

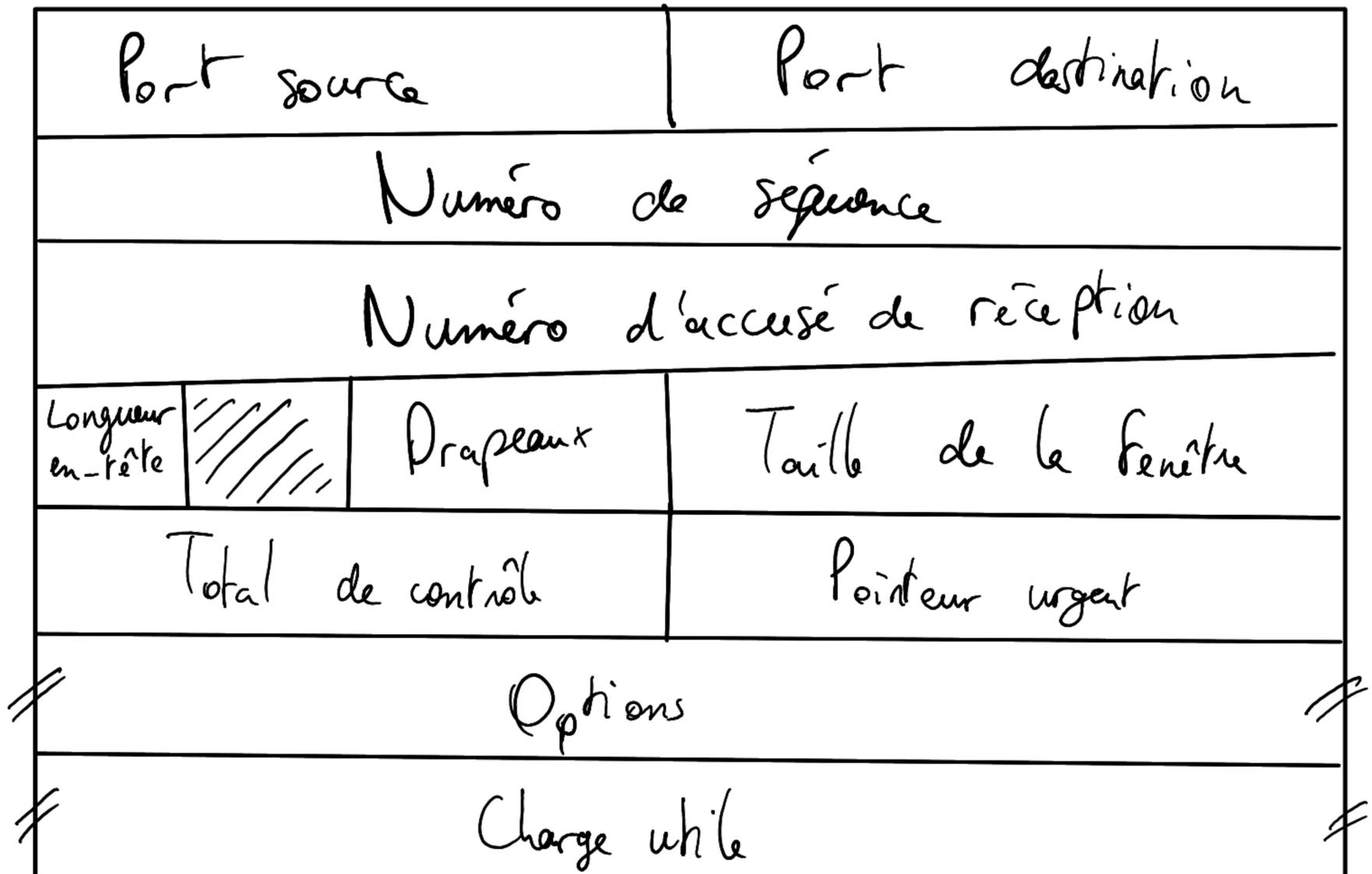
Gestion d'une connexion TCP

Motivation: Comprendre le fonctionnement précis de la vie complète d'une connexion TCP, de son établissement à sa libération

Objectif: Comprendre les mécanismes permettant l'existence des sessions TCP.

TCP → Session: Établissement connexion + Transfert données + Libération

↳ Unité de transfert: Segment



← 32 bits →

Segment TCP

Numéro de séquence → Identifiant unique à chaque segment
↳ Non réutilisé avant un certain temps (dépend de la TTL d'un paquet)

Numéro d'accuse de réception

↳ Indique le prochain octet dans l'ordre attendu

↳ Accuse de réception cumulatif car résume toutes les données reçues en 1 champ.

Longueur en-tête → Nombre de mots hors charge utile

Somme de contrôle → Vérifie la cohérence des données reçues

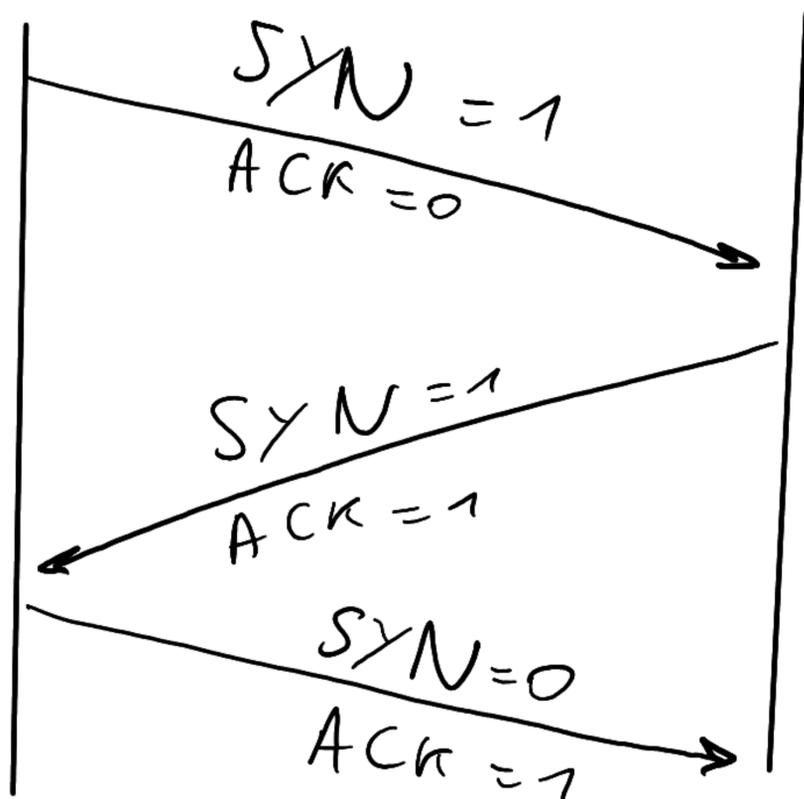
Options → Ex : MSS (Maximum Segment Size) pour indiquer la taille maximale d'un segment prêt à être accepté (défaut : 5360).
↳ Max : 2^{30} .

Taille de la fenêtre → Place restante dans le buffer

Drapeau	Utilisation
ECN	Contrôle de congestion
CWR	
URG	Pointeur d'urgence (rare)
PSH	Force la non utilisation d'un tampon
RST	Réinitialisation de connexion (en cas d'accident par ex)
ACK	Validité du numéro → 1 = prise en compte d'accuse de réception
SYN	Établissement des connexions
FIN	Libération d'une connexion

Établissement d'une connexion

↳ 3 étapes: listen → connect → accept



→ $RST = 1$ pour rejeter la connexion

Host 1

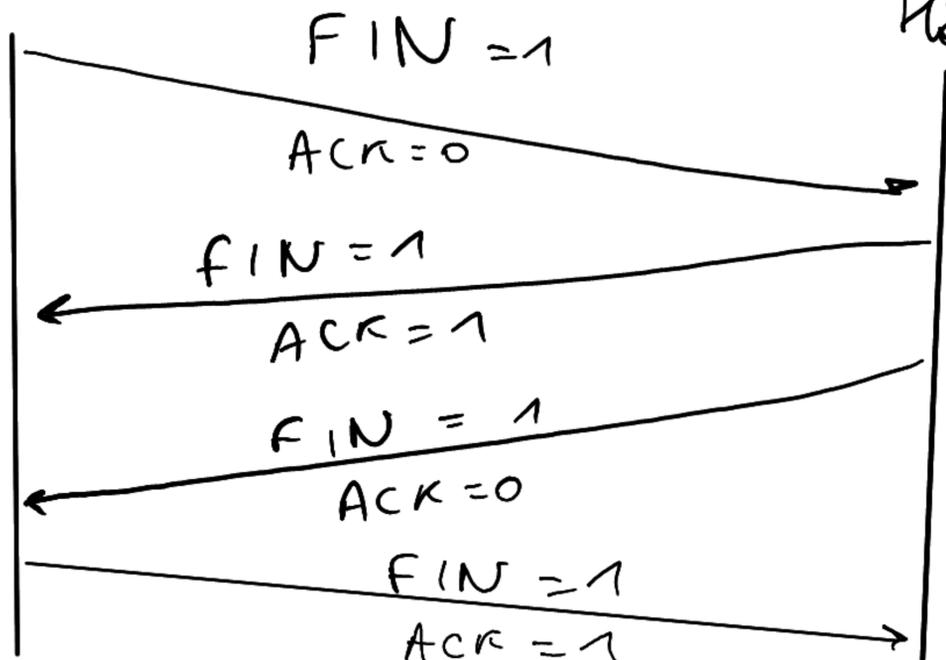
Host 2

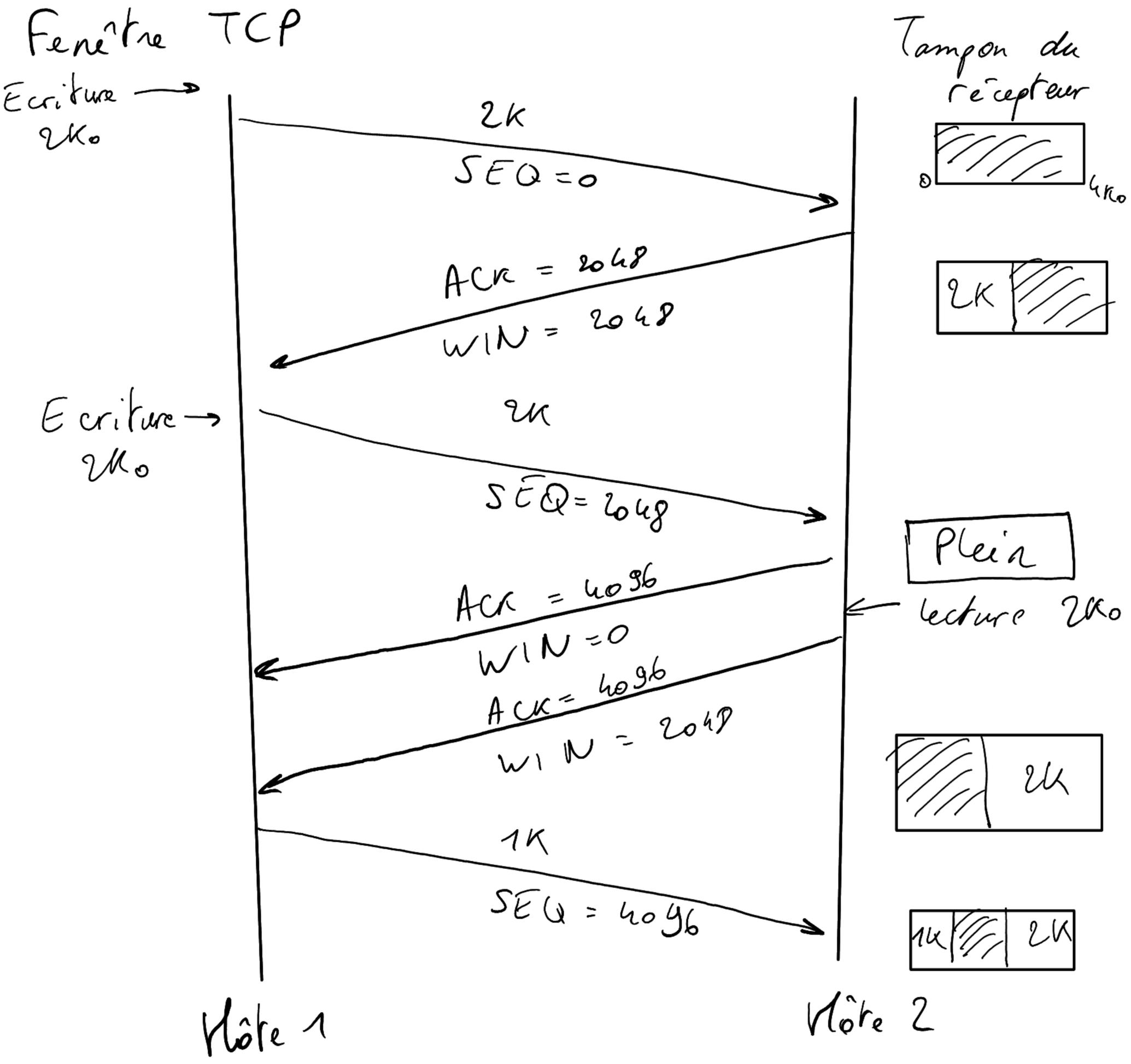
Libération d'une connexion

↳ Les connexions TCP sont bidirectionnelles, mais pour la libération il est plus simple de les comprendre comme 2 connexions unidirectionnelles.

Host 1

Host 2





Limite \rightarrow Syndrome de la fenêtre stupide
 (Annonce espace libre avec 10 de libre)