

Protocole TCP

Introduction aux réseaux locaux

Plan de la présentation

- Rappel
- Détails du protocole TCP

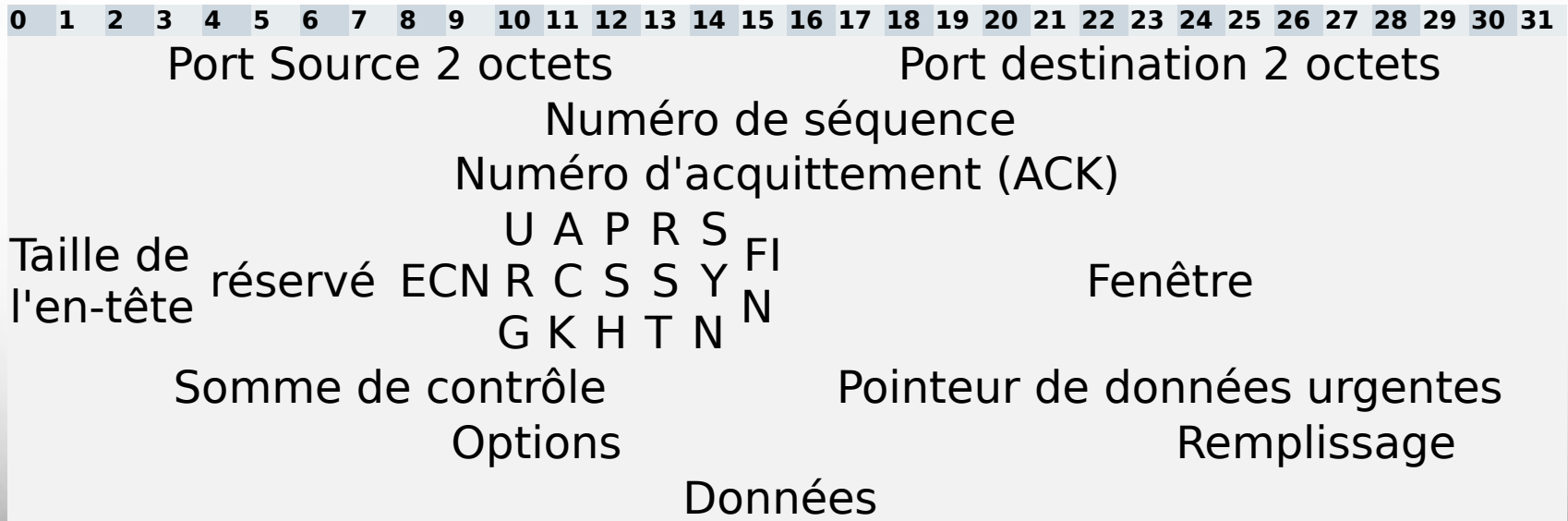
Protocole TCP

Rappel

- TCP offre un transport fiable qui assure:
 - Une connexion entre les deux parties selon une entente précise (Handshaking)
 - Que les deux parties(processus) puissent communiquer (Full duplex-service)
 - Le transport des segments
 - L'ordre des segments
 - La qualité des segments

Structure du segment TCP

- Comparativement à UDP la structure du segment TCP est de 20 octets minimalement au lieu de 8.



En-tête TCP

- Port source 16 bits
 - Numéro du port source
- Port destination 16 bits
 - Numéro du port destination
- Numéro de séquence 32 bits
 - Numéro de séquence du premier octet de ce segment
- Numéro d'acquittement (ACK) 32 bits
 - Numéro de séquence du prochain octet attendu
- Taille de l'en-tête 4 bits
 - Longueur de l'en-tête en mots de 32 bits (les options font partie de l'en-tête)
- Drapeaux (Flags) 12 bits
 - Réservé : Réserve pour un usage futur (4 bits)
 - ECN : signale la présence de congestion (2 bits)
 - URG : Signale la présence de données URGentes
 - ACK : Signale que le paquet est un accusé de réception (ACKnowledgement)
 - PSH : Données à envoyer tout de suite (PuSH)
 - RST : Rupture anormale de la connexion (ReSeT)
 - SYN : Demande de synchronisation (SYN) ou établissement de connexion
 - FIN : Demande la FIN de la connexion

En-tête TCP (suite)

- Fenêtre 16 bits

- Taille de fenêtre demandée, c'est-à-dire le nombre d'octets que le récepteur souhaite recevoir sans accusé de réception

- Checksum 16 bits

- Somme de contrôle calculée sur l'ensemble de l'en-tête TCP et des données, mais aussi sur un pseudo en-tête (extrait de l'en-tête IP)

- Pointeur de données urgentes 16 bits

- Position relative des dernières données urgentes

- Options (0 à 320 bits – divisible par 32 bits)

- Facultatives

- Remplissage (variables)

- Zéros ajoutés pour aligner les champs suivants du paquet sur 32 bits, si nécessaire (voir options)

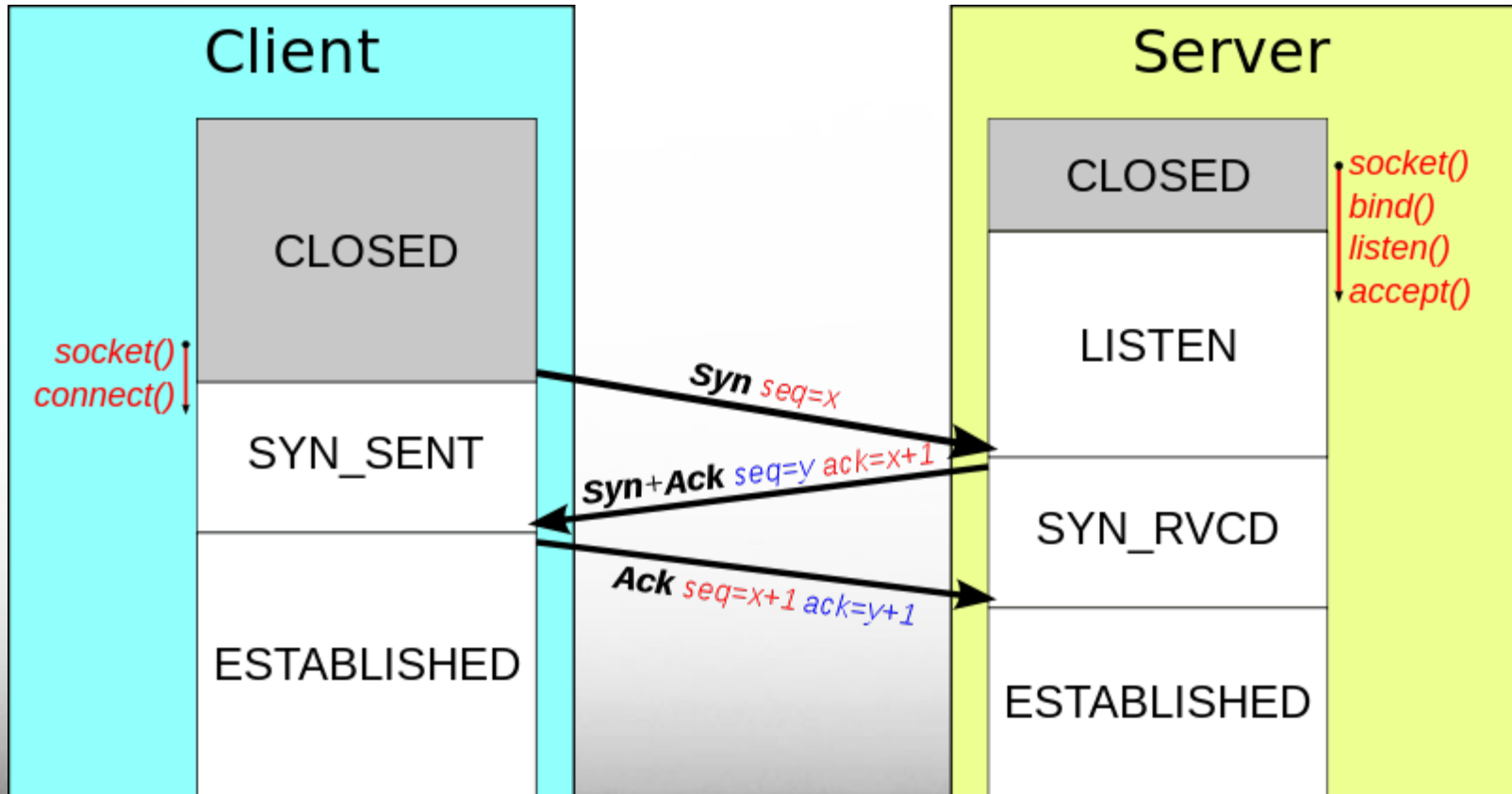
- Données (Les données débute après la taille de l'en-tête)

- Séquences d'octets transmis par l'application

Séquence initiale

- Dans le but d'établir une connexion entre le client et le serveur, les segments suivants seront échangés entre les deux parties:
 - *Étape 1*: Un segment (avec un message vide) et dont le bit SYN est à un est envoyé du client au serveur. (**Segment SYN** avec un numéro tiré au hasard X)
 - *Étape 2*: Le segment SYN est reçu par le serveur et ce dernier prépare les tampons de réception et retourne un nouveau **Segment SYNACK**, avec le bit SYN à 1, le ACK à X+1 et comme numéro Y.
 - *Étape 3*: Le segment SYNACK est reçu par le client et ce dernier prépare aussi ses tampons de réception. Le client envoie un dernier segment avec un ACK de Y+1 et le bit SYN à zéro pour confirmer la connexion. Ce dernier segment peut contenir un message avec le numéro de segment X+1.
- La communication est établie....

Exemple



Ordonnance des segments

- Contrairement au séquençement des segments vue dans les modèles GBN et SR, TCP séquence les octets dans le flux de données.
- Le **numéro de séquence** dans l'en-tête correspondra donc au numéro de l'octet à transférer dans ce segment.
- En considérant, une **taille maximal des données** pouvant être transmise dans un segment (MSS – Maximum segment size) de 1000 octets et un fichier de 50000 octets, le premier segment portera en théorie le numéro de séquence 0, puis le second 1000, jusqu'à 49000.

Taille maximale ?

- Typiquement le MSS équivaut à la taille de transmission maximale d'un trame (MTU - **Maximum transmission unit**) moins les en-tête du segment, du datagramme et de la trame.
- N'oubliez pas que le MSS vaut la taille maximal des données de l'application et non de la taille du segment.

Séquencement des ACK

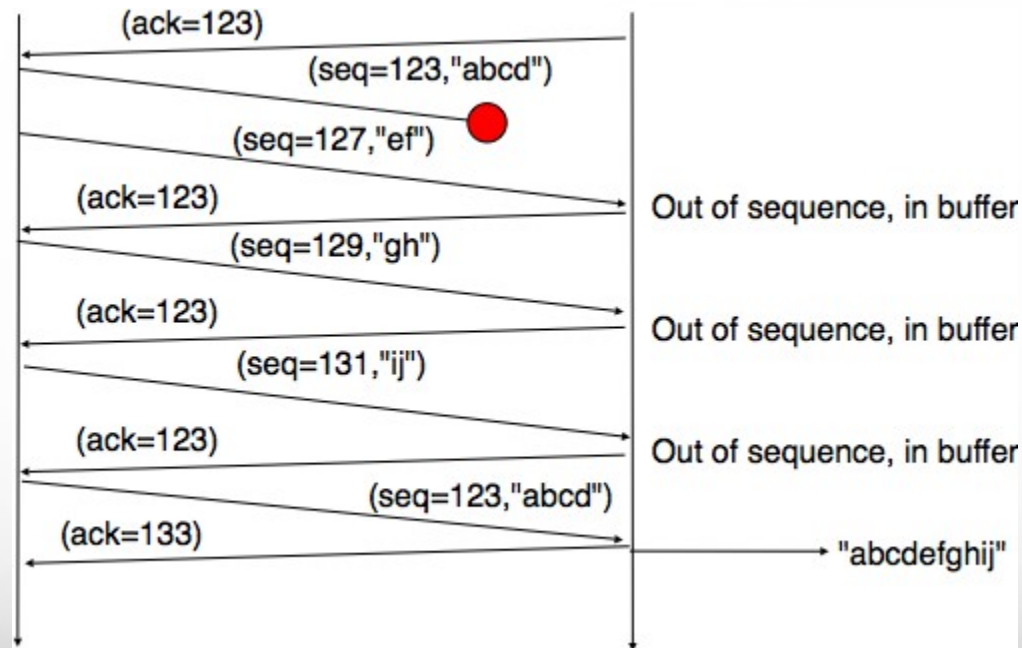
- Étant donné que le service de communication fonctionne dans les deux sens, les numéros de confirmation (ACK) renvoyé équivalra au numéro du prochain octet désiré.
- Exemple: L'hôte A envoie à l'hôte B un segment dont le message est de taille 1000 (octet 0 à 999). L'hôte B enverra un message avec un ACK portant la valeur 1000, expliquant qu'il désire recevoir la suite du flux d'octets à partir de l'octet 1000.
- Si pendant l'attente du segment 1000, l'hôte A envoie un autre segment (numéro 2000, octet 2000 à 2999) c'est à l'hôte B de décider si il retiendra ce paquet ou non, mais dans les deux cas, il doit retourner un ACK avec l'identificateur 1000 (le segment manquant)

Séquence en pratique

- En pratique le numéro de séquence est choisi au hasard (pour éviter la confusion) et démarre donc rarement à zéro.

Exemple avec perte

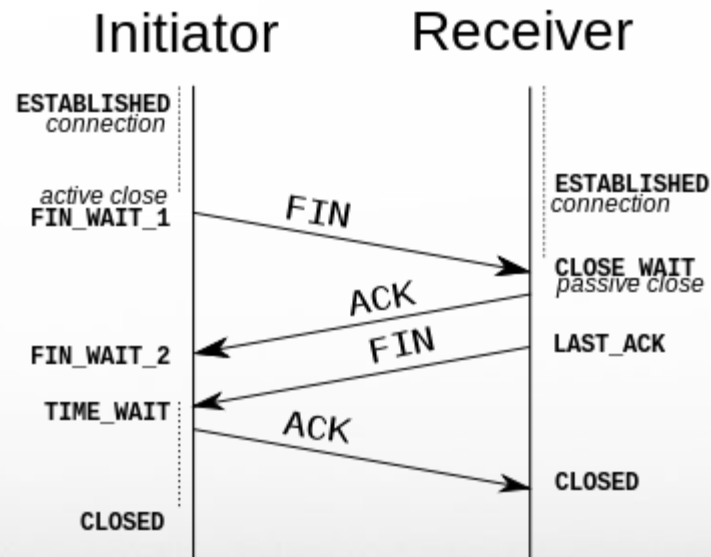
- La séquence 123 sera perdue et 'retrouvée' plus tard.



Terminaison

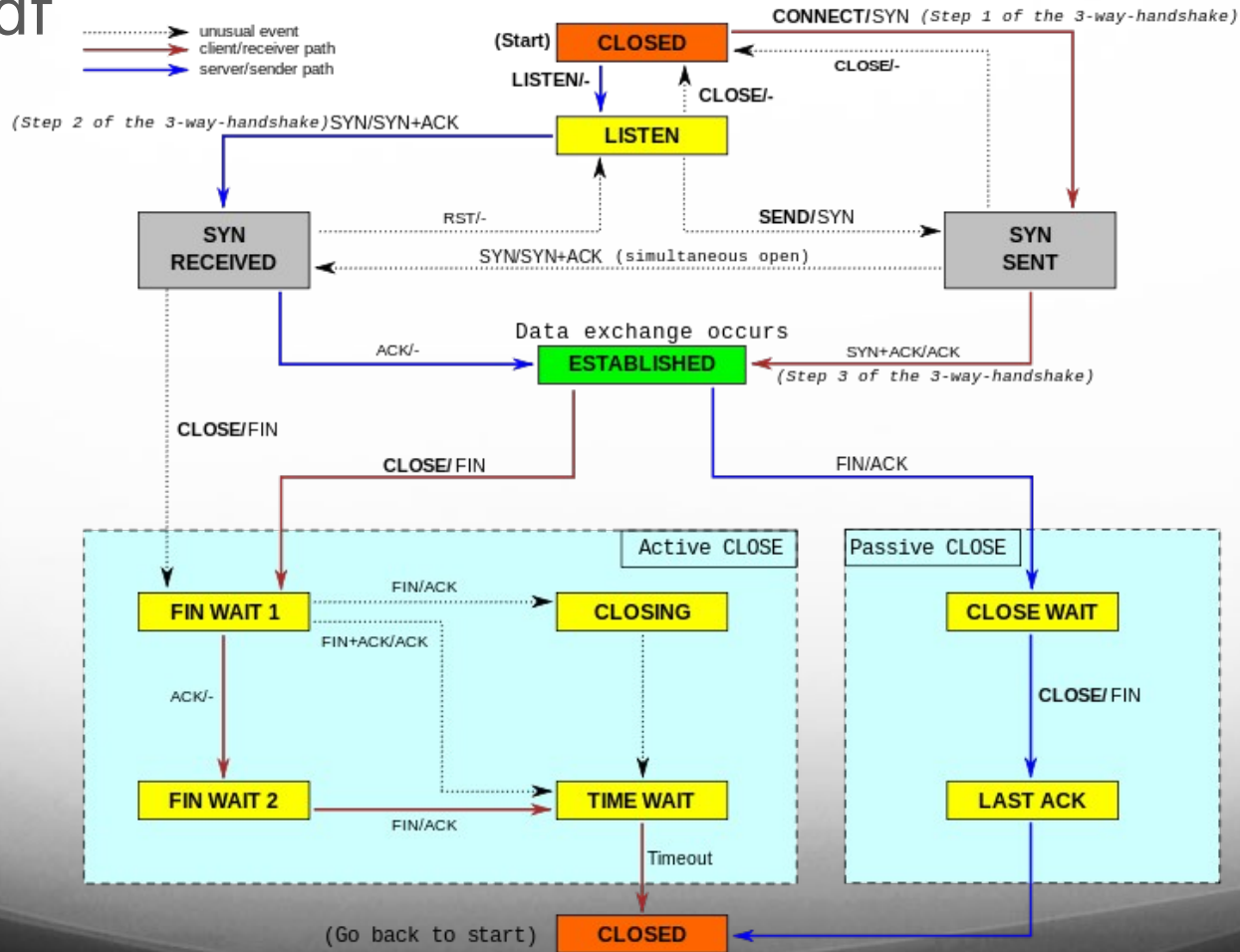
- Pour terminer une connexion, le client suit les étapes suivantes:
 - Étape 1: Le client envoie un message avec le bit FIN à 1
 - Étape 2: Le serveur reçoit le segment FIN et envoie un ACK
 - Étape 3 : Le client reçoit le ACK et attends. Le serveur envoie à son tour un message FIN au client.
 - Étape 4 : Le client reçoit le message FIN du serveur et lui retourne un ACK. Après un certain temps le client libère les ressources (tampons et variables) et ferme définitivement sa connexion.
 - Étape 5 : Le serveur reçoit le ACK du client, libère ses ressources et ferme sa connexion.

Exemple



États TCP

● sadf



À maîtriser

- Les grandes lignes du protocoles TCP
- Les termes :
 - MSS
 - MTU

Des questions?



Sources

- http://fr.wikipedia.org/wiki/Transmission_Control_Protocol
- http://wps.pearsoned.com/ecs_kurose_compnetw_6/216/55463/14198700.cw/index.html
- <http://cnp3book.info.ucl.ac.be/transport/tcp/>