

# Internet avancé

Édouard Lumet - IR2020

## Les relations inter-FAI

---

### Taille d'un AS

- nombre de clients ?
  - AS
  - particuliers
  - entreprises
- plage d'adresses
- nombre de réseaux
- trafic géré
  - intra
  - sortant
  - entrant
- portée/étendu

## Contrôle de congestion en TCP

---

### Les bases

La congestion est un problème de niveau réseau, c'est-à-dire IP. Or IP s'en balles et dit "Tant pis !". On demande alors au niveau transport de faire le contrôle de congestion pour lequel sa seule information est les ACK qu'il reçoit, pourtant il ne la voit pas...

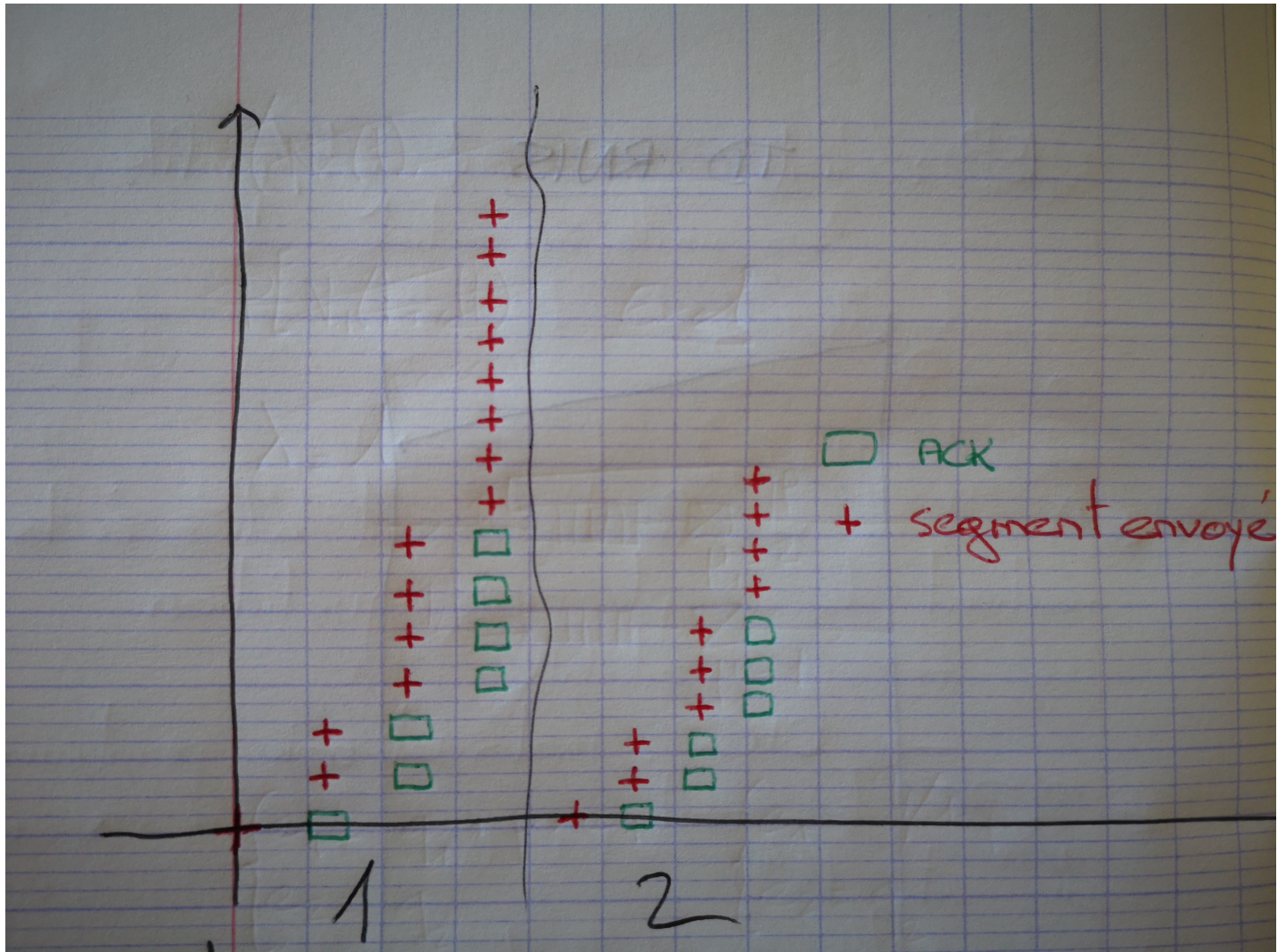
Le premier objectif est de borner la fenêtre d'émission en fonction du contrôle de flux et du contrôle de congestion :  $uwnd$  (usable window) =  $\min(awind, cwnd)$ .

Pour fixer la cwnd, on fait du démarrage lent (slow start). L'objectif étant de ne pas créer de burst au démarrage de la connexion qui viendrait congestionner le réseau. Au départ :

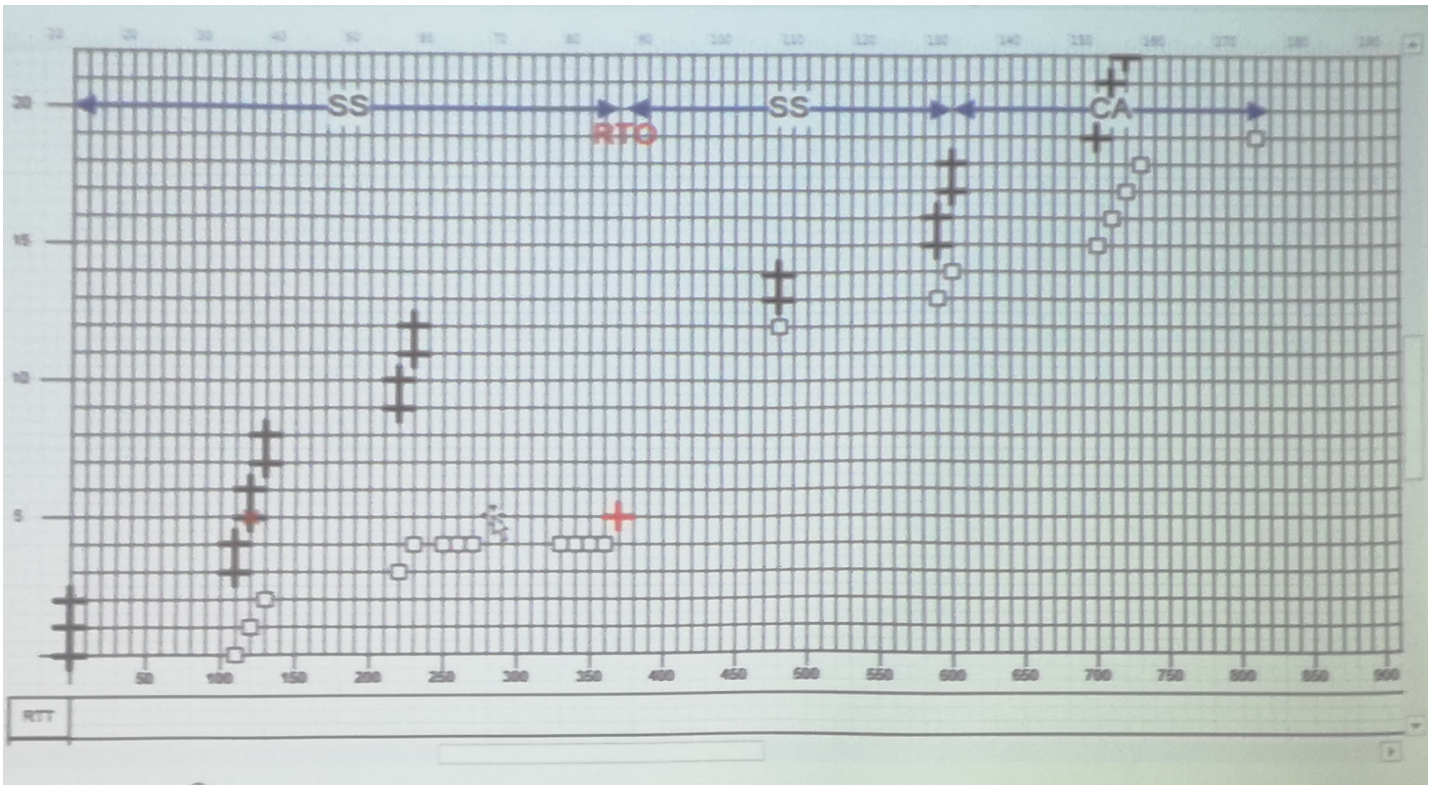
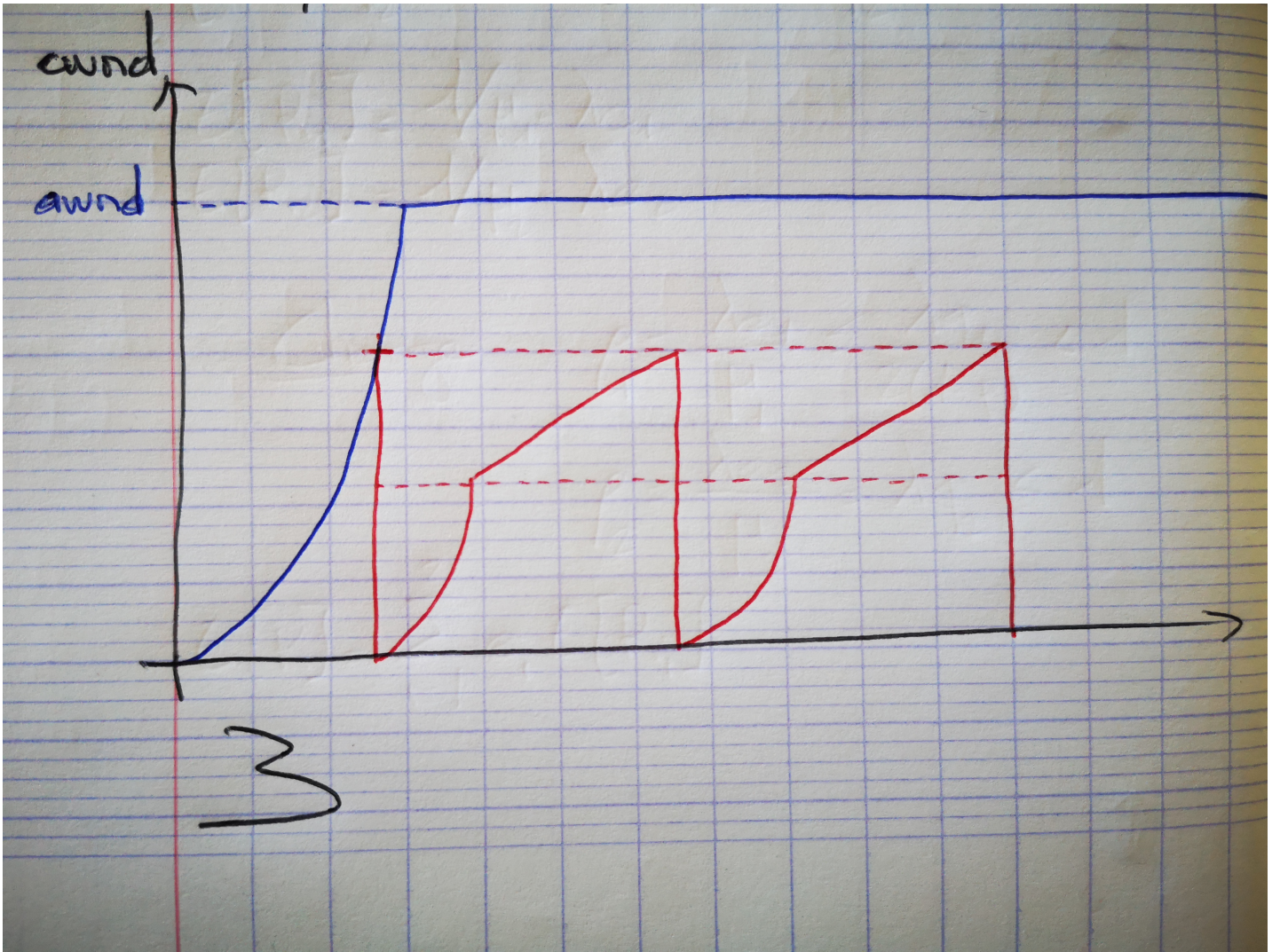
```
cwnd ← InitialWindow avec IW = 1 segment (ou 3 ou 10)
A chaque 'ACK valide'
  cwnd ← cwnd + 1
```

Ensuite il faut prendre en compte la congestion.. comment ? L'objectif est de modifier cwnd en réaction à de la congestion. Mais avant cela, comment détecter de la congestion ? On part du postulat qu'une perte de paquet est due à de la congestion.

Pour détecter la perte sur RTO on considère un RTO comme de la congestion et s'adapter au problème. Au RTO, on sauvegarde la valeur de cwn au moment de la perte dans le Slow Start Threshold :  $ssthresh \leftarrow \min(cwnd/2, awnd)$ .



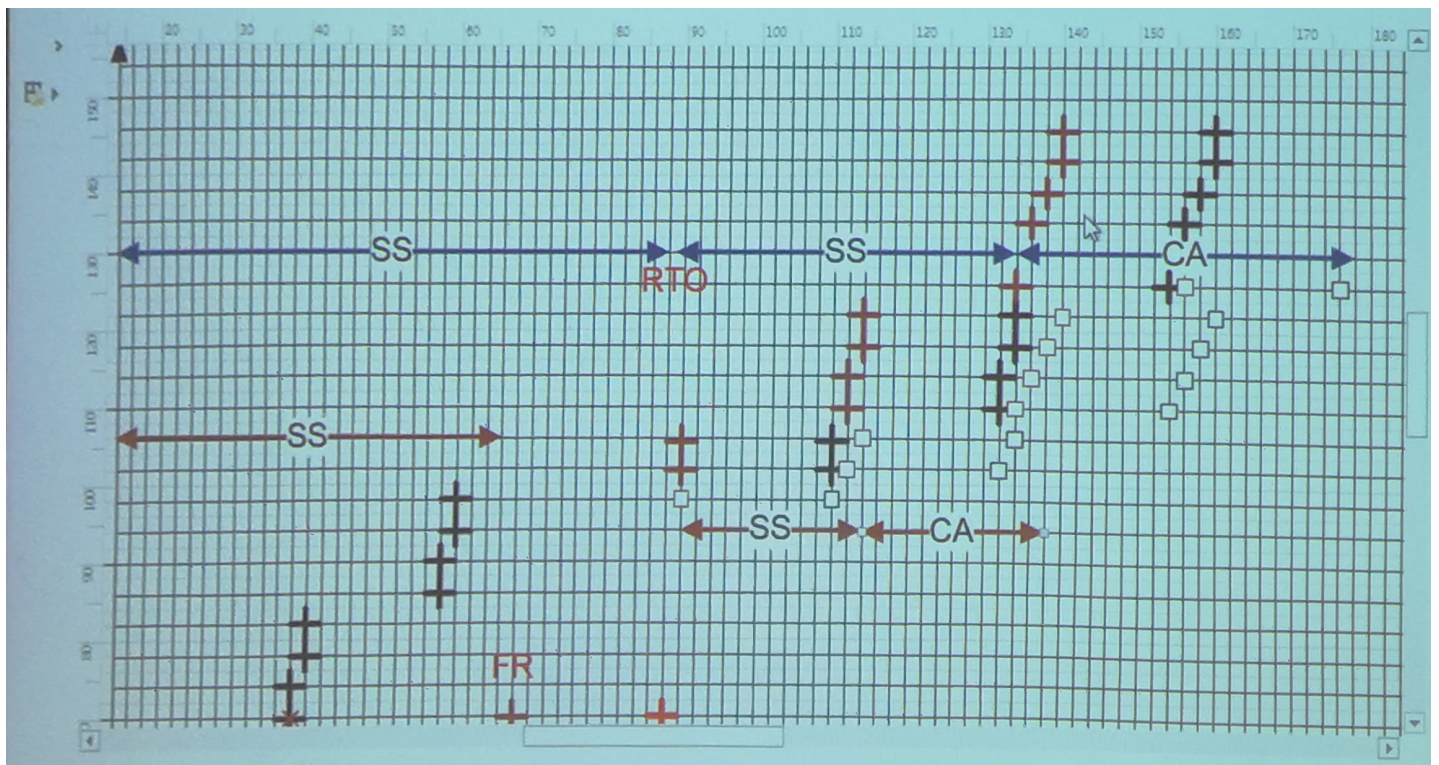
La connexion repart en SS jusqu'à  $ssthresh$ .  $cwnd \leftarrow IW$  (ou 1). Une fois cwnd à la valeur  $ssthresh$ , alors la connexion passe en Congestion Avoidance. En CA, la fenêtre de congestion évolue linéairement :  $cwnd \leftarrow cwnd + 1$



Mise en place du Fast Retransmit afin de détecter plus rapidement un problème de "congestion" sur le réseau. Cependant une perte peut entraîner des accusés dupliqués. Toutefois, les ndups peuvent aussi provenir d'une différence de chemin. Par observation, on fixe le nombre de ndups maximal dus à un changement de chemin à 2. Au delà, ces ndups sont considérés comme dus à une perte, et donc c'est une congestion. A partir de la réception des 3 ndups, on considère qu'il a une perte et on réémet le 1er segment de la fenêtre d'émission

```
ssthresh ← cwnd/2
cwnd ← 1 | IW
```

et on repart en SS jusqu'à SST et on repasse ensuite en CA.



↳ TCP Tahoe

## Évolutions basiques de TCP

Une première évolution est TCP New Reno (suite à TCP Reno). L'objectif est de prendre en compte le débit utile pendant le Fast Retransmit : ne pas surconsidérer une perte ponctuelle. On introduit une nouvelle phase entre Fast Retransmit et la réception de son accusé de réception. Cette phase se nomme Fast Recovery :

```
cwnd ← ssthresh + ndups
```

A la réception de l'ACK, tant qu'il ne valide pas l'intégralité de la fenêtre au moment de la détection de la perte, l'émetteur réémet le 1er segment de sa fenêtre restante et repasse en Fast Recovery. Quand la

fenêtre au moment de la perte est récupérée, l'émetteur passe en CA avec  $cwnd \leftarrow ssthresh$ .

## Exemple

