

Réseaux

Protocoles TCP et UDP

Master Miage 1

Université de Nice – Sophia Antipolis

(Second semestre 2008-2009)

Jean-Pierre Lips (jean-pierre.lips@unice.fr)

(à partir du cours de Jean-Marie Munier)

Sources bibliographiques

- ✓ Comer (D.E.) : TCP/IP architecture, protocoles, applications - 5ème édition - Dunod 2009/01
- ✓ Comer (D.E.) : Réseaux et Internet - CampusPress 2000
- ✓ Pujolle (G.) : Les réseaux - 6ème édition - Eyrolles 2007
- ✓ Servin (C.) : Réseaux et Télécoms - 2ème édition - Dunod 2006
- ✓ Siyan (K.S.) : TCP/IP - 2ème édition - CampusPress 2001
- ✓ Tanenbaum (A.S.) : Réseaux - 4ème édition - Pearson Education 2003

- ✓ RFC 768, 793, 1122, 1106, 1323

- ✓ Cours UREC du CNRS (www.urec.fr)

Protocoles de transport

- Deux protocoles permettant la communication entre applications utilisatrices :
 - **TCP** (*Transmission Control Protocol*) en mode orienté connexion
 - **UDP** (*User Datagram Protocol*) en mode sans connexion
- Similitude avec les protocoles de transport OSI :
 - 5 classes de transport OSI (0, 1, 2, 3, 4) choisies selon la qualité du service de réseau (A, B, C) et les besoins du processus utilisateur
 - UDP similaire au transport classe 0
 - TCP similaire au transport classe 4

Numéros de ports

- Identification d'une application par un numéro de port
 - entier sur 16 bits
 - valeurs < 256 : ports réservés (*well-known ports*)
 - équivalent du **TSAP** (*Transport Service Access Point*) dans le modèle OSI
- **Socket** : combinaison d'une adresse IP et d'un numéro de port
- L'association de 2 sockets définit totalement une connexion TCP ou un échange UDP. Exemple :

(130.190.5.1, 23) et (147.171.150.2, 1094)

connexion entre le processus client numéro 1094 sur l'hôte 147.171.150.2 et le serveur (*daemon*) telnetd sur l'hôte 130.190.5.1 (port 23 réservé au serveur Telnet)

Caractéristiques de TCP

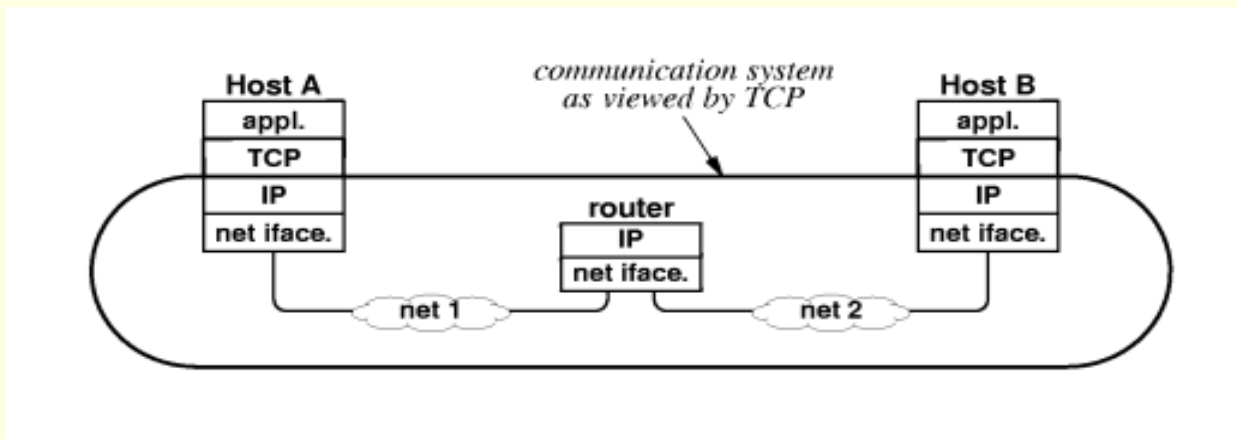
- Transport de flux d'octets non structurés (*stream interface*)
- Service de bout en bout orienté connexion
- Service de transport fiable (accusés de réception, retransmissions)
- Contrôle de flux et de congestion par connexion (mécanisme de fenêtre glissante - *sliding window*)
- Établissement et libération fiables des connexions (mécanismes en trois étapes)

Transport de flux d'octets

- Flux d'octets non structurés : pas de notion d'enregistrement ou de message (propre aux applications)
- Pas de garantie que les données seront remises à l'application réceptrice en morceaux de même taille que ceux qui sont transmis par l'application émettrice
- Découpage de la suite d'octets en **segments TCP**, encapsulés dans des datagrammes IP (champ Protocole IP = 6)
- Nécessité d'un moyen d'envoi des derniers octets (fonction *push*)

Service orienté connexion

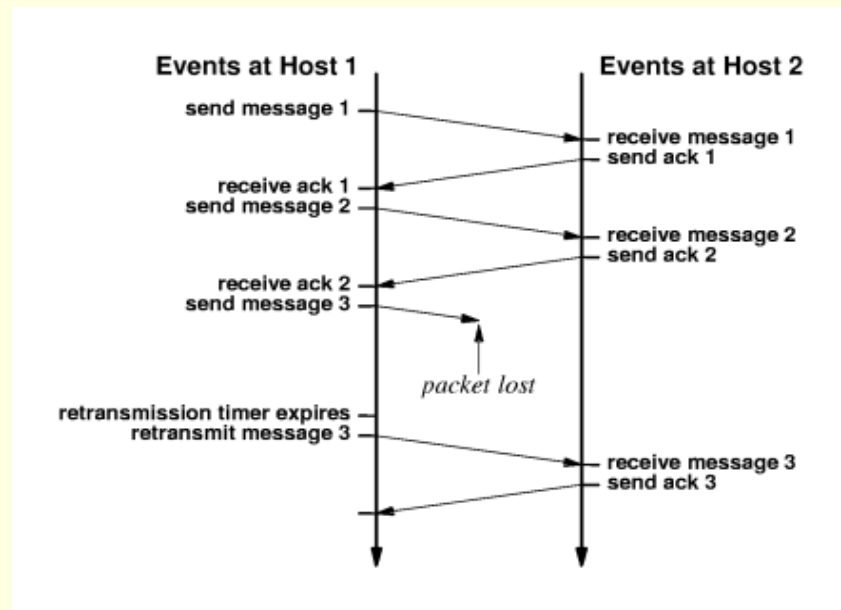
- Etablissement d'une connexion **duplex** point à point (circuit virtuel) de bout en bout entre le port source de l'émetteur et le port de destination du récepteur
- Transmission des données sur cette connexion
- Libération de la connexion



Source : D. E. Comer - Computer Networks and Internets - Prentice Hall 1999

Service de transport fiable

- Accusés de réception (*ACK, Acknowledgement*)
- Retransmission des segments perdus ou en erreur
- Ajustement adaptatif du délai de retransmission (voir plus loin)



Source : D. E. Comer - *Computer Networks and Internets* - Prentice Hall 1999

Fenêtre de transmission

- Mécanisme du *glissement de fenêtre* (*sliding window*)

fenêtre initiale

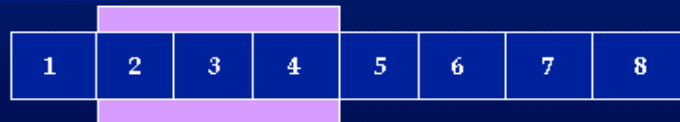
émetteur et destinataire sont concernés



L'émetteur peut envoyer 3 segments avant de recevoir un acquittement

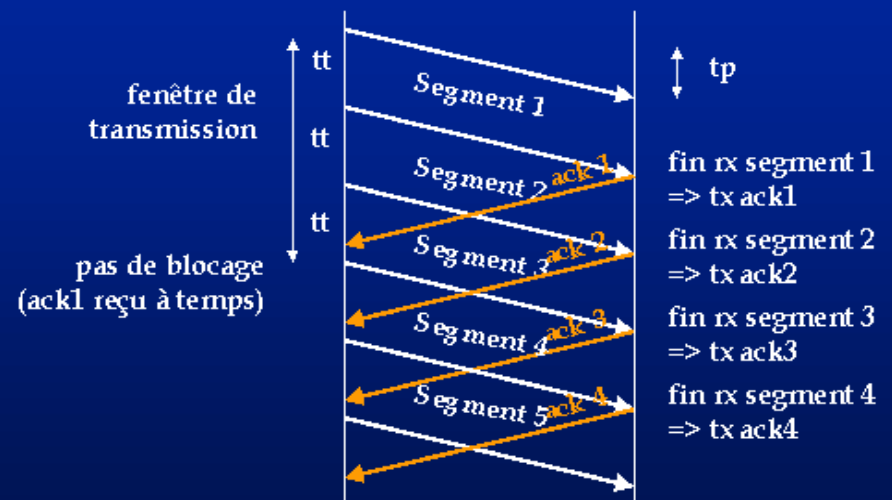
L'acquittement du segment 1 arrive, la fenêtre glisse

glissement



Les performances sont fonctions de la taille de la fenêtre et de

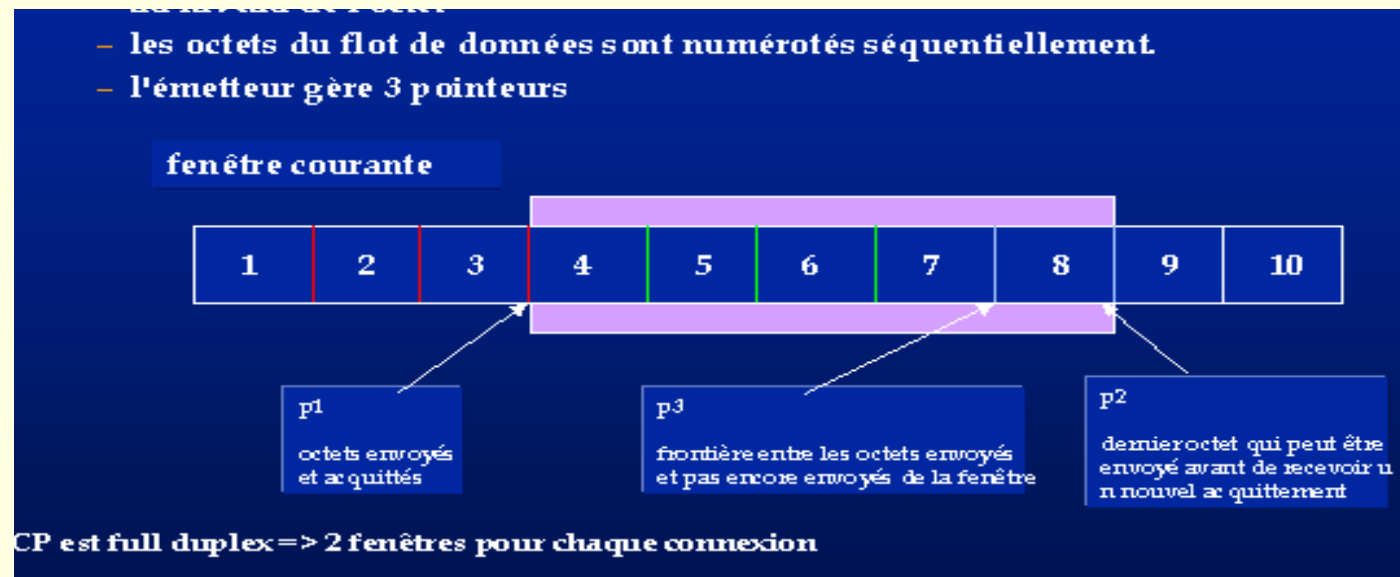
- Si la fenêtre a une taille suffisante, il n'y a pas de blocage



Source : Cours de l'UREC: TCP/IP Protocoles de l'Internet

Fenêtre de transmission (suite)

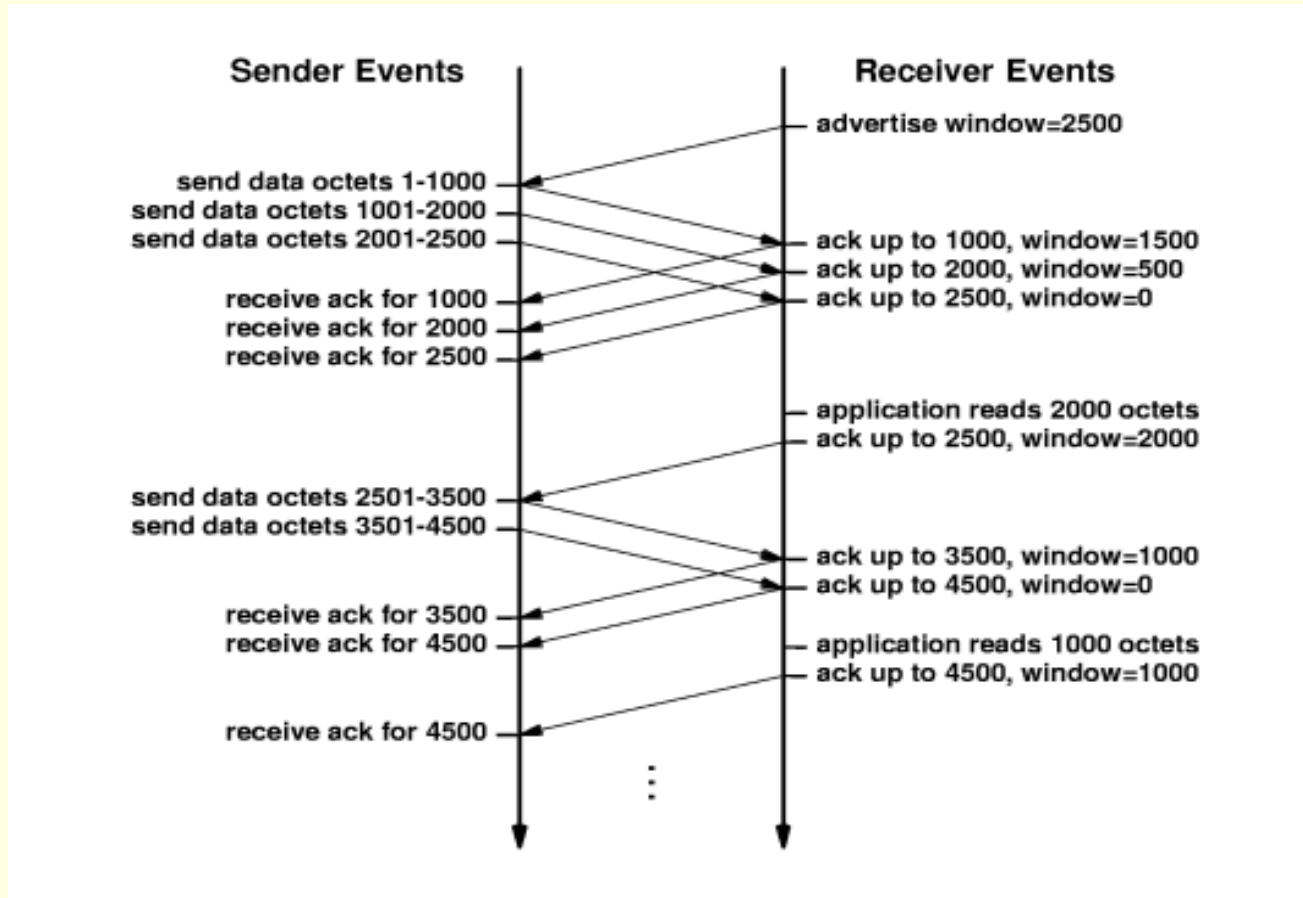
- Permet l'efficacité de la transmission et le contrôle de flux
- Numérotation d'octets (et non de segments)
- Deux fenêtres par connexion (mode duplex)



Source : Cours de l'UREC: TCP/IP Protocoles de l'Internet

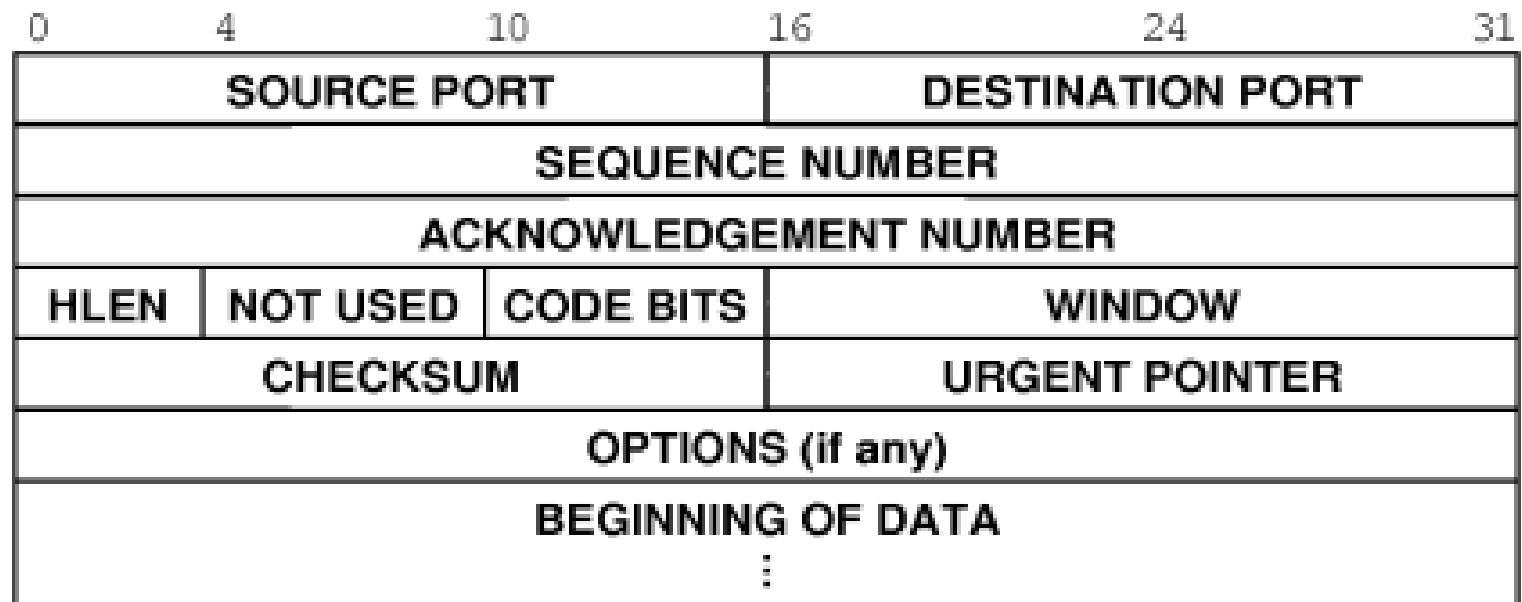
Contrôle de flux

Indication de taille de fenêtre (*window advertisement*)



Source : D. E. Comer - Computer Networks and Internets - Prentice Hall 1999

Format d'un segment TCP



Source : D. E. Comer - Computer Networks and Internets - Prentice Hall 1999

Champs de l'en-tête TCP

- **Port source** (16 bits) et **Port de destination** (16 bits)
 - identificateurs des extrémités de la connexion TCP
- **Numéro de séquence** (32 bits)
 - numéro de séquence du premier octet dans le segment
 - valeur initiale fixée à l'établissement de la connexion
- **Numéro d'acquittement** (32 bits)
 - numéro de séquence du prochain octet attendu par le récepteur (N indique la bonne réception des octets jusqu'à N-1)
 - valeur validée par l'indicateur ACK (voir plus loin)
 - valeur initiale fixée à l'établissement de la connexion
- **HLEN** ou **Décalage de données** (4 bits)
 - taille, en mots de 32 bits, de l'en-tête TCP, y compris la zone optionnelle (indique donc le début des données)

Champs de l'en-tête TCP (suite)

- **Indicateurs (6 bits)**

- URG (*Urgent*) : indique que le champ Pointeur de données urgentes est significatif (Ctrl C, sous Telnet, par exemple)
- ACK (*Acknowledgement*) : indique que le numéro d'acquittement est significatif
- PSH (*Push*) : demande de remise immédiate à l'application, sans attendre le remplissage d'un tampon de réception
- RST (*Reset*) : demande de réinitialisation de la connexion
- SYN (*Synchronization*) : sert à l'établissement d'une connexion
- FIN (*Finished*) : sert à libérer une connexion

- **Fenêtre (16 bits)**

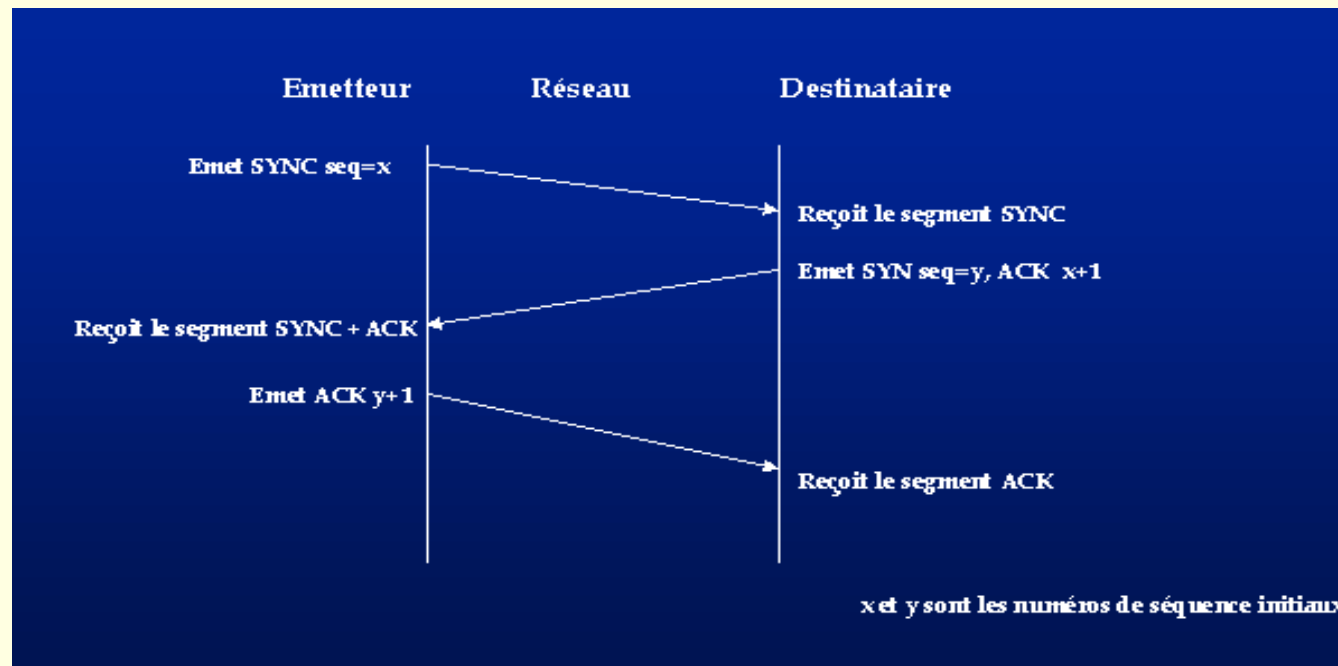
- nombre d'octets, à partir du numéro d'acquittement, que le récepteur est en mesure d'accepter (max : 65535)

Champs de l'en-tête TCP (suite)

- **Total de contrôle** (16 bits)
 - protection de l'en-tête TCP, des données, et d'un pseudo en-tête IP de 3 mots
 - si erreur : pas d'acquittement
- **Pointeur de données urgentes** (16 bits)
 - valeur validée par l'indicateur URG
 - indique la position du dernier octet des données urgentes
- **Options** (taille variable)
 - négociation de la taille maximale des segments (**MSS**, *Maximum Segment Size*)
 - redimensionnement (*window scale*) : taille de fenêtre décalée de 16 bits à gauche
 - réémission sélective (**NAK**, *Negative Acknowledgement*)

Etablissement d'une connexion TCP

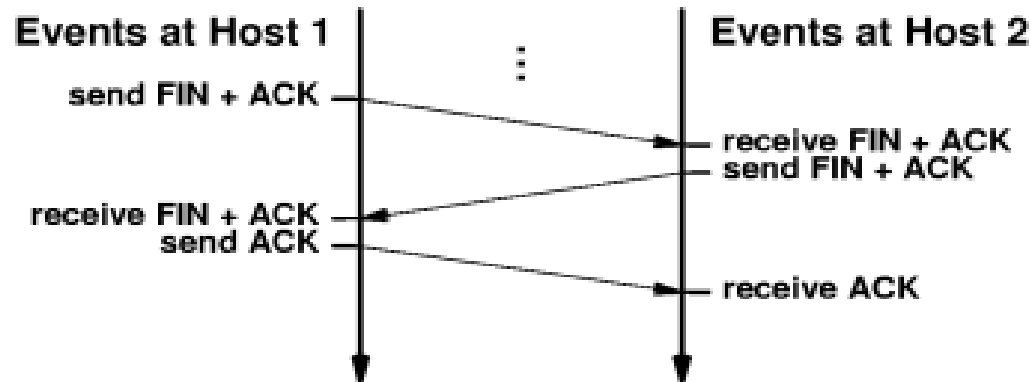
- Mécanisme en 3 étapes (*Three-way Handshake*), chaque message d'établissement étant assimilé à un message de 1 octet



Source : Cours de l'UREC: TCP/IP Protocoles de l'Internet

Libération d'une connexion TCP

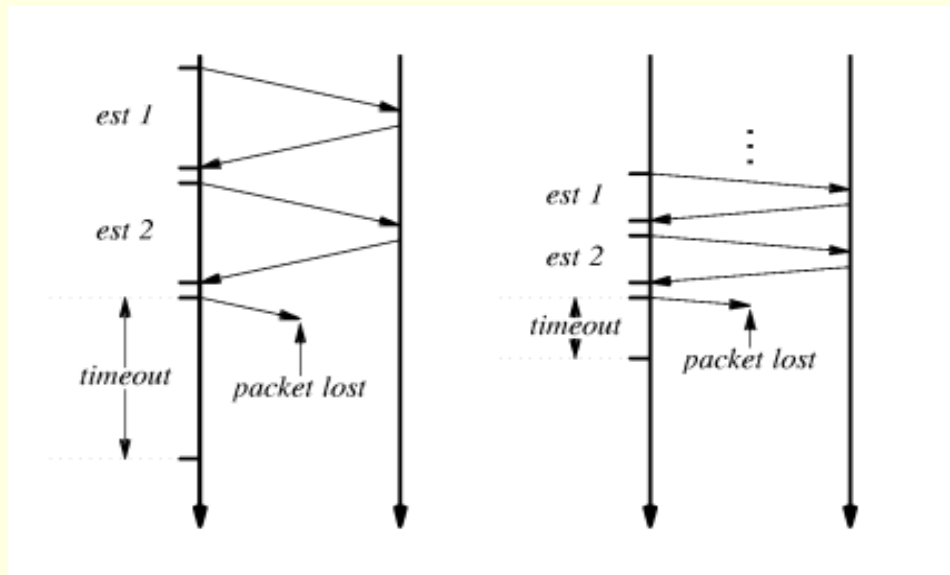
- Mécanisme en 3 étapes (*graceful shutdown*), chaque message de libération étant assimilé à un message de 1 octet
- Autre possibilité : libération brutale par l'indicateur RST



Source : D. E. Comer - Computer Networks and Internets - Prentice Hall 1999

Temporisation adaptative

- Algorithme adaptatif d'ajustement automatique du délai de retransmission en fonction de l'estimation, faite en permanence, du temps de boucle (ou délai aller-retour, ou *round-trip time*)



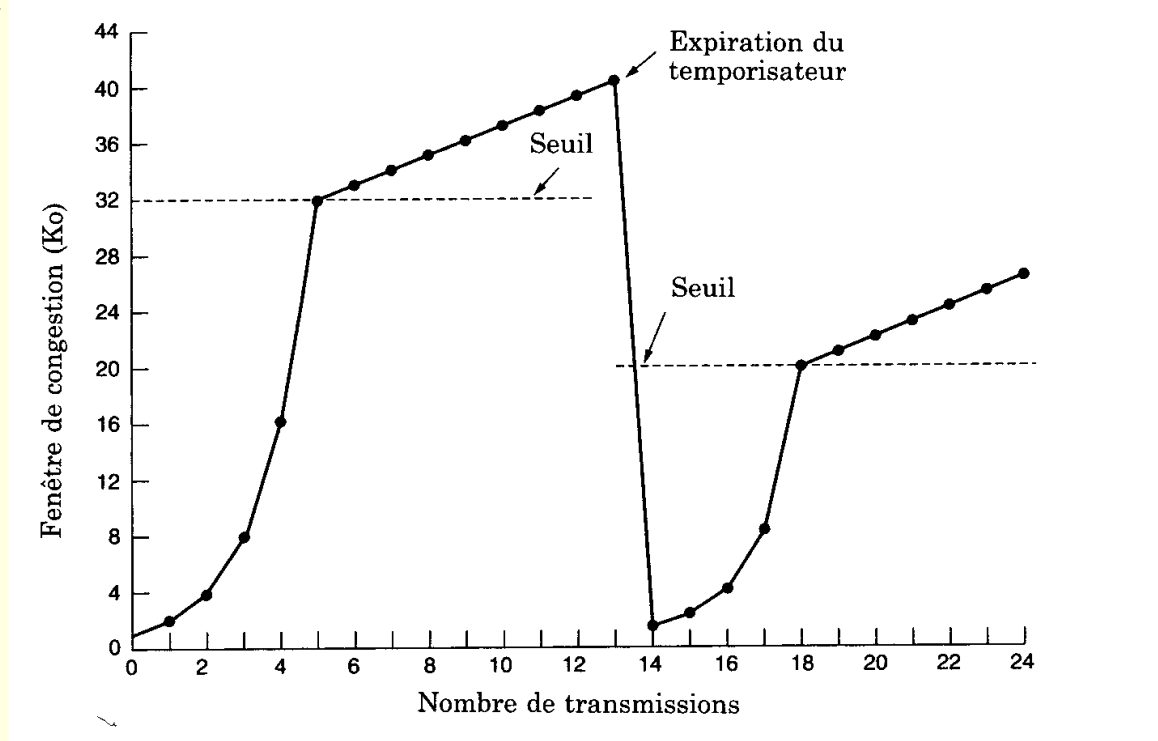
Source : D. E. Comer - Computer Networks and Internets - Prentice Hall 1999

Contrôle de congestion

- Congestion => perte de paquets => éviter de retransmettre trop rapidement : réduction rapide de la taille de fenêtre
- Définition d'une deuxième fenêtre, en plus de la fenêtre de transmission : la fenêtre de congestion (la fenêtre réellement prise en compte est la plus petite des deux)
- Algorithme de '*slow start*' :
 - augmentation exponentielle de la fenêtre de congestion jusqu'à la perte d'un paquet : le seuil de congestion est alors fixé à la moitié de la valeur de la fenêtre et la fenêtre est remise à 1
 - nouvelle augmentation exponentielle jusqu'au seuil, puis augmentation linéaire au-delà, jusqu'à la perte d'un paquet, etc...

Algorithme de *slow-start* (exemple)

- taille maximale de segment = 1024 octets
- initialement : fenêtre de congestion = 64 Ko, mais temporisation expirée
=> seuil à 32 Ko et fenêtre de congestion à 1 Ko (transmission 0)



Source : A. S. Tanenbaum - Réseaux - InterEditions 1997

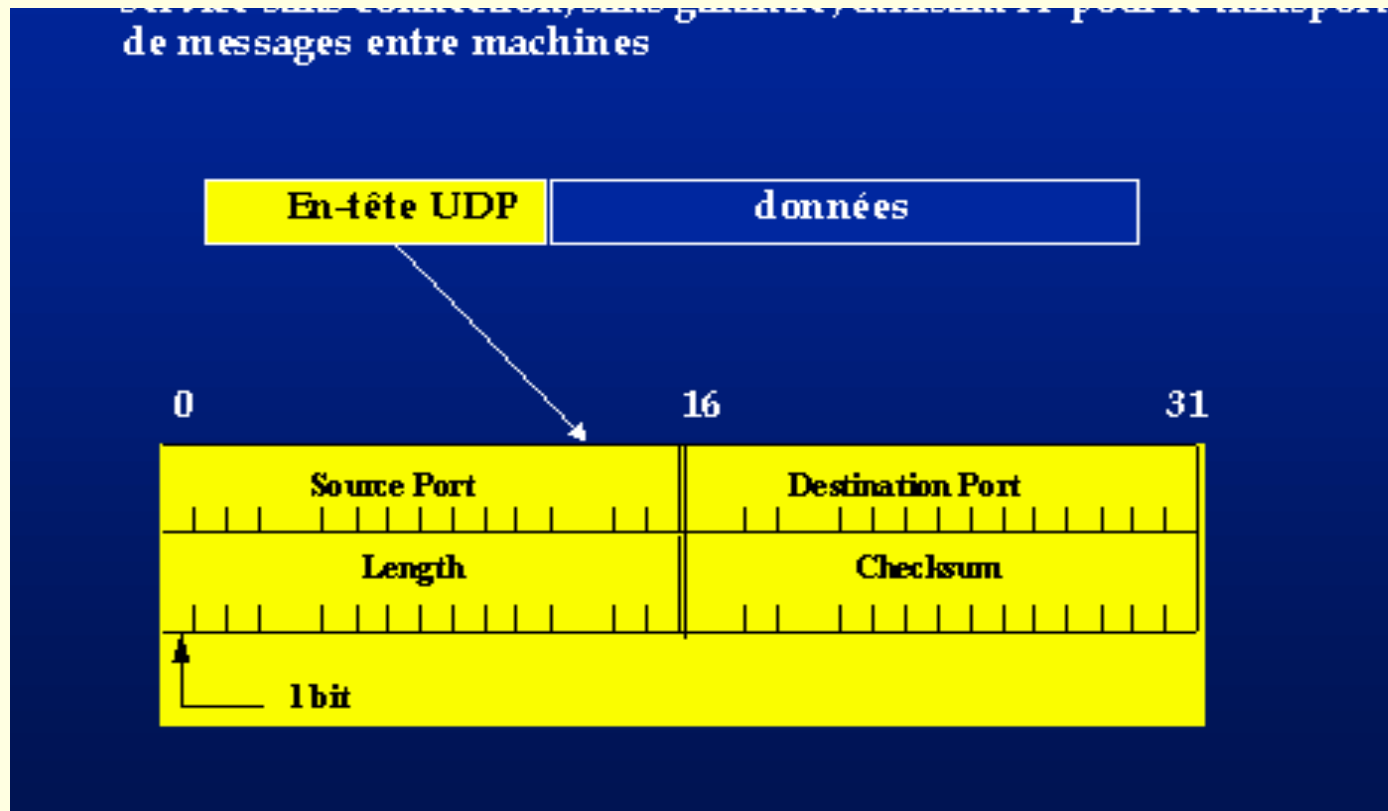
Caractéristiques de UDP

- Transport de datagrammes, en mode non connecté
- Identification des processus d'application (numéros de port UDP)
- Possibilité de vérifier l'intégrité des données (total de contrôle, facultatif)
- Pas d'acquittements, pas de retransmissions, pas de contrôle de flux
- Convient bien aux applications orientées commande/réponse (commandes et réponses contenues chacune dans un seul datagramme)
- Efficace pour les opérations en diffusion

Caractéristiques de UDP (suite)

- Protocole très simple (taille de code limitée) et permettant des temps d'exécution courts
- UDP utilisé par de nombreuses applications :
 - TFTP (*Trivial File Transfer Protocol*)
 - DNS (*Domain Name System*)
 - BOOTP (*BOOTstrap Protocol*)
 - DHCP (*Dynamic Host Configuration Protocol*)
 - SNMP (*Simple Network Management Protocol*)
 - RIP (*Routing Information Protocol*)
 - ...

Format d'un datagramme UDP



Source : Cours de l'UREC: TCP/IP Protocoles de l'Internet

Champs de l'en-tête UDP

- **Port source (16 bits)**
 - identification du processus émetteur
 - optionnel : 0 si non utilisé
- **Port de destination (16 bits)**
 - identification du processus destinataire
- **Longueur (16 bits)**
 - taille du datagramme UDP, en-tête compris (8 octets)
- **Total de contrôle (16 bits)**
 - protection de l'en-tête UDP, des données, et d'un pseudo en-tête IP de 3 mots
 - optionnel : 0 si non utilisé (note : si le total de contrôle est utilisé et la valeur est zéro, elle est représentée par tous les bits à 1)