

Réseaux informatiques

RXI

Chapitre 10

TCP et UDP

2^{ème} partie



Objectifs d'apprentissage

- Savoir expliquer l'utilisation du protocole UDP
- Savoir décrire l'utilisation des numéros de port
- Comprendre les bases des mécanismes de TCP
 - Etablissement et terminaison de connexions
 - Numéros de séquence
 - Acquittement
 - Retransmission
 - Contrôle de congestion



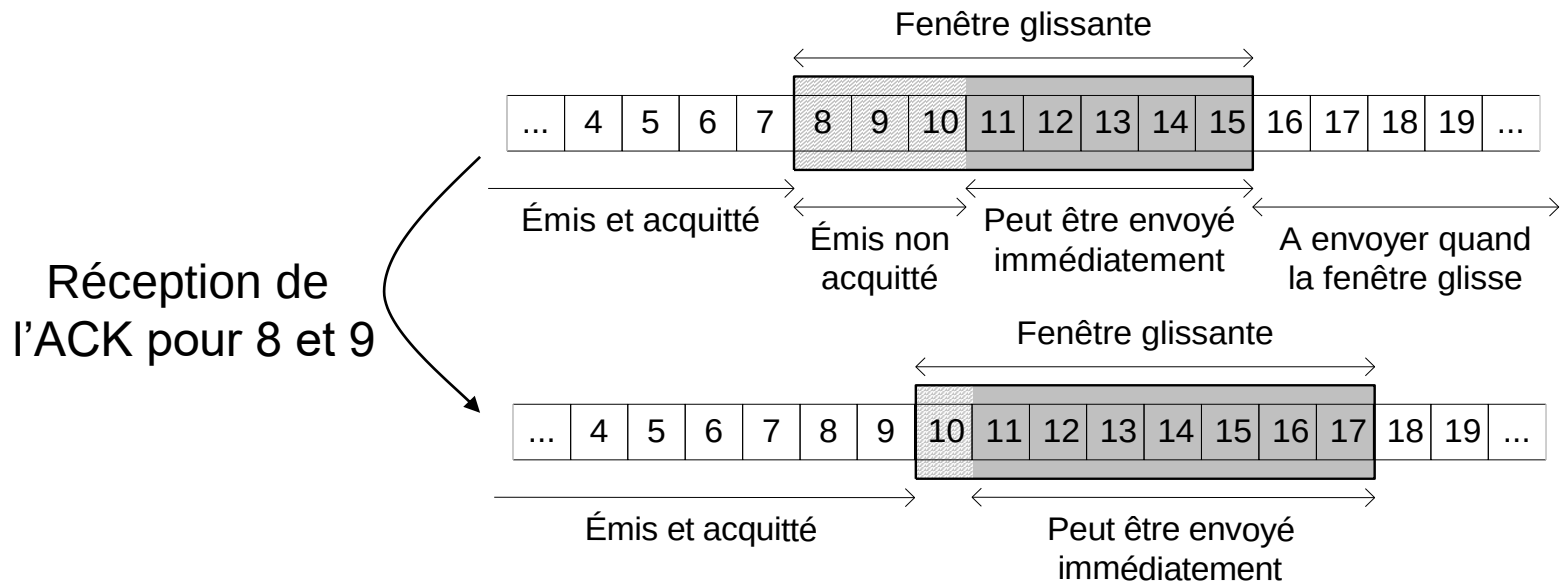
Contrôle de flux et de congestion

- Objectif :
 - Régulation de la vitesse de transmission
- Contrôle de flux
 - Adaptation à la vitesse du récepteur
 - Évite qu'un émetteur rapide surcharge un récepteur lent
- Contrôle de congestion
 - Adaptation à la vitesse du réseau
 - Évite des pertes excessives de paquets à cause d'une surcharge du réseau



Protocole de la fenêtre glissante

- Améliore le taux d'utilisation
 - Permettant à l'émetteur d'envoyer plusieurs paquets avant de devoir attendre un accusé de réception
- Principe
 - Les données dans la fenêtre peuvent être envoyées sans attendre d'acquittement
 - La réception d'un acquittement permet de glisser la fenêtre à droite

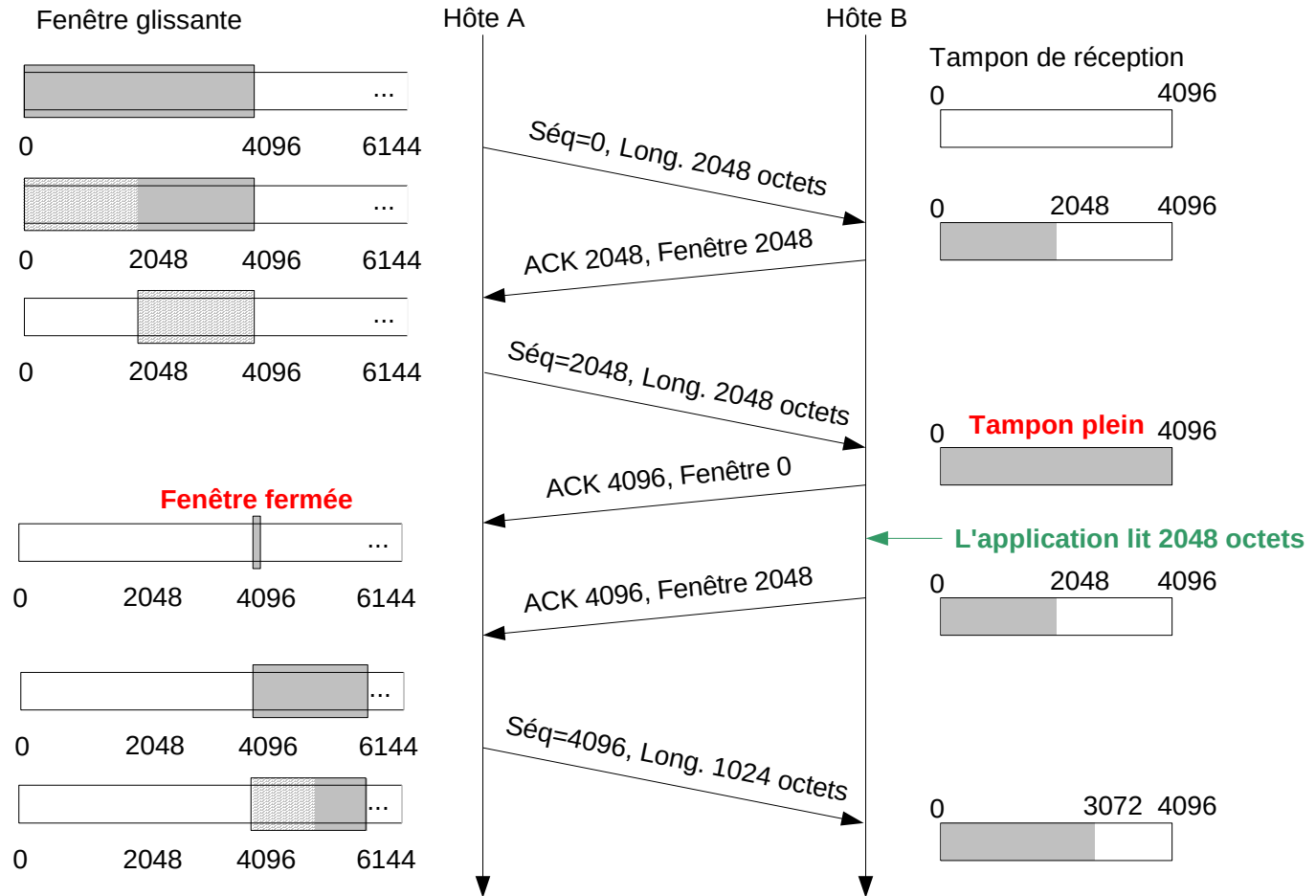


Contrôle de flux dans TCP

- Basé sur un protocole à fenêtre glissante mais avec une **taille de fenêtre variable**
 - La fenêtre utilisable correspond à la place libre dans le tampon du récepteur
 - Chaque accusé de réception indique en plus la taille de la fenêtre (**window advertisement**)
- **En variant la taille de la fenêtre, le récepteur peut contrôler la vitesse de transmission de l'émetteur**



Exemple du contrôle de flux dans TCP



Contrôle de congestion

- État de congestion
 - Le réseau n'est plus en mesure de transporter tout le trafic injecté et perd des paquets
- **Danger de l'amplification d'une congestion** par TCP
 - TCP réagit à une perte avec une retransmission
 - La retransmission peut augmenter la charge du réseau
- Le contrôle de congestion de TCP doit optimiser le débit de transmission sans mettre en danger la stabilité du réseau



Bases du contrôle de congestion de TCP

- La **fenêtre de congestion** (*cwnd*, *congestion window*)
 - Gérée par l'émetteur pour limiter le débit d'émission
 - TCP adapte la taille de *cwnd* au niveau de congestion détecté
- Combinaison du contrôle de flux et de congestion

Fenêtre effective = $\min(\text{Annonce de fenêtre}, cwnd)$
- La taille de la fenêtre est proportionnelle au débit, donc la fenêtre régule le débit



Principe du contrôle de congestion

- L'émetteur adapte valeur cwnd en fonction des acquittements reçus ou des pertes détectées
- Il y a trois phases, avec des règles différentes pour adapter cwnd
 - **Démarrage lent** : Pour détecter le débit optimal, TCP commence avec une petite fenêtre de congestion et augmente rapidement le débit
 - **Évitement de congestion** : Dès que TCP s'approche de la zone de congestion, TCP augmente le débit plus lentement
 - **Accroissement additif - décroissance multiplicative** : Dès qu'une perte est détectée TCP ralentit rapidement puis augmente à nouveau lentement



Démarrage lent (*Slow Start*)

- Motivation
 - TCP doit fonctionner correctement pour n'importe quelle capacité du réseau (entre bits/s et Gb/s)
 - Il faut éviter de surcharger un lien lent au départ
 - Petite fenêtre cwnd au début
 - Il faut rapidement arriver à exploiter la capacité de liens importants
 - Augmentation rapide de cwnd

Algorithme Slow Start

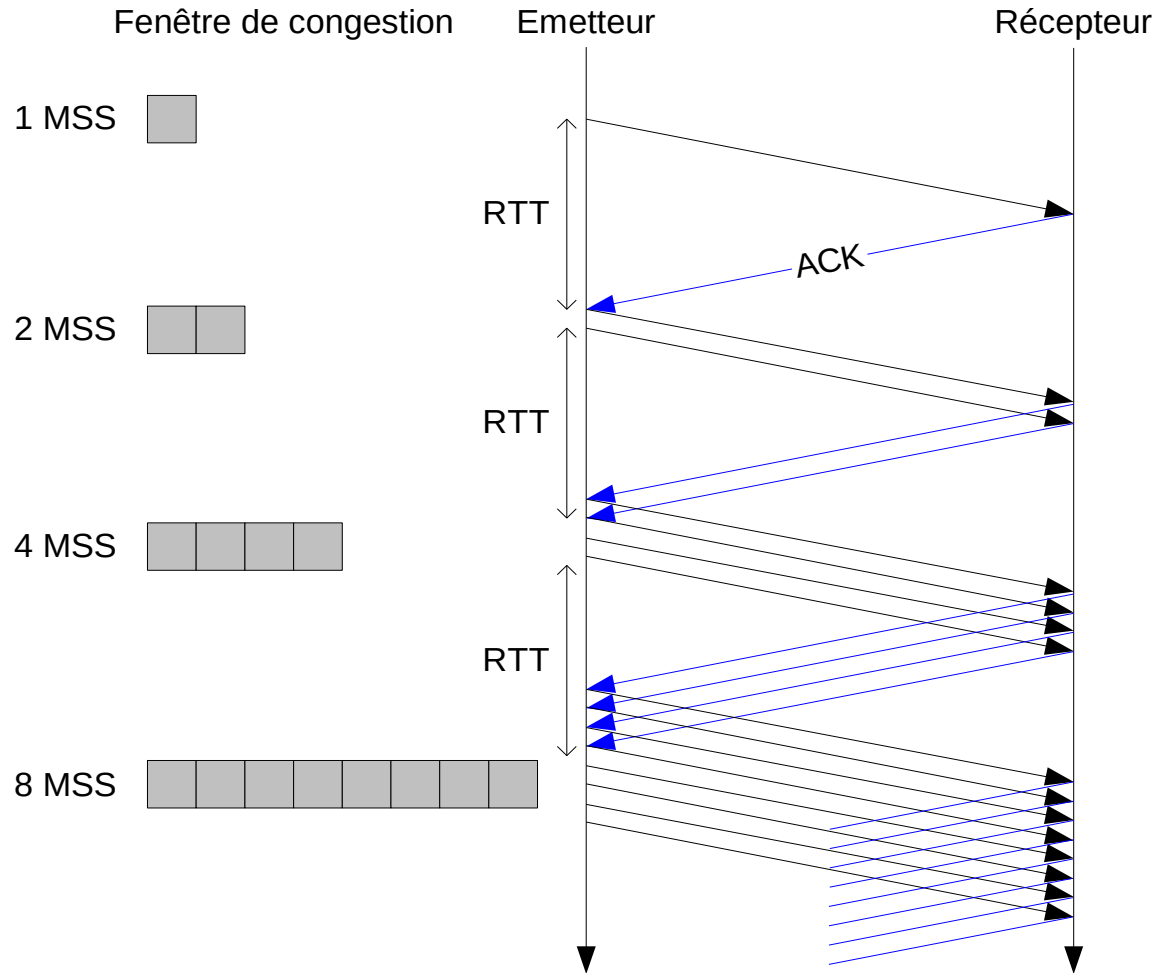
- Initialement cwnd = 1 MSS
- Ensuite cwnd est incrémenté de 1 MSS par acquittement reçu

(MSS: *Maximum Segment Size*: taille maximum d'un paquet)

- Cwnd double chaque RTT (délai aller-retour)



Exemple de Slow Start



Évitement de congestion (*Congestion Avoidance*)

- Dans Slow Start, la taille de `cwnd` augmente exponentiellement
- Augmentation doit ralentir quand TCP s'approche du 'débit optimal'
- Un **seuil d'évitement de congestion** indique quand on s'approche d'une congestion
 - Paramètre `ssthresh` de TCP

Algorithme **Congestion Avoidance**

- Dès que `cwnd > ssthresh`, `cwnd` est agrandie linéairement
- Pour chaque acquittement reçu :

$$cwnd \leftarrow cwnd + \frac{MSS \cdot MSS}{cwnd}$$

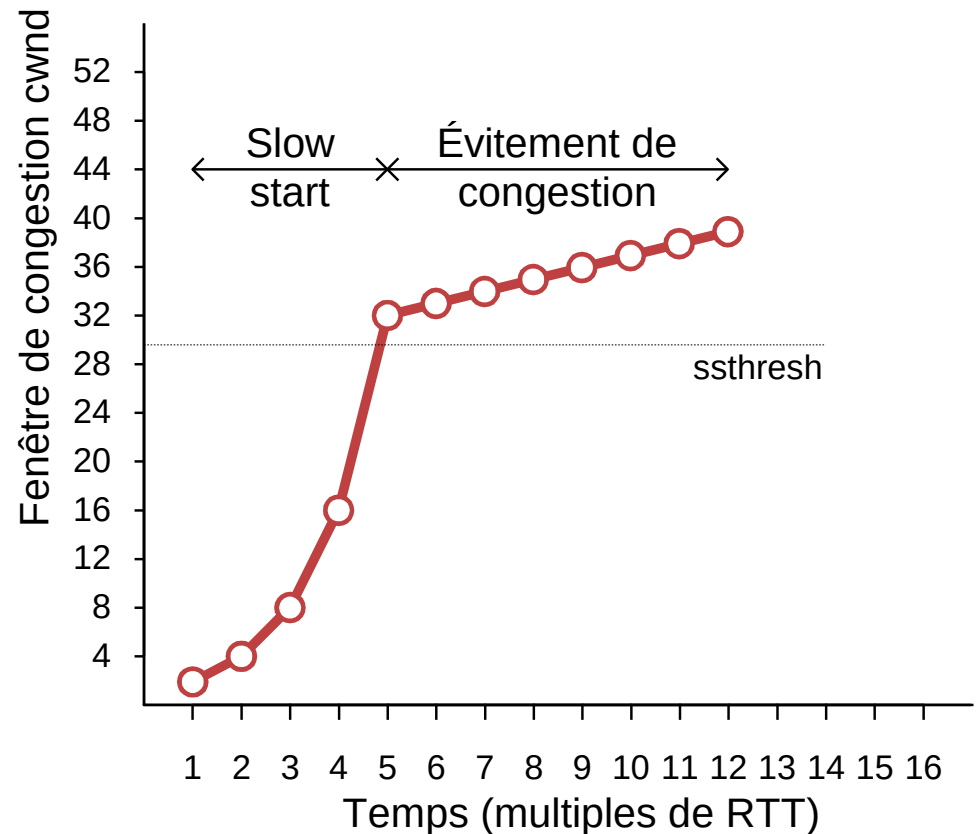
➤ Augmentation d'un MSS par RTT



Exemple

Slow Start et Congestion Avoidance

- **Au-dessous de ssthresh:**
Slow Start
 - Cwnd double chaque RTT
- **Au-dessus de ssthresh:**
Congestion Avoidance
 - Cwnd croît d'un MSS chaque RTT



Variation du seuil d'évitement de congestion

Comment déterminer la valeur du seuil de congestion `ssthresh` ?

1. Lorsqu'il n'y a pas de congestion, `ssthresh` croît linéairement avec `cwnd` → **accroissement additif**
2. Lorsqu'une perte a été détectée, **`ssthresh` est diminué à la moitié**
→ **décroissance multiplicative**
 - Théorie des files d'attente : Pour garantir la stabilité du réseau, le débit doit diminuer de manière exponentielle

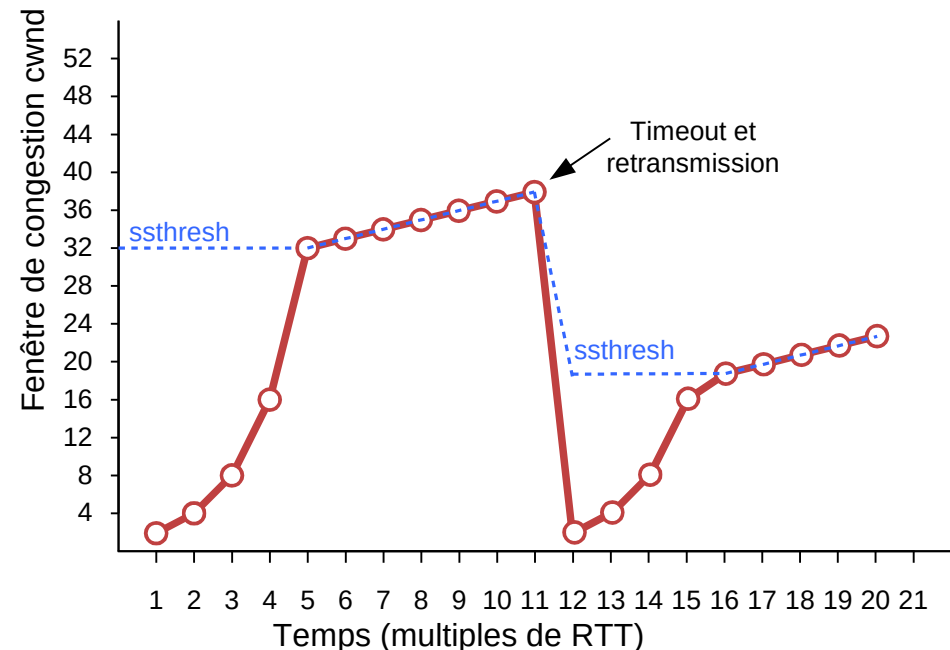
TCP (Reno) recommence avec Slow Start (après un timeout)

- Pour éviter des rafales de retransmissions



Comportement de cwnd et ssthresh

- Trois phases
 - **Slow Start** avec une croissance exponentielle de cwnd
 - Congestion avoidance (**accroissement additif de ssthresh**)
 - Diminution de ssthresh lors d'une perte (**décroissance multiplicative de ssthresh**)



Recouvrement rapide (*Fast Recovery*)

- L'utilisation de Slow Start après des pertes isolées n'est pas optimale
 - Si des segments intermédiaires ont été perdus mais les segments suivants sont arrivés, cela indique une congestion légère et ne justifie pas Slow Start

Recouvrement rapide pour résoudre rapidement la perte de segments isolés

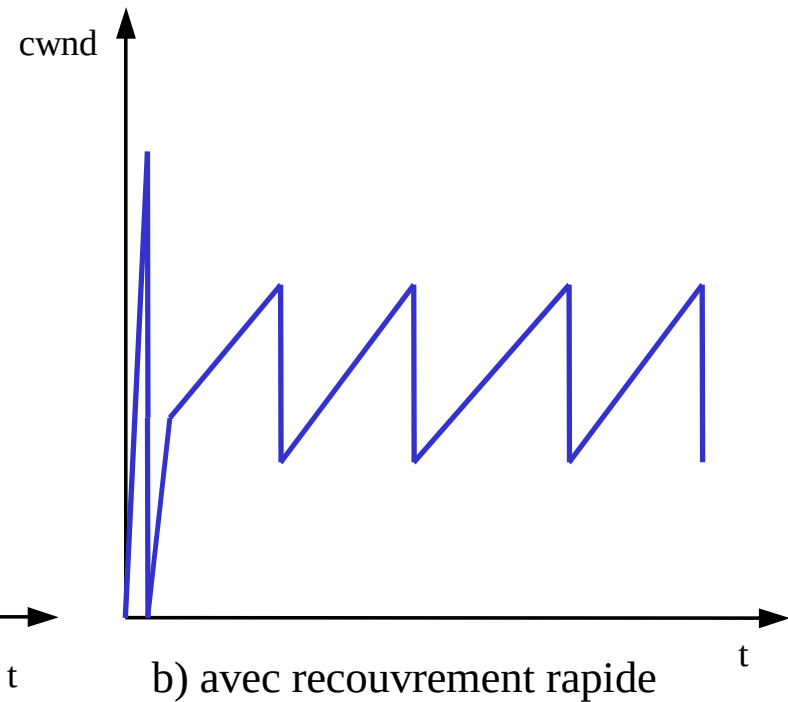
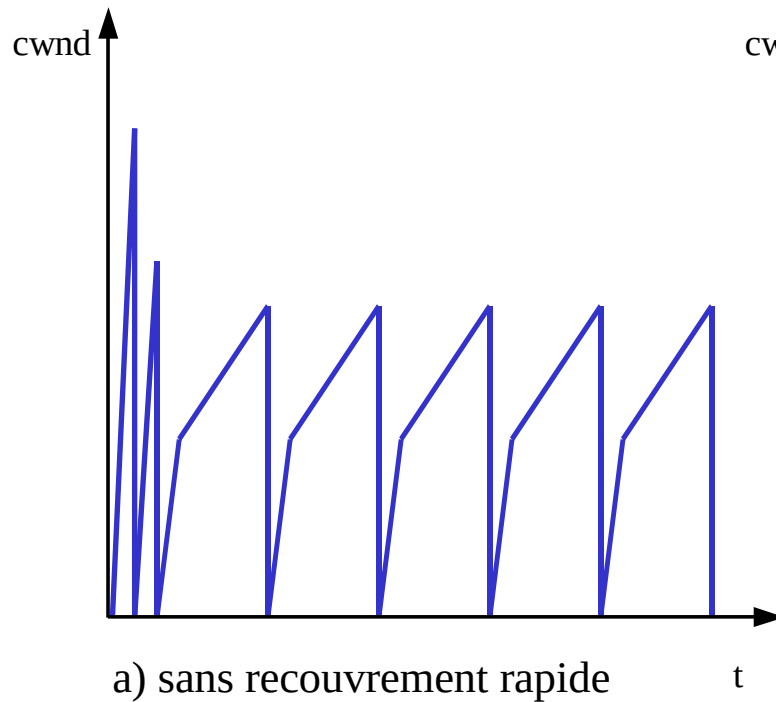
- Utilisé en combinaison avec Retransmission Rapide

Algorithme **Recouvrement Rapide**

- Retransmettre rapidement le segment manquant
- Diminuer la fenêtre à la moitié
- Continuer avec Congestion Avoidance



Effet de recouvrement rapide sur le débit



Résumé du contrôle de congestion dans TCP

- La taille de la fenêtre de congestion est variée en fonction de la congestion du réseau
 - Une congestion est détectée à cause de pertes de paquets
- **Slow Start** : croissance exponentielle du débit
 - Au début de la connexion et après un timeout de retransmission (→ congestion sévère)
 - Permet d'augmenter rapidement le débit
- **Congestion avoidance**: croissance linéaire du débit
 - Dès que le débit s'approche à la capacité disponible
- **Seuil d'évitement de congestion** ssthresh
 - Indique la zone critique où une congestion est possible
 - Accroissement additif et décroissance multiplicative de ssthresh
 - Ssthresh oscille entre le débit maximum et la moitié de ce débit

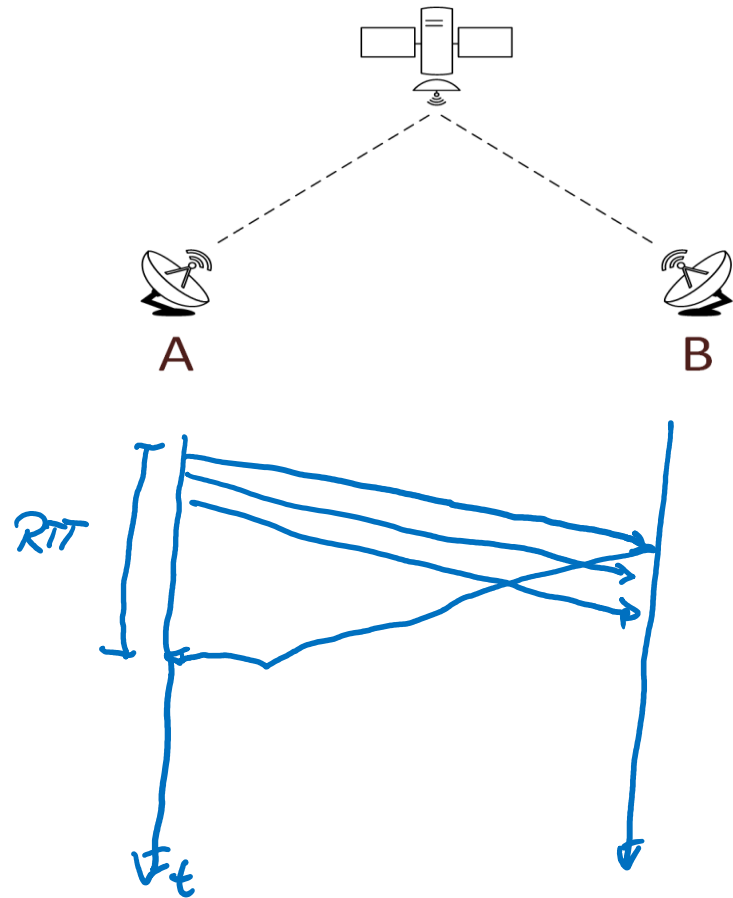


Exercice – débit et fenêtre glissante

Transmission par satellite

- Délai aller-retour: 500 ms
- Taille de la fenêtre glissante: 5'000 octets

$$\begin{aligned}\text{débit} &= \frac{\text{données}}{RTT} = \\ &= \frac{5000 \text{ B} \cdot 8 \text{ b/B}}{0.5 \text{ s}} = \\ &= \frac{40000 \text{ b}}{0.5 \text{ s}} = 80'000 \text{ b/s} \\ &= 80 \text{ kb/s}\end{aligned}$$



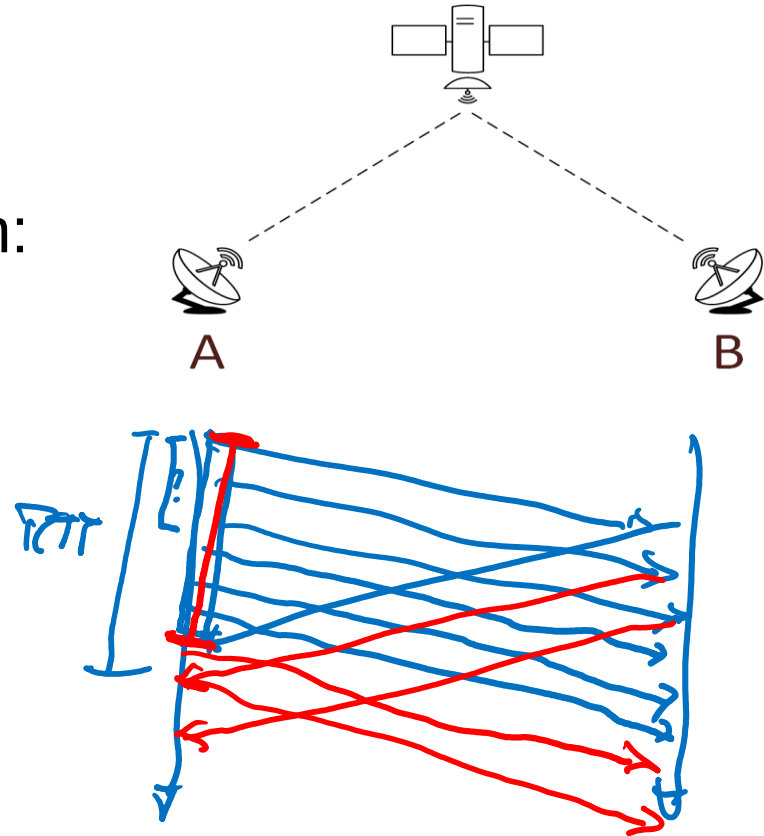
Exercice – débit et fenêtre glissante

Transmission par satellite

- Délai aller-retour: 500 ms
- Débit maximum de la liaison: 2 Mb/s

$$\text{débit} = \frac{\text{données}}{\text{RTT}}$$

$$\begin{aligned}\text{données} &= \text{débit} \cdot \text{RTT} \\ &= 2 \text{ Mb/s} \cdot 0.5 \text{ s} \\ &= 1 \text{ Mb} = \underline{125 \text{ kB}}\end{aligned}$$



Fin du chapitre 10

