

Répétitions du Cours d'Introduction aux Réseaux Informatiques Couche Transport

François Cantin

Département Montefiore
Research Unit in Networking
Université de Liège

Année académique 2008–2009

Contact

- Bureau : Institut Montefiore (B28), I77a.
- Email : cantin@run.montefiore.ulg.ac.be.
- Téléphone : 04 366 26 94.

Table des matières

- 1 Algorithme Slow Start
 - Principes de fonctionnement
 - Vitesse maximale d'une connexion TCP
 - Exercices

- 2 Modélisation du délai de TCP
 - Rappels théoriques
 - Exercices

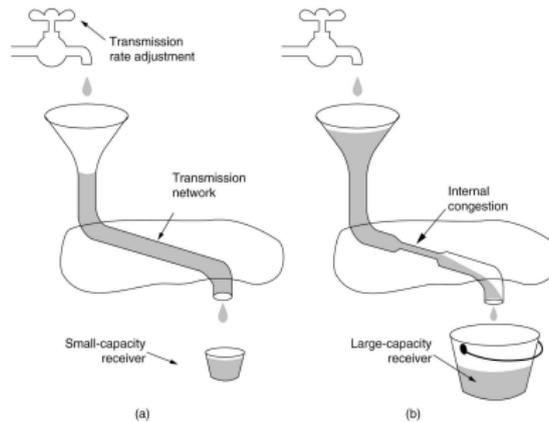
Table des matières

- 1 Algorithme Slow Start
 - Principes de fonctionnement
 - Vitesse maximale d'une connexion TCP
 - Exercices

- 2 Modélisation du délai de TCP
 - Rappels théoriques
 - Exercices

Utilité du contrôle de congestion

Le mécanisme de la fenêtre glissante (*fenêtre de réception*) fournit un contrôle de flux : il empêche l'émetteur de submerger le récepteur. Cependant cela ne permet pas d'empêcher les congestions.



Fenêtre de congestion

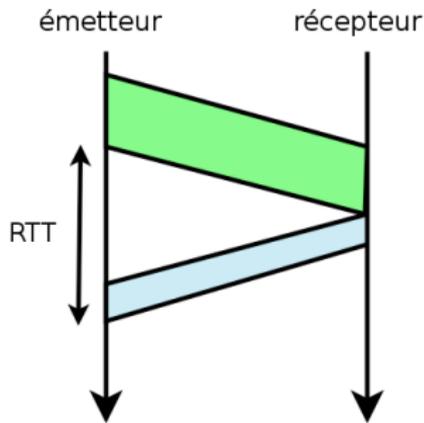
C'est pourquoi on utilise une deuxième fenêtre : la *fenêtre de congestion*. Le nombre de bytes que l'on peut transmettre sans attendre les acquits correspondants est le minimum de la taille des deux fenêtres.

L'algorithme du *Slow Start* est utilisé pour déterminer la taille de la fenêtre de congestion. Il procède en augmentant progressivement la charge imposée au réseau (nbre de bytes envoyés) en se basant sur le MSS (*Maximum Segment Size*) fixé par le récepteur. Le MSS désigne la quantité de données **utiles** maximale que peut contenir un segment (i.e., sans les en-têtes).

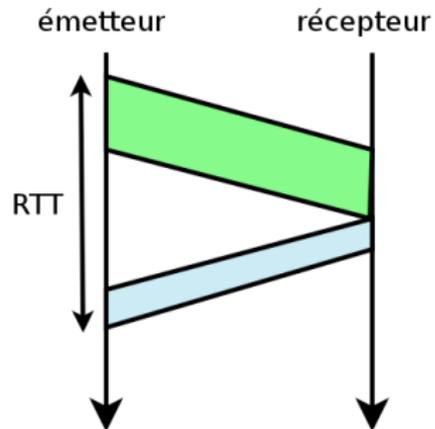
Rappel : Les en-têtes IP et TCP sont toutes deux constituées de 20 octets de données.

Définition du RTT

Le RTT (*Round Trip Time*) peut être défini de plusieurs façons.



Au cours théorique



Aux répétitions

Slow Start (1)

Fonctionnement

- A l'ouverture d'une connexion, la taille de la fenêtre est de 1 MSS (ce n'est pas toujours le cas en pratique).
- La taille de la fenêtre augmente d'une unité par segment acquitté. Si n segments sont acquittés, la taille de la fenêtre augmente donc de n .
- Pour des transferts en rafale, la taille de la fenêtre double à chaque RTT (croissance exponentielle).
- Quand la taille de la fenêtre de congestion \geq la taille de la fenêtre de réception, alors la taille de la fenêtre de congestion = la taille de la fenêtre de réception.
- Quand la taille de la fenêtre atteint un seuil imposé (*threshold*), la croissance de la fenêtre devient linéaire. La taille de la fenêtre augmente donc d'une unité **seulement** quand le nombre de segments acquittés est égal à la taille de la fenêtre.

Slow Start (2)

Détection et gestion des pertes

Pour TCP, une perte d'un segment est due à une congestion. Dans les réseaux filaires où la probabilité d'erreur est très faible, on peut considérer cette affirmation comme vraie. Cela n'est pas le cas dans les réseaux sans fil et l'étude de ce problème fait d'ailleurs l'objet de plusieurs travaux.

Il y a deux façons différentes de détecter une perte :

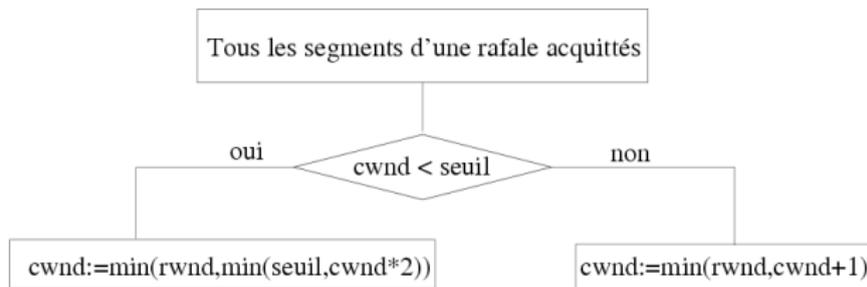
- Si un segment n'a pas été acquitté après un certain temps (expiration du timer), la valeur du seuil devient la moitié de la taille **de la fenêtre de congestion**. Ensuite la taille de la fenêtre de congestion est réinitialisée à 1 MSS.
- Si l'émetteur reçoit un certain nombre (typiquement 3) de doublons d'acquets, la valeur du seuil devient également la moitié de la taille de la fenêtre de congestion **mais** la taille de la fenêtre de congestion est réinitialisée **à la nouvelle valeur du seuil**.

Slow Start (3)

Résumé (1)

Soient $cwnd$ (au cours théorique, $cwnd = CongWin = Congestion Window$) la taille de la fenêtre de congestion et $rwnd$ la taille de la fenêtre de réception.

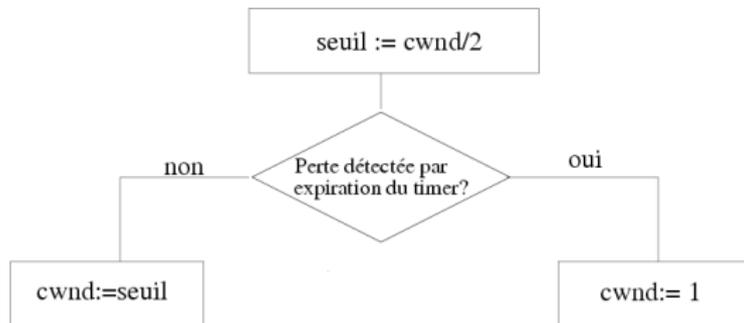
Augmentation de la taille de la fenêtre de congestion :



Slow Start (4)

Résumé (2)

Diminution de la taille de la fenêtre de congestion :



Slow Start (5)

Illustration

Connexion TCP où le seuil initial est de 20 MSS et la taille de la fenêtre de réception est de 28 MSS. Une station émet en rafale des données et à la 8^{ème} transmission (22 MSS en rafale) le timer expire.

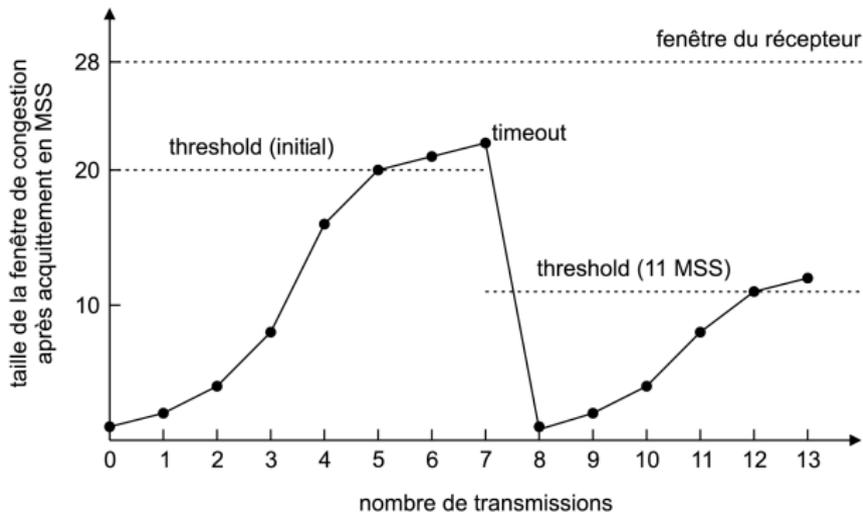


Table des matières

- 1 Algorithme Slow Start
 - Principes de fonctionnement
 - **Vitesse maximale d'une connexion TCP**
 - Exercices

- 2 Modélisation du délai de TCP
 - Rappels théoriques
 - Exercices

Déterminer la vitesse maximale

La vitesse maximale peut être limitée par la capacité du réseau ou par la taille de la fenêtre de réception.

Si

$$\underbrace{\frac{rwnd \cdot (MSS + \text{en-têtes})}{RTT}}_{\text{débit brut maximal}} \leq \text{débit ligne la + lente}$$

alors c'est la fenêtre du récepteur qui va limiter la vitesse maximale.

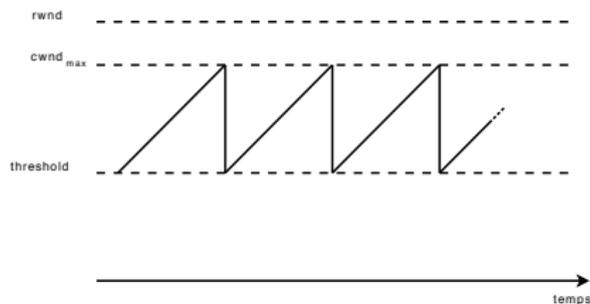
Sinon, c'est la capacité du réseau. Dans ce cas, il faut résoudre l'inéquation

$$\frac{cwnd \cdot (MSS + \text{en-têtes})}{RTT} \leq \text{débit ligne la + lente}$$

pour déterminer la taille maximale de la fenêtre de congestion (et donc pour déterminer la vitesse maximale). Remarquez que la solution de cette inéquation doit être $< rwnd$.

Déterminer la vitesse maximale (2)

Remarque : La valeur $cwnd_{max}$ obtenue en résolvant la seconde inéquation est le débit maximal autorisé sur le réseau entre les entités communicantes. On considère qu'aller au delà de cette valeur entraînera des pertes de paquets et une réduction de la taille de la fenêtre de congestion.



En moyenne, sur la durée de l'échange, le débit de la source sera inférieur à cette valeur.

Table des matières

- 1 Algorithme Slow Start
 - Principes de fonctionnement
 - Vitesse maximale d'une connexion TCP
 - Exercices

- 2 Modélisation du délai de TCP
 - Rappels théoriques
 - Exercices

Exercice 1

Considérons l'effet de l'utilisation du Slow Start sur une ligne avec un RTT de 10 ms sans congestion. La fenêtre du récepteur est de 24 Kb avec un MSS de 2 Kb. Le seuil est au départ fixé à 32 Kb et on suppose que la source désire émettre en continu.

Combien de temps faut-il attendre pour que la taille de la fenêtre de congestion soit maximale sans tenir compte des délais d'émission et de réception ?

Résolution exercice 1

- Pas de congestion \Rightarrow c'est *rwnd* qui limite *cwnd*.
- Initialement, $cwnd = 1$ MSS, $rwnd = 12$ MSS et seuil = 16 MSS.
- Chaque émission de segments en rafale et la réception des acquits correspondants se font en un RTT càd 10 ms.

temps	reçu	taille fenêtre	émis
0	—	1	1 MSS
1 RTT	1 ACK	+1 \rightarrow 2	2 MSS
2 RTT	2 ACK	+2 \rightarrow 4	4 MSS
3 RTT	4 ACK	+4 \rightarrow 8	8 MSS
4 RTT	8 ACK	+4 \rightarrow 12	12 MSS

La taille de la fenêtre de congestion a atteint celle de la fenêtre de réception et ne pourra donc plus augmenter. Pour que la taille de la fenêtre de congestion soit maximale, il faut 4 RTT = 40 ms.

Exercice 2

Supposons que la taille de la fenêtre de congestion de TCP soit égale à 18 Kb ($MSS = 1 \text{ Kb}$) et qu'un timeout se déclenche. Quelle sera la *taille maximale* de la fenêtre de congestion quand 4 transmissions en rafale auront été acquittées normalement ?

Résolution exercice 2

Pour que la taille soit maximale, il faut que les rafales successives remplissent la fenêtre de congestion, sinon la taille de la fenêtre croîtra plus lentement que par doublements successifs, ce qui ne permettra pas d'atteindre la taille maximale.

Après le timeout, la taille de la fenêtre de congestion est de 1 MSS et le threshold vaut la moitié de la taille de la fenêtre de congestion avant le crash (càd 9 MSS).

temps	reçu	taille fenêtre	émis
0	—	1 MSS	1 MSS
1 RTT	1 ACK	+1 → 2 MSS	2 MSS
2 RTT	2 ACK	+2 → 4 MSS	4 MSS
3 RTT	4 ACK	+4 → 8 MSS	8 MSS
4 RTT	8 ACK	+1 → 9 MSS	...

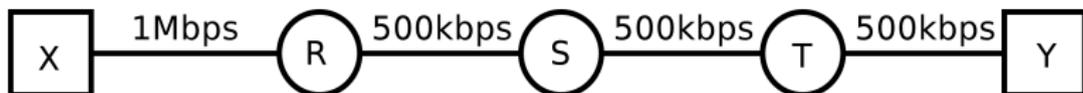
La taille maximale de la fenêtre après 4 rafales vaut 9 MSS = 9 Kb.

Exercice 3 (août 2002)

X et Y sont reliés par une connexion sur laquelle se trouvent 3 routeurs R , S , et T . La liaison de X à R est de 1 Mbps et les autres liaisons sont à 500 kbps. Tous les temps de propagation sont négligeables. X transmet des données à Y à l'aide d'une connexion TCP. La fenêtre de réception de Y est de 12 MSS, sachant que la taille maximale des segments a été négociée à 512 Bytes. Le temps de réaction des routeurs est de 10 ms, et le temps de traitement des stations (vérification d'un segment et génération d'un acquit, vérification d'un acquit et préparation d'émission d'un nouveau segment) est de 15 ms. Chaque segment reçu par Y donne lieu à l'émission immédiate d'un acquit (segment TCP vide). Le *timeout* avant que X ne considère un segment comme perdu est de 2 s. On prendra en compte l'overhead introduit par TCP et IP.

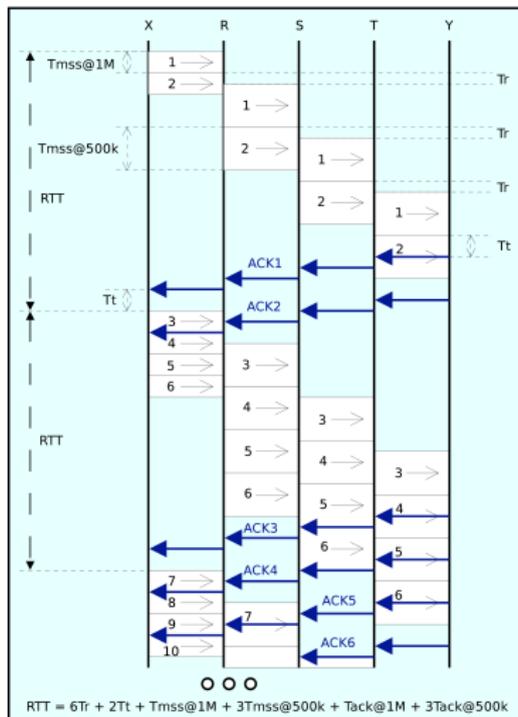
Exercice 3 (août 2002) (2)

- 1 Quel temps faut-il à X et Y dans le meilleur des cas pour atteindre le débit maximal, sachant que le *threshold* est initialisé à 4 KBytes ?
- 2 Quelle est l'efficacité des différentes lignes au débit maximal ?
- 3 Pour un *threshold* initial de 20 MSS, sachant que tous les acquits d'une rafale viennent d'être perdus, combien de temps faudra-t-il avant que le débit maximum ne soit récupéré ?



Résolution exercice 3

Calcul du RTT



Résolution exercice 3 (2)

Calcul du RTT (2)

Le *RTT* est le délai entre l'émission du premier bit d'un paquet et la réception du dernier bit de son acquit (ne pas oublier d'inclure les en-têtes dans le calcul du temps de transmission).

$$T_{MSS@500k} = \frac{(512 + 40) \cdot 8}{500 \cdot 10^3} = 8.83 \text{ ms}$$

$$T_{ACK@500k} = \frac{40 \cdot 8}{500 \cdot 10^3} = 0.64 \text{ ms}$$

$$\begin{aligned} RTT &= 3.5T_{MSS@500k} + 3.5T_{ACK@500k} + 6T_r + 2T_t \\ &= 3.5 \cdot 8.83 + 3.5 \cdot 0.64 + 6 \cdot 10 + 2 \cdot 15 \\ &= 123.15 \text{ ms} \end{aligned}$$

$$\frac{rwnd \cdot (MSS + \text{en-têtes})}{RTT} = \frac{12 \cdot ((512 + 40) \cdot 8)}{0.12315} \approx 430304 \text{ bps}$$

Résolution exercice 3 (3)

Temps pour atteindre le débit maximal

Comme $\frac{rwnd \cdot (MSS + \text{en-têtes})}{RTT}$ est inférieur au débit de la ligne la plus lente (500000 bps), c'est la fenêtre du récepteur qui va limiter la vitesse maximale ($\Rightarrow cwnd_{max} = rwnd = 12$). Nous avons donc :

temps	reçu	taille fenêtre	émis
0	—	1	1MSS
1 RTT	1 ACK	+1 \rightarrow 2	2 MSS
2 RTT	2 ACK	+2 \rightarrow 4	4 MSS
3 RTT	4 ACK	+4 \rightarrow 8	8 MSS
4 RTT	8 ACK	+1 \rightarrow 9	9 MSS
5 RTT	9 ACK	+1 \rightarrow 10	10 MSS
6 RTT	10 ACK	+1 \rightarrow 11	11 MSS
7 RTT	11 ACK	+1 \rightarrow 12	12 MSS

Il faut donc 7 RTT pour atteindre le débit maximal, soit 862.05 ms.

Résolution exercice 3 (4)

Efficacité au débit maximal et temps de récupération après la perte

Efficacité au débit maximal

A la vitesse maximale, le débit est de

$$D_{max} = \frac{cwnd_{max} \cdot MSS}{RTT} = \frac{12 \cdot 512 \cdot 8}{0.12315} = 399123 \text{ bps}$$

Sur les lignes à 500kbps, cela correspond à une efficacité de 79.82%, et à une efficacité de 39.91% sur la ligne à 1Mbps.

Temps de récupération après la perte

Puisque tous les acquits d'une rafale viennent d'être perdus, il faudra attendre l'expiration d'un timer pour que la perte soit détectée, après quoi un nouveau *Slow Start* sera nécessaire. Le seuil (*threshold*) pour ce nouveau *Slow Start* est fixé à la moitié de la fenêtre de congestion actuelle :

$$thresh = \frac{12 \cdot MSS}{2} = 6 \text{ MSS}$$

Résolution exercice 3 (5)

Temps de récupération après la perte (2)

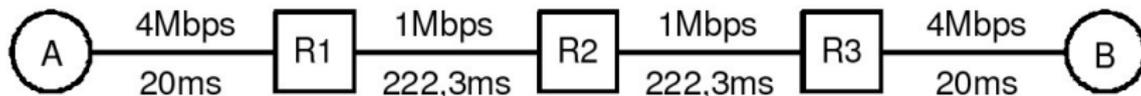
temps	reçu	taille fenêtre	émis
0	—	1	1 MSS
1 RTT	1 ACK	+1 → 2	2 MSS
2 RTT	2 ACK	+2 → 4	4 MSS
3 RTT	4 ACK	$\min(8, thr) = 6$	6 MSS
4 RTT	6 ACK	+1 → 7	7 MSS
...
8 RTT	10 ACK	+1 → 11	11 MSS
9 RTT	11 ACK	+1 → 12	12 MSS

9 RTT seront cette fois nécessaires pour atteindre le débit maximal, ce qui donne donc un délai global

$$T_{\text{recovery}} = T_{\text{timeout}} + 9 \text{ RTT} = 3.108 \text{ s}$$

Exercice 4 (août 2004)

On a la topologie suivante (les nombres présents sur la figure sont les débits et les délais de propagation des lignes) :



Toutes les lignes sont en full duplex, et A a une infinité de données à transmettre à B à l'aide d'une connexion TCP. La fenêtre de réception de B est de 64 MSS et la taille maximale des segments est de 1460 octets. On suppose que les équipements ont un temps de traitement nul.

Exercice 4 (août 2004) (2)

Chaque segment reçu par B donne lieu à l'émission immédiate d'un acquit (segment TCP vide). Si A ne reçoit pas d'acquit dans les 2 secondes qui suivent l'envoi d'un paquet, il considère celui-ci comme perdu. On prendra en compte l'overhead introduit par TCP et IP.

- Sachant que le seuil initial est fixé à 32 MSS, quel temps faut-il à A pour atteindre le débit maximal ?
- Quelle est l'efficacité des différentes lignes (dans le sens $A \rightarrow B$) au débit maximal ?

Résolution exercice 4

A réaliser comme exercice personnel. Il sera corrigé au début de la prochaine séance d'exercices.

Exercice 5 (août 2005)

Une entité A établit une connexion TCP avec une entité B et souhaite envoyer des données à celle-ci. La taille maximale d'un segment a été négociée à 2 Kb et le débit offert par le réseau est de 1 Mbps. La taille de la fenêtre de réception de B est de 50 Kb, le RTT est de 20 ms et le seuil initial de l'algorithme de Slow Start est fixé à 64 Kb. Un timer de retransmission de 1s est enclenché à chaque début d'envoi d'une rafale. On ne prendra pas en compte l'overhead introduit par TCP et IP.

- 1 Sachant que la 4^{ème} rafale est entièrement perdue, combien de temps faut-il pour arriver à une fenêtre de congestion de taille maximale après la perte ?
- 2 Supposons maintenant que le débit offert par le réseau soit de 10 Mbps, quelle est alors l'efficacité maximale ?

Résolution exercice 5

Seuil après la perte

temps	reçu	taille fenêtre	émis
0	–	1	1 MSS
1 RTT	1 ACK	+1 → 2	2 MSS
2 RTT	2 ACK	+2 → 4	4 MSS
3 RTT	4 ACK	+4 → 8	8 MSS

La dernière rafale est perdue \Rightarrow le nouveau seuil est égal à 4 MSS, *cwnd* est remis à 1 et on doit attendre l'expiration du timer de retransmission pour recommencer à émettre. Soit t_{max} le temps recherché. On a :

$$\begin{aligned}
 t_{max} &= t_o + t_{rec} \\
 &= 1 + t_{rec}
 \end{aligned}$$

où t_o désigne le timer de retransmission et t_{rec} le temps pour parvenir au débit maximal **après** la perte.

Résolution exercice 5 (2)

Taille maximale de la fenêtre de congestion

$$\frac{rwnd \cdot MSS}{RTT} \stackrel{?}{\leq} 1 \text{ Mbps}$$

$$\frac{25 \cdot 2 \cdot 10^3}{20 \cdot 10^{-3}} \stackrel{?}{\leq} 1 \cdot 10^6$$

$$\frac{50}{20} \not\leq 1$$

La taille maximale de la fenêtre de congestion est donc limitée par le réseau.

$$\frac{cwnd_{max} \cdot MSS}{RTT} \leq 1 \cdot 10^6$$

$$cwnd_{max} \leq \frac{1 \cdot 10^6 \cdot 20 \cdot 10^{-3}}{MSS}$$

$$cwnd_{max} \leq 10 < rwnd$$

Résolution exercice 5 (3)

Temps pour parvenir au débit maximal après la perte

temps	reçu	taille fenêtre	émis
0	–	1	1 MSS
1 RTT	1 ACK	+1 → 2	2 MSS
2 RTT	2 ACK	+2 → 4	4 MSS
3 RTT	4 ACK	+1 → 5	5 MSS
4 RTT	5 ACK	+1 → 6	6 MSS
...
8 RTT	9 ACK	+1 → 10	10 MSS

Après 8 RTT, la fenêtre de congestion est maximale

$$\Rightarrow t_{rec} = 8 \cdot 20 \cdot 10^{-3} = 0,160 \text{ s.}$$

On a donc $t_{max} = 1 + 0,160 = 1,160 \text{ s.}$

Résolution exercice 5 (4)

Efficacité maximale si le débit offert est de 10 Mbps

Si le débit offert par le réseau est de 10 Mbps, on a :

$$\frac{rwnd \cdot MSS}{RTT} \stackrel{?}{\leq} 10 \cdot 10^6$$

$$\frac{25 \cdot 2 \cdot 10^3}{20 \cdot 10^{-3}} \stackrel{?}{\leq} 10 \cdot 10^6$$

$$\frac{5}{2} \leq 10$$

D'où, dans ce cas-ci, $cwnd_{max} = 25$. On a alors :

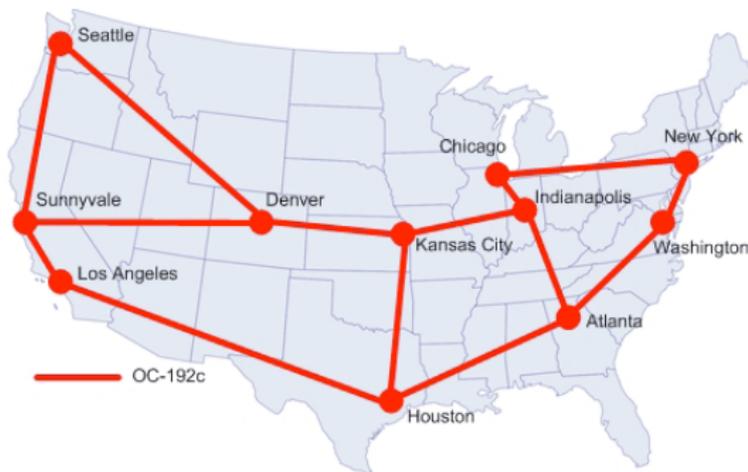
$$D_{max} = \frac{cwnd_{max} \cdot MSS}{RTT} = \frac{25 \cdot 2 \cdot 10^3}{20 \cdot 10^{-3}} = 2,5 \cdot 10^6 \text{ bps} = 2,5 \text{ Mbps}$$

et

$$\eta_{max} = \frac{2,5}{10} = 0,25 = 25\%$$

Exercice 6 (août 2006)

Le réseau Abilene est le réseau américain de la recherche interconnectant les différentes universités et centres de recherche américains. Le réseau, composé de 11 nœuds et de 14 liens bidirectionnels et full duplex d'une capacité de 10 Gbps, est illustré ci-dessous (source : <http://abilene.internet2.edu/images/abilene-current.gif>).



Exercice 6 (août 2006) (2)

Considérons une session TCP entre le routeur de Seattle et le routeur de New-York (on supposera qu'il y a une infinité de données à transférer).

On sait que :

- tous les liens ont un poids de 1 ;
- pour relier 2 points, on choisit le chemin de poids minimum (si plusieurs chemins de poids minimum existent, on choisira toujours le même de façon arbitraire) ;
- la taille maximale des segments (MSS) a été négociée à 1460 octets ;
- les temps de traitement, de transmission et de queuing sont négligeables ;
- le délai de propagation de chaque lien est de 5ms ;
- la réception d'un segment par le routeur de New-York donne lieu à l'émission immédiate d'un acquit (segment TCP vide) ;
- la taille de la fenêtre de réception du routeur de New-York est de 32 MSS ;

Exercice 6 (août 2006) (3)

- le seuil initial du routeur de Seattle est de 16 MSS ;
- si le routeur de Seattle ne reçoit pas d'acquit dans la seconde qui suit l'envoi d'un paquet, il considère celui-ci comme perdu.

En prenant en compte l'overhead introduit par TCP et IP, on demande :

- 1 Quel est le chemin emprunté entre Seattle et New-York ? Justifiez.
- 2 Quel temps faudra-t-il au routeur de Seattle pour atteindre son débit maximal dans le meilleur des cas ?
- 3 Quelle est, au débit maximal, l'efficacité des liens utilisés par le transfert des données de Seattle à New-York ?
- 4 10 min après l'établissement de la session entre Seattle et New-York, le lien entre Chicago et New-York tombe en panne. Le RTT de la session TCP changera-t-il ? Si oui, quelle sera sa nouvelle valeur ?

Résolution exercice 6

Chemin choisi et RTT

Le chemin choisi est Seattle – Denver – Kansas City – Indianapolis – Chicago – New-York. En effet, c'est celui dont la somme des poids qui le composent est minimum (les autres chemins entre Seattle et New-York sont composés de plus de liens et donc leur coût est plus élevé).

$$\begin{aligned}RTT &= T_{MSS} + T_{ACK} + 2 \cdot T_{prop} + 2 \cdot T_{tr} \\ &= 0 + 0 + 2 \cdot T_{prop} + 0 \\ &= 2 \cdot (5 \cdot 5 \cdot 10^{-3}) = 50\text{ms}\end{aligned}$$

Résolution exercice 6 (2)

Taille maximale

$$\frac{rwnd \cdot (MSS + \text{en-têtes})}{RTT} \stackrel{?}{\leq} 10 \cdot 10^9$$

$$\frac{32 \cdot (1460 \cdot 8 + 40 \cdot 8)}{50 \cdot 10^{-3}} \stackrel{?}{\leq} 10 \cdot 10^9$$

$$7,68 \cdot 10^6 \leq 10 \cdot 10^9$$

D'où $cwnd_{max} = rwnd = 32$.

Résolution exercice 6 (3)

Slow Start

temps	reçu	taille fenêtre	émis
0	–	1	1 MSS
1 RTT	1 ACK	+1 → 2	2 MSS
2 RTT	2 ACK	+2 → 4	4 MSS
3 RTT	4 ACK	+4 → 8	8 MSS
4 RTT	8 ACK	+8 → 16	16 MSS
5 RTT	16 ACK	+1 → 17	17 MSS
...
20 RTT	31 ACK	+1 → 32	32 MSS

D'où le temps recherché est égal à $t = 20 \cdot 50 \cdot 10^{-3} = 1\text{s}$.

Résolution exercice 6 (4)

Efficacité maximale et nouveau RTT

$$D_{max} = \frac{cwnd_{max} \cdot MSS}{RTT} = \frac{32 \cdot 1460 \cdot 8}{50 \cdot 10^{-3}} = 7475200 \text{bps}$$

$$\eta_{max} = \frac{1}{2} \cdot \frac{7475200}{10 \cdot 10^9} = 0,037\%$$

Si le lien entre Chicago et New-York tombe en panne, le chemin utilisé par la connexion TCP n'est plus valide et doit être changé \Rightarrow le RTT va changer. Comme il existe maintenant plusieurs plus courts chemins, choisissons-en un de façon arbitraire. Soit Seattle – Denver – Kansas City – Indianapolis – Atlanta – Washington – New-York le nouveau chemin emprunté par la connexion TCP. On a :

$$\begin{aligned} RTT_{new} &= T_{MSS} + T_{ACK} + 2 \cdot T_{prop} + 2 \cdot T_{tr} \\ &= 0 + 0 + 2 \cdot T_{prop} + 0 \\ &= 2 \cdot (6 \cdot 5 \cdot 10^{-3}) = 60 \text{ms} \end{aligned}$$

Table des matières

- 1 Algorithme Slow Start
 - Principes de fonctionnement
 - Vitesse maximale d'une connexion TCP
 - Exercices

- 2 Modélisation du délai de TCP
 - Rappels théoriques
 - Exercices

Objectifs

- Calculer le temps qu'il faut à TCP pour envoyer un objet (une image, un fichier texte, un MP3).
- Pour un objet donné, on définit la **latence** comme le temps entre le moment où le client initie une connexion TCP et le moment où le client reçoit l'objet demandé dans sa totalité.
- Nous allons envisager trois cas :
 - ① La fenêtre est de taille fixée (émission en continu).
 - ② La fenêtre est de taille fixée (émission en rafales).
 - ③ La taille de la fenêtre est dynamique (utilisation du Slow Start).
- Les modèles ci-dessous mettent en évidence les éléments clés de la latence : le TCP handshaking initial, le temps de transmission de l'objet et le Slow Start.

Hypothèses

- Pas de partage de la bande passante avec d'autres flux
- Réseau composé d'un seul lien d'une capacité de R bps reliant le client et le serveur
- La fenêtre du récepteur est toujours plus grande que la fenêtre de congestion
- Pas de retransmission
- Les en-têtes sont négligeables
- Soient O la taille de l'objet à transmettre et S la taille du MSS (ces deux valeurs sont en **bits**). On a $O = X \cdot S$ avec $X \in \mathcal{N}$.
- Les temps de transmission sont négligeables sauf pour les paquets transportant des segments de taille maximale
- Dans le cas du Slow Start, on suppose que le contrôle de congestion est toujours dans sa phase de croissance exponentielle (le seuil de l'algorithme n'est jamais atteint)

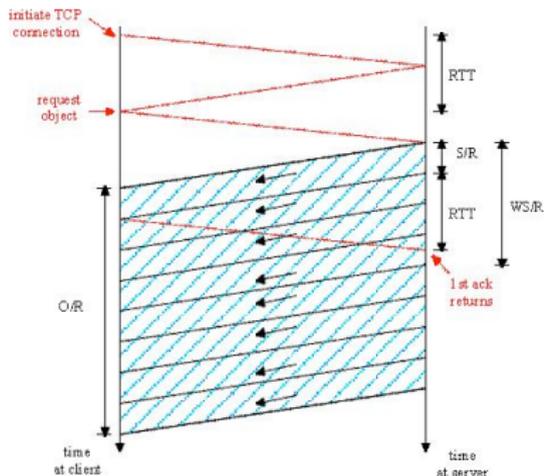
RTT

Dans la suite de cette section, on reprend la définition du RTT telle que donnée dans le cours théorique (cf. transparent intitulé «Définition du RTT»).

Formules pour calculer la latence

Taille de fenêtre fixée - Premier cas

- Soit $W > 0$ la taille de la fenêtre de congestion. On a $W \in \mathcal{N}$.
- $WS/R \geq RTT + S/R$: l'ACK du premier paquet est reçu avant que la fenêtre soit remplie \Rightarrow contrôle de congestion inactif et latence minimale.

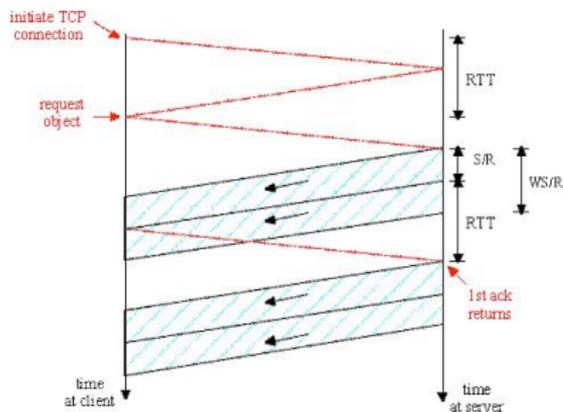


$$\text{latence} = 2RTT + \frac{O}{R}$$

Formules pour calculer la latence (2)

Taille de fenêtre fixée - Deuxième cas

- $WS/R < RTT + S/R$: la fenêtre est remplie avant l'arrivée de l'ACK du premier paquet \Rightarrow le serveur doit s'arrêter.
- Soit $K = \lceil \frac{O}{WS} \rceil$, le nombre de fenêtres nécessaires pour transmettre l'objet $\Rightarrow K - 1$ arrêts du serveur.



$$\text{latence} = 2RTT + \frac{O}{R} + (K - 1) \left(\frac{S}{R} + RTT - \frac{WS}{R} \right)$$

Formules pour calculer la latence (3)

Taille de fenêtre fixée - Formule générale

- Dans le cas où la fenêtre est de taille fixe, on a donc

$$\text{latence} = 2RTT + \frac{O}{R} + (K - 1) \left[\frac{S}{R} + RTT - \frac{WS}{R} \right]^+$$

- avec $[x]^+ = \max(x, 0)$.

Formules pour calculer la latence (4)

Fenêtre dynamique (Slow Start)

- K est le nombre de fenêtres nécessaires pour transmettre l'objet (attention, la taille de la fenêtre augmente maintenant de manière exponentielle).
- Soit Q le nombre de fois que le serveur s'arrêterait si l'objet était de taille infinie (dans ce cas, la fenêtre finira par atteindre une taille telle que le serveur n'aura plus à s'arrêter).
- Nombre d'arrêts P du serveur lors d'une transmission TCP :
 - Soit l'objet est de taille suffisante pour que la fenêtre puisse atteindre une taille assez grande pour que le serveur ne doive plus s'arrêter. Dans ce cas, $P = Q$ (avec $Q < K - 1$).
 - Sinon, $P = K - 1$ (avec $K - 1 < Q$).
- Donc $P = \min(Q, K - 1)$.

Formules pour calculer la latence (5)

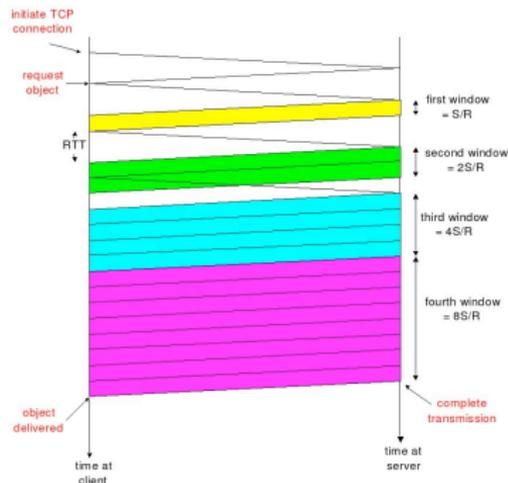
Fenêtre dynamique (Slow Start)

- Nous allons montrer que :

$$\text{latence} = 2RTT + \frac{O}{R} + P \left[RTT + \frac{S}{R} \right] - (2^P - 1) \frac{S}{R}$$

Composantes de la latence :

- $2RTT$ pour établir la connexion,
- O/R pour transmettre l'objet et
- les temps d'arrêt du serveur



Formules pour calculer la latence (6)

Fenêtre dynamique (Slow Start)

- Temps entre le début de l'envoi d'un segment et la réception de l'ACK correspondant : $\frac{S}{R} + RTT$.
- Temps pour transmettre la fenêtre k : $2^{k-1} \frac{S}{R}$.
- Temps d'arrêt après la fenêtre k : $[\frac{S}{R} + RTT - 2^{k-1} \frac{S}{R}]^+$.

$$\begin{aligned}
 \text{latence} &= 2RTT + \frac{O}{R} + \sum_{k=1}^P \text{temps d'arrêt}_k \\
 &= 2RTT + \frac{O}{R} + \sum_{k=1}^P \left[\frac{S}{R} + RTT - 2^{k-1} \frac{S}{R} \right] \\
 &= 2RTT + \frac{O}{R} + P \left[RTT + \frac{S}{R} \right] - (2^P - 1) \frac{S}{R}
 \end{aligned}$$

Formules pour calculer la latence (7)

Fenêtre dynamique - Calcul de K

- K est le nombre de fenêtres nécessaires pour transmettre l'objet.

$$\begin{aligned}
 K &= \min \{ k : 2^0 S + 2^1 S + \dots + 2^{k-1} S \geq O \} \\
 &= \min \left\{ k : 2^0 + 2^1 + \dots + 2^{k-1} \geq \frac{O}{S} \right\} \\
 &= \min \left\{ k : 2^k - 1 \geq \frac{O}{S} \right\} \\
 &= \min \left\{ k : k \geq \log_2 \left(\frac{O}{S} + 1 \right) \right\} \\
 &= \left\lceil \log_2 \left(\frac{O}{S} + 1 \right) \right\rceil
 \end{aligned}$$

Formules pour calculer la latence (8)

Fenêtre dynamique - Calcul de Q

- Q est le nombre de fois que le serveur s'arrêterait si l'objet était de taille infinie.

$$\begin{aligned}
 Q &= \max \{ q : \text{le } q^{\text{ième}} \text{ temps d'arrêt} \geq 0 \} \\
 &= \max \left\{ q : 2^{q-1} \leq \frac{\frac{S}{R} + RTT}{\frac{S}{R}} \right\} \\
 &= \left\lfloor \log_2 \left(1 + \frac{RTT}{S/R} \right) \right\rfloor + 1
 \end{aligned}$$

Formules pour calculer la latence (9)

Fenêtre dynamique - Remarque

On a :

$$\frac{\text{latence}}{\text{latence minimale}} \leq 1 + \frac{P}{[(O/R)/RTT] + 2}$$

Donc, si $RTT \ll \frac{O}{R}$ (i.e., si le RTT est beaucoup plus petit que le temps de transmission de l'objet), le Slow Start n'augmentera pas de façon significative la latence.

Table des matières

- 1 Algorithme Slow Start
 - Principes de fonctionnement
 - Vitesse maximale d'une connexion TCP
 - Exercices

- 2 Modélisation du délai de TCP
 - Rappels théoriques
 - Exercices

Exercice 7

Kurose, 3rd ed., problème 34

Considérons l'envoi d'un objet de taille $O = 100000$ bytes d'un serveur vers un client. Soient $S = 536$ bytes et $RTT = 100$ ms. Supposons que le protocole de transport utilise une fenêtre de congestion de taille fixe W .

- 1 Donnez la latence minimale pour un taux de transmission R de 28 kbps. Déterminez également la taille de fenêtre minimale qui aboutit à cette latence.
- 2 Répétez l'étape 1 pour $R = 100$ kbps.
- 3 Répétez l'étape 1 pour $R = 1$ Mbps.
- 4 Répétez l'étape 1 pour $R = 10$ Mbps.

Résolution exercice 7

- La latence minimale est $2RTT + \frac{O}{R}$
- Pour que la latence soit minimale, il faut que $(K - 1) \left[\frac{S}{R} + RTT - \frac{WS}{R} \right]^+ = 0$. D'où, le W_{min} recherché sera égal à $\min(W_1, W_2)$ avec

$$\begin{aligned}
 W_1 &= \min_W \left\{ \frac{S}{R} + RTT - \frac{WS}{R} \leq 0 \right\} \\
 &= \min_W \left\{ \frac{S}{R} + RTT \leq \frac{WS}{R} \right\} \\
 &= \min_W \left\{ W \geq 1 + \frac{RTT}{S/R} \right\} \\
 &= 1 + \left\lceil \frac{RTT}{S/R} \right\rceil \\
 W_2 &= \left\lceil \frac{O}{S} \right\rceil
 \end{aligned}$$

Résolution exercice 7 (2)

On a $W_2 = 187$. Dès lors, on a le tableau suivant :

R	latence minimale	W_1	W_{min}
28 Kbps	28,77s	2	2
100 Kbps	8,2s	4	4
1 Mbps	1s	25	25
10 Mbps	0,28s	235	187

Exercice 8

Kurose, 3rd ed., problème 36

On considère un scénario dans lequel $RTT = 1$ s, $O = 100$ Kbytes et $S = 536$ bytes. Au moyen d'un tableau, comparez la latence minimale à la latence obtenue avec le Slow Start dans les cas où le débit R est égal à 28 Kbps, 100 Kbps, 1 Mbps et 10 Mbps.

Résolution exercice 8

On a $K = \lceil \log_2 \left(\frac{O}{S} + 1 \right) \rceil = 8$ qui ne dépend pas de R et qui restera donc constant.

On obtient les valeurs suivantes pour $Q = \left\lfloor \log_2 \left(1 + \frac{RTT}{S/R} \right) \right\rfloor + 1$ et $P = \min(Q, K - 1)$:

R	Q	P
28 Kbps	3	3
100 Kbps	5	5
1 Mbps	8	7
10 Mbps	12	7

Résolution exercice 8 (2)

On peut alors calculer les quantités suivantes pour chaque valeur de R :

R	$\frac{S}{R}$	$P \left[RTT + \frac{S}{R} \right]$	$(2^P - 1) \frac{S}{R}$
28 Kbps	0,153	3,459	1,071
100 Kbps	0,043	5,215	1,333
1 Mbps	0,0043	7,0301	0,546
10 Mbps	0,0004	7,0028	0,051

Résolution exercice 8 (3)

On peut alors obtenir le tableau final :

R	O/R	P	Latence min.	Latence S. S.
28 Kbps	29,26s	3	31,26s	33,65s
100 Kbps	8,19s	5	10,19s	14,07s
1 Mbps	819 ms	7	2,82s	9,3s
10 Mbps	82 ms	7	2,08s	9,03s

On peut voir que quand $RTT \ll \frac{O}{R}$, la latence est proche de la latence minimale (l'effet du Slow Start est faible). Plus R augmente, plus $\frac{O}{R}$ diminue, donc plus le RTT se rapproche de $\frac{O}{R}$ et donc plus la latence diverge de la latence minimale.