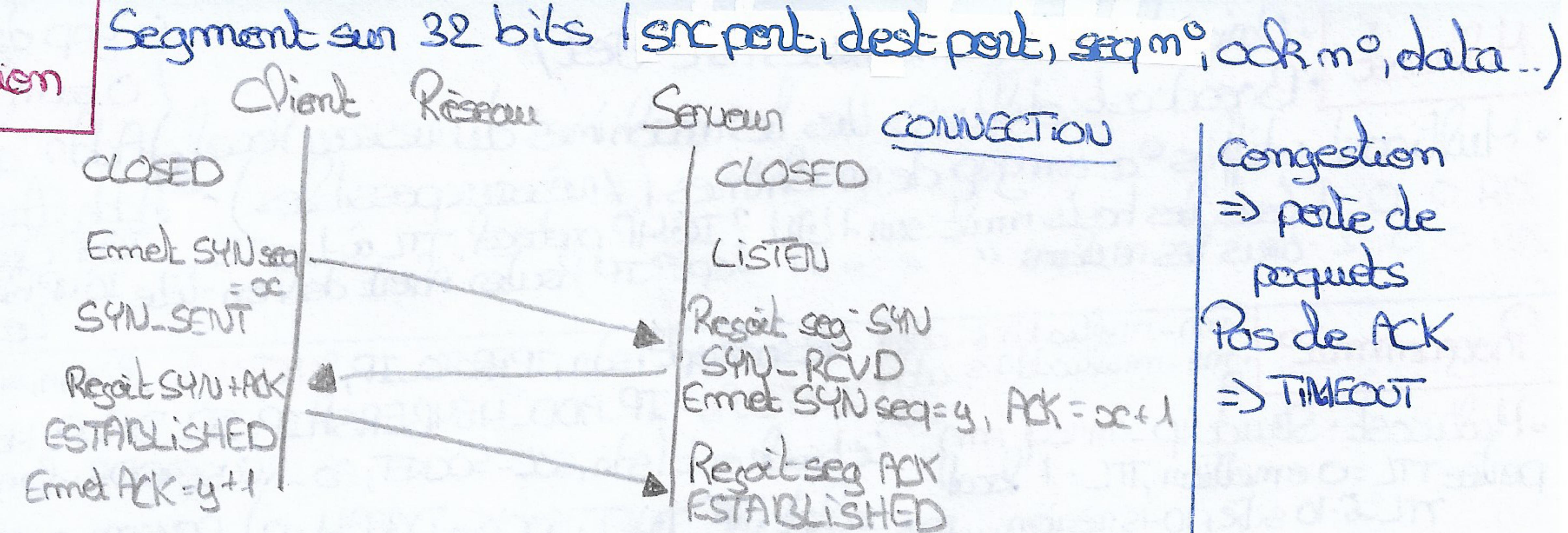


TCP - Contrôle de congestion

TCP

- Point à point
- Orienté connexion
- Fiable et ordonné
- Pas de frontière de msg
- Full duplex
- Flux contrôlé



Contrôle de congestion : Effet de la taille de la fenêtre

Fenêtre émetteur = nb paquets en transit dans le réseau.

Contrôle de bout en bout | Débit transmis limité par taille fenêtre

$$\rightarrow \omega \text{ seg. MSS octets émis en 1 RTT} : \text{débit} = \frac{\omega \times \text{MSS octets}}{\text{RTT}}$$

=> Objectif = taille de fenêtre max sans perte

Congestion window : fenêtre petite si congesto importante | fen grande si congesto faible

TCP Tahoe : vers origine de TCP qui impl. contrôle de congesto initialisé: cwnd = 1

- AIMD (Add-Increase, Mult-Decr.)
- Slow Start (ss)
- Congestion Avoidance (ca)

Algo AIMD + ca :

- Incrémentation CongestWindow d'un paquet par RTT (linear increase)
- Réduire CongestThresh et CongestWindow (mult-decrease) quand perte

Algo évitement de congesto :

- /* slowstart terminé */
- /* cwnd >= thresh */
- jusqu'à (ut de perte) {

 pour chaque seg ACKed {

 cwnd++

$$\text{thresh} = \text{cwnd}/2$$

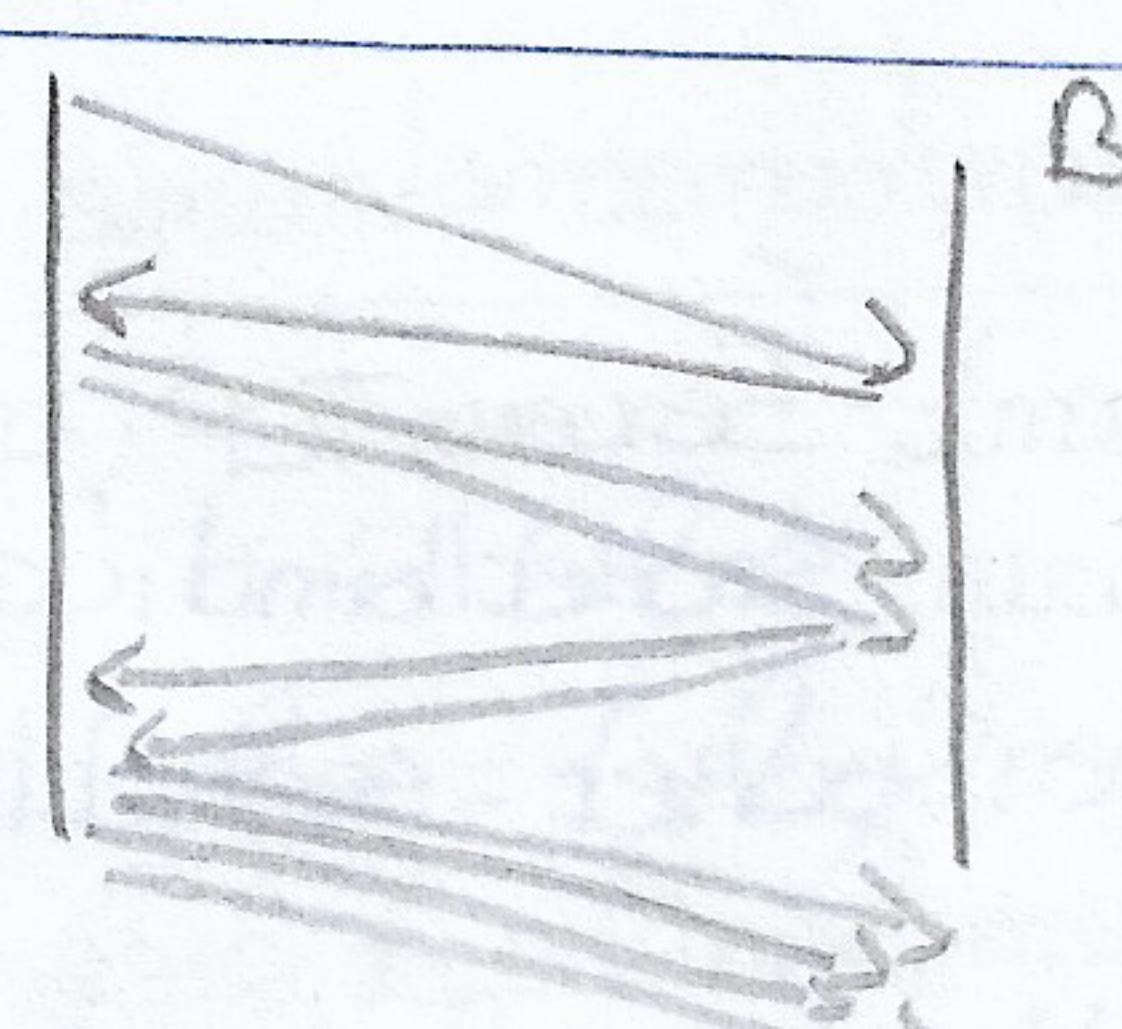
$$\text{cwnd} = 1$$

effectuer slowstart

Algo Slow Start : pour chaque seg ACKed

$\times 2$ jusqu'à atteindre seuil pris +1 par RTT jusqu'à ce que l'eut perte ou cwnd >= threshod)

Evt perte : TIMEOUT ou / et 3 ACKs dupliqués



TCP Reno : vers la plus utilisée obj

- Fast Retransmit : ACK dupliqué si perte, si 3 eut plus seg non-transmis
- Fast Recovery : supprime la phase de ss après fast Retr. (sans att-fin timer) gène la transmission jusqu'à l'arrivée d'ACK non dupliqué

Algo fast Recovery :

- ① quand 3^e ACK dupliqué reçu:
 - ssthresh = cwnd / 2
 - retransmettre seg manquant

- ② chaque fois qu'un autre ACK dupliqué arrive:
 - cwnd += MSS
 - transmettre paquets si possible

- ③ quand ACK arrive qui ACKed ma donnée
 - cwnd = ssthresh

MSS = Maximum Segment Size

=> Eviter le slow start si possible : TIMEOUT + lancement cwnd/2 après fast Retransmit

- Stratégie TCP : - contrôle une fois détectée,
 - augmente charge jusqu'au pt de congesto puis diminue quand congesto.

- Stratégie alternative : - prévient la congesto
 - réduire débit avant de perdre
 - 2 poss : Rouleur ; méca RED
 - : Note : TCP Vegas

TCP Vegas : principe = observer si RTT ou débit fluctue

Base RTT = min(RTTs) mesurés

ExpedRate = cwnd / BaseRTT , se calcule ActualRate 1 fois par RTT

Diff = ExpRate - ActRate

Si diff < a : augmente cwnd linéairement

Si diff > b : diminue cwnd linéairement

Si non : cwnd inchangée

Si N session sur le même lien alors dimensionne à 1/n capacité

cwnd

Taux émiss.

Taux émiss.

Rouleur Congestionné

Liaison surchargée

Taux émiss. = cwnd / RTT

Taux émiss. = cwnd / RTT