

# Couche Transport

# Introduction

## Modèle TCP/IP



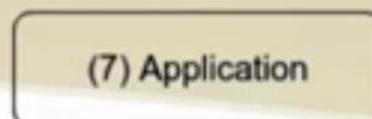
Application

Transport

Internet

Accès réseau

## Modèle OSI



(7) Application

(6) Présentation

(5) Session

(4) Transport

(3) Réseau

(2) Liaison de données

(1) Physique

Assure un transfert de bout en bout, des données issues de la couche supérieure.

# Introduction

## Modèle OSI

(7) Application

(6) Présentation

(5) Session

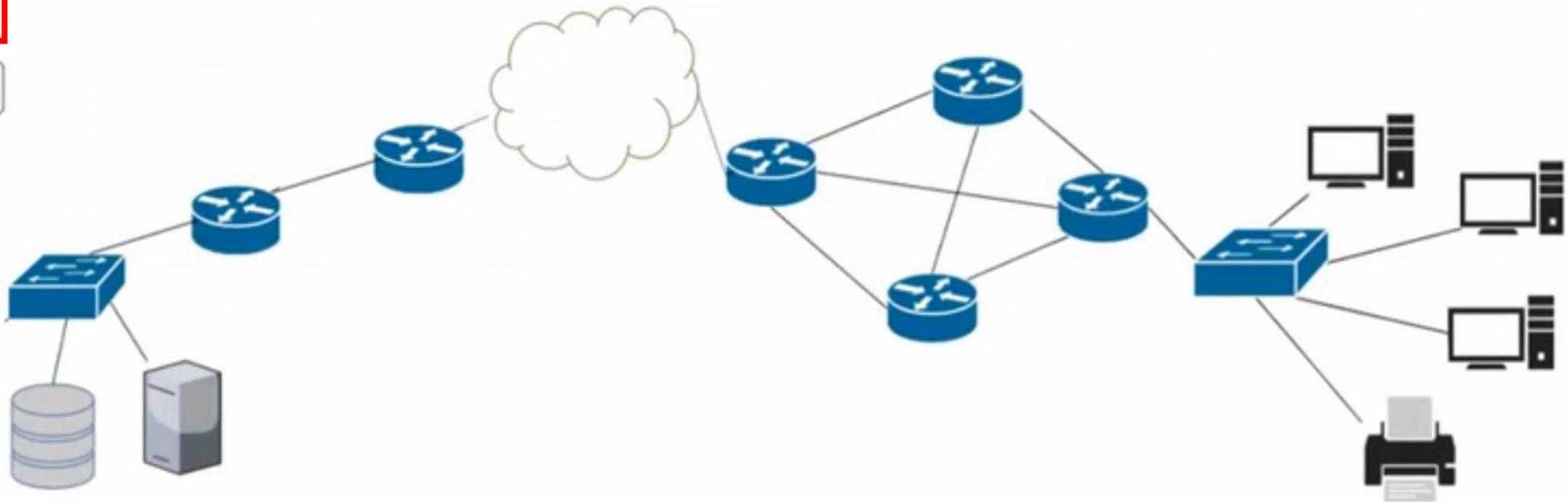
(4) Transport

(3) Réseau

(2) Liaison de données

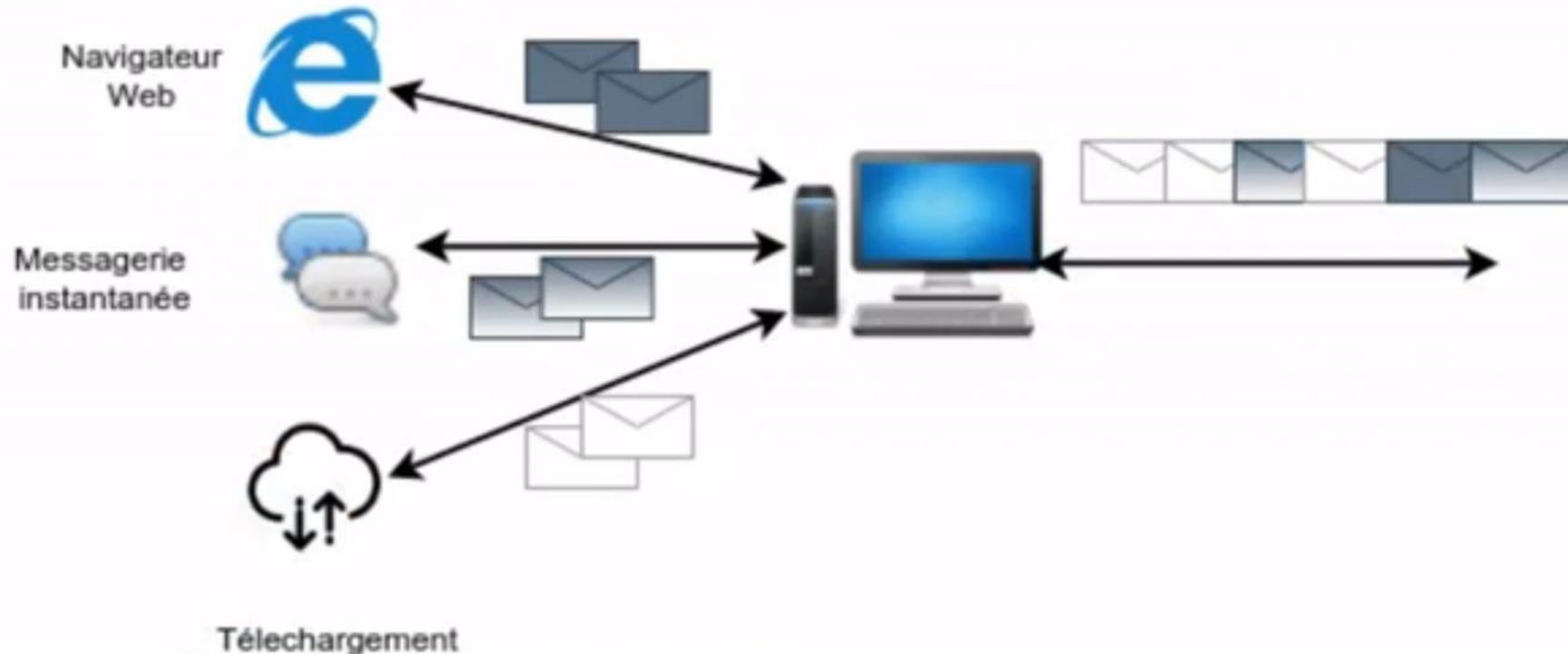
(1) Physique

Transfer de bout en bout



# Services de la couche transport

- Segmenter (réassembler) les données de la couche application
- Identifier la donnée de chaque application : numéro de port (3<sup>eme</sup> type d'adresse)



# Mode d'acheminement

Potocoles de la couche Transport (modèle TCP/IP)



TCP  
(Transmission Control Protocol)

UDP  
(User Datagram Protocol)

Le choix entre ces deux protocoles est suivant l'importance accordée à la fiabilité ou à la charge imposée au réseau.



**TCP**

**Transmission Control Protocol**

# TCP

Le protocole TCP est un protocole de transport fiable.

Nom PDU : **Segment**

# Caractéristiques du TCP

1. **Encapsulation;**

2. **Reconstitution ordonnée** : les segments envoyés peuvent emprunter des routes différentes, ainsi arriver dans le désordre. Certaines applications demandent que les segments doivent arriver dans un ordre donné pour être traités correctement.

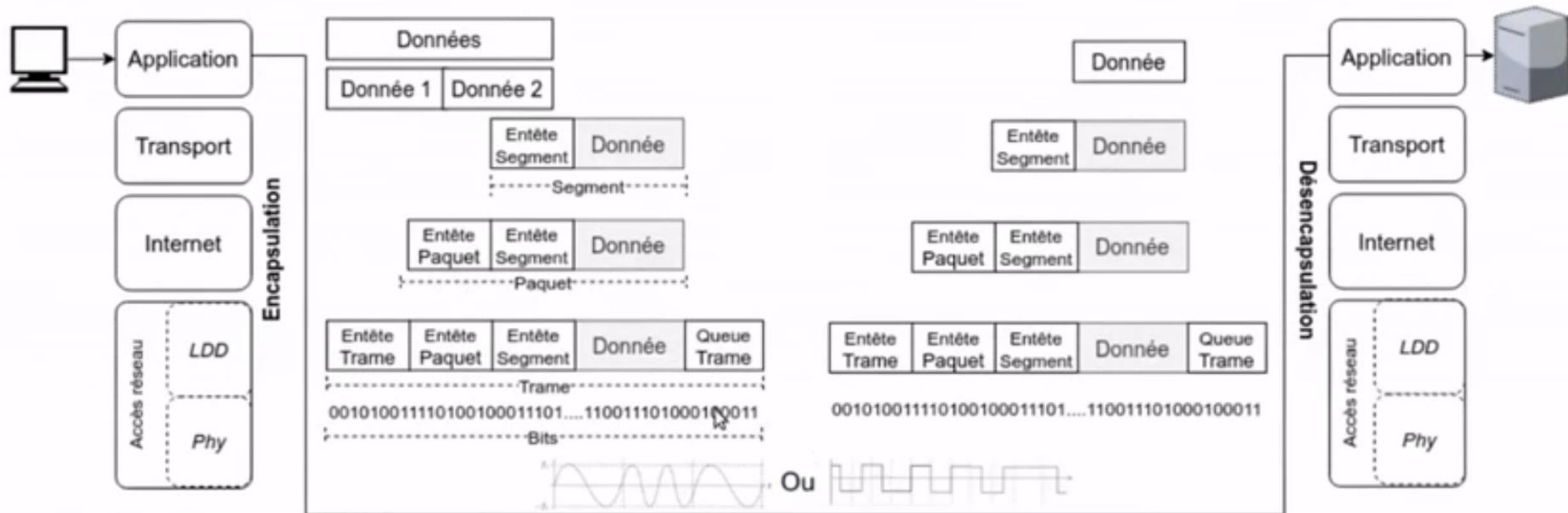
3. **Acheminement fiable** : le récepteur doit envoyer à l'émetteur un accusé de réception;

4. **Contrôle de flux** : le récepteur peut demander à l'émetteur de réduire le nombre de segments transférés et à demander l'accusé de réception;

5. **Orienté connexion** : avant l'envoi des segments, l'émetteur doit établir une connexion permanente avec le récepteur;

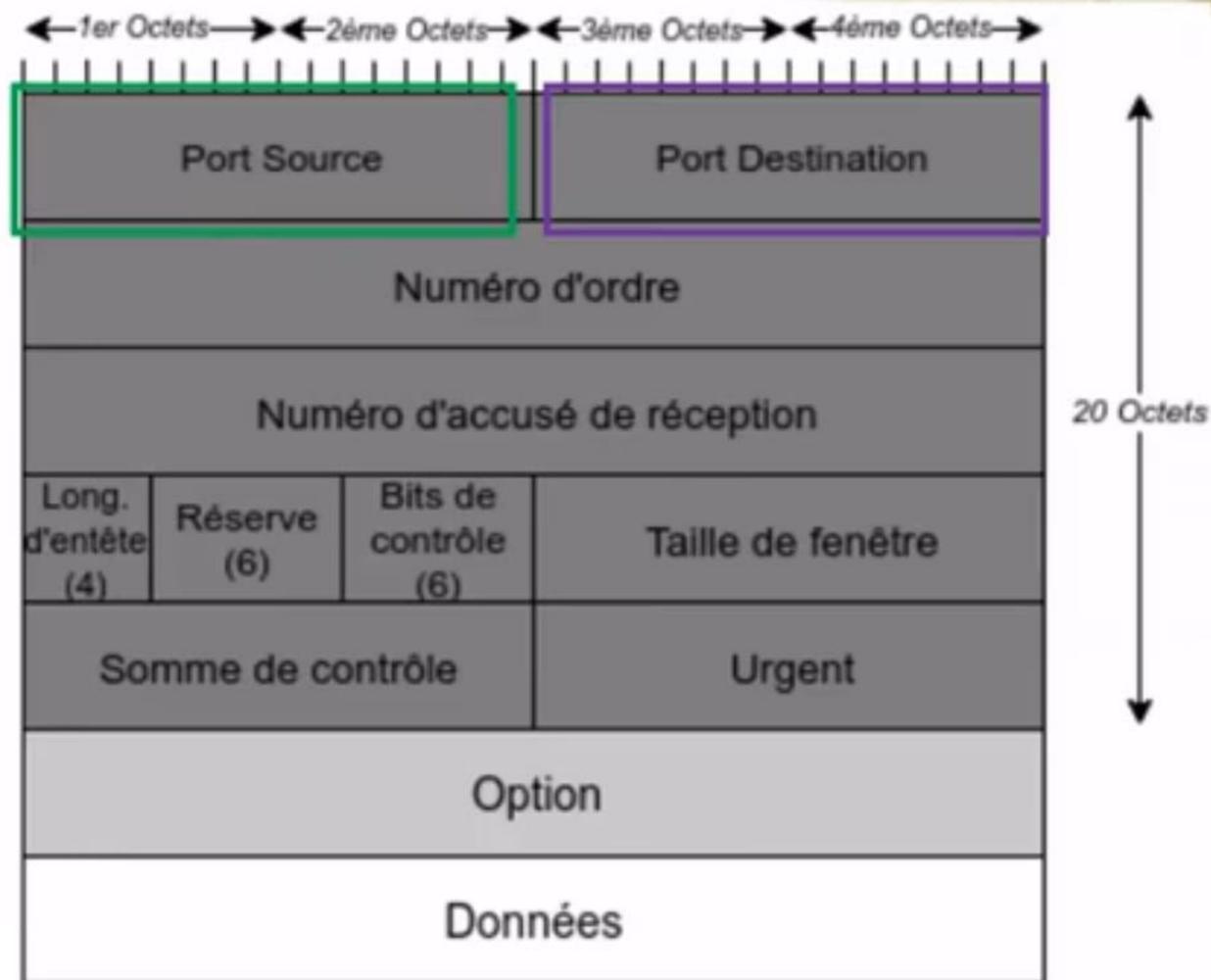
# Caractéristiques du TCP

## 1. Encapsulation;

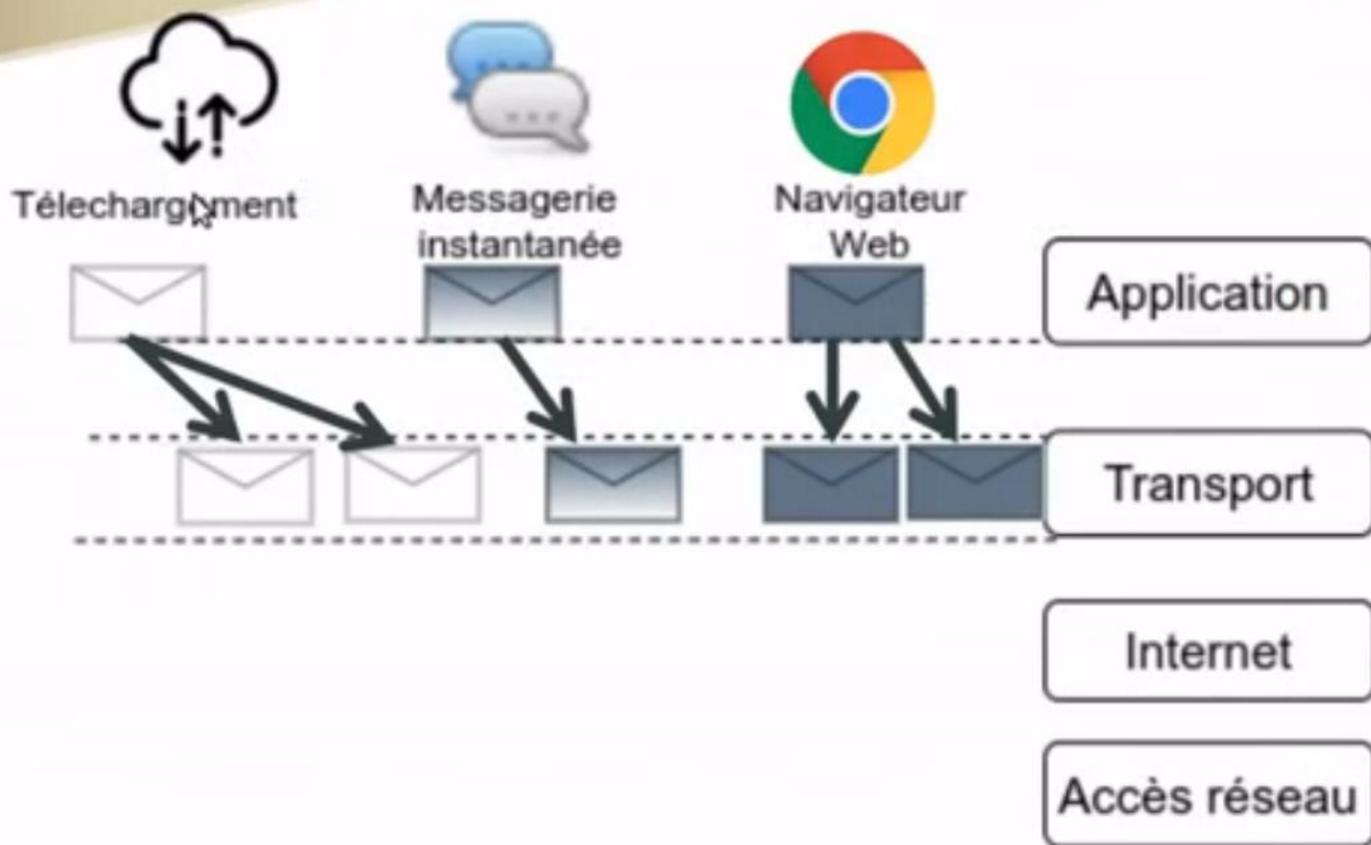


# En-tête TCP

- **Numéro de port** : c'est un identifiant unique utiliser pour différencier les segments de chaque application.



# En-tête TCP

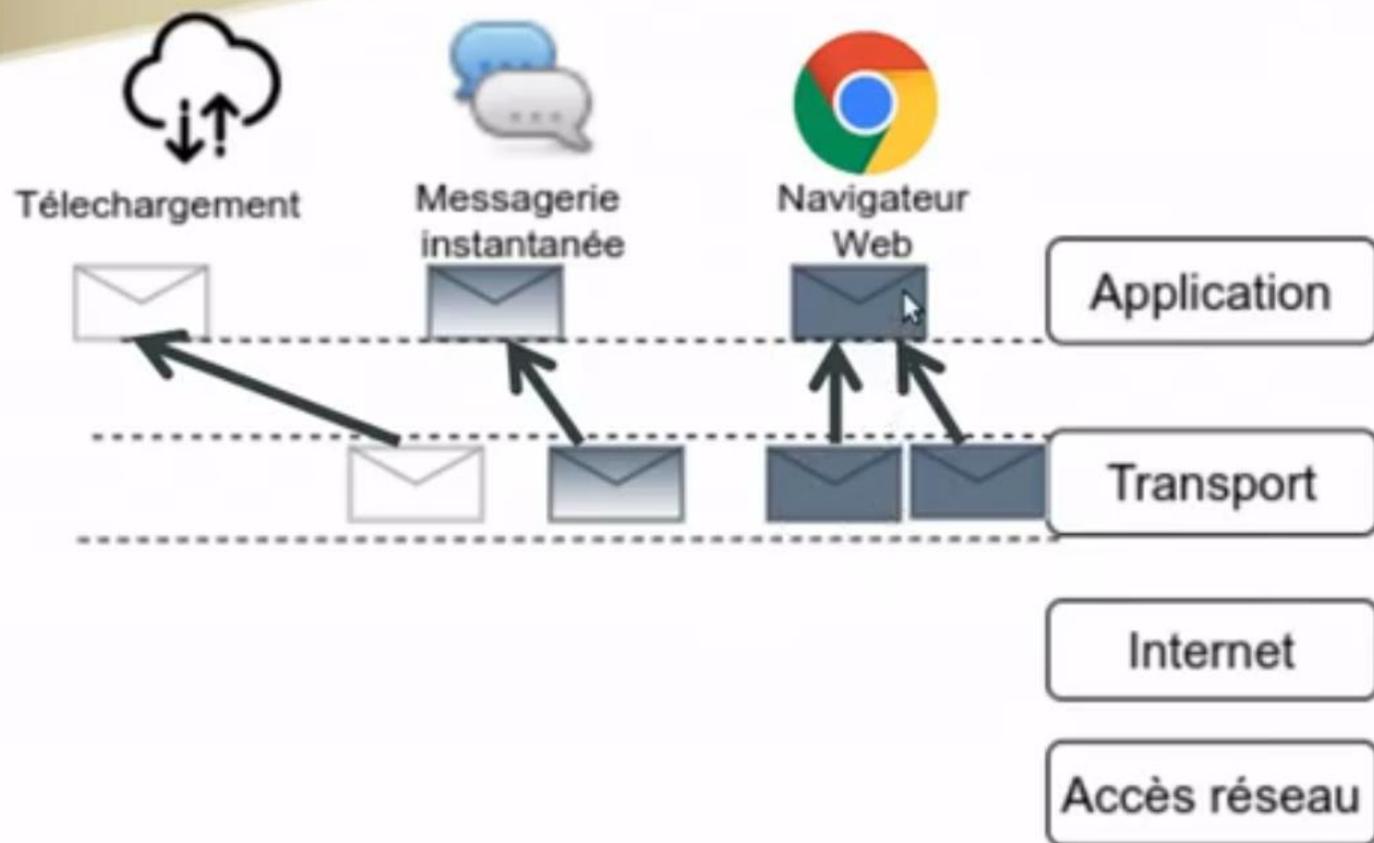


Le numéro de port source est associé à l'application de l'émetteur

Le numéro de port de destination à celle du destinataire.

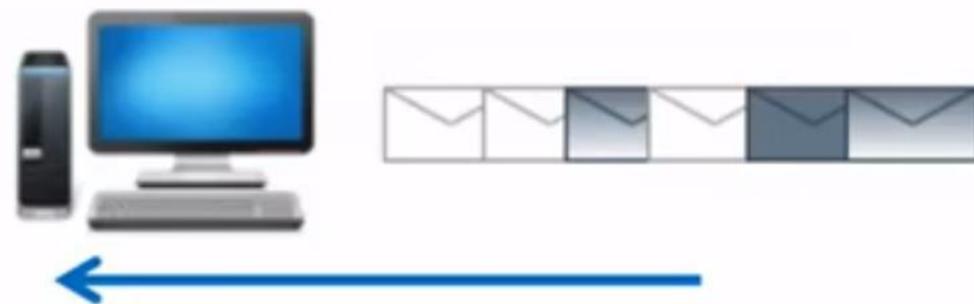


# En-tête TCP



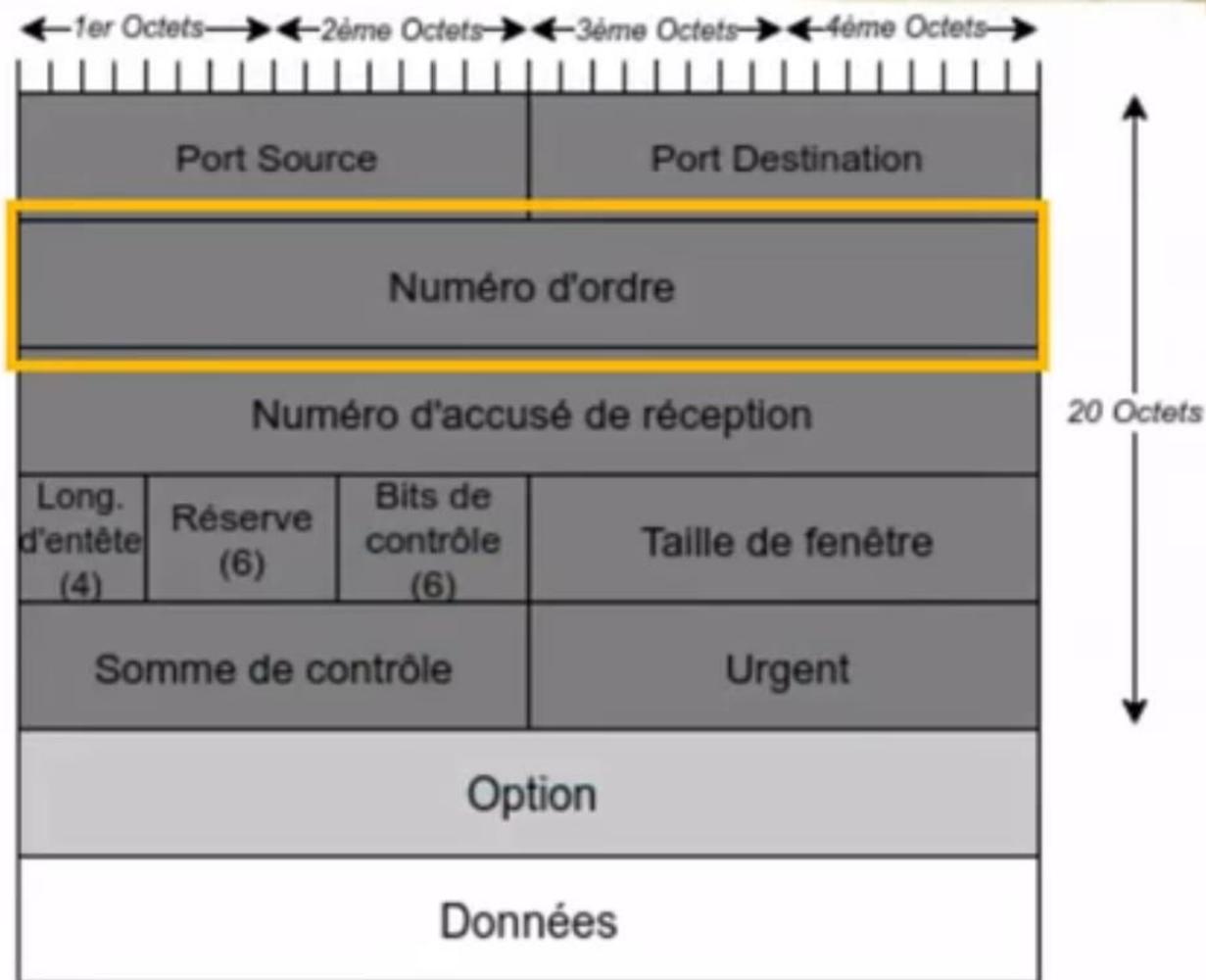
Le numéro de port source est associé à l'application de l'émetteur

Le numéro de port de destination à celle du destinataire.



# En-tête TCP

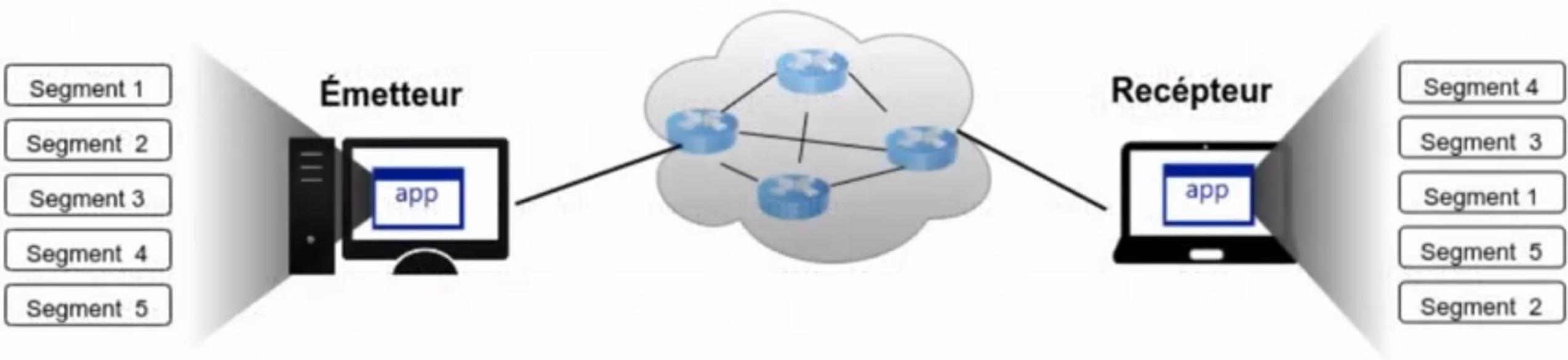
- **Un numéro d'ordre (SEQ)** : pour identifier la position de chaque segment dans le message d'origine.



# Caractéristiques du TCP

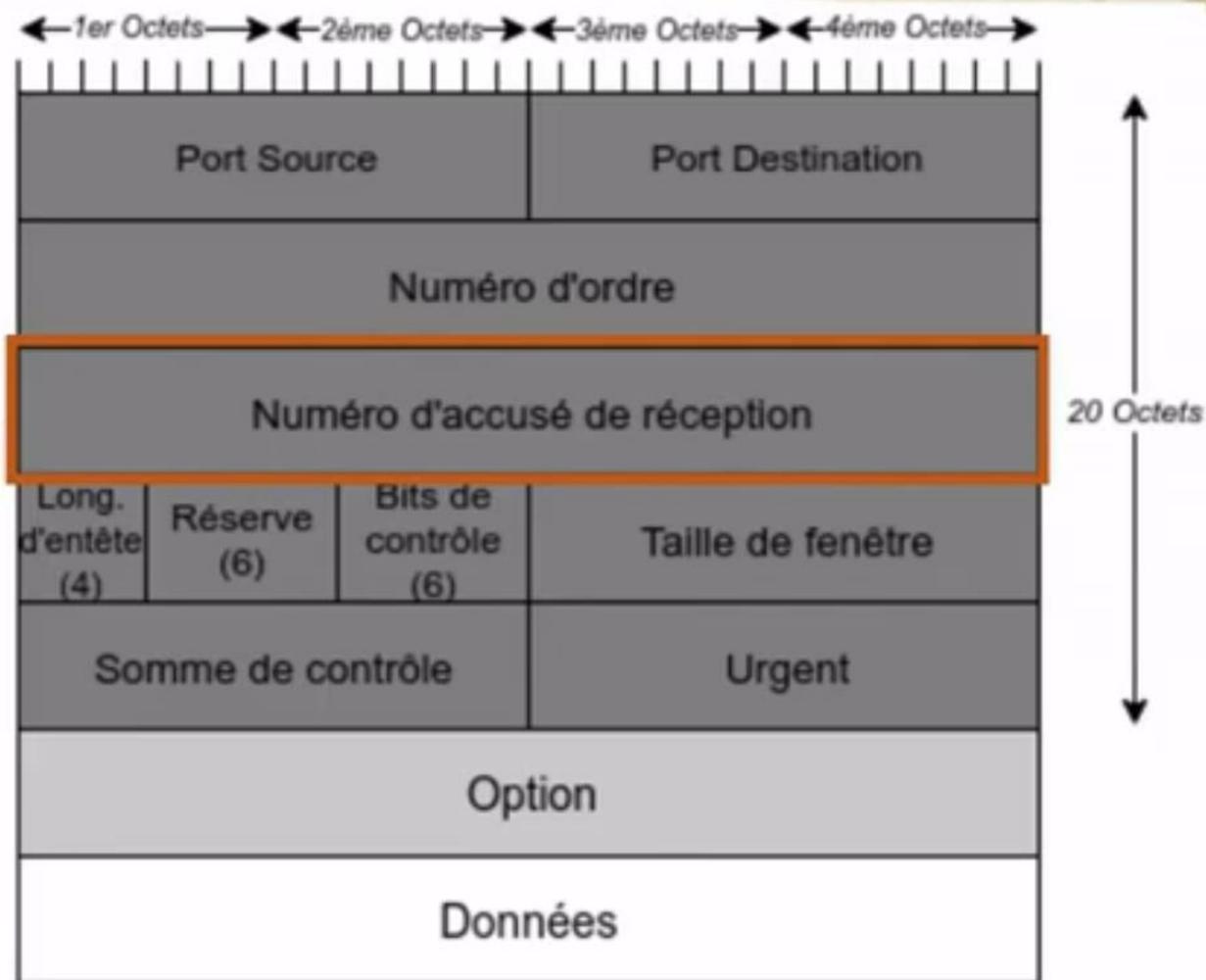
1. Encapsulation;
2. **Reconstitution ordonnée** : les segments envoyés peuvent emprunter des routes différentes, ainsi arriver dans le désordre. Certaines applications demandent que les segments doivent arriver dans un ordre donné pour être traités correctement.
3. **Acheminement fiable** : le récepteur doit envoyer à l'émetteur un accusé de réception;
4. **Contrôle de flux** : le récepteur peut demander à l'émetteur de réduire le nombre de segments transférés et à demander l'accusé de réception;
5. **Orienté connexion** : avant l'envoi des segments, l'émetteur doit établir une connexion permanente avec le récepteur;

# En-tête TCP



# En-tête TCP

- **Un numéro d'accusé de réception (ACK)** : indique les données qui ont été reçues.



# Caractéristiques du TCP

1. **Encapsulation;**

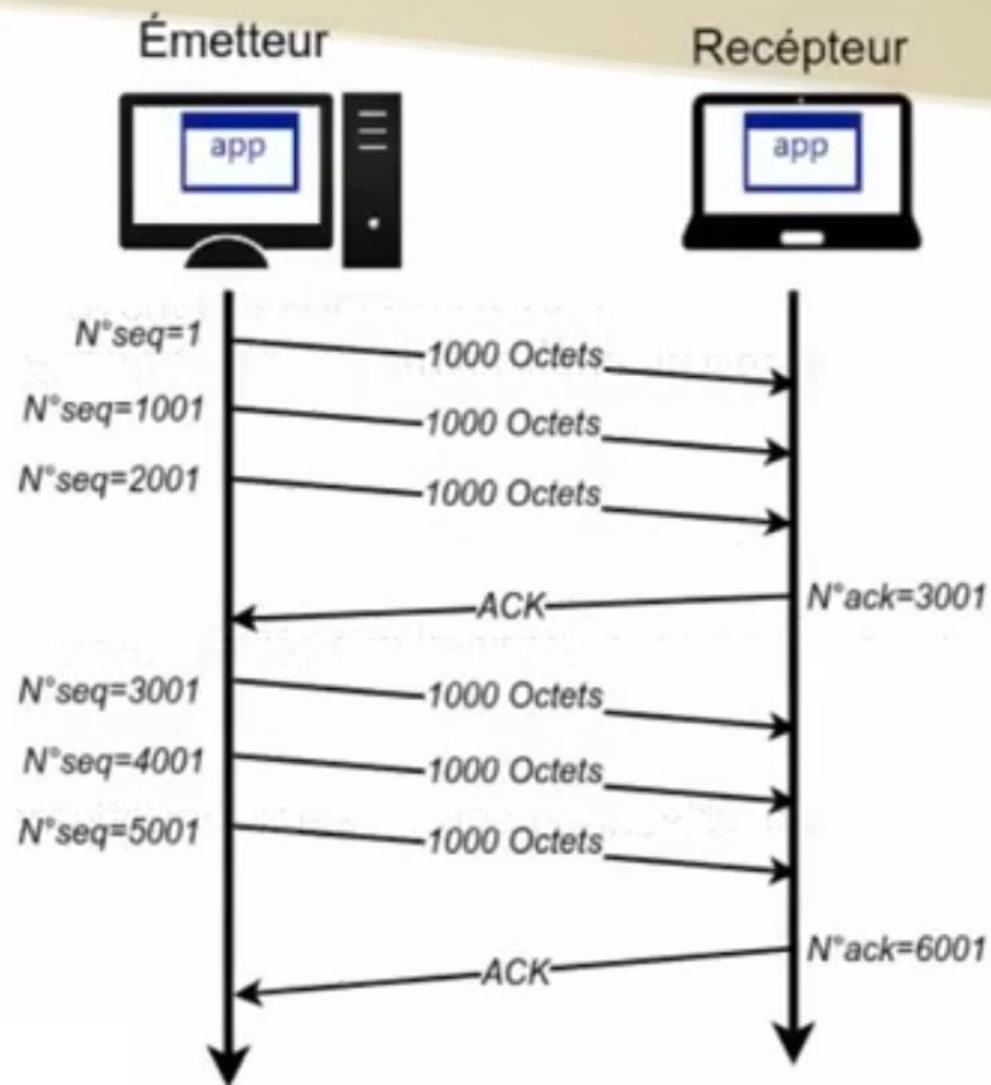
2. **Reconstitution ordonnée** : les segments envoyés peuvent emprunter des routes différentes, ainsi arriver dans le désordre. Certaines applications demandent que les segments doivent arriver dans un ordre donné pour être traités correctement.

3. **Acheminement fiable** : le récepteur doit envoyer à l'émetteur un accusé de réception;

4. **Contrôle de flux** : le récepteur peut demander à l'émetteur de réduire le nombre de segments transférés et à demander l'accusé de réception;

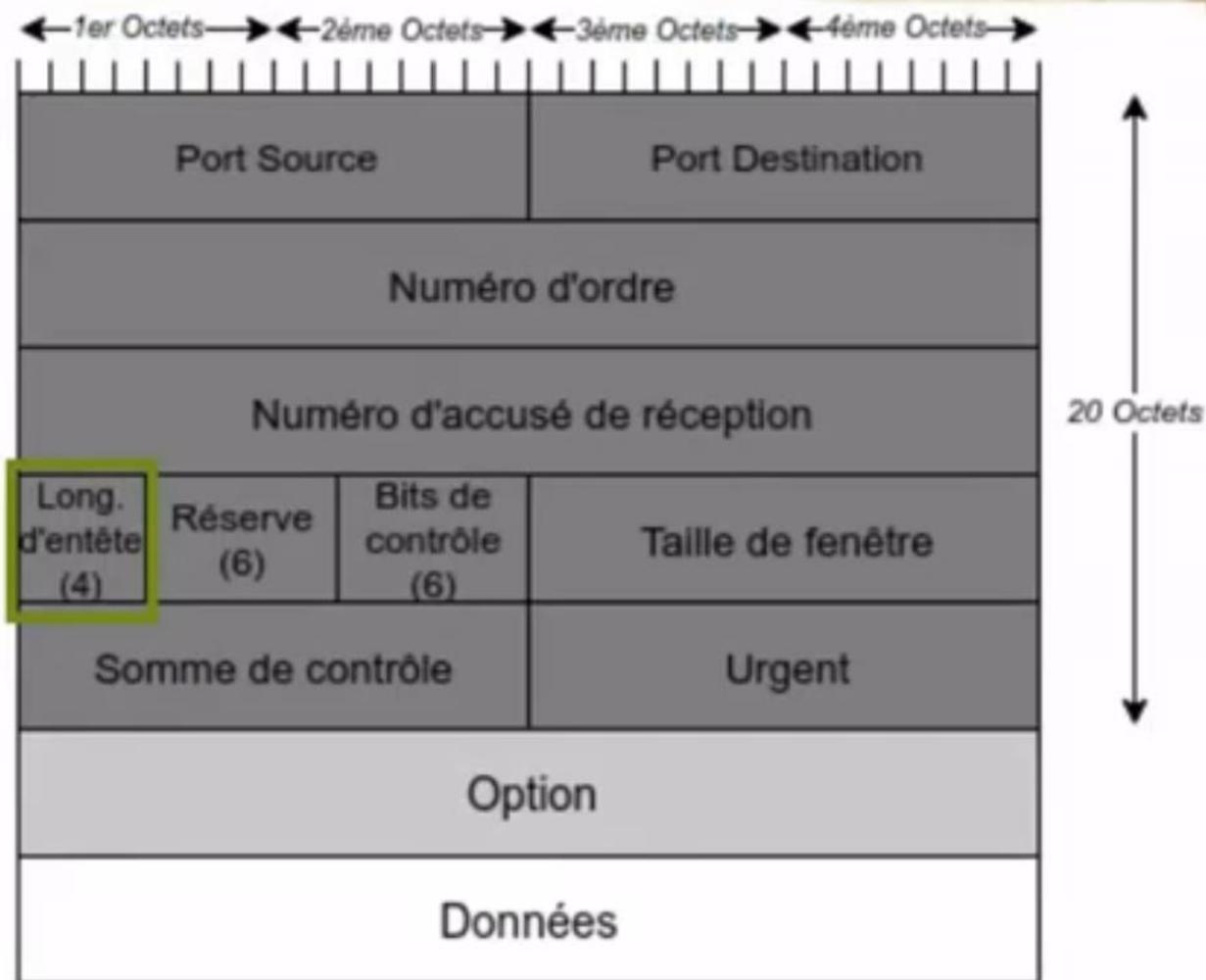
5. **Orienté connexion** : avant l'envoi des segments, l'émetteur doit établir une connexion permanente avec le récepteur;

# En-tête TCP



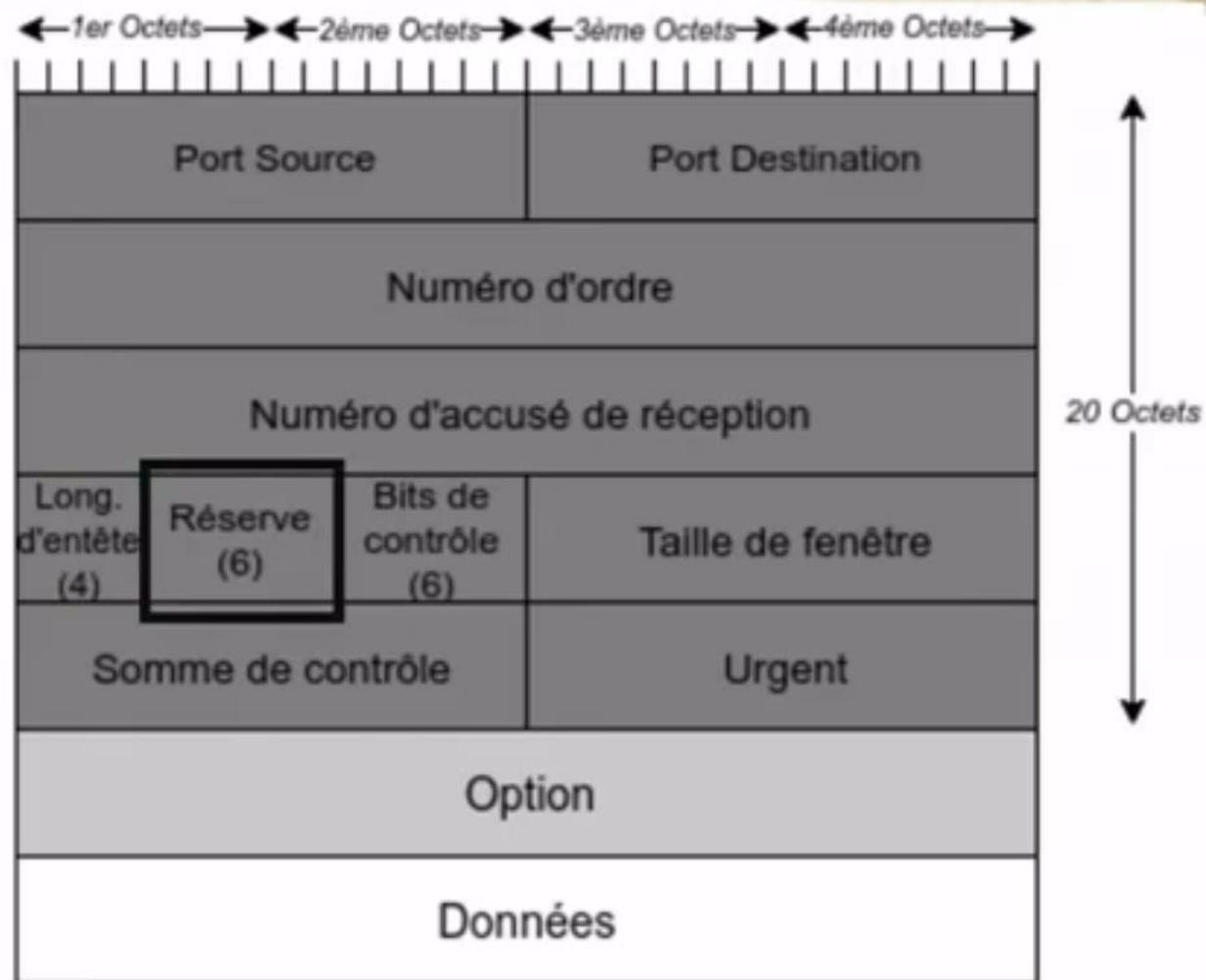
# En-tête TCP

- **Longueur d'en-tête ou OFFSET** : le nombre de mots de 4 octets qui composent l'entête.



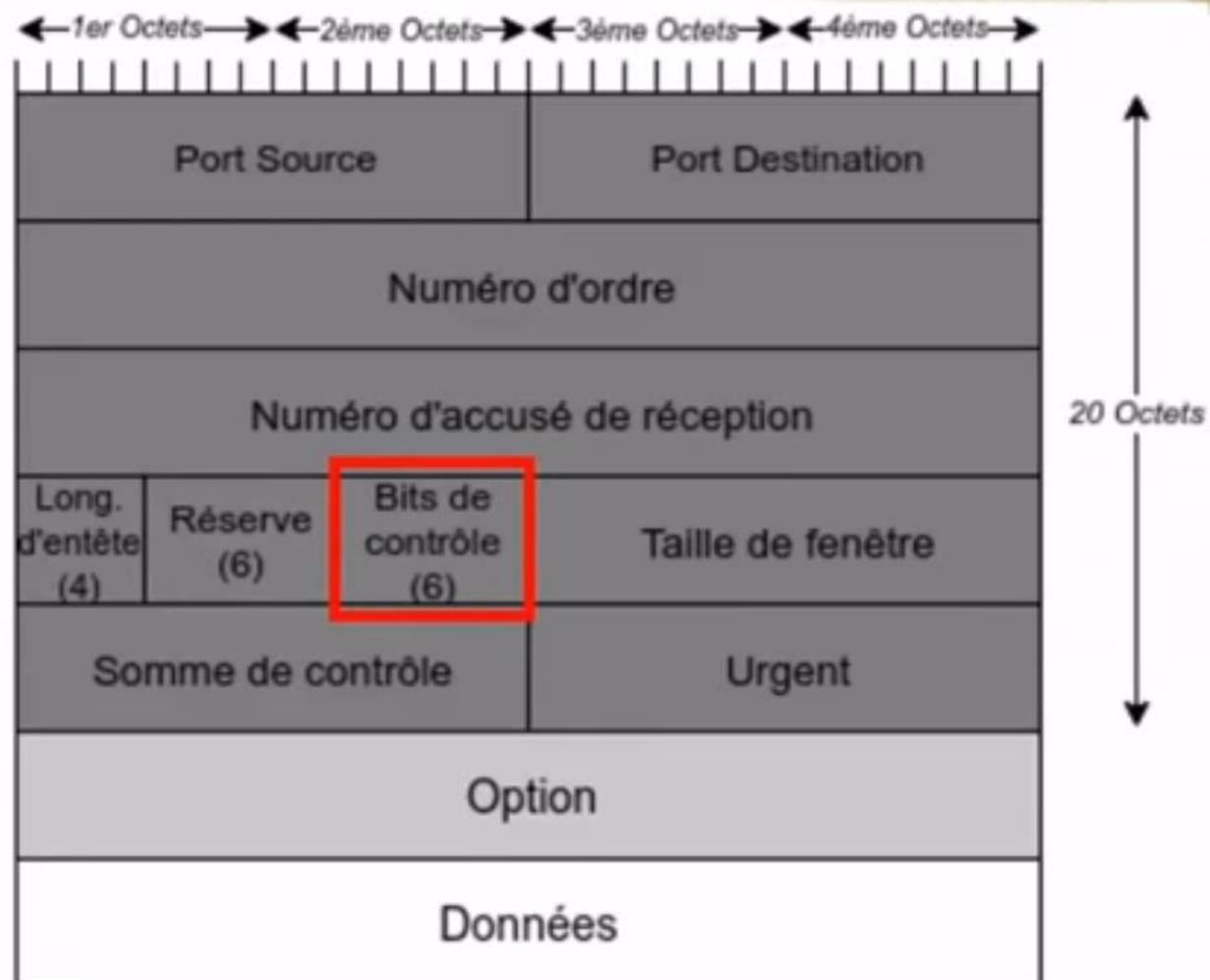
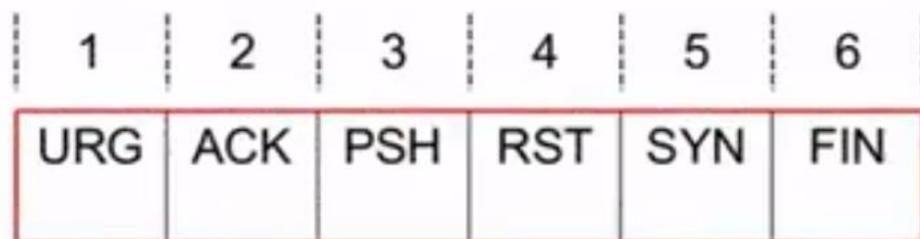
# En-tête TCP

- **Réservé** : réservé pour un futur usage.

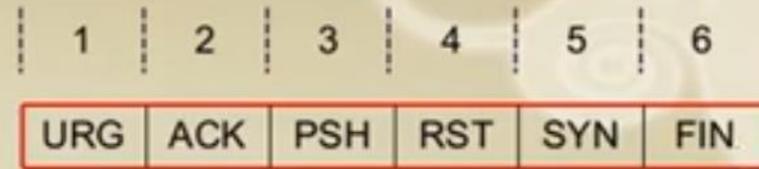


# En-tête TCP

• **Bits de contrôle** ou **code** ou **flags**: comprennent un code sur 6 bits indiquant la fonction du segment (la nature du segment):



# En-tête TCP



**URG** (urgent pointer) : le segment transporte des données urgentes dont la place est indiquée par le champ Pointeur d'urgence;

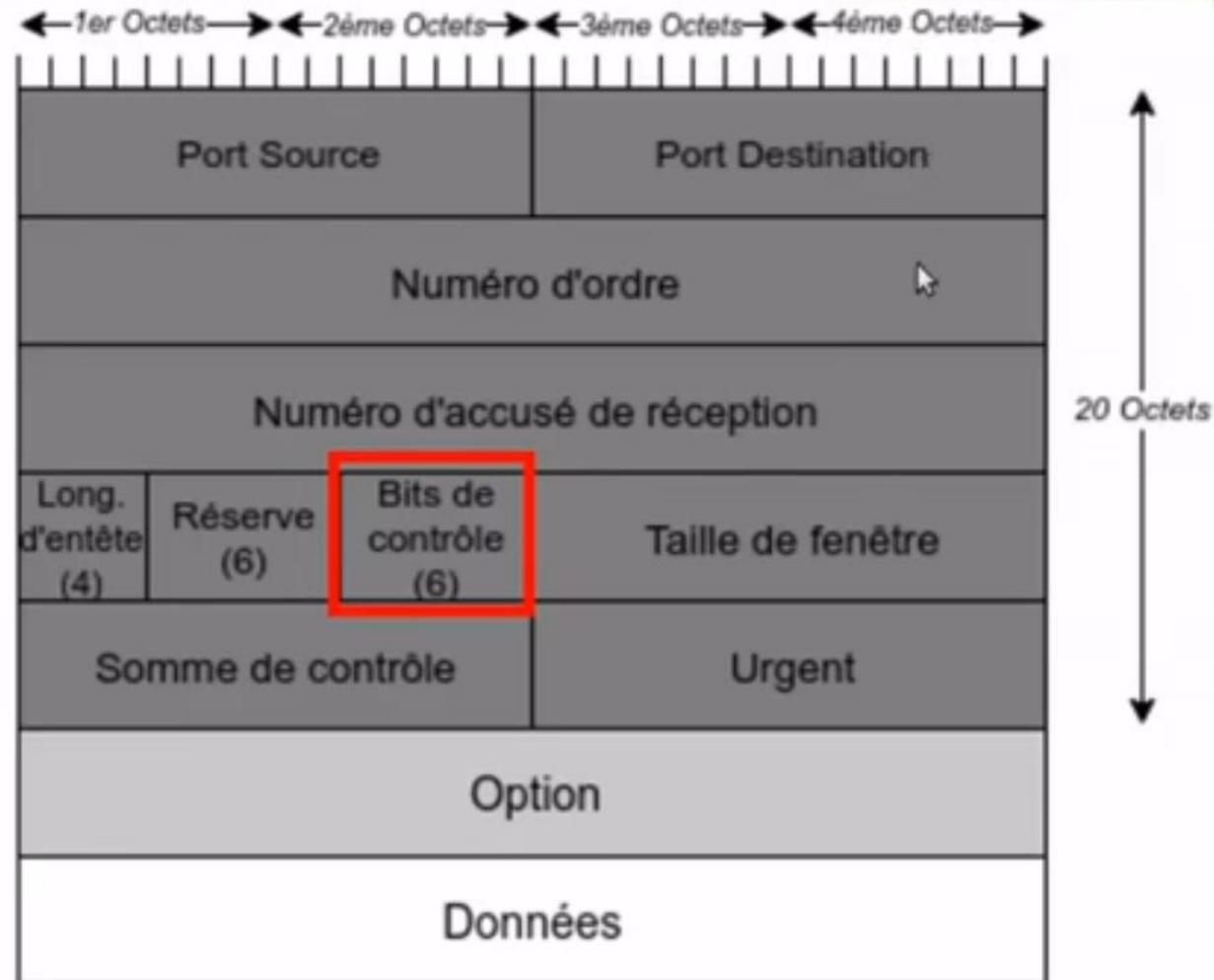
**ACK** (Acknowledgement): le segment transporte un accusé de réception.

**PSH** (Push) : le segment devra être délivrer immédiatement. Le récepteur ne doit pas attendre que son tampon de réception soit plein pour délivrer les données à l'application.

**RST** (Reset) : réinitialisation de la connexion.

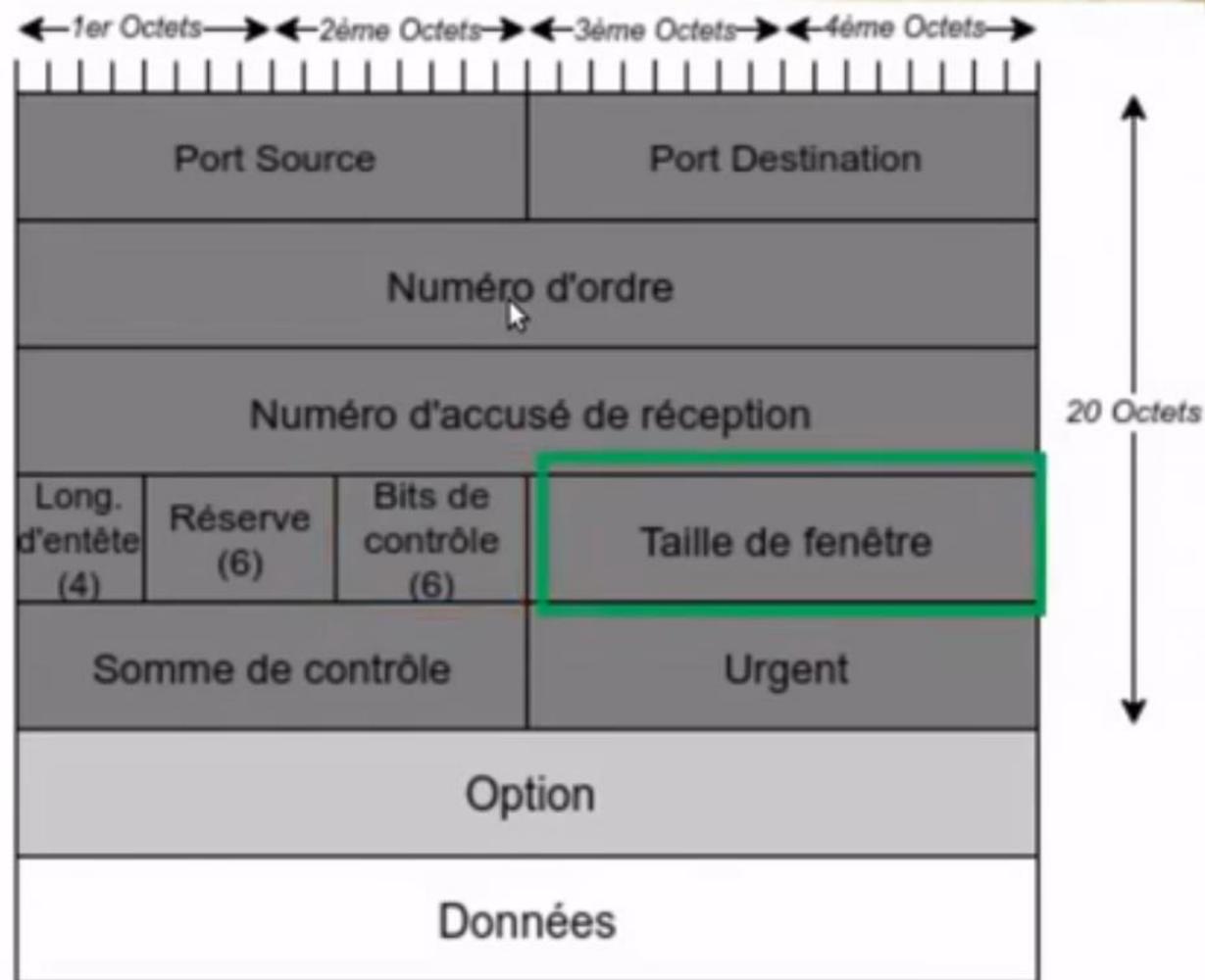
**SYN** (Synchronize) : indiquer qu'il s'agit de l'ouverture de connexion. Le champ « sequence number » contient la valeur de début de la connexion.

**FIN** (Final): terminer l'émission des segments.



# En-tête TCP

- La taille de fenêtre : c'est la taille du buffer de réception.



# Caractéristiques du TCP

1. **Encapsulation;**

2. **Reconstitution ordonnée** : les segments envoyés peuvent emprunter des routes différentes, ainsi arriver dans le désordre. Certaines applications demandent que les segments doivent arriver dans un ordre donné pour être traités correctement.

3. **Acheminement fiable** : le récepteur doit envoyer à l'émetteur un accusé de réception;

4. **Contrôle de flux** : le récepteur peut demander à l'émetteur de réduire le nombre de segments transférés et à demander l'accusé de réception;

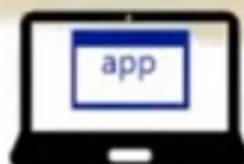
5. **Orienté connexion** : avant l'envoi des segments, l'émetteur doit établir une connexion permanente avec le récepteur;

# En-tête TCP

Émetteur

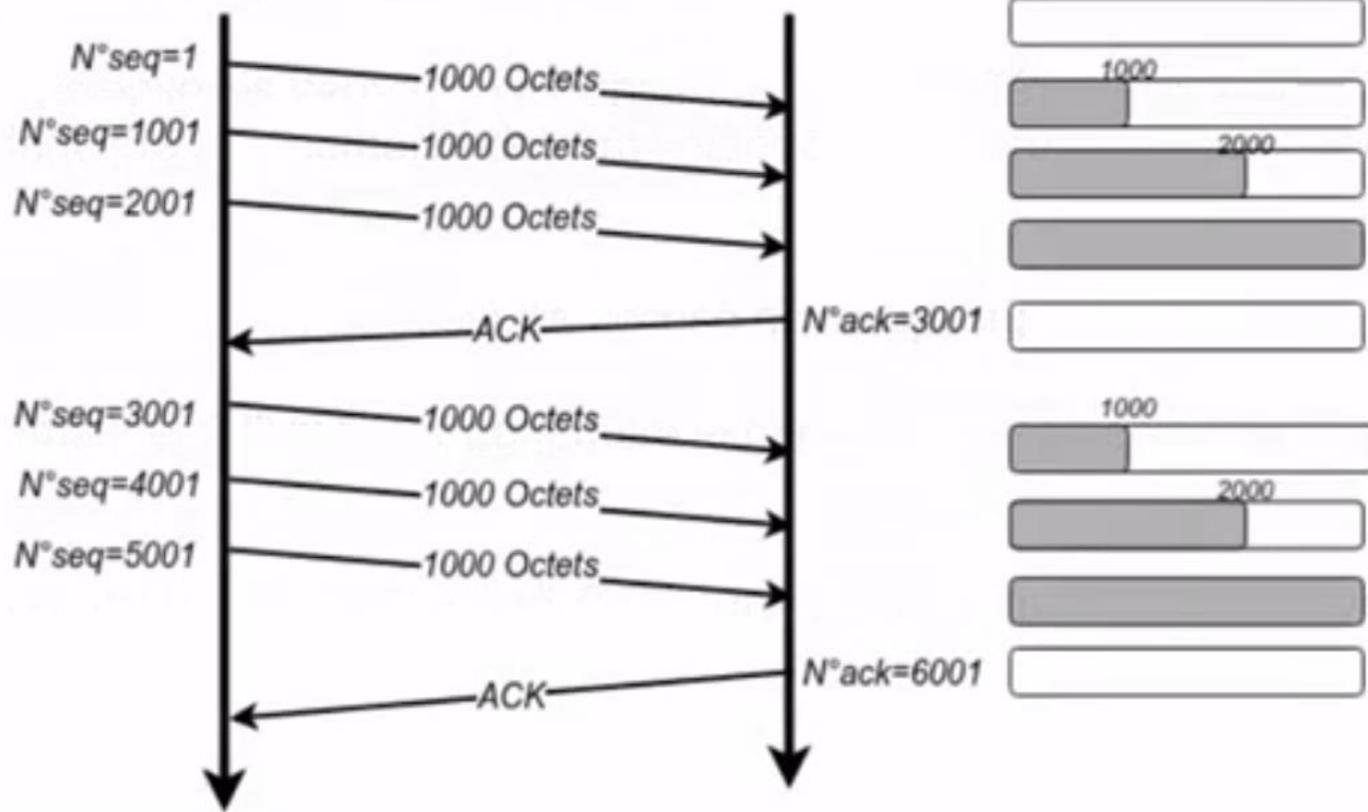


Recépteur



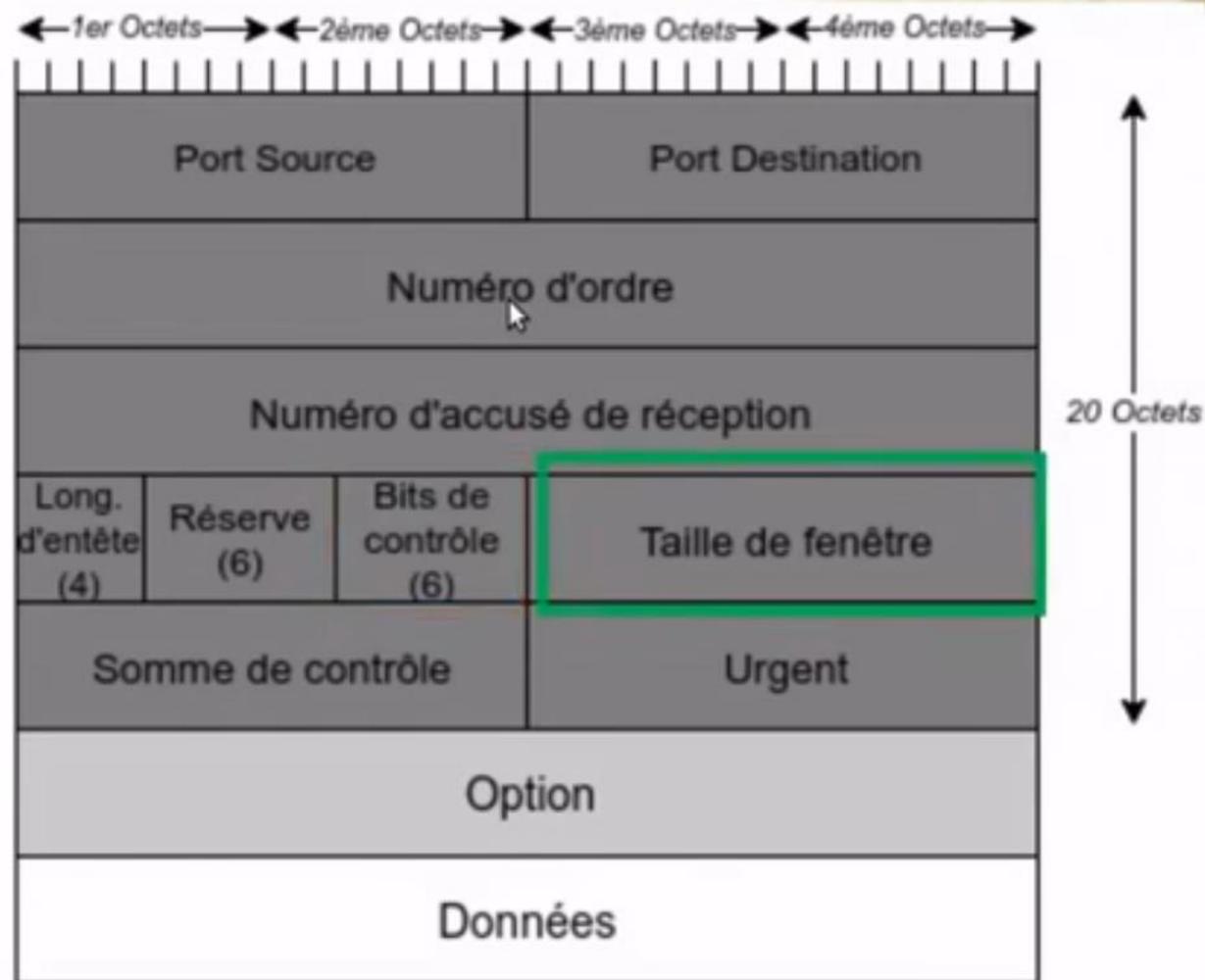
Taille de fenêtre

3000 Octets



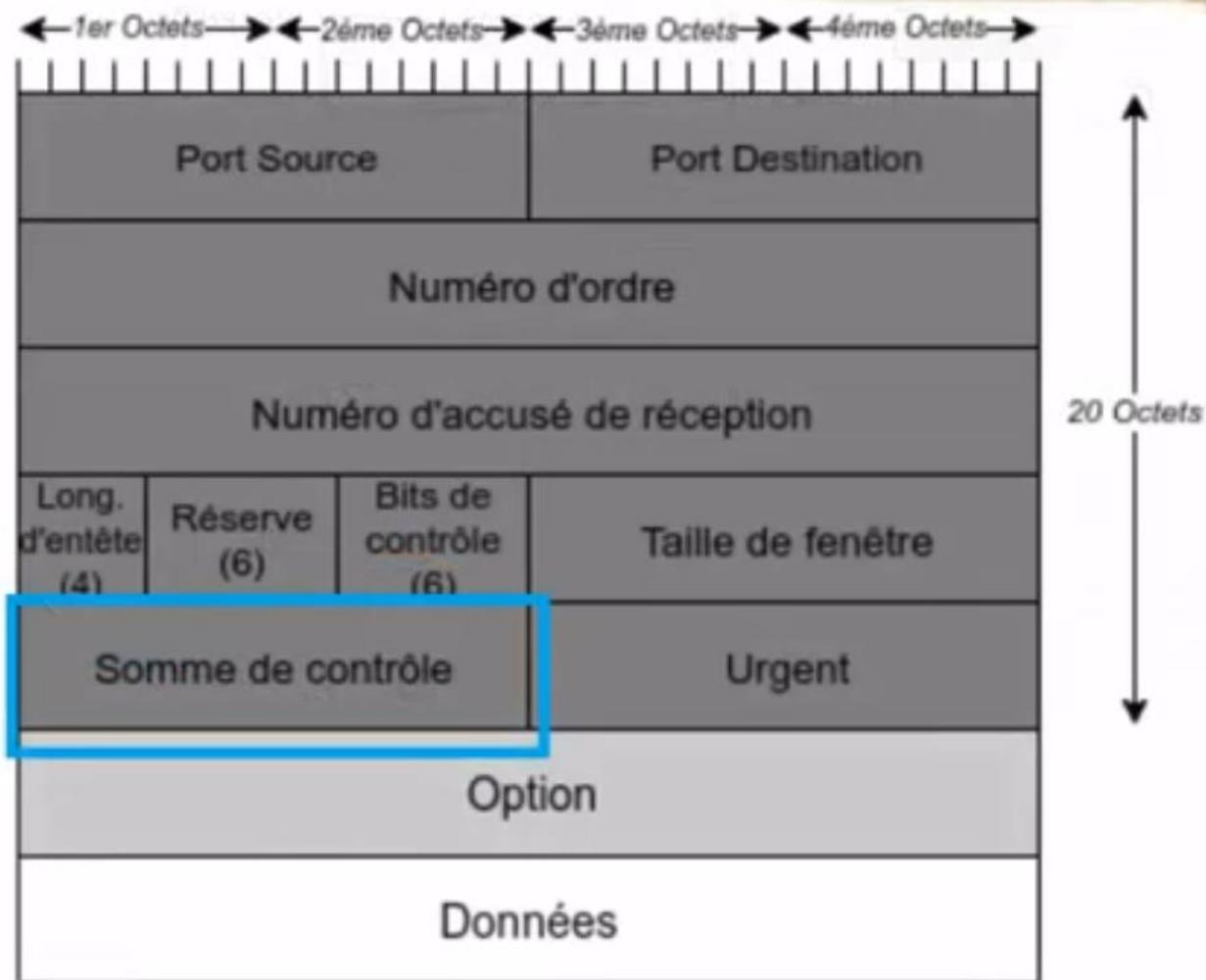
# En-tête TCP

- La taille de fenêtre : c'est la taille du buffer de réception.



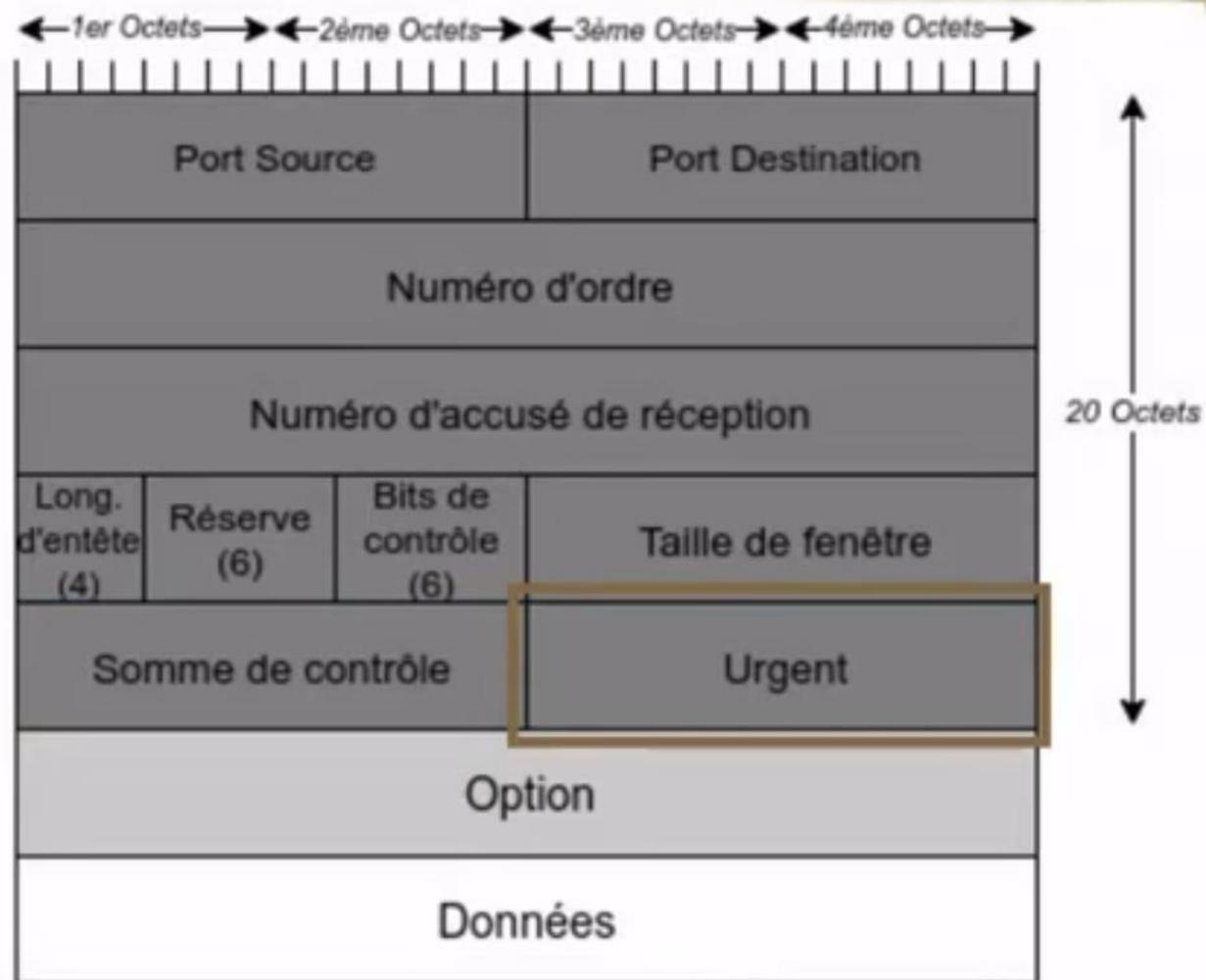
# En-tête TCP

- **La somme de contrôle** : utilisée pour le contrôle des erreurs sur l'en-tête et les données de segment.



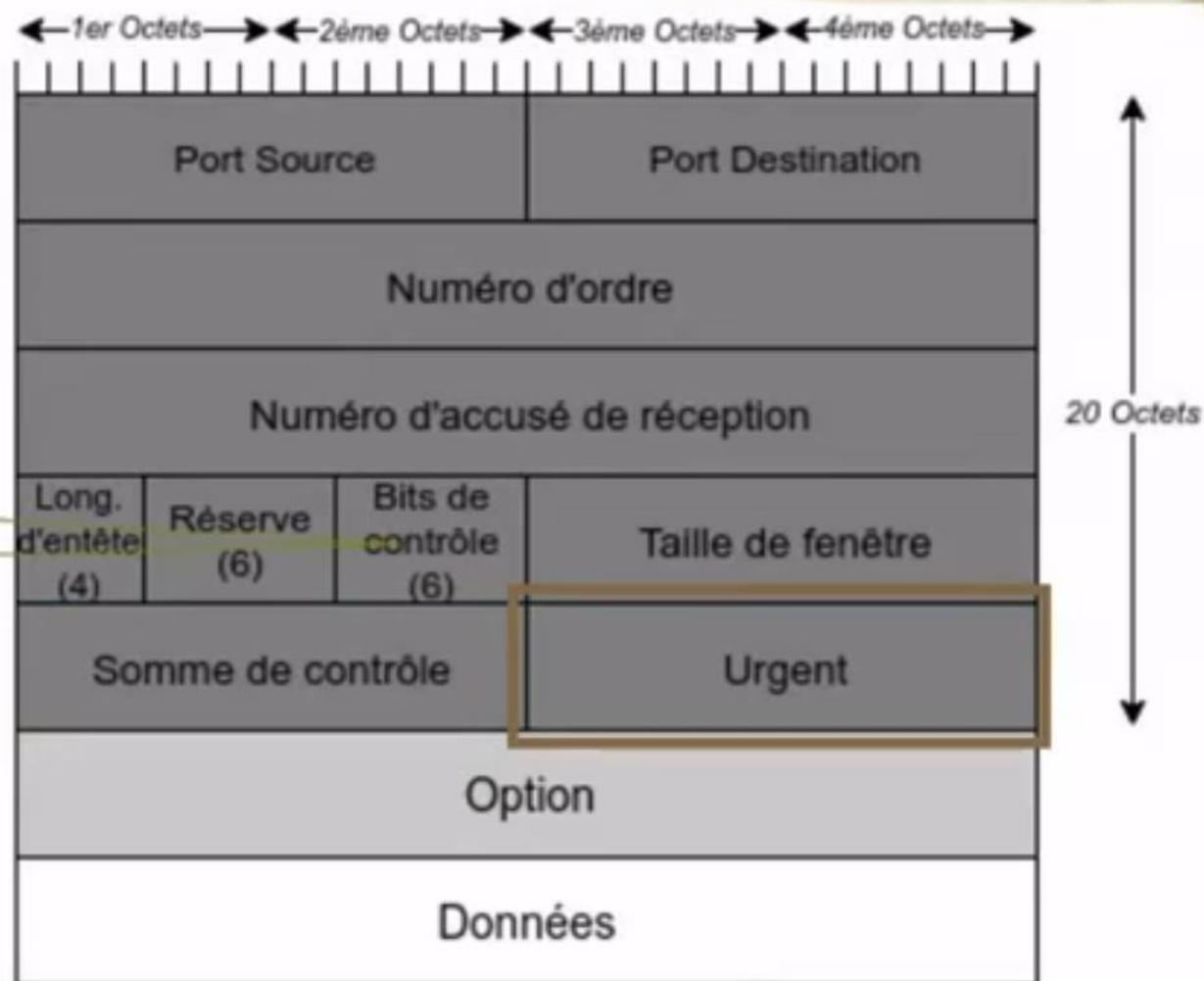
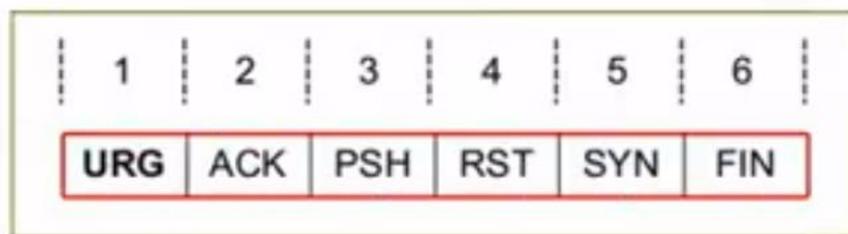
# En-tête TCP

- **Urgent** : ce champ est valide lorsque le bit URG est activé. Indique la position des données urgentes.



# En-tête TCP

- **Urgent** : ce champ est valide lorsque le bit URG est activé. Indique la position des données urgentes.



# En-tête TCP

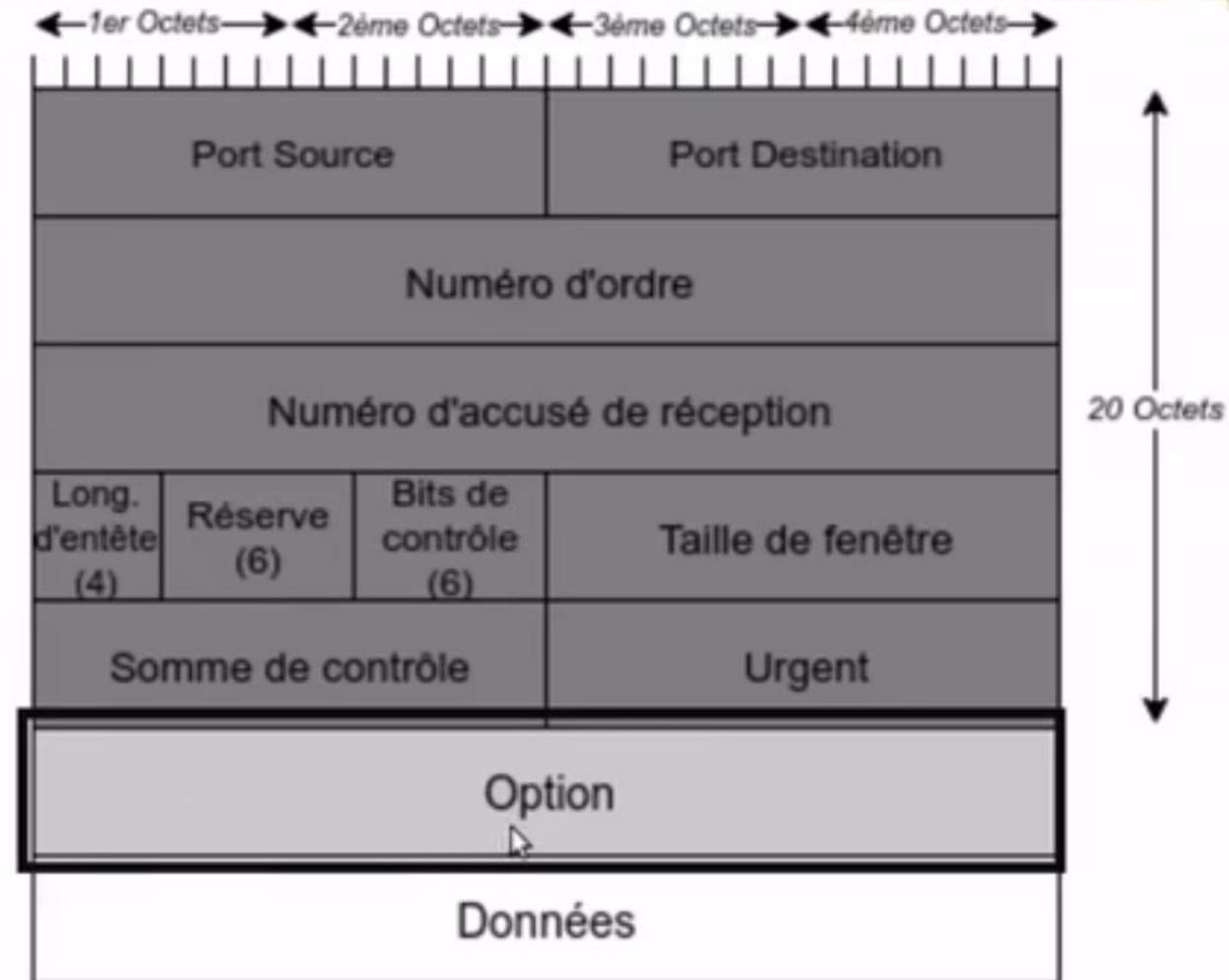
- **Options** : (taille variable), sa présence est détectée lorsque l'offset est supérieur à 5.

On trouve dans ce champ :

- **MSS** : au moment de l'établissement d'une connexion, chaque partie annonce sa taille de MSS (Maximum Segment Size) ou longueur maximum de segment.

- **Timestamp** (estampille temporelle) : permet de calculer la durée qu'un segment prend aller et retour (RTT, Round Trip Time).

- **Wscale** (Window Scale ou facteur d'échelle) : pour augmenter la taille de la fenêtre.



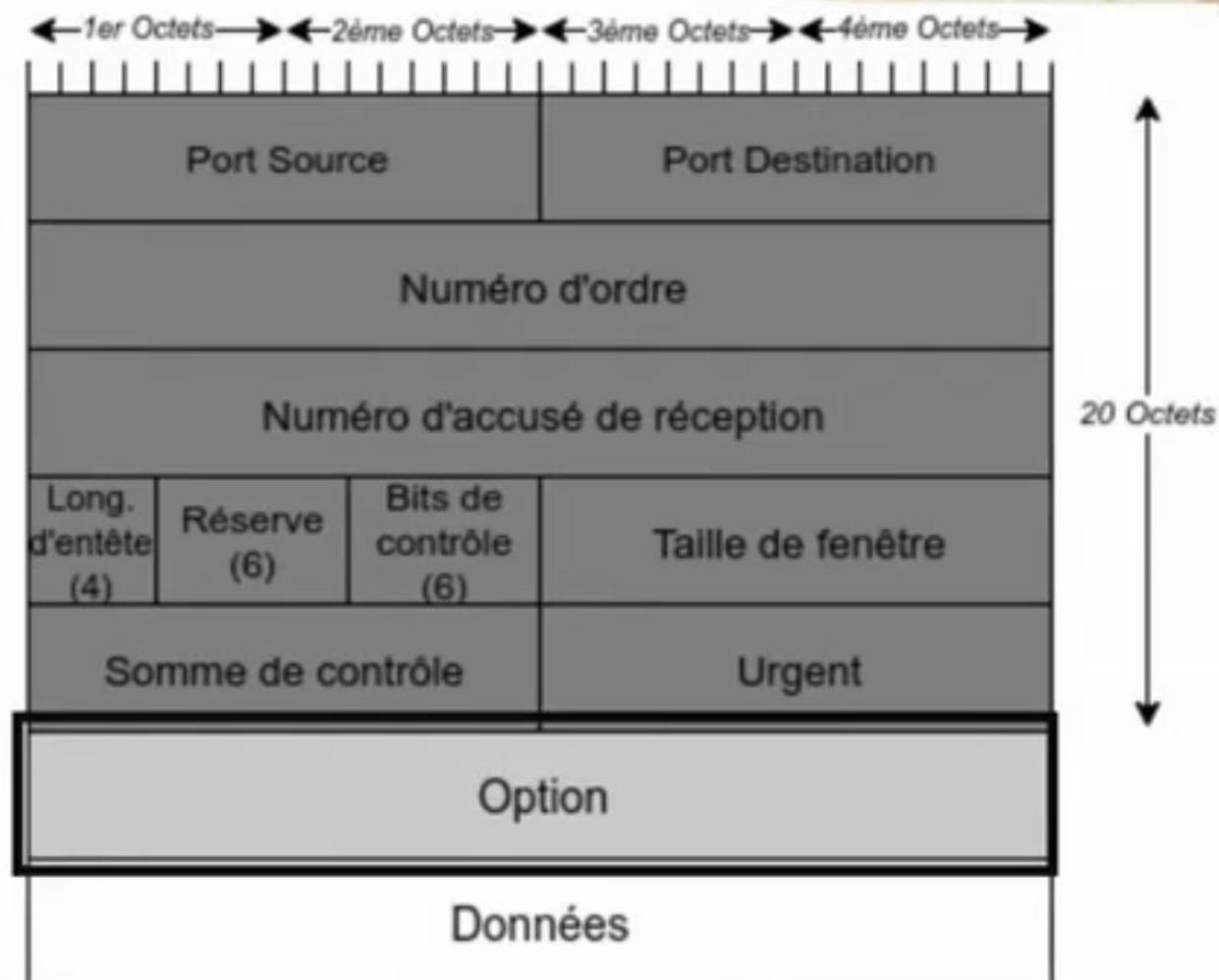
# En-tête TCP

• **Options** : (taille variable), sa présence est détectée lorsque l'offset est supérieur à 5.

Ce champ peut avoir jusqu'à **10 mots** de 32 bits (4 Octets).

Avec les 5 premiers mots, l'entête TCP peut atteindre au maximum 15 mots .

-> (**Taille Max entête TCP = 60 octets**)



# Caractéristiques du TCP

1. **Encapsulation;**

2. **Reconstitution ordonnée** : les segments envoyés peuvent emprunter des routes différentes, ainsi arriver dans le désordre. Certaines applications demandent que les segments doivent arriver dans un ordre donné pour être traités correctement.

3. **Acheminement fiable** : le récepteur doit envoyer à l'émetteur un accusé de réception;

4. **Contrôle de flux** : le récepteur peut demander à l'émetteur de réduire le nombre de segments transférés et à demander l'accusé de réception;

5. **Orienté connexion** : avant l'envoi des segments, l'émetteur doit établir une connexion permanente avec le récepteur;

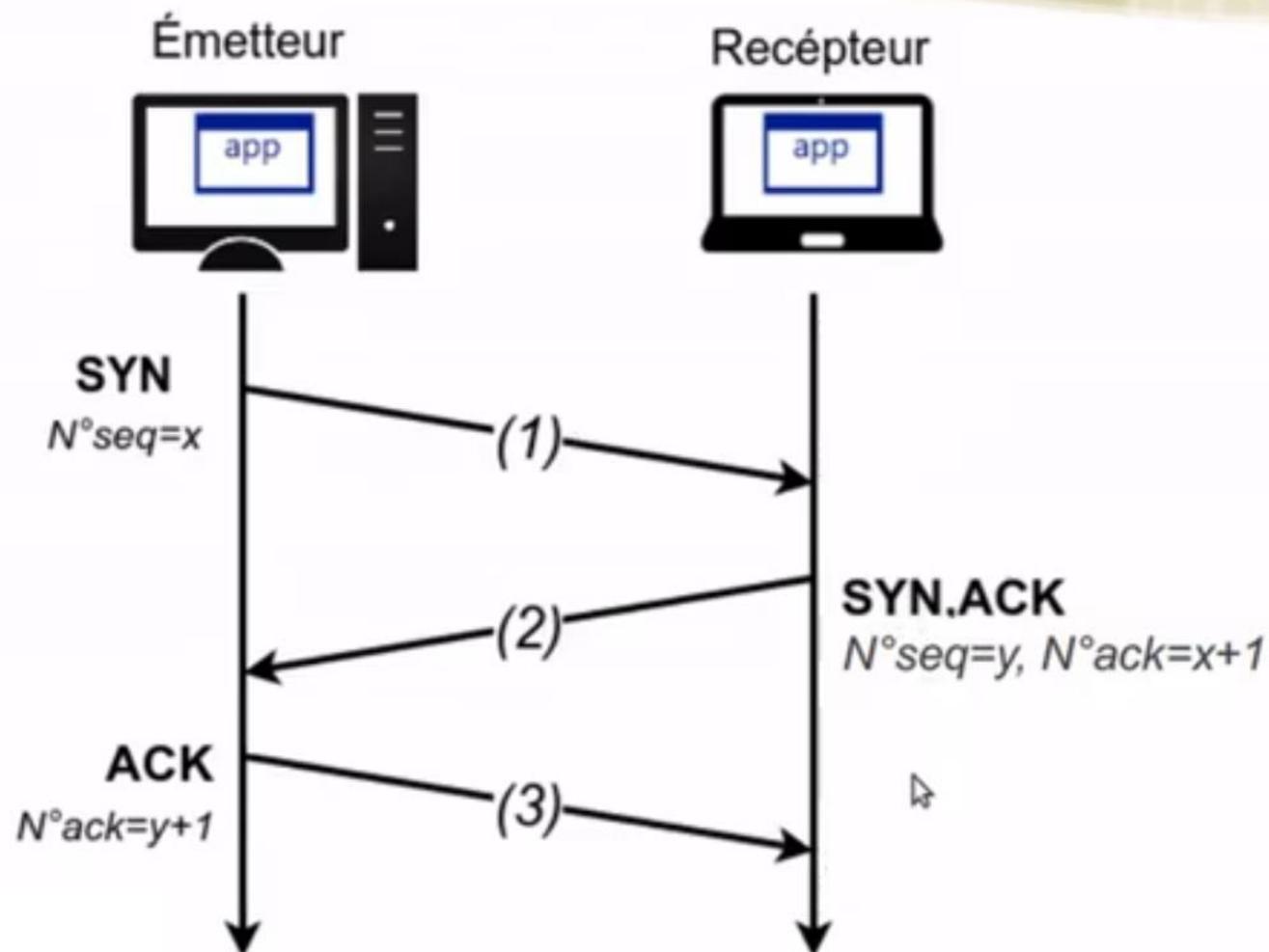
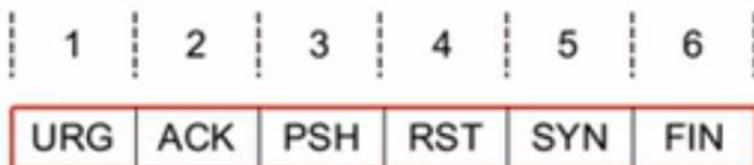
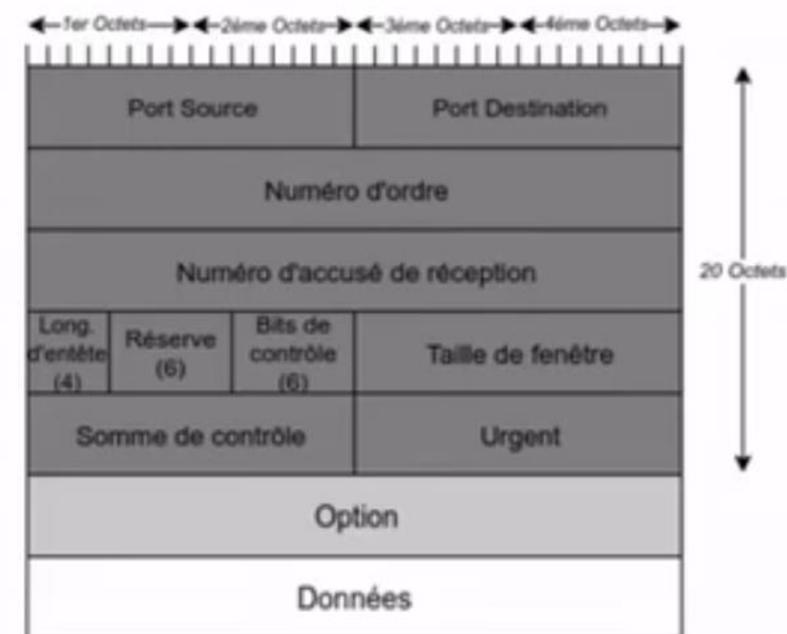
## Communication TCP

Elle est orientée connexion:

- 1- Établir la connexion;
- 2- Envoi des segments et la réception des ACK;
- 3- Déconnexion.

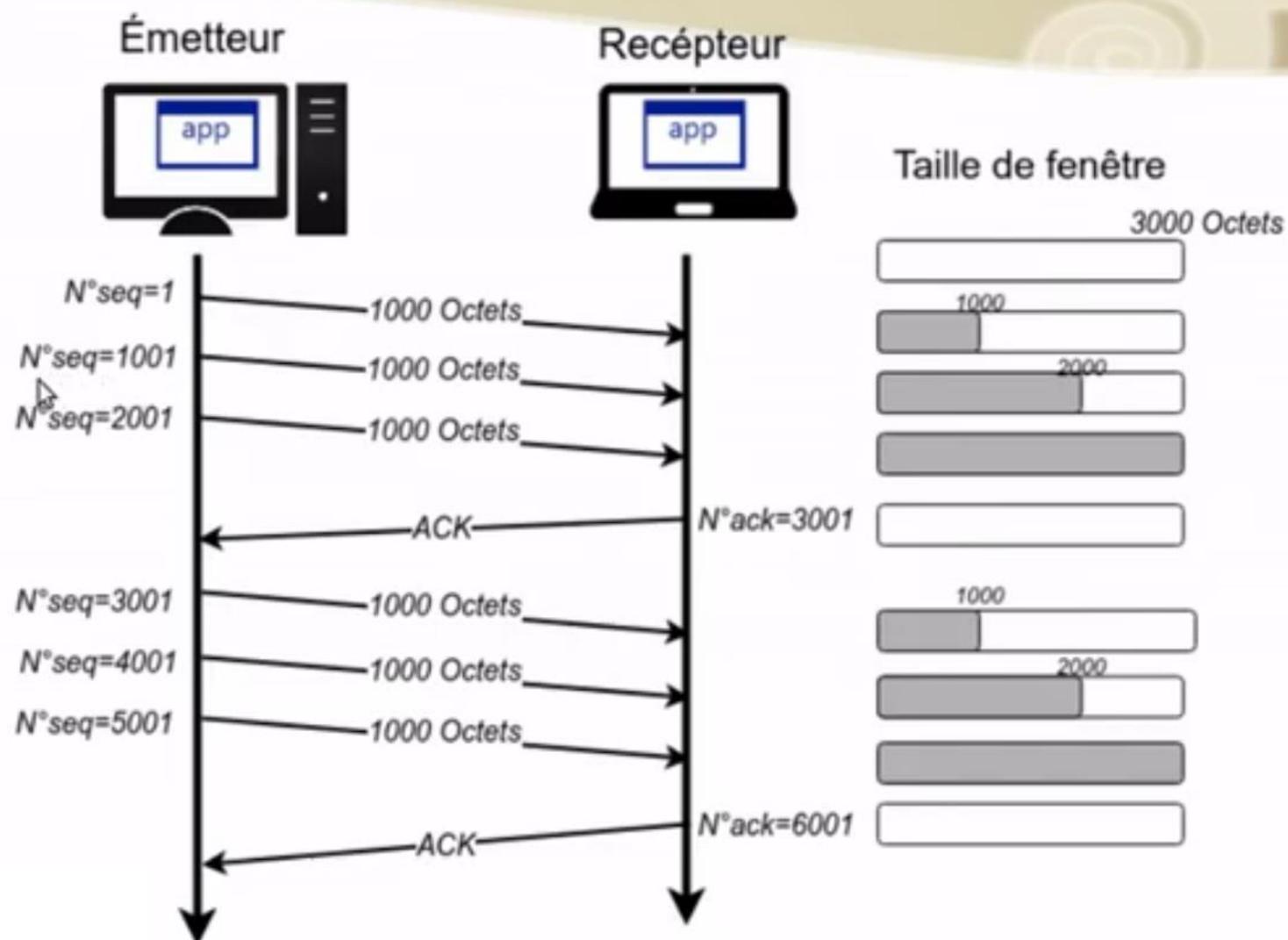
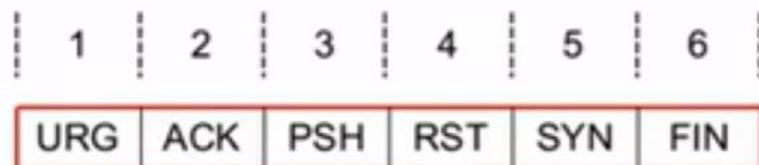
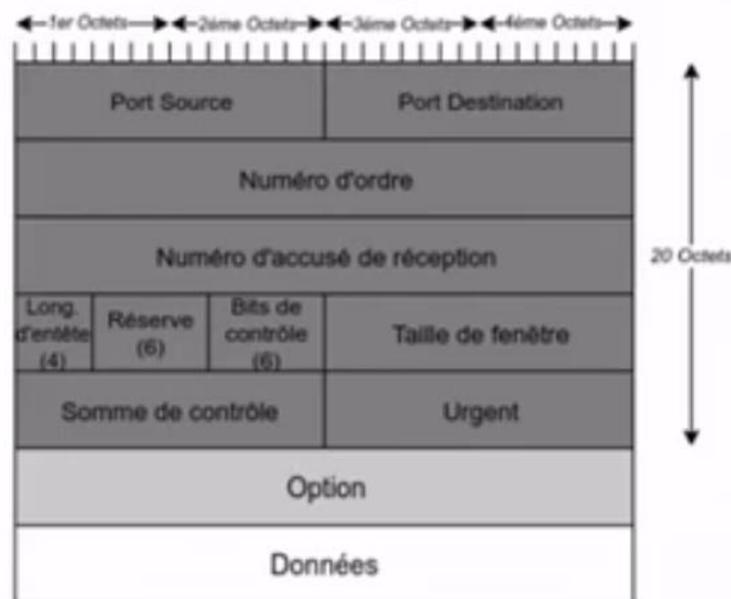
# Communication TCP

## 1- Établir la connexion



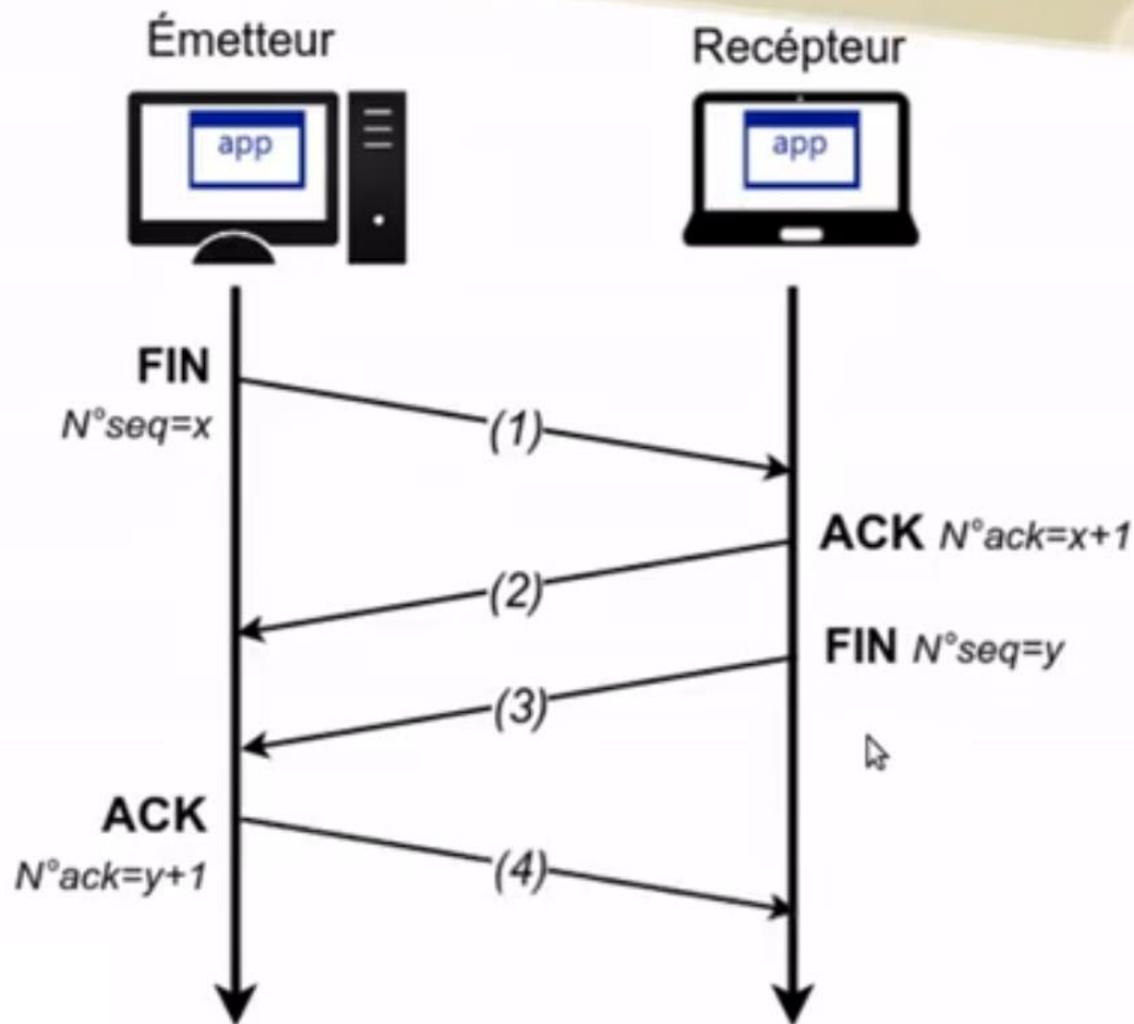
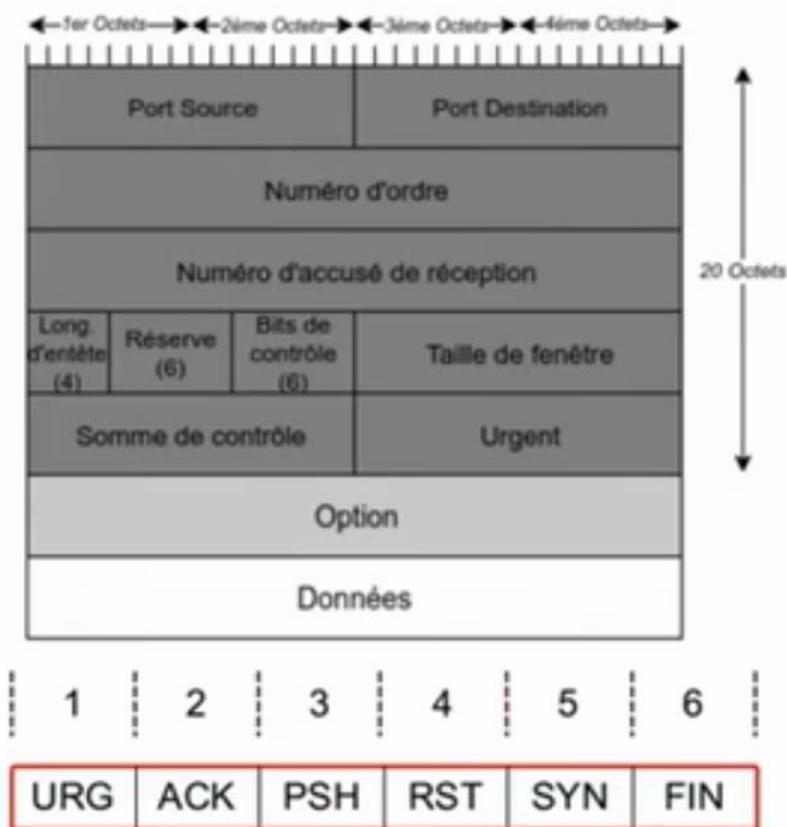
# Communication TCP

## 2) Envoie de segments



# Communication TCP

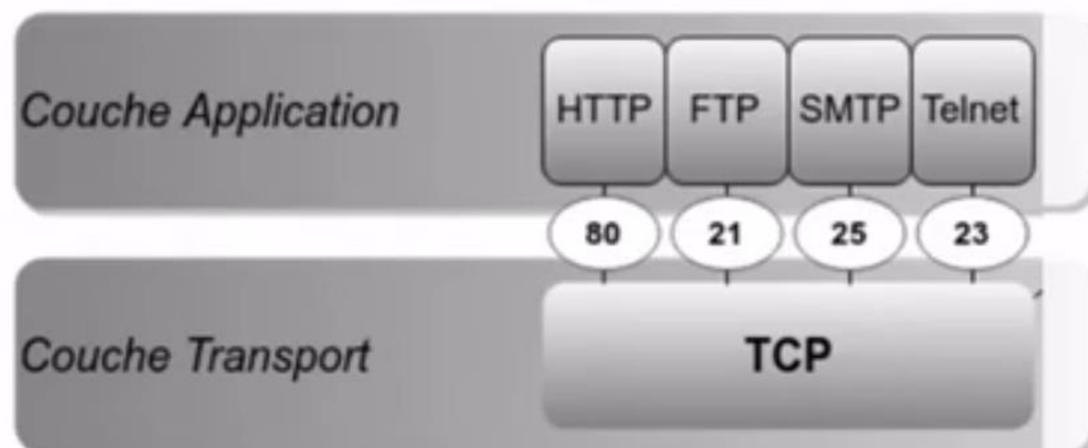
## 3- Déconnexion

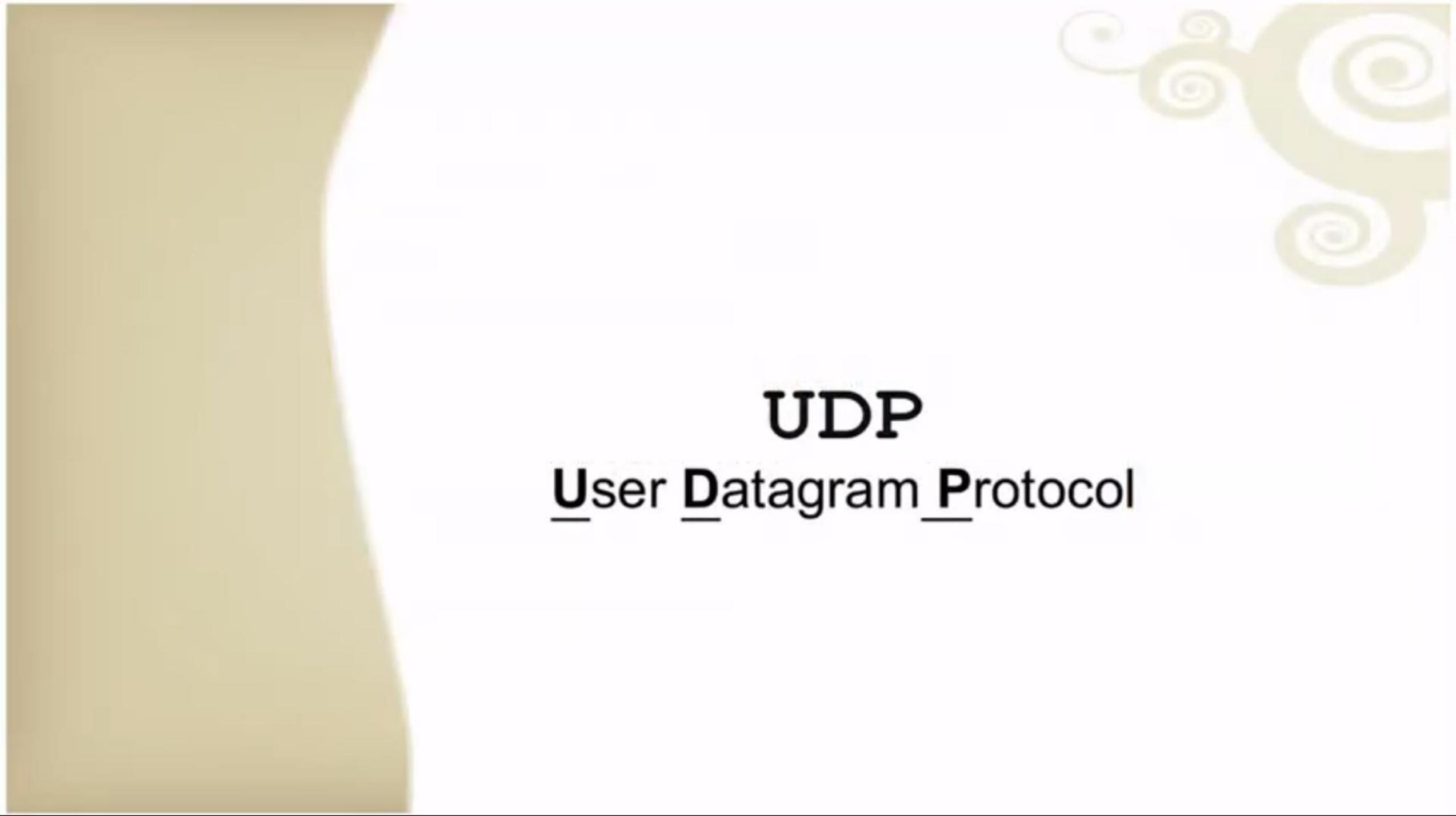


# TCP

## Exemple d'applications utilisant le protocole TCP:

- Les navigateurs Web : avec le protocole HTTP et FTP;
- La messagerie: avec le protocole SMTP et IMAP;
- Applications telles que les bases de données.





**UDP**

**User Datagram Protocol**

# UDP

Le protocole UDP fournit uniquement les fonctions de transport de base entre les applications appropriées.

Transport non fiable

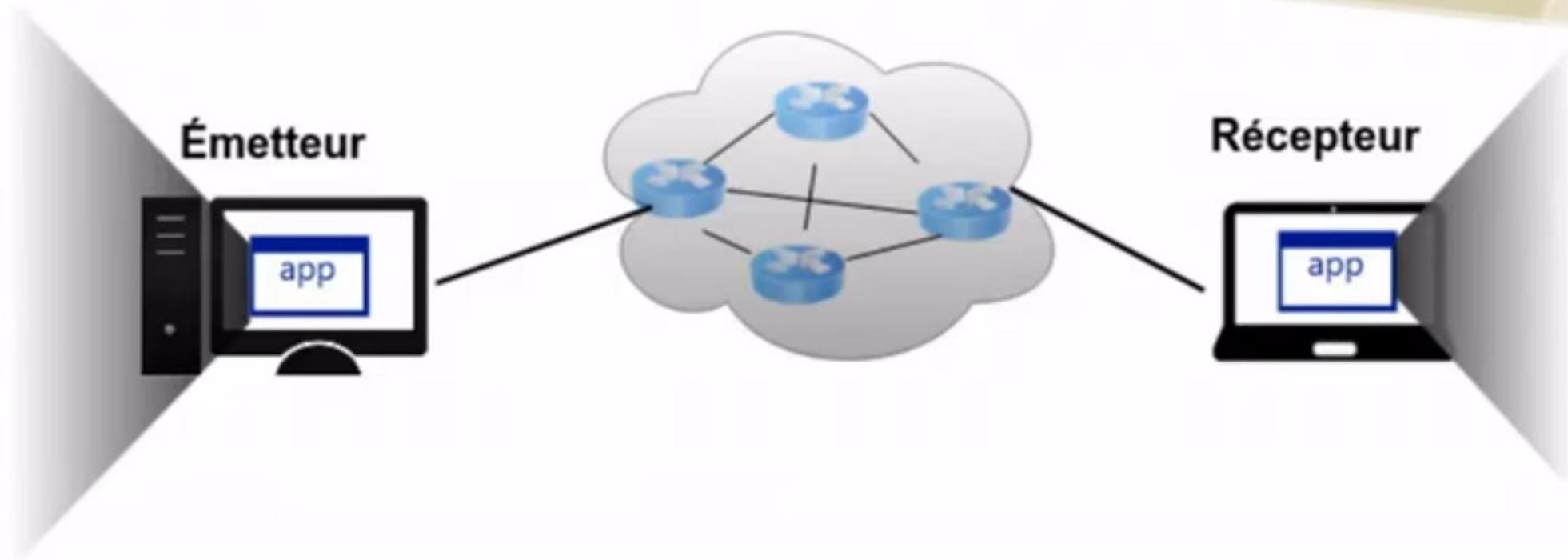
Nom PDU -> Datagramme

# Caractéristiques du UDP

1. **Encapsulation;**
2. **Sans connexion** : avant l'envoi des datagrammes, l'émetteur n'établit pas de connexion avec le récepteur;
3. **Acheminement au mieux ( non fiable)** : certaines applications peuvent tolérer une certaine perte lors de la transmission de données sur le réseau, mais pas les retards de transmission. Avec l'UDP il n'y a pas de processus permettant de faire retransmettre à l'émetteur les datagrammes perdues ou endommagées. Ainsi, pas de surcharge supplémentaire;
4. **Pas de contrôle de flux** : si les ressources du récepteur sont surexploitées, il abandonne les datagrammes reçus jusqu'à ce que ses ressources soient disponibles;
5. **Pas d'ordonnement des datagrammes** :

# Caractéristiques du UDP

- Datagramme 1
- Datagramme 2
- Datagramme 3
- Datagramme 4
- Datagramme 5
- Datagramme 6
- Datagramme 7
- Datagramme 8



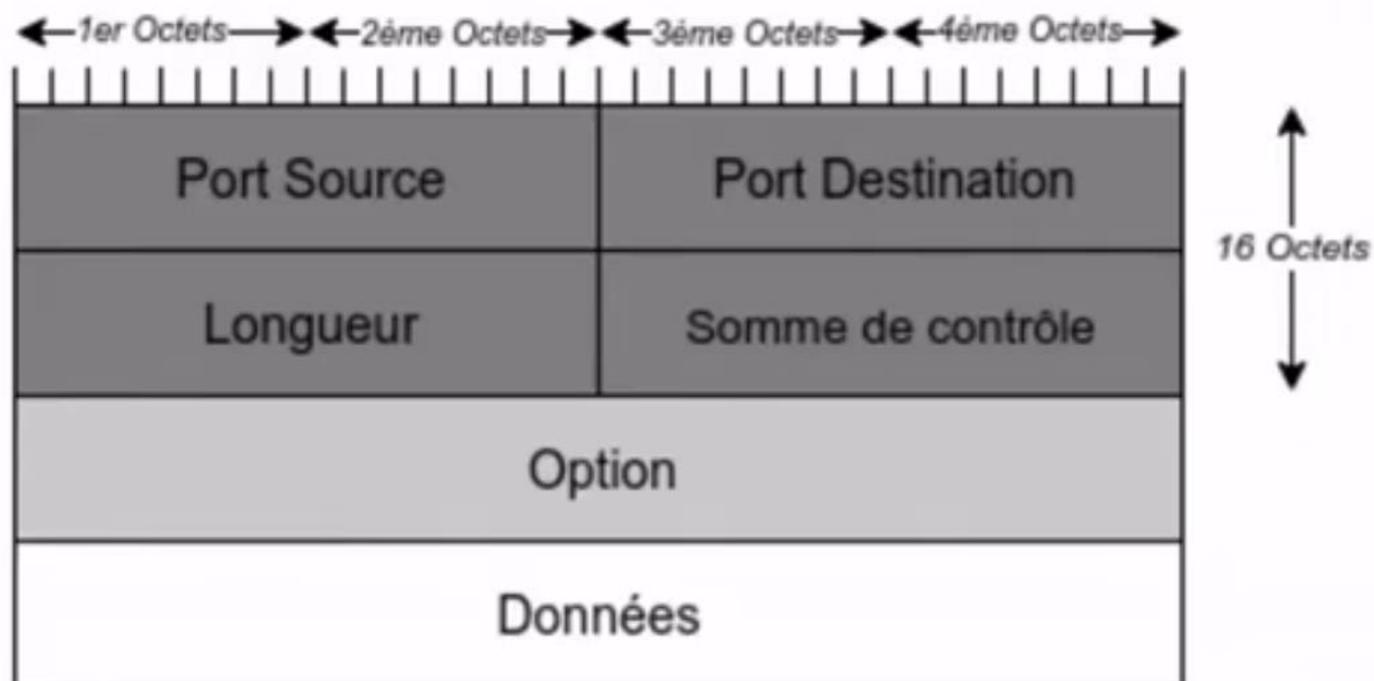
- Datagramme 1
- Datagramme 4
- Datagramme 3
- Datagramme 8
- Datagramme 5
- Datagramme 7

# En-tête UDP

Taille de l'entête réduite:

Supprimer:

- Service avec connexion,
- La fiabilité;
- Contrôle du flux.



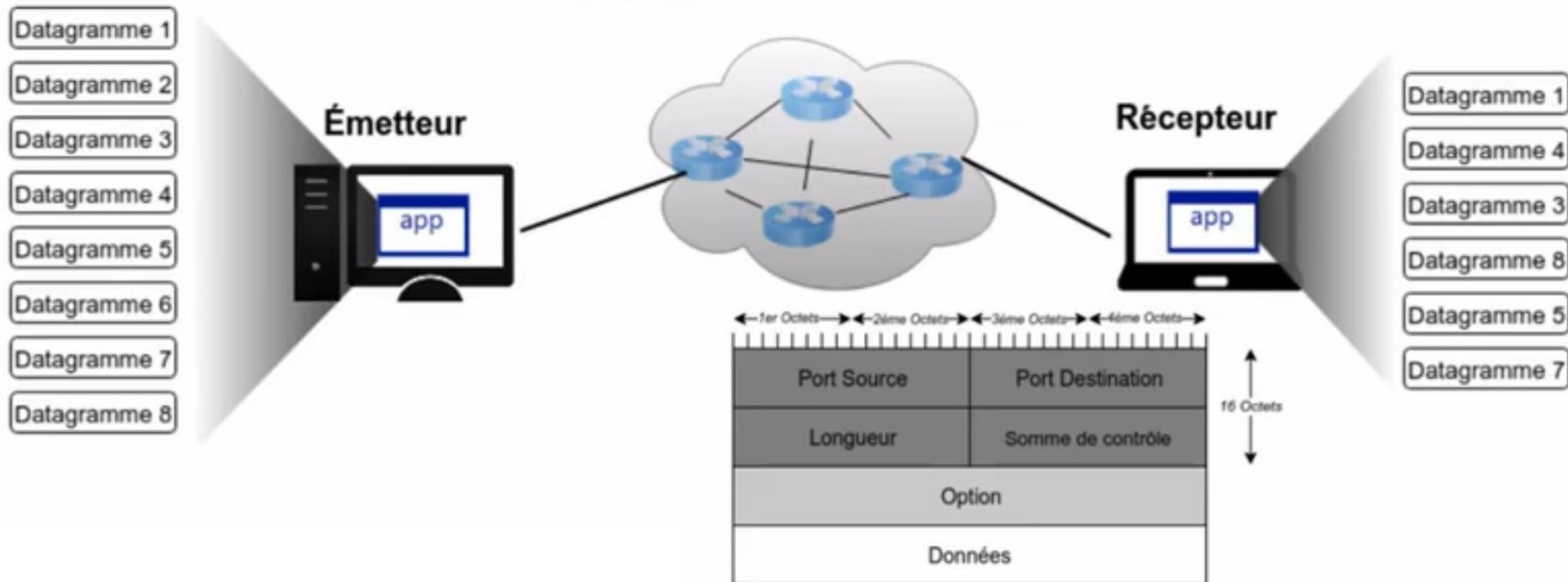
# UDP

Le choix entre TCP et UDP est suivant l'importance accordée à:

- la **fiabilité**
- la **charge imposée au réseau**.

Exemple d'applications utilisant le protocole UDP:

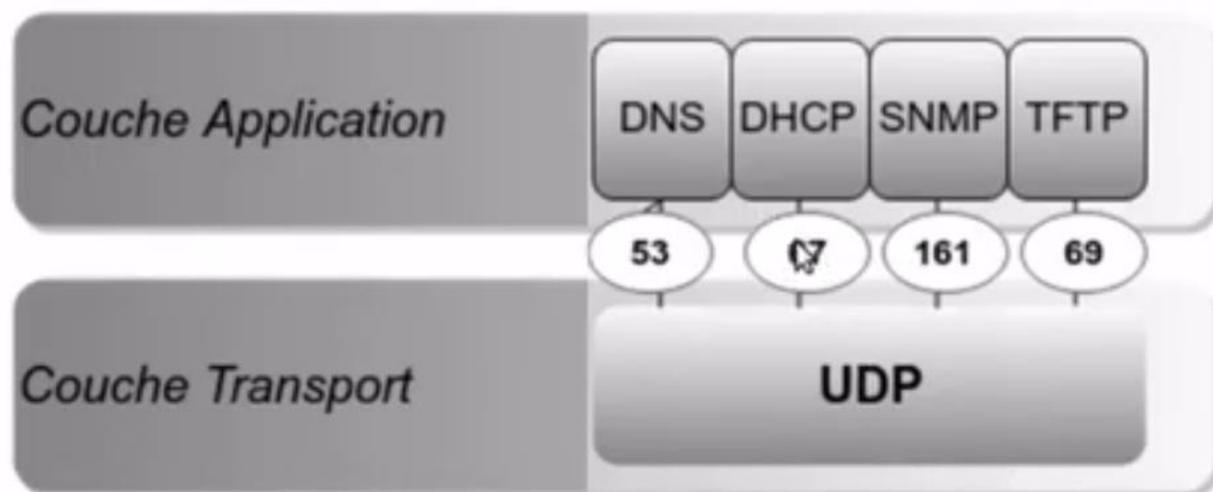
- Lecture audio/vidéo en continu (VoIP, Jeux en ligne) :



# UDP

Exemple d'applications utilisant le protocole UDP:

- Système de noms de domaines: DNS;
- Administration réseau : avec le protocole SNMP;
- Configuration automatique des paramètres IP : Protocole DHCP;





# **Adressage de la couche transport**

# Introduction

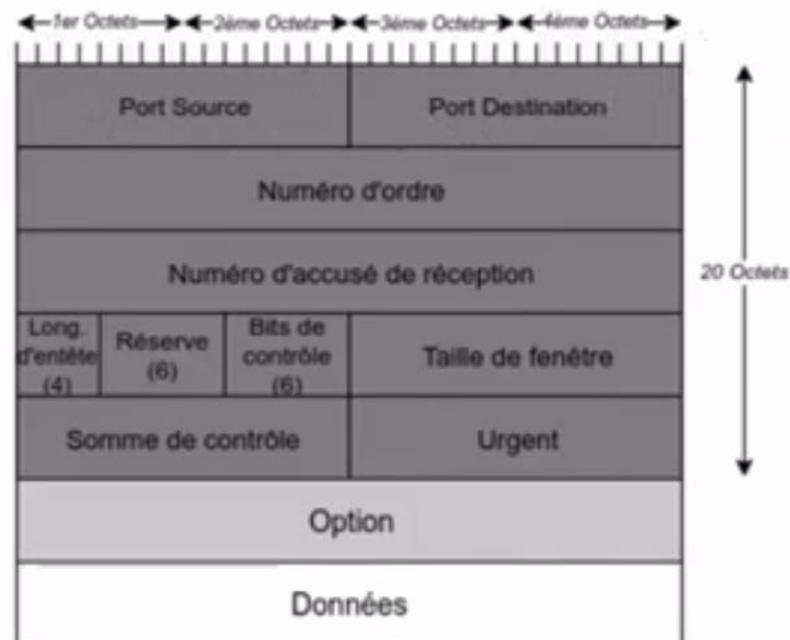
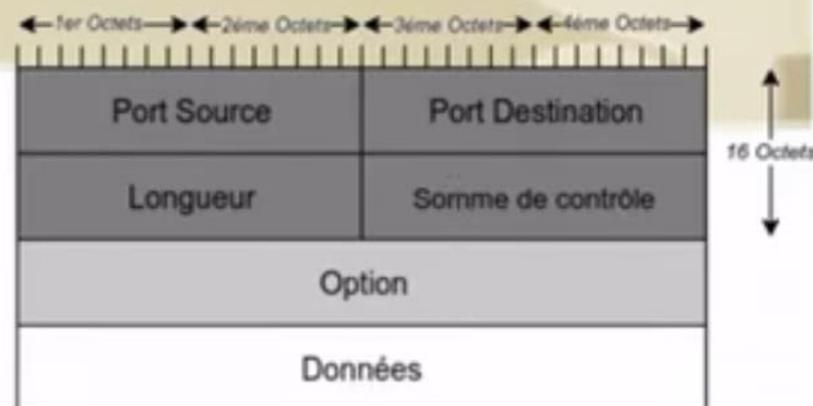
Il est considéré comme étant le 3ème type d'adresse.

Le numéro de port source est associé à l'application de l'émetteur

Le numéro de port de destination à celle du destinataire.

Un port est un identifiant numérique de valeur **décimale**

$$2^{16} = 65\ 536$$



# Les différents numéros de port

L'Internet Assigned Numbers Authority (IANA) attribue les numéros de port.

Il existe 3 catégories de numéros de port :

- 1. Ports réservés [0 - 1 023]** : sont des ports connus ou réservés (well known ports). Ils sont affectés aux processus système ou aux *programmes exécutés par les serveurs*;
- 2. Ports inscrits ou enregistrés [1 024 - 49 151]** : ils sont attribués par l'IANA pour le service spécifique à la suite d'une demande d'un utilisateur ou d'une entreprise de développement d'applications;
- 3. Ports privés ou dynamiques [49 152 - 65 535]** : ne sont pas enregistrés auprès de l'IANA. Ils sont utilisés pour des applications privées ou pour des besoins temporaires. Ils sont affectés de façon dynamique à des *applications clientes* lorsqu'une connexion à un service est initiée par un client. Certains services de partage de fichiers peer-to-peer utilisent ces ports.

## Les différents numéros de port

	<b>TCP</b>	<b>UDP</b>
<b>Ports réservés</b>	21 : FTP	67 : DHCP
	32 : Telnet	69 : TFTP
	25 : SMTP	520 : RIP
	80 : HTTP	
	143 : IMAP	
	443 : HTTPs	
<b>Ports inscrits</b>	1863 : MSN Messenger	5004 : RTP
	2000 : Cisco SCCP VoIP	5040 : SIP VoIP

# Utilisation du protocole TCP et du protocole UDP

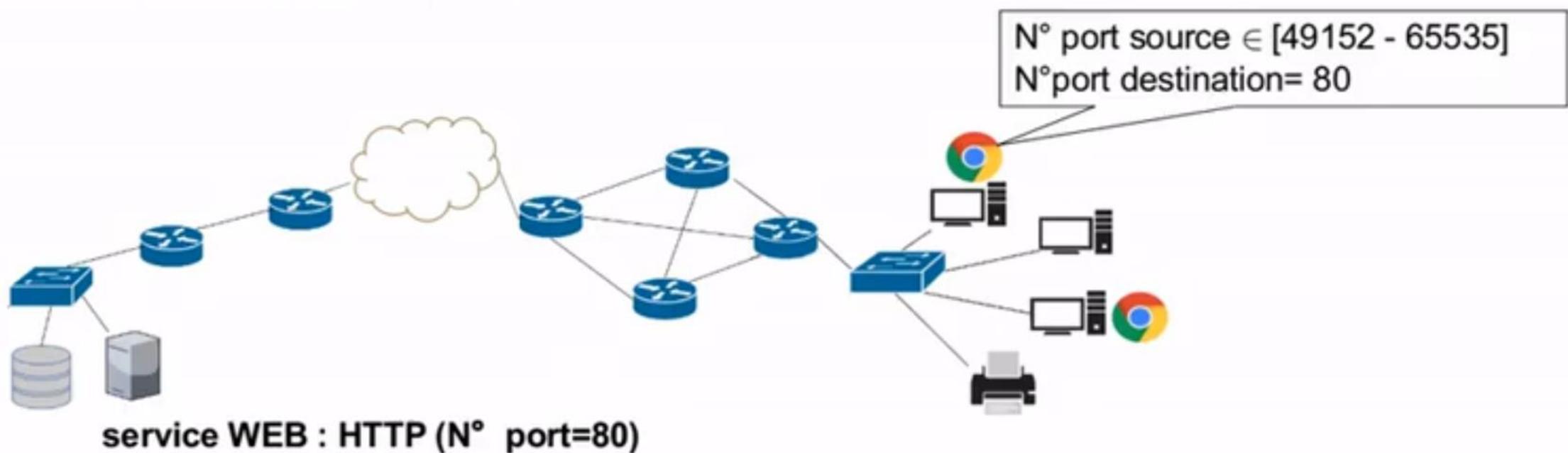
Certaines applications ou services peuvent utiliser le protocole TCP dans certains cas et le protocole UDP dans d'autres.

La faible surcharge du protocole UDP permet au service DNS de gérer très rapidement de nombreuses requêtes de clients. Mais dès que la taille de la donnée augmente, ce protocole utilise le mode TCP.

	<b>TCP</b>	<b>UDP</b>
<b>Ports réservés</b>	53 : DNS	
	161 : SNMP	
<b>Ports inscrits</b>	1433 : MS SQL	

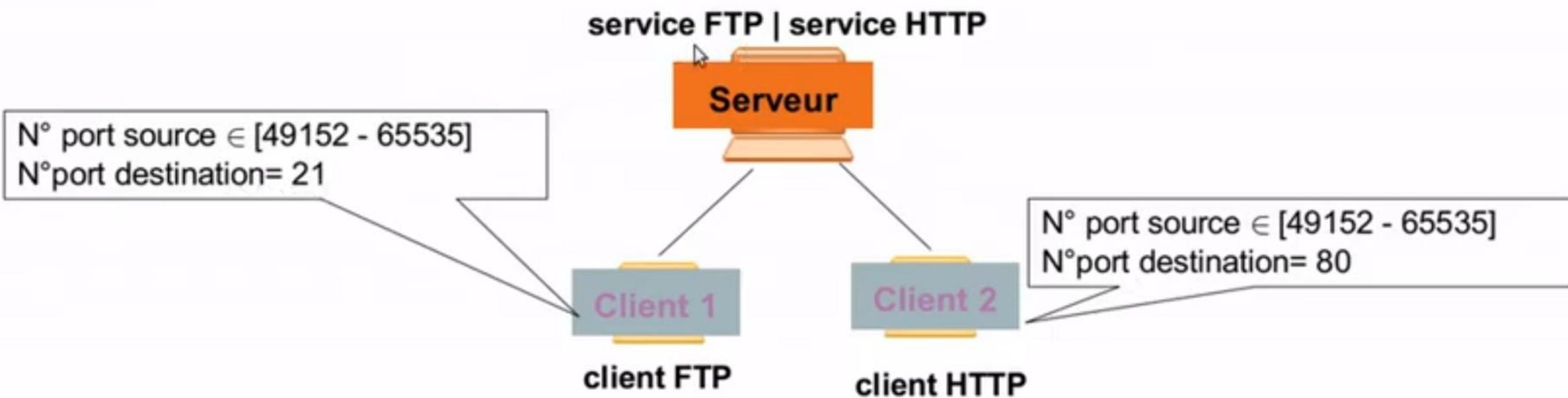
# Les différents numéros de port

À savoir que les numéros de port des applications clients **sont attribués, de façon plus ou moins aléatoire**, par le système d'exploitation de la machine hôte.



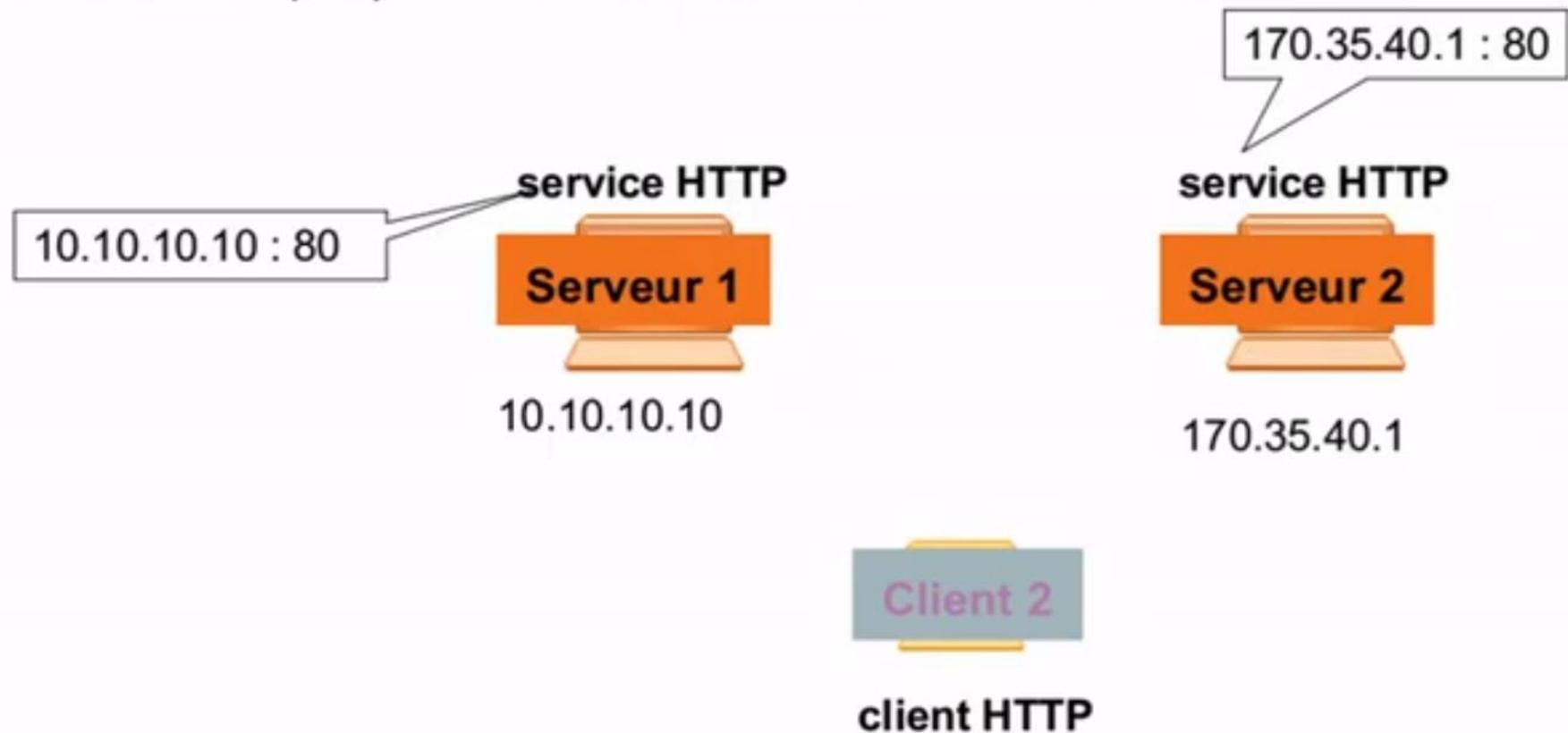
# Les différents numéros de port

Le port de destination (côté client) permet d'informer le serveur de destination du service demandé.



# Les différents numéros de port

combinaison (adresse IP + N° port) -> **Socket** ou **Interface de connexion**



# Utilitaire de connexions

L'utilitaire netstat permet de connaître quelles connexions TCP actives sont ouvertes et s'exécutent sur un hôte en réseau.

```
C:\> netstat
```

```
Active Connections
```

Proto	Local Address	Foreign Address	State
TCP	kenpc:3126	192.168.0.2:netbios-ssn	ESTABLISHED
TCP	kenpc:3158	207.138.126.152:http	ESTABLISHED
TCP	kenpc:3159	207.138.126.169:http	ESTABLISHED
TCP	kenpc:3160	207.138.126.169:http	ESTABLISHED
TCP	kenpc:3161	sc.msn.com:http	ESTABLISHED
TCP	kenpc:3166	www.cisco.com:http	<b>ESTABLISHED</b>

```
C:\>
```