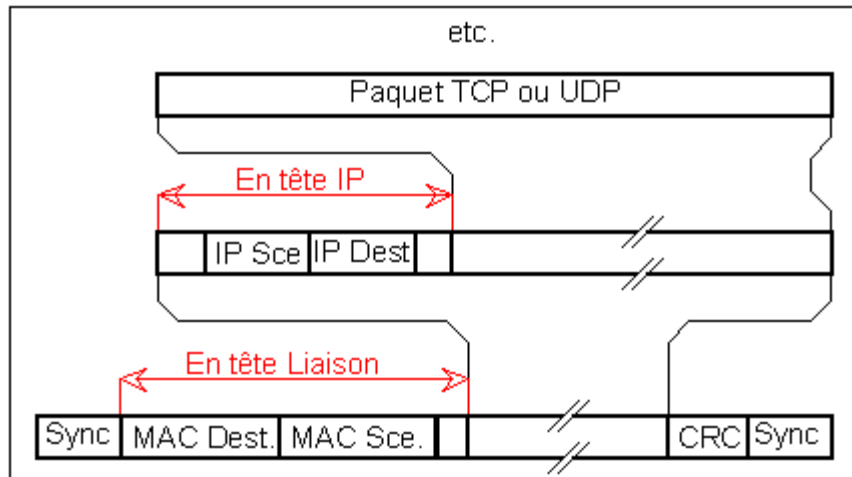




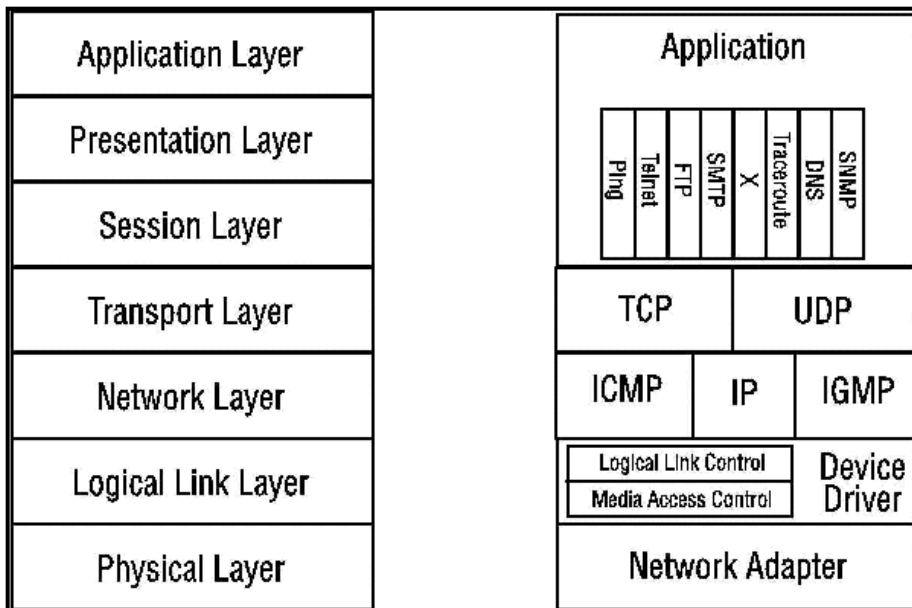
**INSTITUT DE NOUVELLES TECHNOLOGIES**  
**معهد التكنولوجيات الحديثة**

***Présentation***  
***de l'architecture de la suite de protocole TCP / IP***  
**Couche Transport**  
**Protocole TCP UDP**

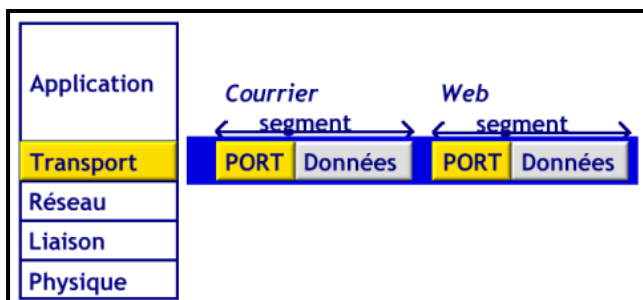
## La couche transport



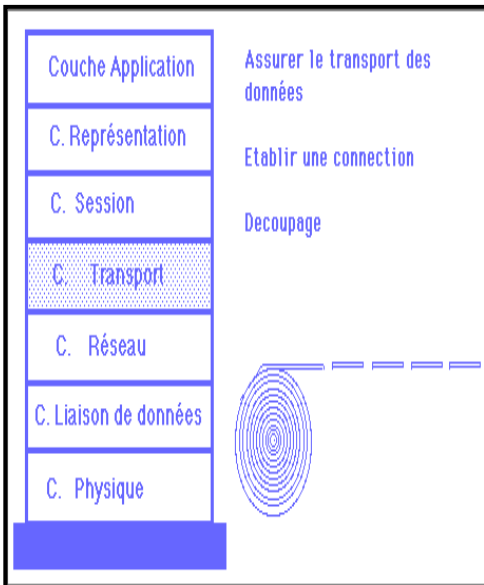
## Les protocoles de la couche transport



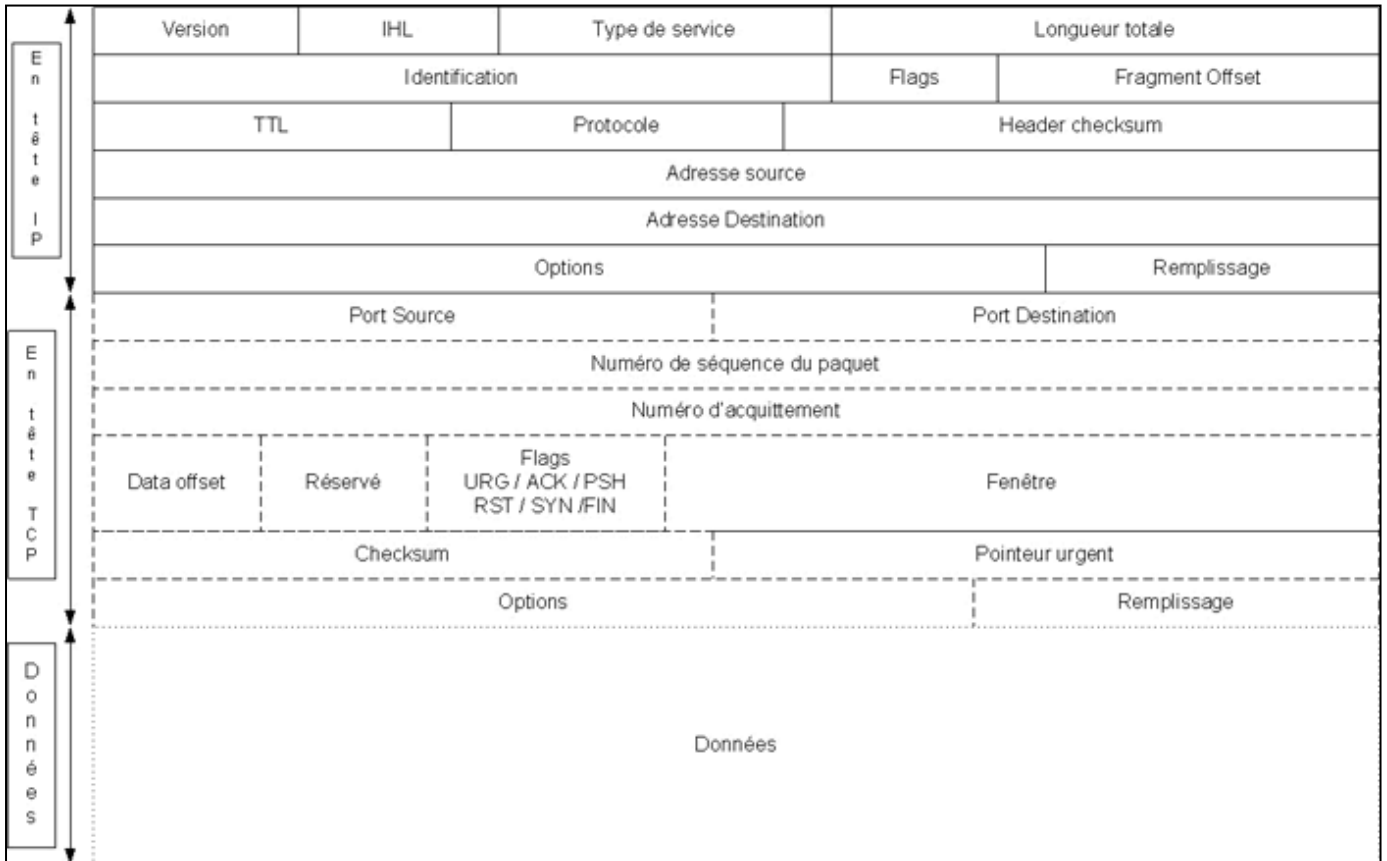
## La segmentation des données



## Couche transport : transfert des messages de bout en bout



## TCP



# TCP: Survol

RFCs: 793, 1122, 1323, 2018, 2581

- **point-à-point:**
  - Un émetteur, un récepteur
- **Fiable, réception dans l'ordre du flot d'octet:**
  - Pas de « frontières de message »
- **Fenêtre d'émission:**
  - Les mécanismes de contrôle de flot et de congestion règlent la taille de la fenêtre
- **Tampons de réception et d'émission**
- **Transfert bidirectionnel des données:**
  - Transfert bi-directionnel de données sur une connexion
  - MSS: maximum segment size
- **Orienté connexion:**
  - handshaking (échange de msgs de control) initialise l'état de l'émetteur et du récepteur avant l'émission de données
- **Contrôle de flot :**
  - L'émetteur ne submerge pas le récepteur



2: Application Layer 8

## Le protocole TCP

- **Notion de port (source et destination)**
  - telnet 23/tcp, smtp 25/tcp, http 80/tcp
- **Notion de session**
  - retransmission des paquets non acquittés
  - n° de séquence unique pour chaque paquet TCP
  - sens d'établissement de la connexion
    - Notion de flag (SYN, ACK, FIN, RST, ...)

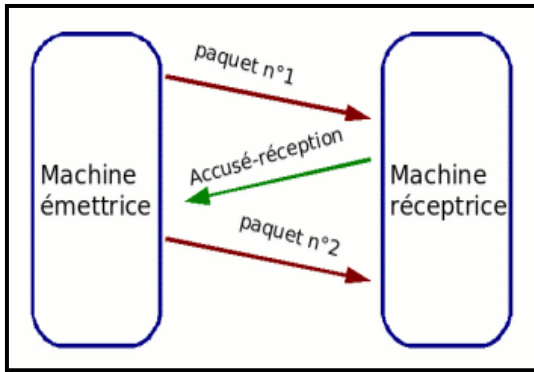


HSC  
0

Stéphane AUBERT © Hervé Schauer Consultants

- 16 -

## L'établissement de la connexion




---

---

---

---

---

---

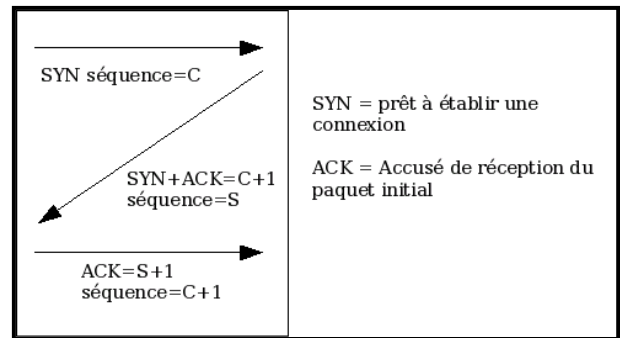
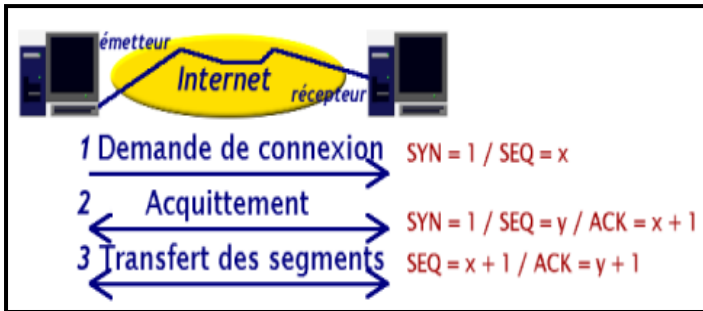


---

---

---

---




---

---

---

---

---

---

---

---

## Flags TCP

---

---

---

---

---

---

---

---

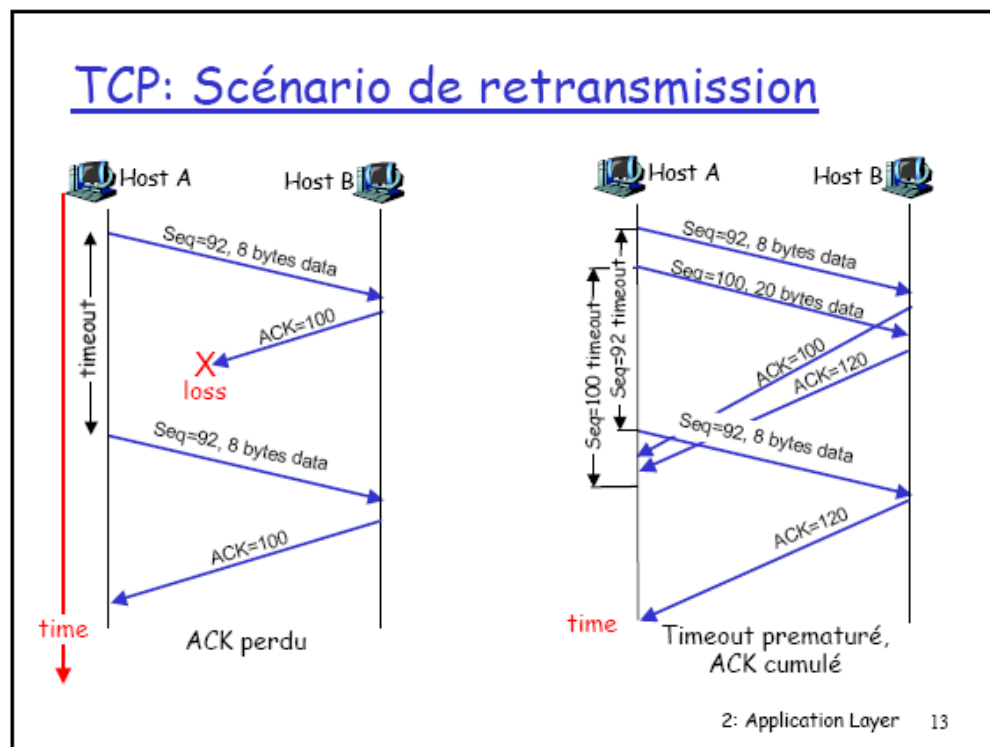
## Les drapeaux (flag) TCP

- **SYN** : pour établir une connexion
- **ACK** : pour acquitter un paquet
- **RST** : pour détruire une session
- **FIN** : pour mettre fin à une session
- **PSH** : interdire le stockage temporaire
- **URG** : pour indiquer une urgence

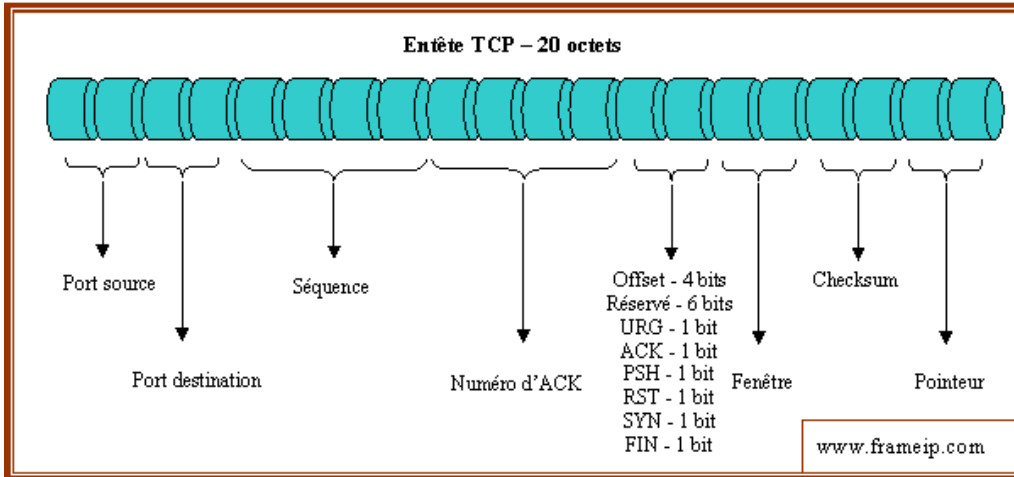
### Exemple :



## Retransmission TCP



## Les champs TCP



## Port et socket

Attribution des ports ``ancienne méthode/nouvelle méthode''

## TCP: Transfert fiable de données

**evenement:** réception de données de l'application  
Créer un segment



**evenementt:** ACK reçu, avec ACK # y  
Traitement du ACK

### Émetteur simplifié

- Transmission dans un sens
- Pas de contrôle de flot et de congestion

## TCP: Gestion de la Connexion

- « L'émetteur et le récepteur doivent initier une « connexion TCP » avant d'échanger des données
- « Initialiser les variables TCP:
  - « seq. #s
  - « Tampons, information de contrôle de flot (e.g. RcvWindow)
- « *client*: initiateur de la connexion

```
Socket clientSocket = new
Socket("hostname", "port
number");
```
- « *server*: contacté par le client

```
Socket connectionSocket =
welcomeSocket.accept();
```

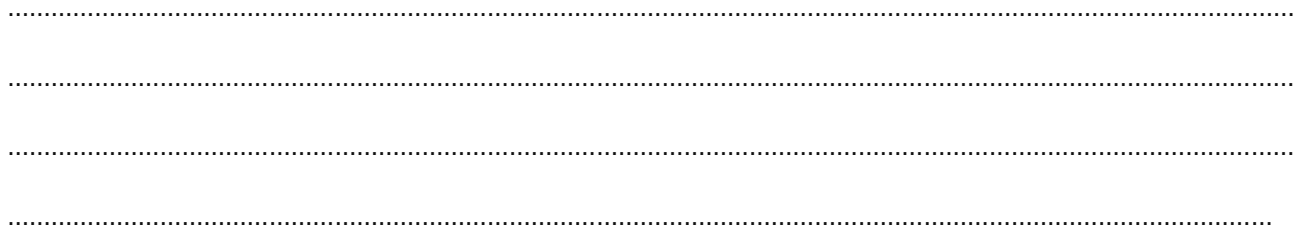
### Poignée de main tripartite:

- « **1**: le client envoie un segment de contrôle TCP SYN
  - « Défini le seq # initial
- « **2**: le serveur reçoit le SYN, et répond par un segment de contrôle SYNACK
  - « ACKs le SYN reçu
  - « alloue des tampons
  - « spécifie le seq.# initial du serveur-> récepteur

2: Application Layer 17



## Contrôle de flux



### Contrôle de Flot TCP

#### Contrôle de flot

L'émetteur ne submerge pas le récepteur en émettant trop rapidement

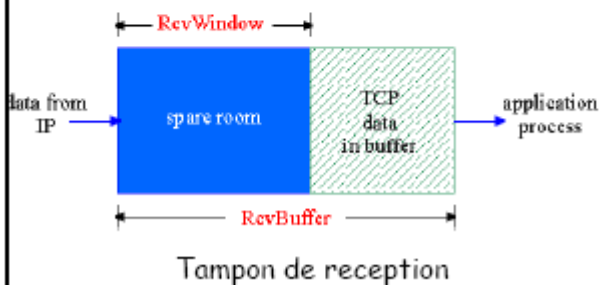
**Recepteur:** informe explicitement l'émetteur de l'espace vide (qui change dynamiquement)

✦ champs **RcvWindow**

dans l'entête TCP

**émetteur:** borne la taille des données transmises et non acquittées à **RcvWindow**

RcvBuffer = Taille du tampon de réception TCP  
RcvWindow = espace vide dans le tampon



2: Application Layer 14

### Principes du contrôle de Congestion

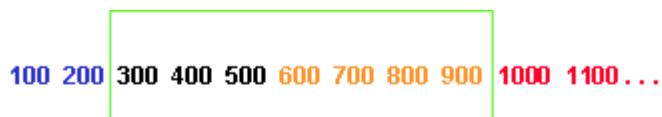
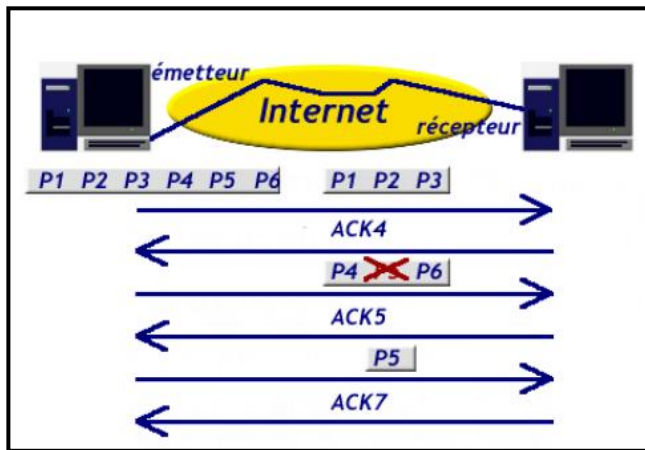
#### Congestion:

- ✦ "trop de sources envoient trop de données trop rapidement dans le *réseau*"
- ✦ Différent du contrôle de flot!
- ✦ manifestations:
  - ✦ Pertes de paquets (débordement des routeurs)
  - ✦ délais important (file d'attente dans les routeurs)

2: Application Layer 21

## Fenêtrage « window »

### Gestion de Fenêtrage TCP



**Exemple : Fenêtre de 700 octets**

## UDP