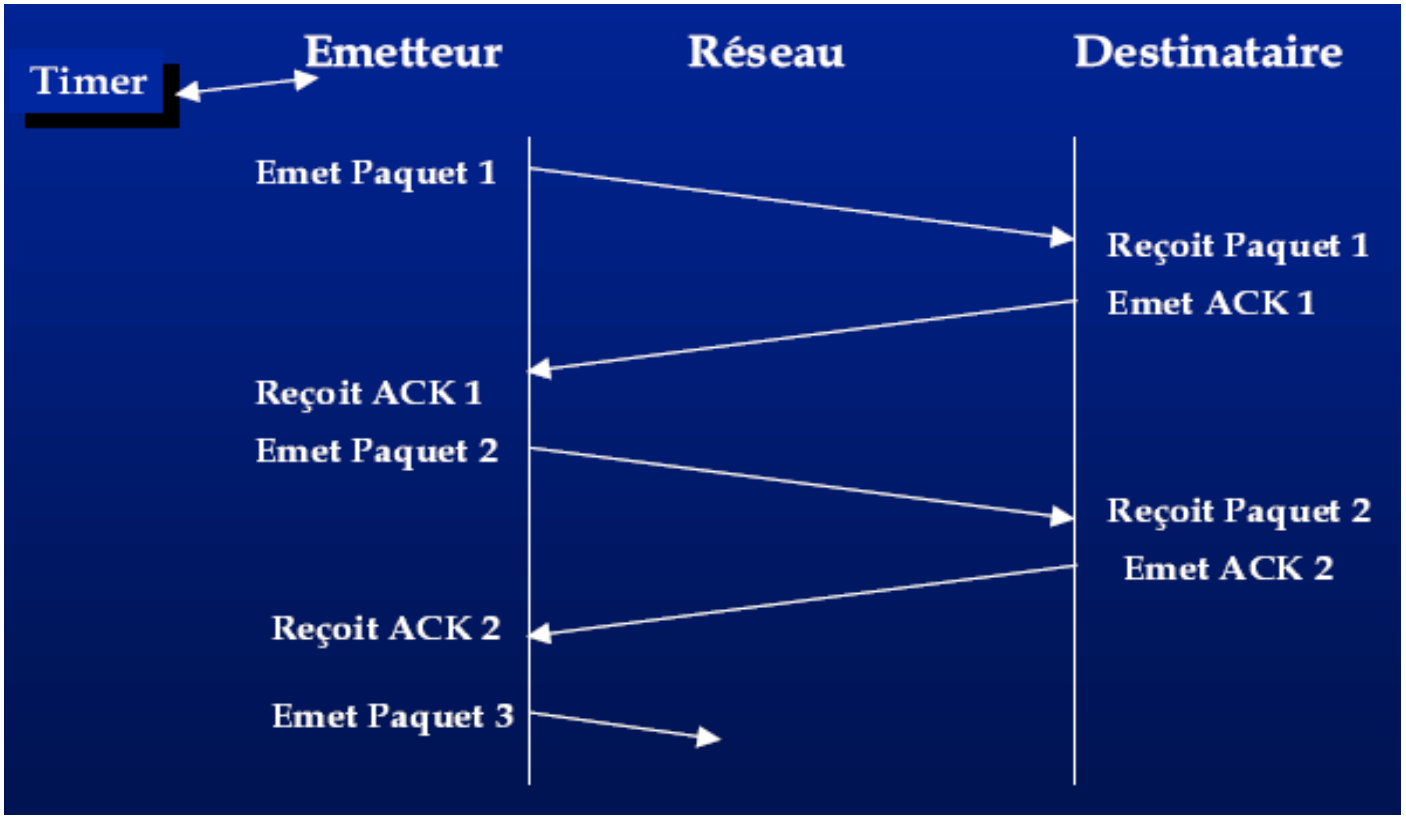
3. Contrôle de flux

- Un contrôle de flux est nécessaire pour prévenir la congestion des transferts :
  - Un serveur performant peut générer plus de trafic que le réseau ne peut en supporter.
  - Un serveur peut être sollicité par un nombre très élevé de clients.



Fiabilité d'un service réseau

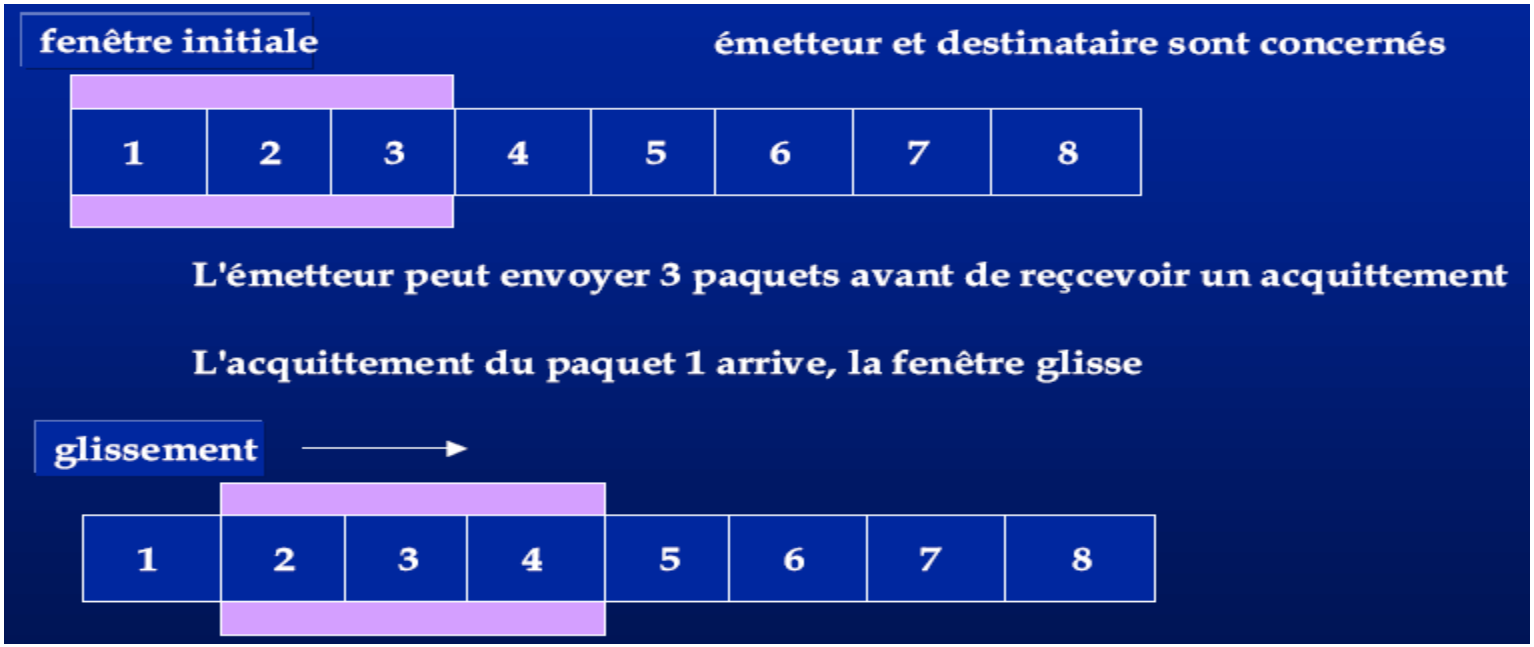
Un Mécanisme naïf : "**Send and Wait**" : on transmet un segment puis on attend l'acquittement avant de transmettre le suivant.



☹ Mauvaise efficacité !!!!

Fiabilité d'un service réseau

- Mécanisme de fenêtre glissante :



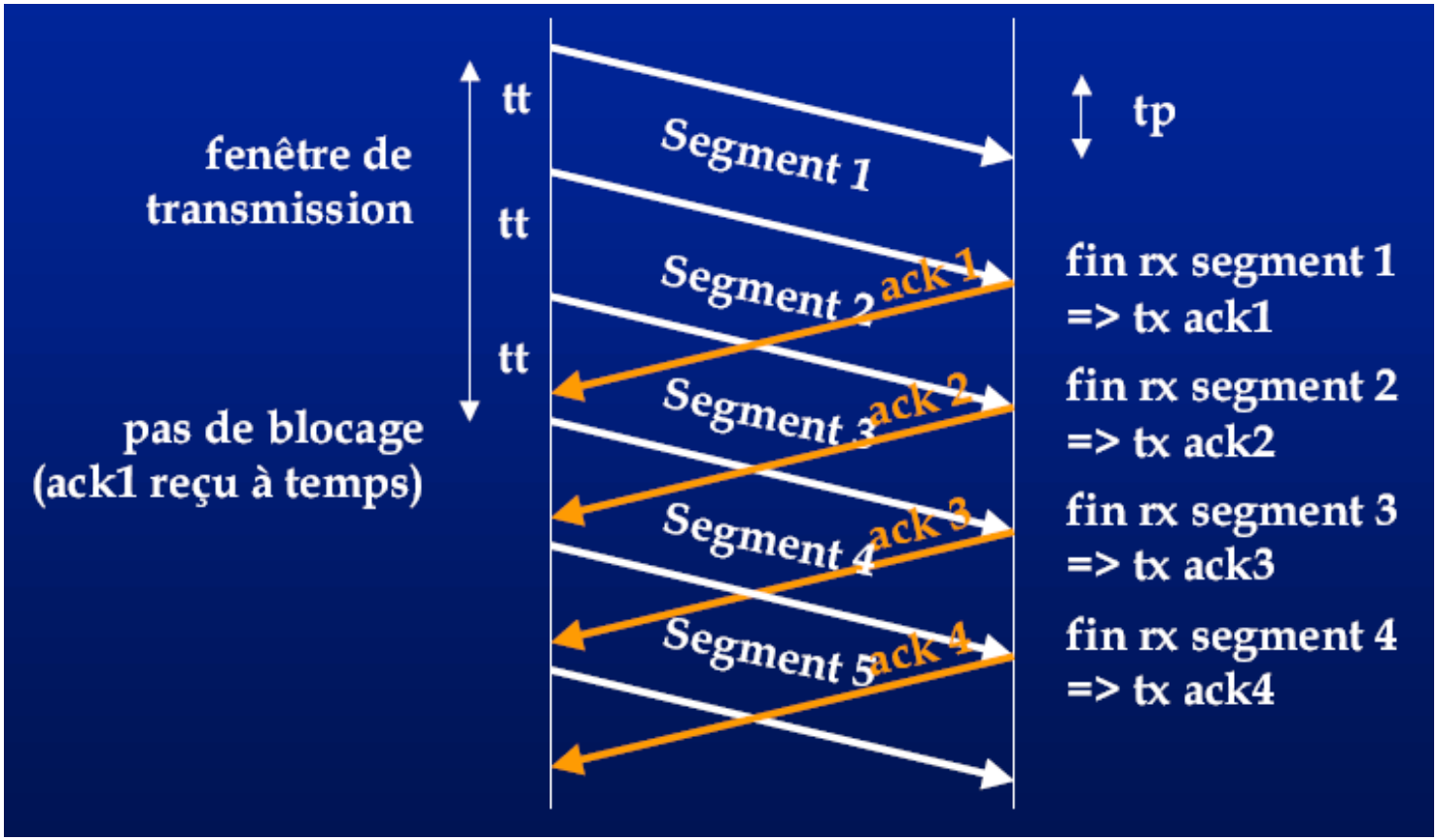
- Les performances sont fonction de la taille de la fenêtre et de la vitesse à laquelle le réseau accepte les paquets.

Fiabilité d'un service réseau

- Utilisation de fenêtres glissantes, dont la taille est adaptée en fonction des conditions (pertes observées).
- Le récepteur peut indiquer à l'émetteur la quantité de données qu'il peut accepter à un moment donné (contrôle de flux).
- Bonne efficacité et contrôle de flux permettant de s'adapter à la grande variabilité des conditions observées sur l'Internet.

4. Fenêtrage

Si la fenêtre a une taille suffisante, il n'y a pas de blocage !

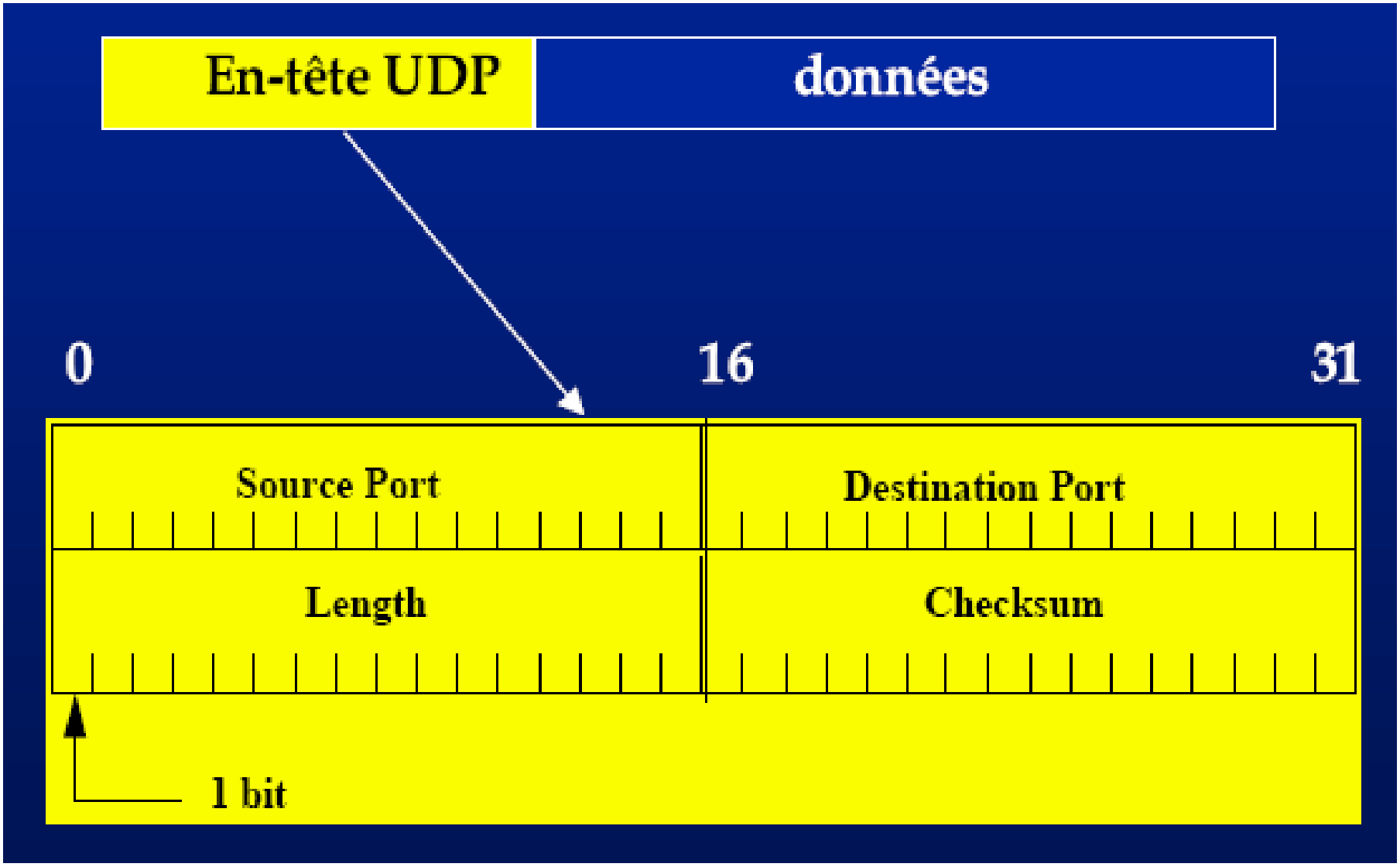


UDP : User Datagram Protocol *rfc 768*

Services offerts :

- Un Transfert de données non fiable (ne garantit ni la remise ni l'ordre des datagrammes délivrés),
- Service de datagrammes sans connexion (ne fournit pas établissement de connexion),
- Pas de contrôle de flux,
- Pas de contrôle de congestion,
- Les sommes de contrôle des données sont facultatives dans le protocole UDP.

En-tête UDP (1)


En-tête UDP
données
0
16
31
Source Port
Destination Port
Length
Checksum
1 bit

En-tête UDP (2)

- **Source Port (16 bits) :** Numéro du port source. Ce champ est optionnel.
- **Destination Port (16 bits) :** Numéro du port destination.
- **Length (16 bits) :** Longueur en octets du datagramme UDP incluant l'en-tête et les données.
- **Checksum (16 bits) :** Somme de contrôle sur 16 bits de l'en-tête et des données.

Quelle couche de transport pour quelle application?

Application	Protocole de couche Applicative	Protocole de transport
E-mail	SMTP [RFC 2821] pop	TCP
Accès terminal distant	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
Transfert de fichiers	FTP [RFC 959]	TCP
Streaming multimédia Dhcp dns	Propriétaire (e.g. YouTube)	TCP ou UDP Udp udp
Téléphonie Internet	Propriétaire (e.g. Skype)	UDP

Protocoles de la couche application

- Les protocoles de couche application TCP/IP les plus connus sont ceux permettant l'échange d'informations entre les utilisateurs. Ces protocoles spécifient les informations de