

# Chapitre 7 : La couche transport



## Initiation aux réseaux



# Chapitre 7

7.1 Protocoles de la couche transport

7.2 TCP et UDP

7.3 Résumé



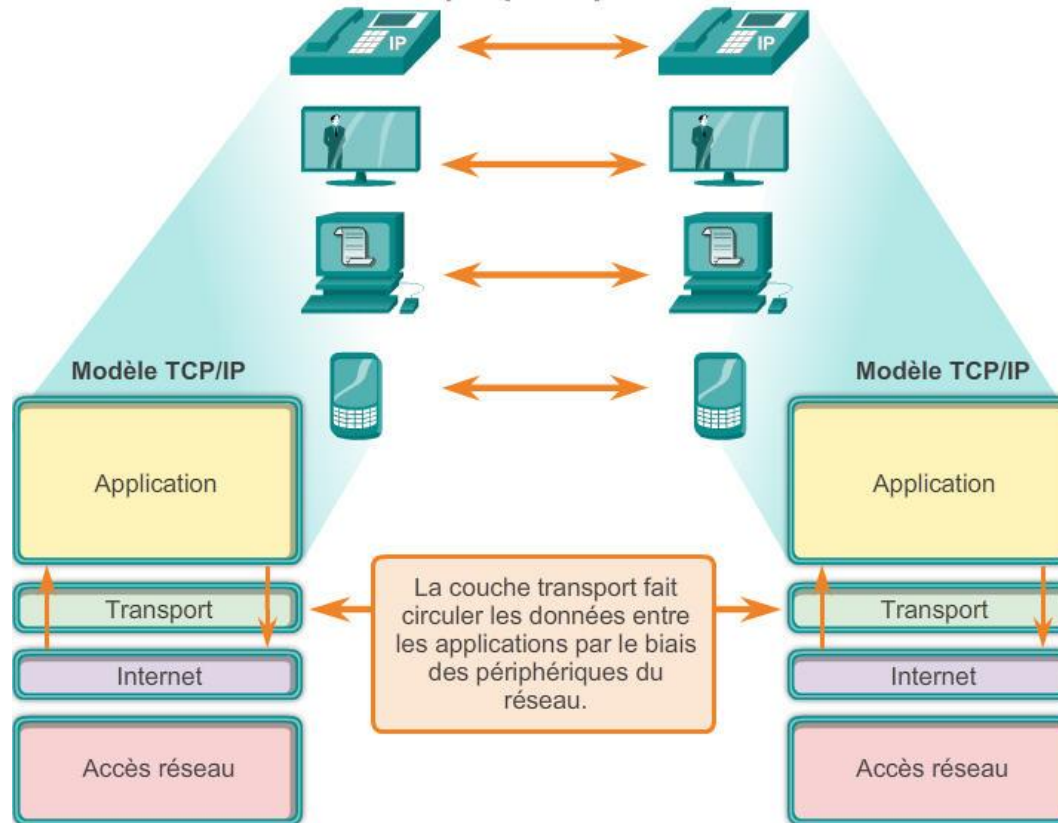
# Chapitre 7 : Les objectifs

- Décrire le rôle de la couche transport dans la gestion du transport des données dans la communication de bout en bout
- Décrire les caractéristiques des protocoles TCP et UDP, y compris les numéros de port et leur utilisation
- Expliquer comment les processus d'établissement et d'interruption de session TCP garantissent la fiabilité des communications
- Expliquer comment les unités de données de protocole TCP sont transmises et comment leur réception est confirmée pour garantir l'acheminement des données
- Expliquer les processus client UDP permettant d'établir la communication avec un serveur
- Déterminer si des transmissions TCP haute fiabilité ou des communications UDP non garanties sont les plus adaptées aux applications fréquemment utilisées



# Rôle de la couche transport

Activation de la communication des applications sur les périphériques





## Transport des données

# Rôle de la couche transport

- Le rôle de la **couche transport** est d'établir une session de communication temporaire entre deux applications pour acheminer les données entre elles.
- TCP/IP utilise deux protocoles pour cela :
  - TCP (Transmission Control Protocol)
  - UDP (User Datagram Protocol)
- Fonctions principales des protocoles de la couche transport :
  - Suivre les communications individuelles entre les applications résidant sur les hôtes source et de destination
  - Segmenter les données pour faciliter la gestion et réassembler les données segmentées en flux de données d'application vers la destination
  - Identifier l'application appropriée pour chaque flux de communication

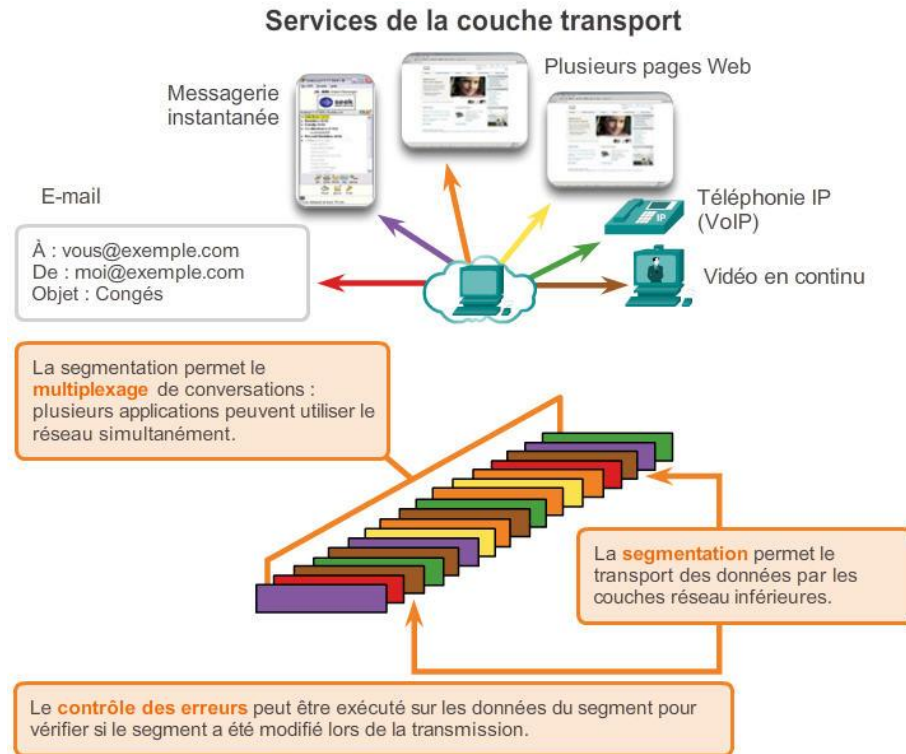


## Transport des données

# Fonctions de la couche transport

### La segmentation des données

- Permet de multiplexer sur le même réseau différentes communications provenant de nombreux utilisateurs.
- Permet d'envoyer et de recevoir des données tout en exécutant plusieurs applications.
- Un en-tête est ajouté à chaque segment pour l'identifier.
- La segmentation facilite la reprise sur erreur et la retransmission des données endommagées.





## Transport des données

# Fiabilité de la couche transport

- La couche transport est également responsable de la gestion des exigences de fiabilité d'une conversation
- Toutes les applications n'ont pas besoin du même degré de fiabilité.
- TCP/IP fournit deux protocoles de la couche transport, **TCP et UDP**.

## Transmission Control Protocol (TCP)

- Assure un acheminement fiable – Toutes les données arrivent à destination
- Surcharge le réseau avec les contrôles

## User Datagram Protocol (UDP)

- Fournit juste les fonctions de base pour la transmission, sans aucune garantie
- Moins de surcharge



## Présentation des protocoles TCP et UDP

# Présentation du protocole TCP

## Transmission Control Protocol (TCP) RFC 793

- Orienté connexion : création d'une session entre la source et la destination (prépare les périphériques à communiquer entre eux)
  - les périphériques négocient la quantité de trafic pouvant être transmise à un moment donné
  - les données de communication peuvent être étroitement gérées
- Acheminement fiable : retransmission des données perdues ou endommagées
- Utilise les accusés de réception et d'autres mécanismes pour garantir la transmission
- Garantie l'ordre : numérotation et séquençement des segments
- Contrôle de flux: quand le protocole TCP détermine que ces ressources sont surexploitées, il peut demander à l'application qui envoie les données d'en réduire le flux
- Protocole avec état : il conserve une trace du nombre de segments qui sont été envoyés à un hôte donné à partir d'une application spécifique

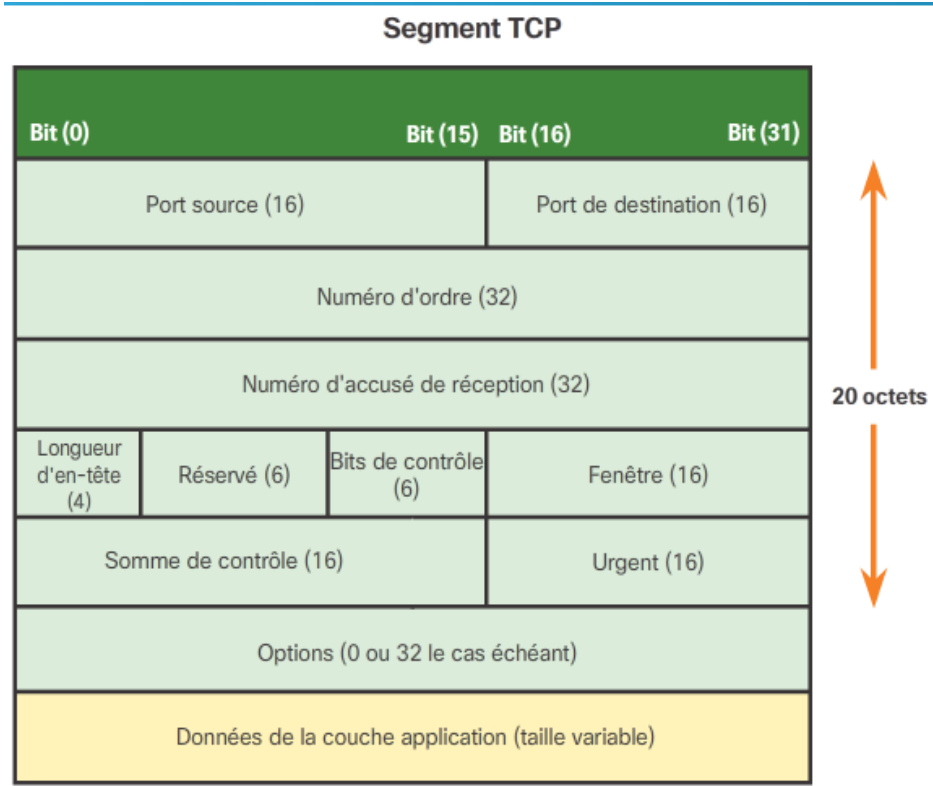




# Présentation des protocoles TCP et UDP

## Le protocole TCP

- Ports source- destination** – identifient les processus source et destination
- numéro d'ordre (32 bits)** – numéro du message envoyé.
- numéro d'accusé de réception (32 bits)** – indique les données qui ont été reçues et le message reçu.
- champ Longueur d'en-tête (4 bits)** – Indique la longueur de l'en-tête du segment TCP (exprimé en mots de 4 octets).
- champ Réserve (6 bits)** - champ réservé pour les futures évolutions.
- Des bits de contrôle (6 bits)** – indiquant l'objectif et la fonction du segment TCP.
- La taille de fenêtre (16 bits)** – indique la taille des données qui peuvent être acceptées en même temps.
- somme de contrôle (16 bits)** – contrôle des erreurs sur l'en-tête et les données de segment.
- Le champ Urgent (16 bits)** – indique la position des données urgentes.





## Présentation des protocoles TCP et UDP

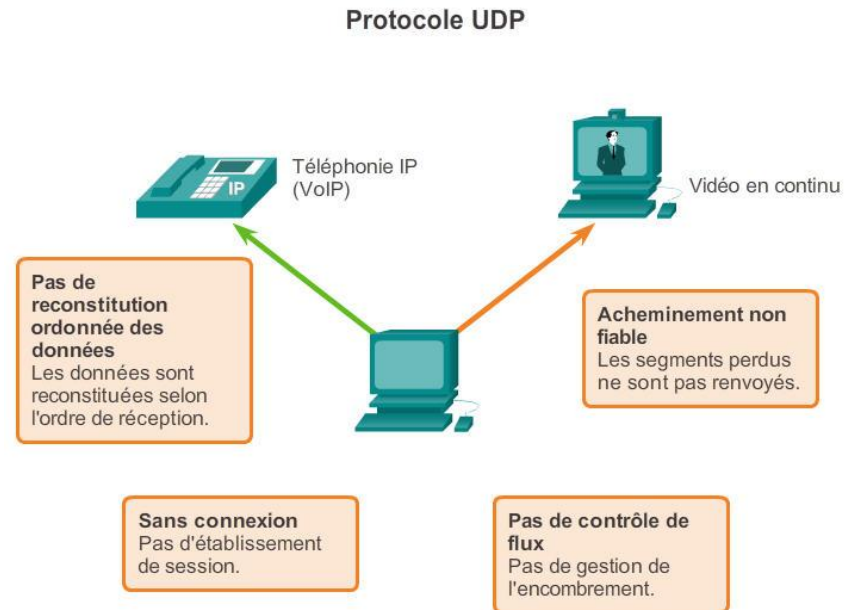
# Présentation du protocole UDP

## User Datagram Protocol (UDP)

- RFC 768
- Mode non connecté
- Sans négociation préalable
- Sans garantie de remise
- Sans reconstitution ordonnée des données
- Sans contrôle de flux
- Acheminement au mieux

### Applications utilisant UDP :

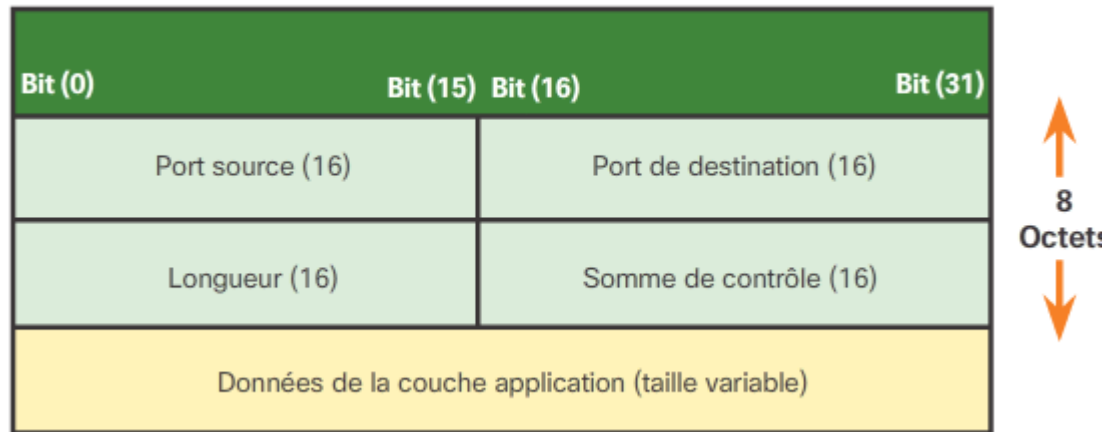
- Système de noms de domaine (DNS)
- Lecture vidéo en continu
- Voix sur IP (VoIP)





## Présentation des protocoles TCP et UDP

# Présentation du protocole UDP

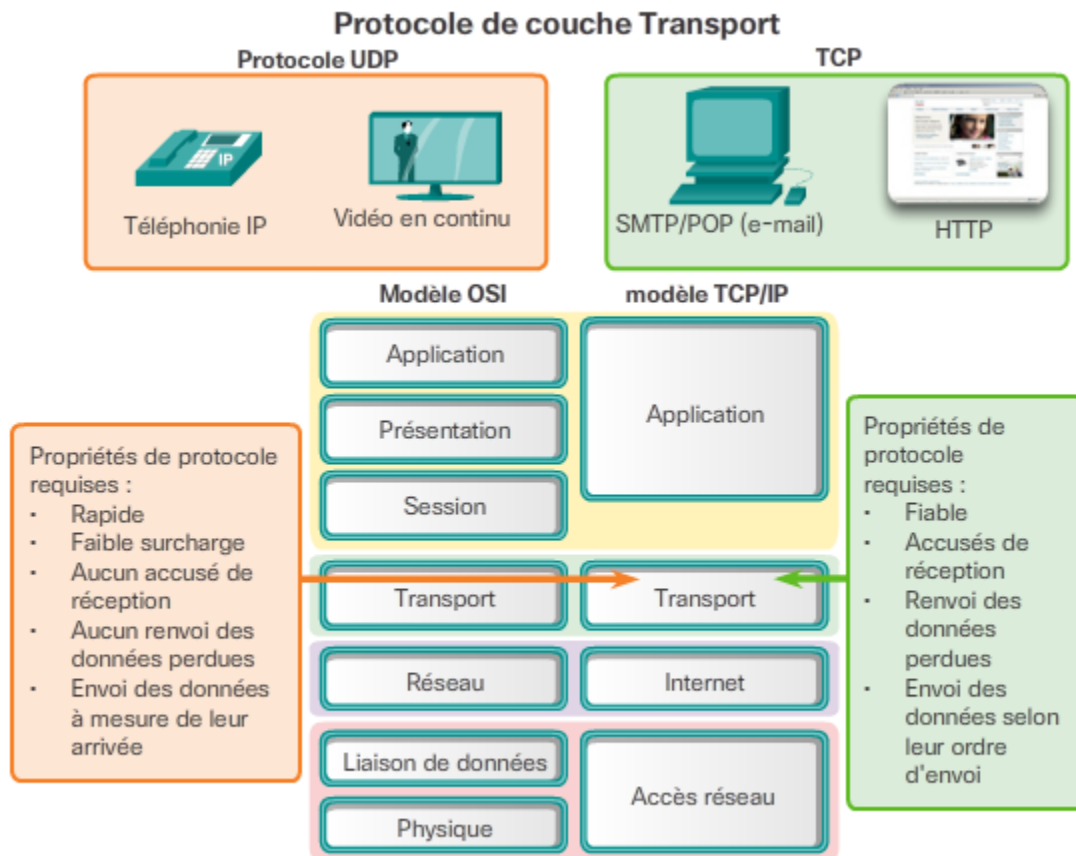


- Les unités de données du protocole UDP sont appelés des datagrammes
  - Ports source et destination: identifient les processus en communication
  - Longueur du datagramme
  - Contrôle d'erreur facultatif



# Présentation des protocoles TCP et UDP

## Le bon protocole de couche transport pour la bonne application



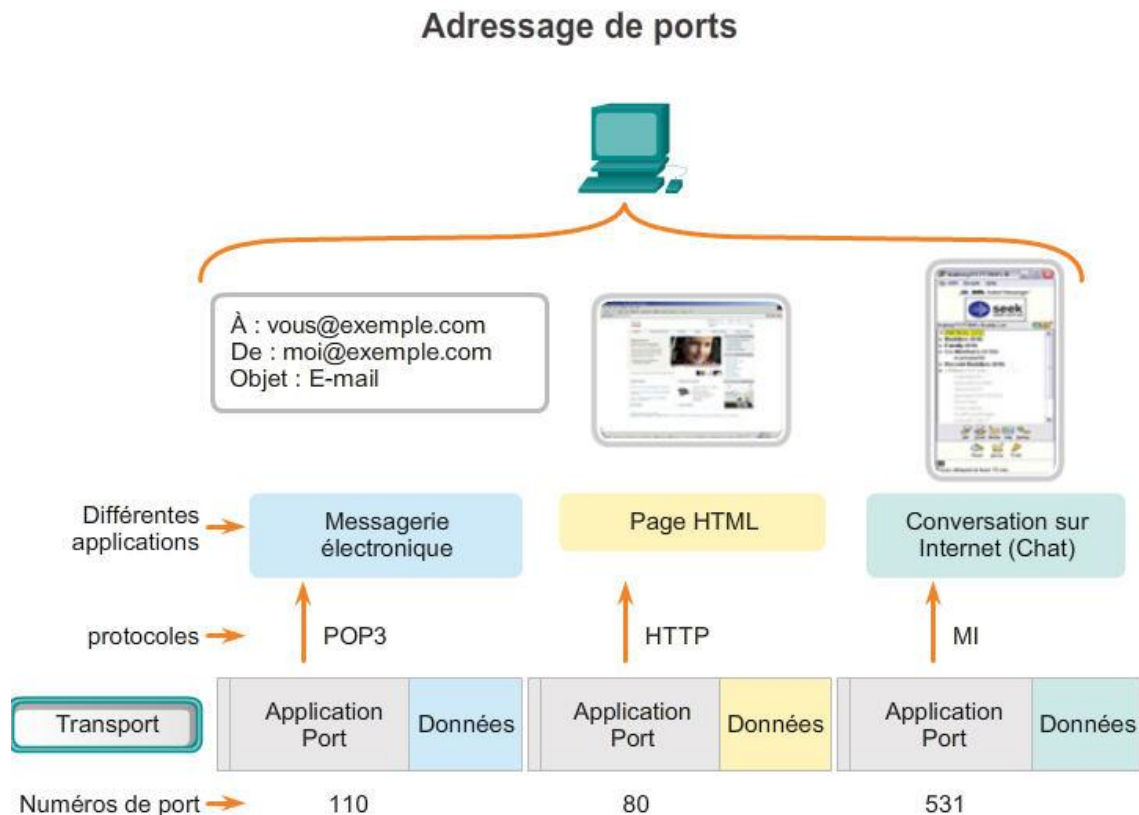
Les développeurs d'applications choisissent le protocole de couche transport approprié en fonction de la nature de l'application.



## Initiation aux protocoles TCP et UDP

# Séparation des communications multiples

Les numéros de port sont utilisés par les protocoles TCP et UDP pour différencier les applications.



Les données des différentes applications sont dirigées vers l'application adéquate car chaque application dispose d'un numéro de port unique.

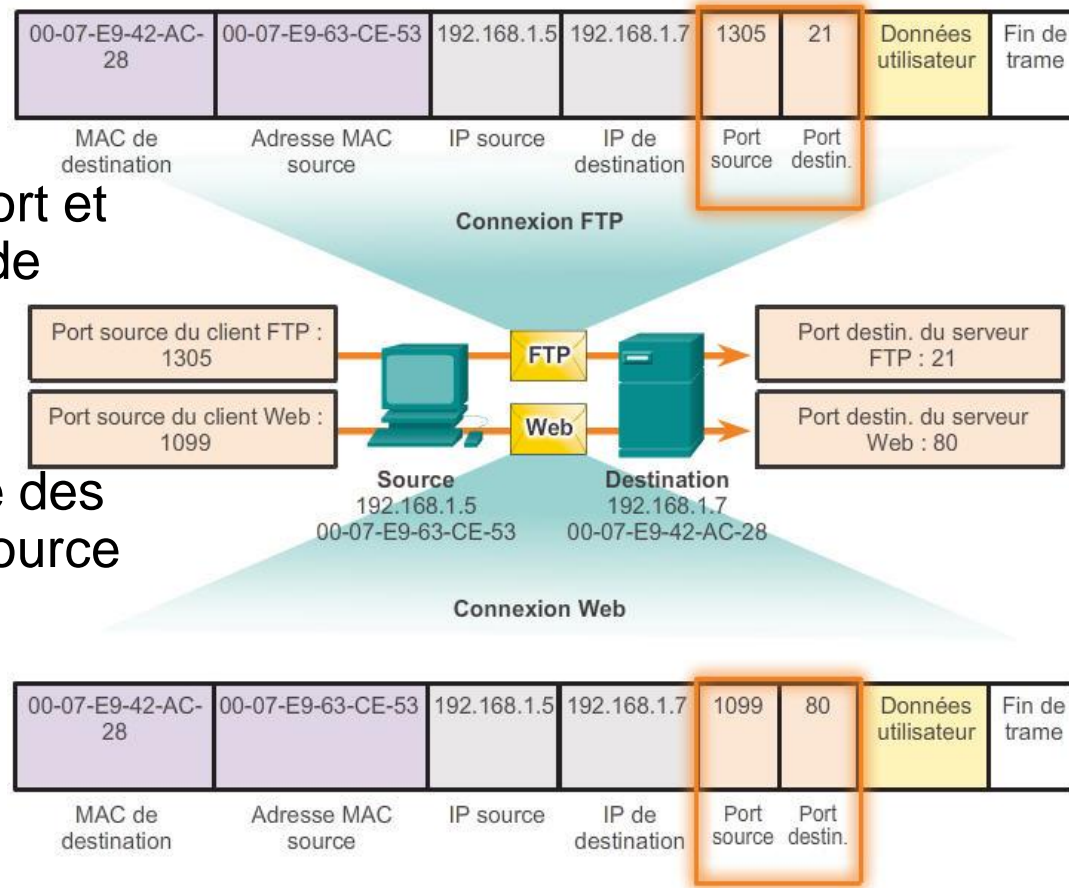


## Initiation aux protocoles TCP et UDP

# Adressage de ports TCP et UDP

■ la combinaison du numéro de port et de l'adresse IP de l'hôte identifie de manière unique un processus d'application (socket)

■ Une paire de sockets, composée des adresses IP et numéros de port source et de destination identifie d'une manière unique la conversation spécifique entre deux hôtes.







## Initiation aux protocoles TCP et UDP

# Adressage de ports TCP et UDP

### Numéros de port

Plage de numéros de port	Groupe de ports
0 à 1023	Ports réservés
De 1024 à 49151	Ports inscrits
49152 à 65535	Ports dynamiques et/ou privés

- Ports réservés: numéros réservés à des services et applications connus
- Ports Inscrits: numéros affectés à des processus ou applications particulières d'utilisateurs. Aussi utilisés par les processus client (si non affecté à un service)
- Ports privés ou dynamiques: généralement affectés de façon dynamique à des applications clientes lorsqu'une connexion à un service est initiée par un client.

#### Ports TCP réservés :

21 FTP  
 23 Telnet  
 25 SMTP  
 80 HTTP  
 143 IMAP  
 194 Internet Relay Chat (IRC)  
 443 Secure HTTP (HTTPS)

#### Ports UDP réservés :

69 TFTP  
 520 RIP

#### Ports TCP/UDP réservés courants :

53 DNS  
 161 SNMP  
 531 AOL Instant Messenger, IRC

#### Ports TCP inscrits :

1863 MSN Messenger  
 2000 Cisco SCCP (VoIP)  
 8008 Alternate HTTP  
 8080 Alternate HTTP

#### Ports UDP inscrits :

1812 RADIUS Authentication Protocol  
 5004 RTP (Voice and Video Transport Protocol)  
 5040 SIP (VoIP)



## Initiation aux protocoles TCP et UDP

# Adressage de ports TCP et UDP

## Netstat

- Permet d'examiner les connexions TCP qui sont ouvertes et actives sur un hôte connecté au 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  ESTABLISHED

C:\>
```

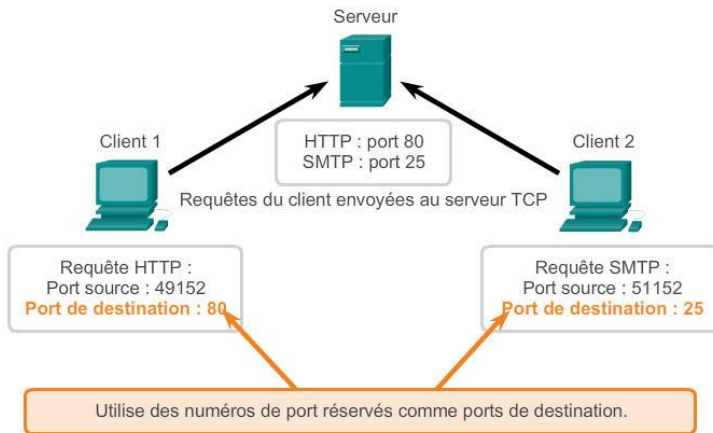




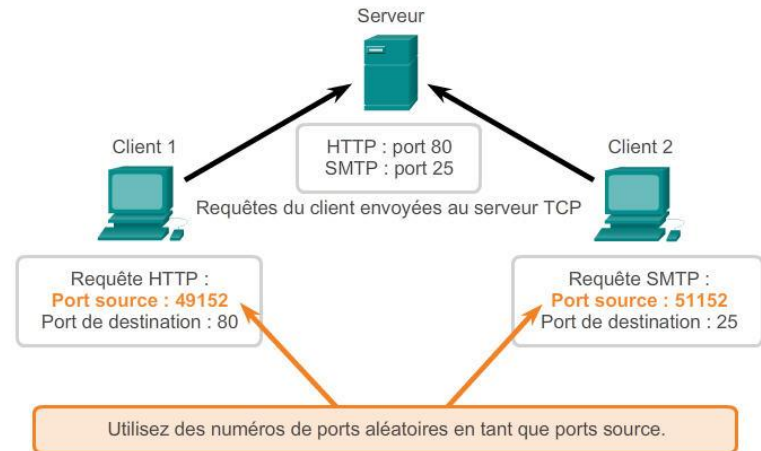
# Communication TCP

## Processus serveur TCP

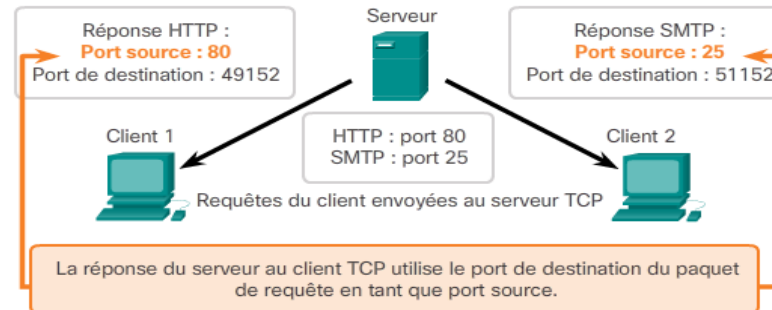
Ports de destination des requêtes



Ports source des requêtes



Ports source des réponses



Requête HTTP :  
Port source : 49152  
Port de destination : 80

Requête SMTP :  
Port source : 51152  
Port de destination : 25



# Établissement et fermeture d'une connexion TCP

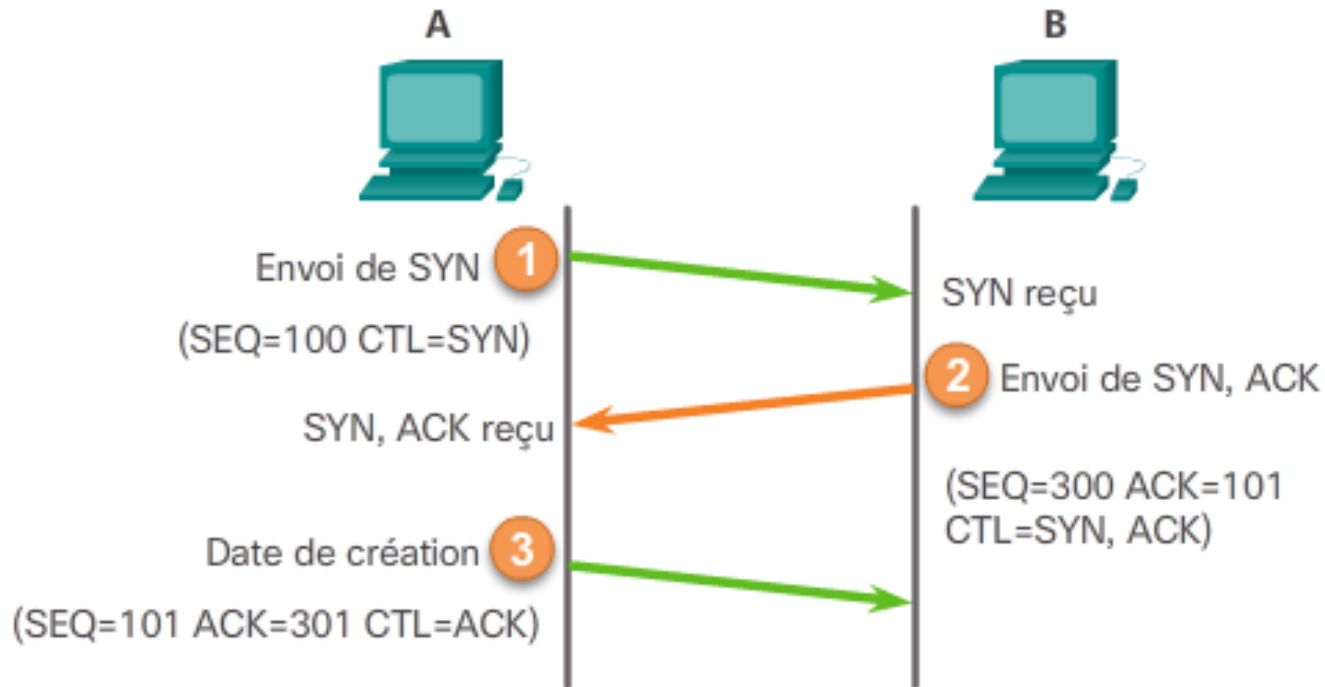
## Connexion en trois étapes

- Vérifie que le périphérique de destination est bien présent sur le réseau
- S'assure que le périphérique de destination a un service actif sur le numéro de port de destination utilisé par le client
- Informe le périphérique de destination que le client source souhaite établir une session de communication sur ce numéro de port



## Communication TCP

# Connexion TCP en trois étapes

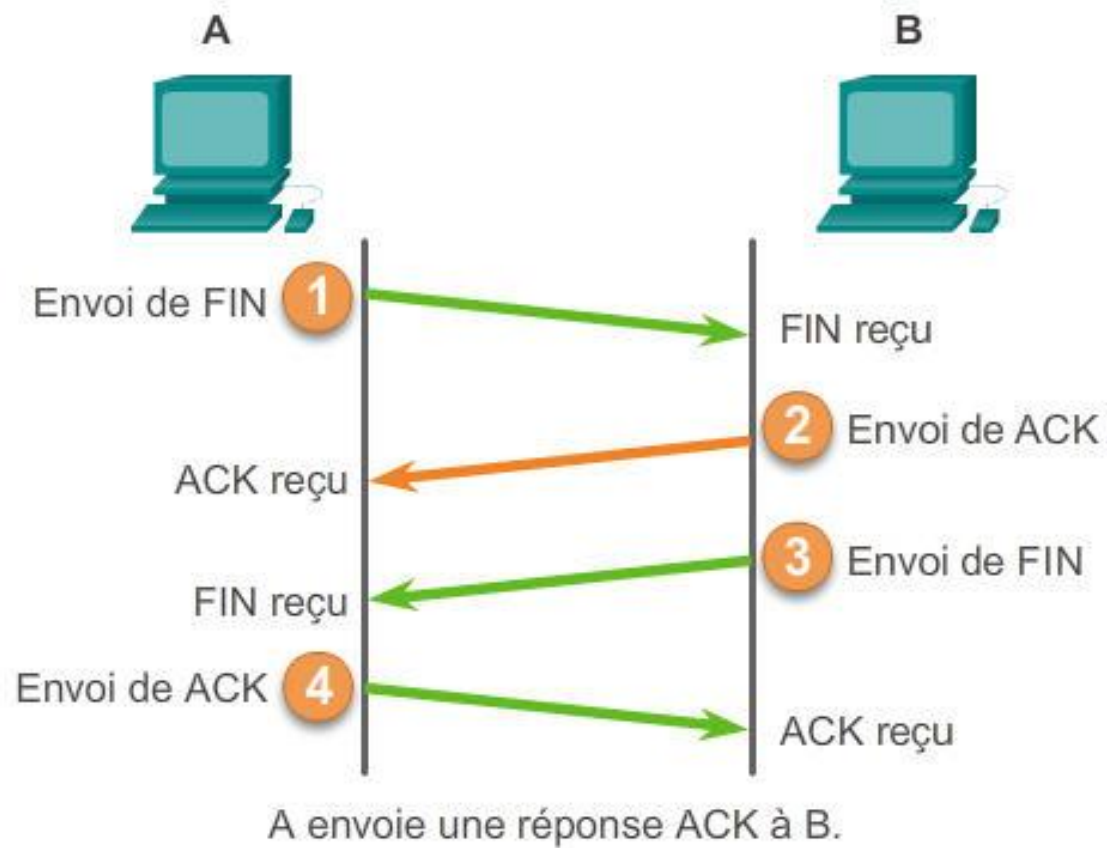


CTL = nature des bits de contrôle de l'en-tête TCP définis sur 1  
A envoie une réponse ACK à B.



## Communication TCP

# Fermeture de la session TCP

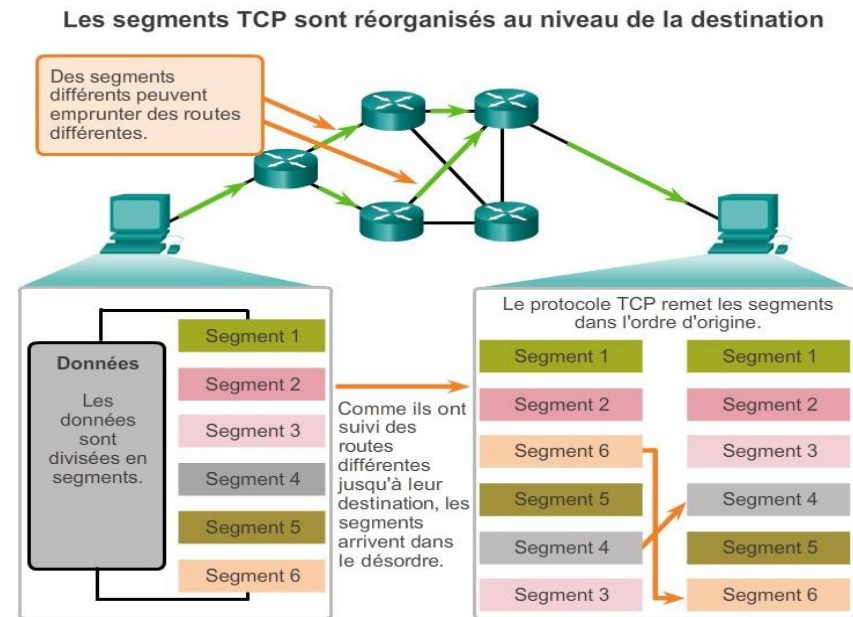




# Fiabilité et contrôle de flux

## Fiabilité du protocole TCP – Livraison ordonnée

- Numéros d'ordre utilisés pour remettre les segments dans l'ordre d'origine
- Lors de la configuration de la session, un numéro d'ordre initial, ou ISN, est défini
- Le premier segment a un numéro d'ordre ISN+1
- le numéro d'ordre est incrémenté du nombre d'octets ayant été transmis
- TCP récepteur place les données d'un segment dans une mémoire tampon.
- Les segments sont remis dans l'ordre correct et sont transmis à la couche application une fois qu'ils ont été réassemblés.

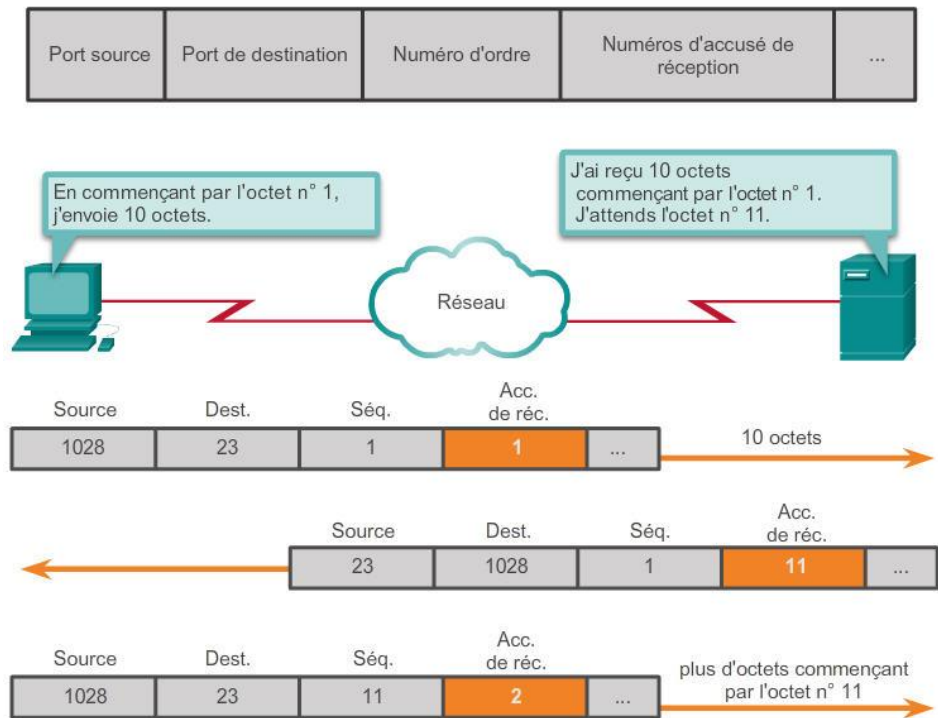




# Fiabilité du protocole TCP – Accusé de réception et taille de fenêtre

Le numéro d'ordre et celui de l'accusé de réception sont utilisés ensemble pour confirmer la réception.

Accusé de réception des segments TCP





# Fiabilité du protocole TCP – Accusé de réception

- TCP ne génère d'accusé de réception que pour les séquences contiguës d'octets
- Dans une implémentation TCP classique, un hôte transmet un segment, place une copie du segment dans une file d'attente de retransmission et lance un minuteur
- Quand l'accusé de réception des données est reçu, le segment est supprimé de la file d'attente.
- Si l'accusé de réception n'est pas reçu avant l'écoulement du délai prévu, le segment est considéré perdu et donc il sera retransmis.



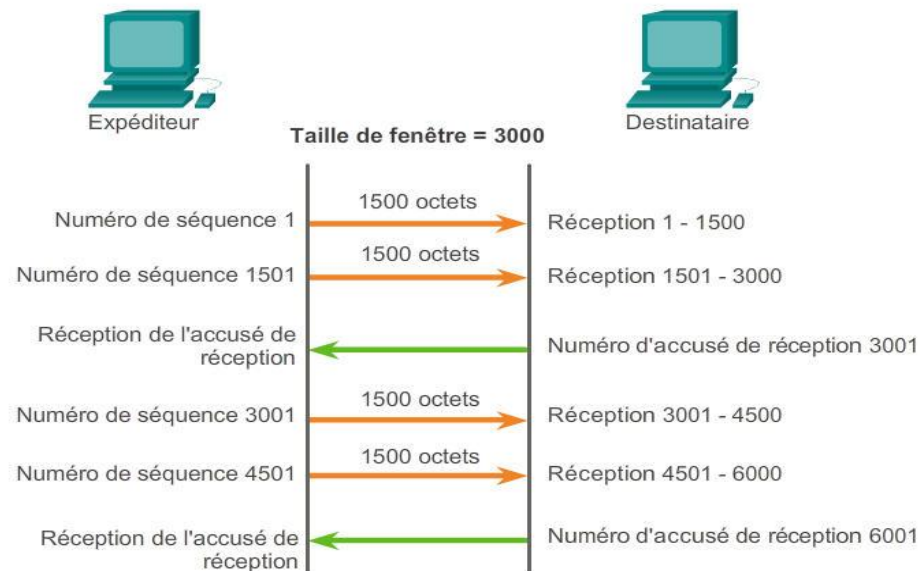


## Contrôle de flux et fiabilité du protocole TCP

# Taille de fenêtre et accusés de réception

- **Taille de fenêtre** : la quantité de données qu'une source peut transmettre avant qu'un accusé de réception arrive.
- **Le contrôle de flux** consiste à limiter la quantité de données de segments transférées en une seule fois et à demander des accusés de réception avant de transmettre davantage de données

Accusé de réception de segment TCP et taille de fenêtre



La **taille de fenêtre** détermine le nombre d'octets envoyés avant l'attente d'un accusé de réception.

Le numéro d'**accusé de réception** est le numéro du prochain octet attendu.

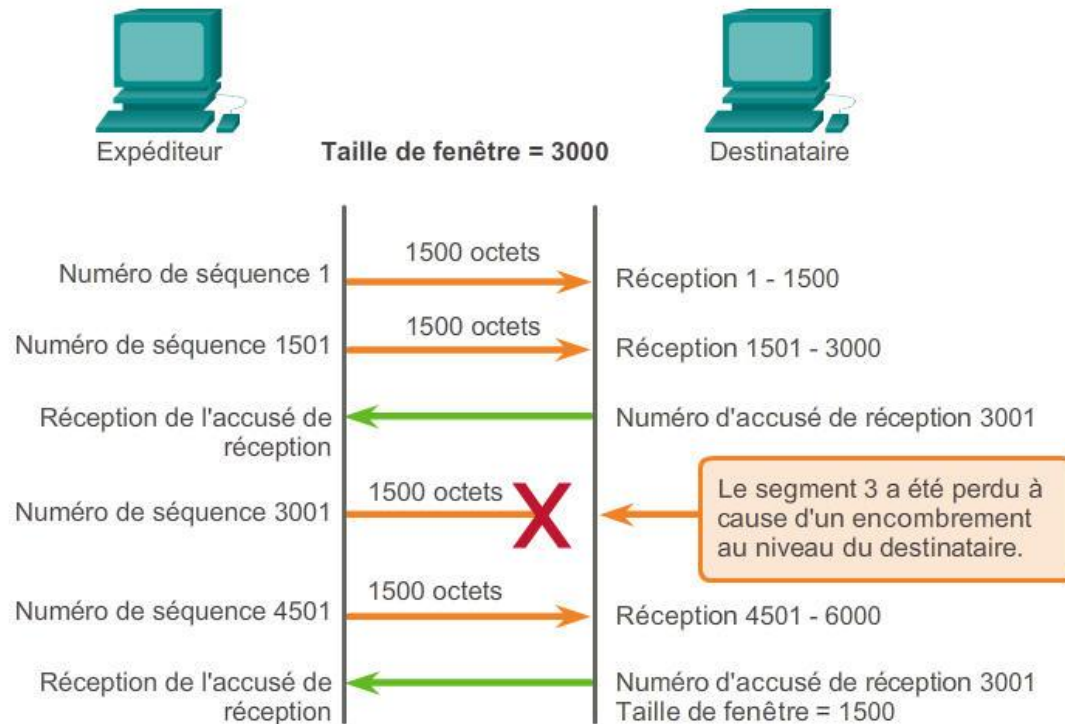




## Fiabilité et contrôle de flux

# Contrôle de flux TCP – Éviter l'encombrement

### Encombrement TCP et contrôle de flux

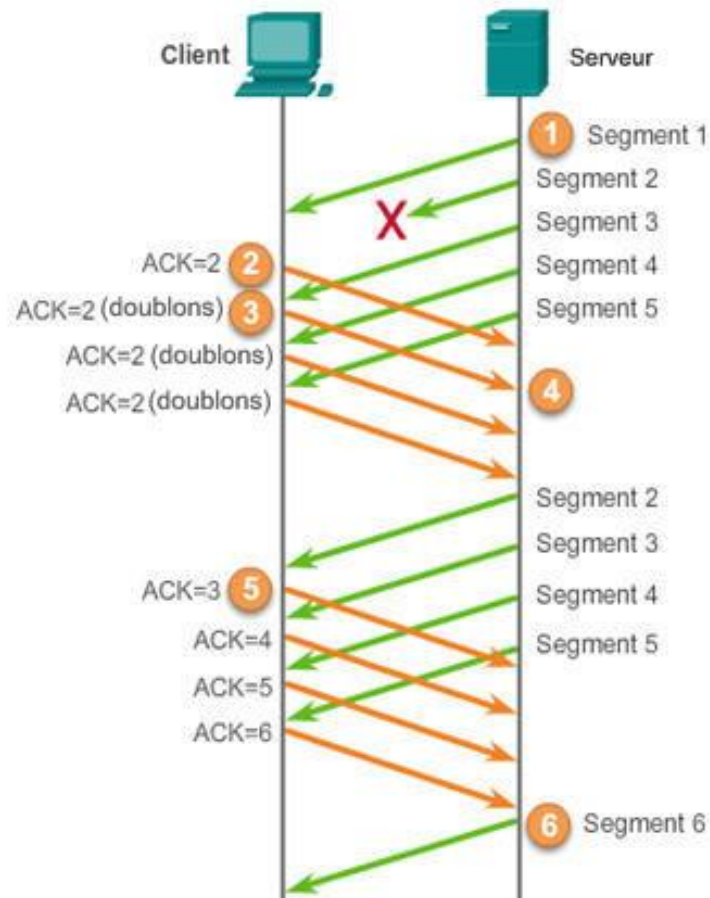


Si des segments sont perdus du fait d'un encombrement, le destinataire enverra un accusé de réception pour le dernier segment séquentiel reçu et répondra en utilisant une taille de fenêtre réduite.



Fiabilité et contrôle de flux

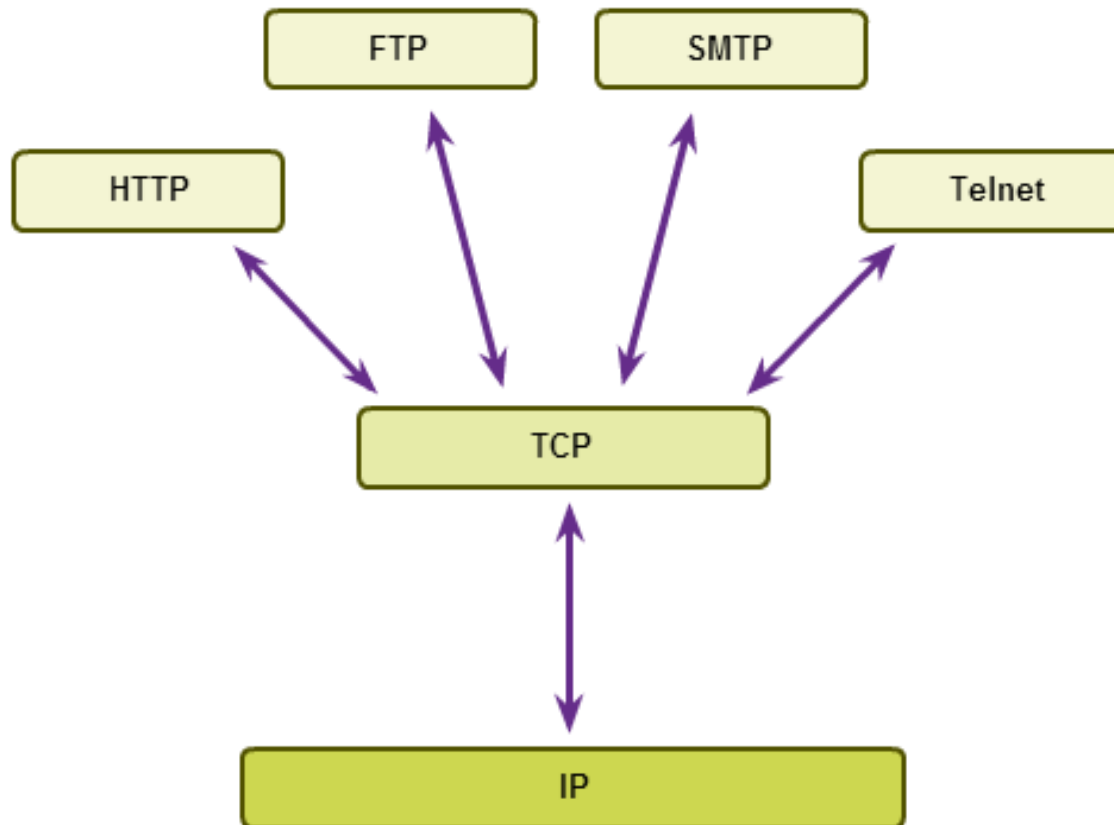
# Fiabilité TCP - Accusés de réception





TCP ou UDP

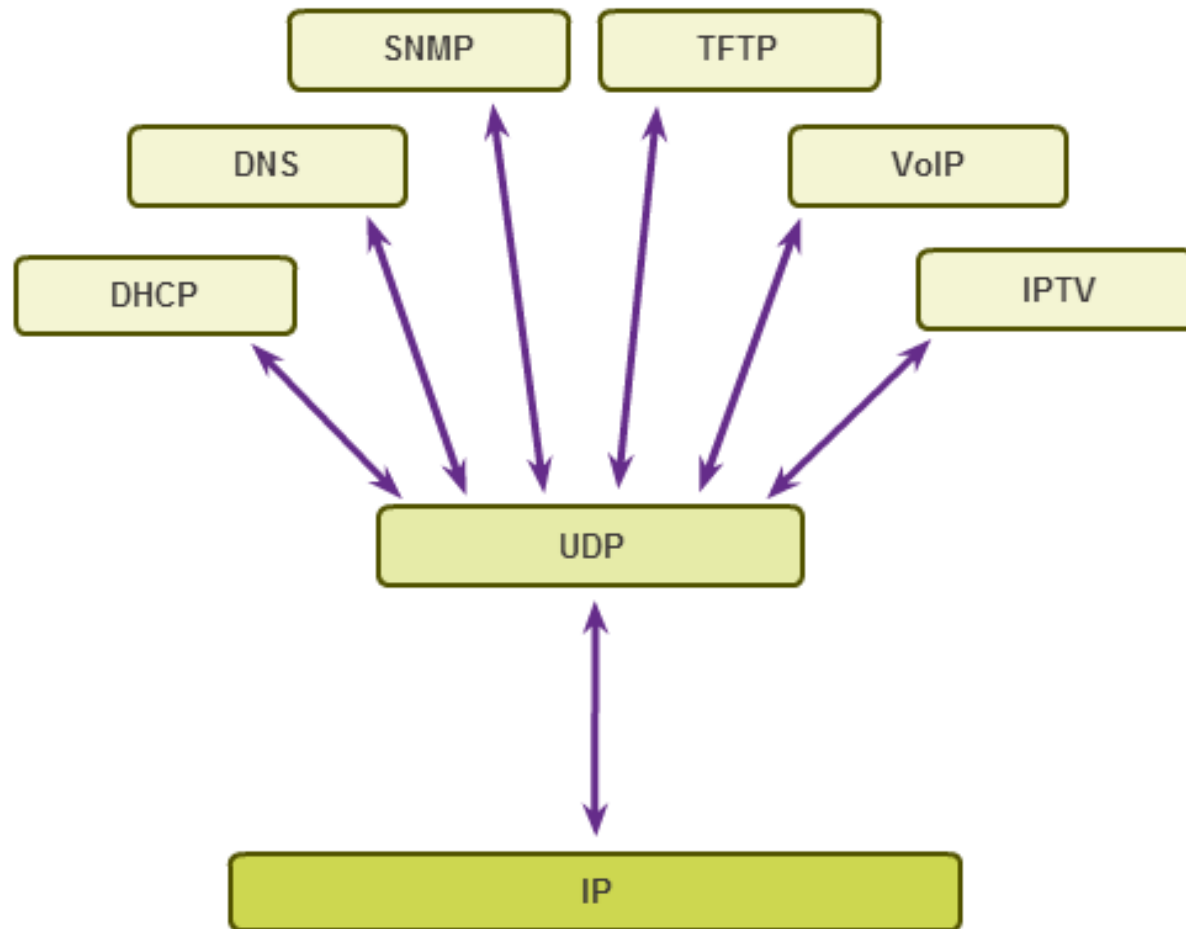
# Applications utilisant le protocole TCP





TCP ou UDP

# Applications utilisant le protocole UDP





## Chapitre 7 : Résumé

- La couche transport assure trois fonctions essentielles : le multiplexage, la segmentation et la reconstitution, et le contrôle des erreurs.
- Celles-ci sont indispensables pour assurer la qualité de service et la sécurité sur les réseaux.
- Il faut savoir comment les protocoles TCP et UDP fonctionnent et quelles applications courantes utilisent chacun d'eux pour garantir la qualité de service et créer des réseaux plus fiables.
- Les ports fournissent un « tunnel » qu'empruntent les données pour passer de la couche transport à l'application appropriée au niveau de la destination.

