

## TD 3 : CONTROLE DE CONGESTION TCP

### 1. COMPORTEMENT DYNAMIQUE DE TCP

#### 1.1 Rappels

Dans la licence vous avez appris que TCP est utilisé pour le transport fiable de données dans l'Internet. Vous avez appris comment est établi et comment est libéré une connexion TCP, et comment TCP effectue des retransmissions afin d'assurer la fiabilité. Dans cette exercice vous allez étudier un autre comportement fondamental de TCP : la contrôle de congestion.

TCP date des années 70. Des algorithmes de contrôle de congestion ont été ajoutés dans ce protocole à la suite des recherches réalisées par Van Jacobsen et publiée en 1988. Ces algorithmes continuent à évoluer, avec différentes variantes de TCP (TCP Tahoe, TCP Reno, etc.) Cette exercice est fondée sur la dernière mise à jour de la norme proposée à l'IETF, le RFC 2581 (avril 1999).

#### 1.2 Exercices

- Pour TCP, quel phénomène indique une congestion dans le réseau ? Comment précisément peut-on constater cette congestion ?
- Que se passe t-il dans le routeur pour susciter ce phénomène ?
- Pour TCP ce phénomène donne suite à l'inférence de la congestion. Mais ce phénomène peut se produire même quand il n'y a pas de congestion dans le réseau. Dans quels autres cas un tel phénomène peut-il se produire ?
- Si ce phénomène n'indique pas toujours une congestion, pourquoi est-ce que l'IETF base la norme TCP là-dessus ? Pourquoi est-ce que l'IETF n'a pas défini une norme dans laquelle un routeur constate lui-même un état de congestion et envoi un message explicite à l'émetteur ?
- Pour le contrôle de congestion, TCP utilise un seuil qui indique le débit au dessus duquel on risque de rencontrer de la congestion. Ce seuil est exprimé par un paramètre `ssthresh`, qui indique un nombre d'octets. Pour obtenir le débit seuil on divise `ssthresh` par la période aller retour entre l'émetteur et le récepteur (un « RTT » ou « round trip time » en anglais).

Le débit peut varier en dessous et au dessus du seuil `ssthresh` / RTT. L'émetteur maintient un paramètre `cwnd` qui indique le nombre d'octets qu'il peut envoyer dans le réseau avant de recevoir un acquittement. (Le nom `cwnd` est un raccourci pour « congestion window » en anglais, qui veut dire « fenêtre de congestion ».) Quand `cwnd` > `ssthresh`, l'émetteur fait particulièrement attention à ne pas provoquer de congestion.

Supposons que `ssthresh` est à 5000 octets, `cwnd` est à 6000 octets, et la taille d'un paquet est de 500 octets. Un émetteur envoi douze paquets de 500 octets dans une période RTT, et reçoit douze acquittements (un pour chaque paquet). Que devient les valeurs de `ssthresh` et `cwnd` ? Comment s'appellent ces changements de valeurs ?

- f) Supposons que `ssthresh` est toujours à 5000 octets, que `cwnd` est maintenant à 14.000 octets, que l'émetteur envoie  $14.000/500 = 28$  paquets, et que l'émetteur reçoit une indication de congestion avant de recevoir le premier acquittement. Que devient les valeurs de `ssthresh` et `cwnd` ? Comment s'appellent ces changements de valeurs ?
- g) Nous venons de voir comment on augmente et comment on diminue `cwnd` en fonction de l'absence ou la présence d'indicateurs de congestion. Comment s'appelle cet algorithme ? Quelle est l'idée derrière cet algorithme ?
- h) Au démarrage, et après avoir reçu une indication de congestion, la valeur de `cwnd` est plus petite que la valeur de `ssthresh`. Décrivez la manière permettant d'augmenter `cwnd` quand  $cwnd < ssthresh$ , en fonction de l'exemple suivant.

Supposons que `ssthresh` soit égal à 3000 octets et que `cwnd` soit égal à 500 octets, la taille d'un paquet. L'émetteur a plusieurs paquets prêts à être envoyés. Combien de paquets envoie l'émetteur pendant la première période RTT ? S'il reçoit des acquittements pour tous ses paquets, que devient la valeur de `cwnd` ? Combien de paquets envoie l'émetteur pendant la deuxième période RTT ? S'il reçoit des acquittements pour tous ses paquets, que devient la valeur de `cwnd` ? En générale, comment évolue la taille de `cwnd` ?

- i) Comment s'appelle la période pendant laquelle `cwnd` est plus petit que `ssthresh` ?
- j) Que devient la valeur de `ssthresh` si l'émetteur reçoit une indication de congestion pendant que `cwnd` est plus petit que `ssthresh` ?
- k) Supposez que nous souhaitons effectuer un gros transfert de données sur TCP. En négligeant la période pendant laquelle `cwnd` est plus petit que `ssthresh`, montrez que le débit moyen d'une connexion TCP est égal à

$$\bar{R} = \frac{3W.MSS}{4RTT}$$

où  $W$  est la taille de la fenêtre au moment de la perte de paquets, et  $RTT$  est le délai aller-retour (supposé constant durant la période de la transmission).

- l) Montrer que le taux de pertes est égal à :

$$p = \frac{1}{\frac{3}{8}w^2 + \frac{3}{4}w}$$

- m) Montrer que si le taux de pertes observé par une connexion TCP est  $p$  alors, son débit moyen peut être approximé par :

$$\bar{R} = \frac{1.22MSS}{RTT\sqrt{p}}$$

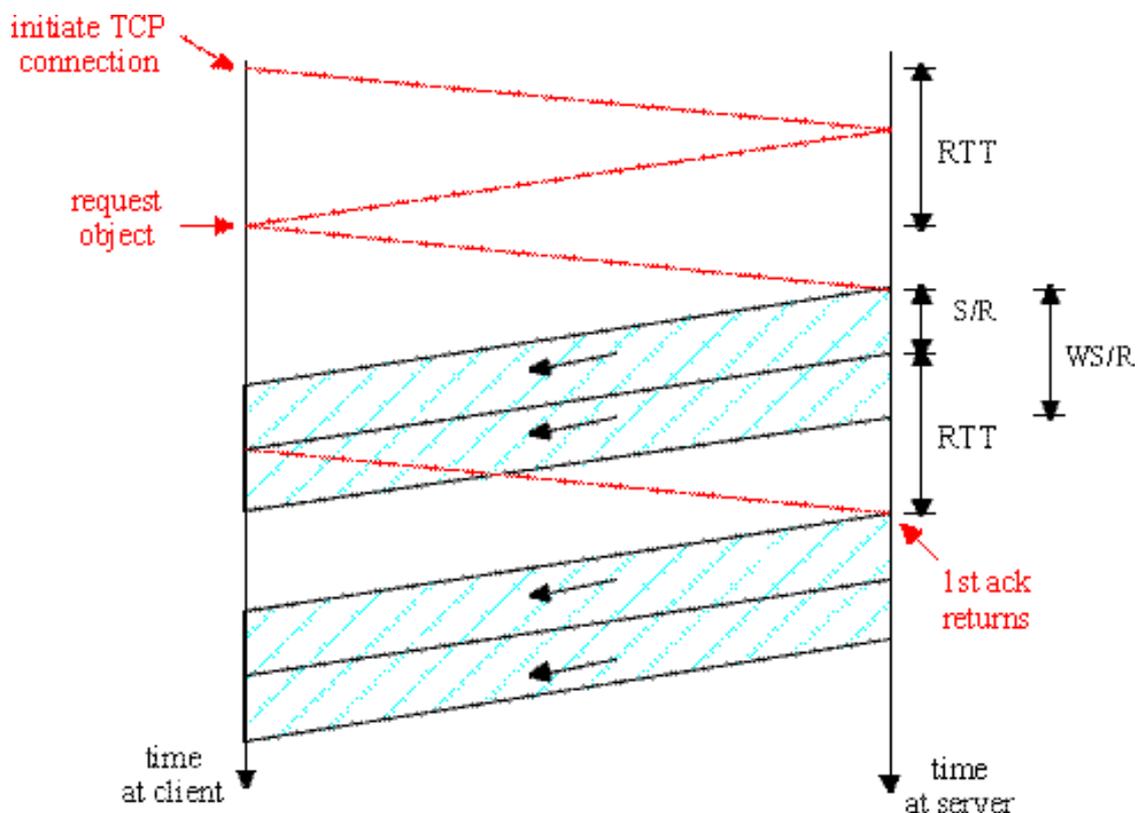
- n) Quels autres paramètres peuvent influencer sur le débit d'une connexion TCP ?
- o) Quelle utilité voyez-vous à la relation calculée dans le m ? Explicitez la notion de TCP amical (*TCP Friendly*).

## 2. ETUDE DE LA LATENCE D'UN SERVEUR WEB

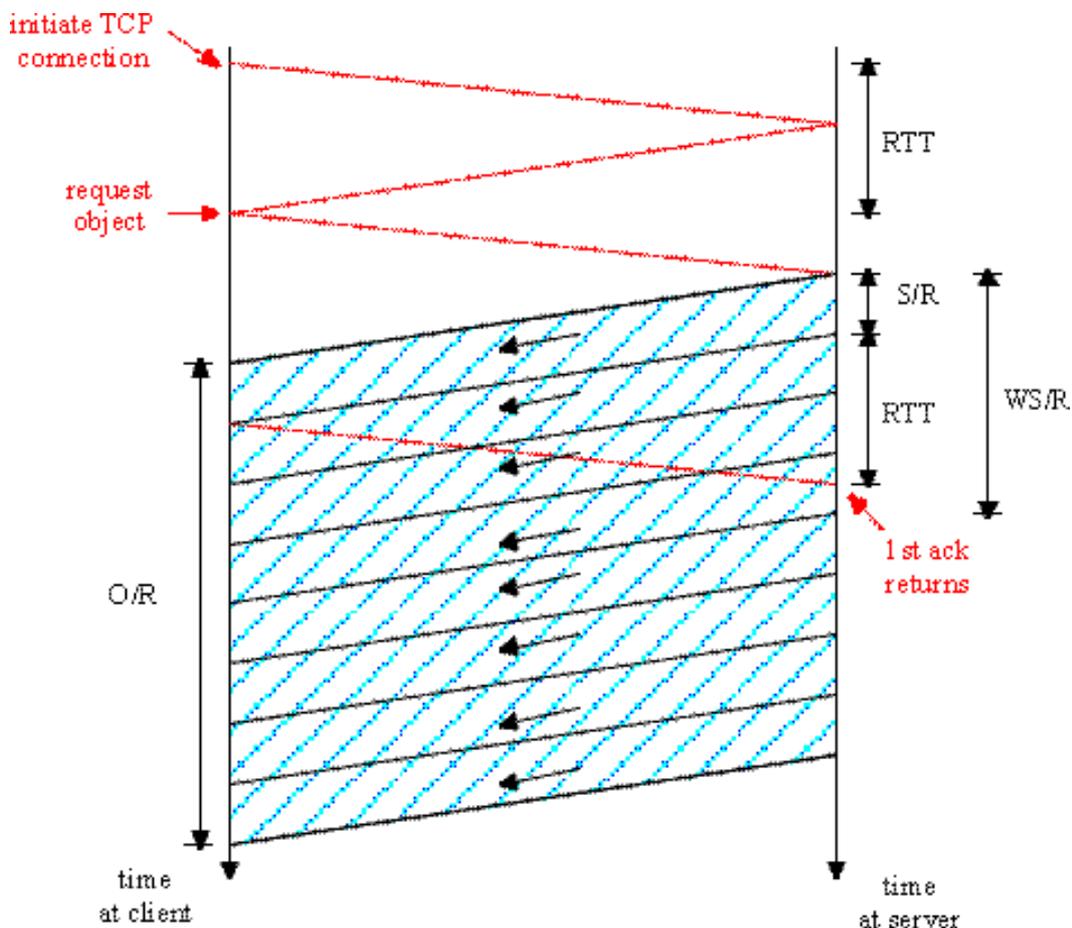
2.1 Dans cet exercice nous souhaitons étudier la latence d'un serveur web. La latence est définie par le temps nécessaire à pour répondre à une requête HTTP. Nous supposons ici que le réseau n'est pas congestionné, que la taille de la page à recevoir du serveur est  $O$  bits, le MSS est égal à  $S$  bits, le débit de la liaison connectant le client au serveur est  $R$  bps et que le délai aller-retour est  $RTT$ .

- Supposons que n'avons pas de fenêtre de contrôle de congestion. Dans ce cas montrer que la latence est égale à  $L = 2RTT + O/R$ .
- Nous supposons une fenêtre de congestion statique de taille fixe égale à  $W$ . Calculer la latence dans les deux cas suivants :

1-  $WS/R < RTT + S/R$



2-  $WS / R > RTT + S / R$



c) Comparer la latence obtenue avec une fenêtre de contrôle de congestion dynamique (avec slow-start) est la latence sans aucun contrôle de congestion.

d) Application numérique :  $S= 536$  octets,  $RTT= 100$  msec,  $O=100$  Koctets et  $O=5$  Koctets

R	O/R	P	Latence Min	Latence
28 kbps				
100 Kbps				
1 Mbps				
100 Mbps				

$P$  est le nombre d'octets envoyés avant de passer au cas 2. Latence Min est la latence s'il n'y aura pas de fenêtre de congestion.

