

Majeure Informatique - X03

Contrôle écrit : Réseaux, protocoles (INF 586)

Vendredi 31 mars 2006

Le contrôle dure 2h. Les documents sont autorisés.

Question 1 : Efficacité du protocole Stop and Wait (4 points)

Nous considérons deux stations A et B reliées par un lien de transmission ayant un délai de propagation fixe d secondes et une capacité C bits par seconde. Tous les paquets ont la même taille L bits. Les accusés de réception ont la même taille l bits. Nous nous proposons d'étudier l'efficacité d'un protocole de transmission fiable très simple : le protocole Stop and Wait. Selon ce protocole, la source émet les paquets un à un ; quand le récepteur reçoit correctement un paquet il renvoie un accusé de réception ; la source renvoie le paquet émis si elle ne reçoit pas un accusé de réception avant un "time out" de T secondes.

L'efficacité d'un protocole de transmission fiable est définie comme étant le rapport entre le débit effectif moyen obtenu avec ce protocole et le débit maximal théorique (égal à la capacité du lien). Pour calculer l'efficacité du protocole Stop and Wait, considérons le temps X entre la première transmission d'un paquet numéro n et celle du paquet de numéro $n + 1$. Soit p la probabilité qu'une transmission soit réussie (paquet et accusé de réception bien reçus).

1. Calculer le temps S entre l'émission d'un paquet et la réception de l'accusé de réception dans le cas d'une transmission réussie.
2. Calculer la valeur moyenne $E[X]$ en fonction de S , p et T .
3. En déduire une expression de l'efficacité du SWP.

Question 2 : Routage CIDR (3 points)

Soient P, Q et R trois fournisseurs de services réseaux avec les attributions d'adresses CIDR $C1.0.0.0/8$, $C2.0.0.0/8$ et $C3.0.0.0/8$. La notation "/8" dans $C1.0.0.0/8$ signifie qu'on a un masque de réseau de 8 bits à 1 : $FF.0.0.0$. Chaque client du fournisseur reçoit initialement des adresses qui sont un sous-ensemble du fournisseur.

P a les clients suivants : PA avec les adresses $C1.A3.0.0/16$ et PB avec les adresses $C1.B0.0.0/12$. Q a les clients suivants : QA avec les adresses $C2.0A.10.0/20$ et QB avec les adresses $C2.0B.0.0/16$. Nous faisons l'hypothèse qu'il n'y a pas d'autres fournisseurs ni d'autres clients.

a) Donnez les tables de routage pour P, Q, R en faisant l'hypothèse que chaque fournisseur se connecte aux deux autres fournisseurs.

b) Faites l'hypothèse maintenant que P est connecté à Q et que Q est connecté à R, mais que P et R ne sont pas directement connectés. Donnez les tables de routage pour P et R.

c) Supposons que le client PA acquière une ligne directe à Q et que le client QA acquière une ligne directe à P, en plus des lignes existantes. Donnez les tables pour P et Q, en ignorant R.

Question 3 : Analyse du Protocole TCP sur un lien non fiable(5 points)

Cet exercice vise à effectuer une analyse simple des performances de TCP sur un lien non fiable. Il est inspiré d'une analyse due à Towsley. La fenêtre de transmission est augmentée de 1 tous les RTTs et elle est divisée par 2 à chaque perte. On suppose que toute perte est détectée juste après l'envoi de la fenêtre entière. On ne se soucie pas de la perte des acquittements.

a. Considérons tout d'abord un lien qui perd un paquet tous les $Y = 1/p$. En supposant que la fenêtre revient à la même valeur w après chaque perte, estimer le débit moyen obtenu. On négligera 1 devant w .

b. Considérons maintenant un lien qui perd un paquet avec probabilité constante p très faible (on négligera p devant 1). Toutes les pertes de paquets sont indépendantes. Estimer le nombre moyen de paquets envoyés entre deux pertes. On rappelle que $\sum_{k=0}^{\infty} q^k k = q \left(\frac{1}{1-q} \right)' = \frac{q}{(1-q)^2}$.

Soit W_i la taille de la fenêtre juste après la i ème perte de paquet. Soit X_i le nombre d'envois de fenêtres jusqu'à la perte suivante. Estimer le nombre Y_i de paquets envoyés. En déduire une expression du débit D_i entre la i ème perte et la suivante. On négligera 1 devant X_i .

Exprimer W_{i+1} en fonction de W_i et X_i . On suppose qu'un régime stationnaire est atteint quand $i \rightarrow \infty$. En première approximation, on supposera W et X indépendantes. En déduire une relation sur leur moyennes. En négligeant de plus la variance de X (soit $E[X^2] = E[X]^2$), estimer finalement le débit moyen.

c. Quelle serait en théorie le débit maximal atteignable sur un lien de débit d avec taux de perte p ? Exprimer un taux de perte maximal garantissant un fonctionnement satisfaisant de TCP. Comment doit varier la fiabilité d'un lien en fonction de son débit ?

Question 4 : Contrôle de congestion TCP (5 points)

a) Le protocole TCP réduit la fenêtre de congestion `ccwnd` de façon exponentielle (`ccwnd` est divisée par deux) quand une perte est soupçonnée. Quel est l'impact de ce mécanisme sur les performances de TCP dans le cas où le réseau contient des liens haut débit hétérogènes (en terme de taux d'erreurs en particulier) ?

b) Le seuil `ssthresh` est par ailleurs initialisé à une valeur arbitraire (qui peut varier de quelques "Maximum Segment Size" à une valeur maximale `MAXSSSTHRESH` selon les implémentations). Quel est l'impact d'un mauvais choix de `ssthresh` sur les performances du protocole TCP dans le cas où le réseau contient les liens avec un produit (bande passante)(délai) important (large Bandwidth Delay Product) ?

c) La bande passante disponible varie à cause de plusieurs facteurs : les autres flux partageant le même goulot d'étranglement, l'utilisation du canal et la dynamique du protocole MAC, le handoff et les interférences dans le cas des réseaux mobiles. Expliquer comment l'ajustement de `ccwnd` et de `ssthresh` permet d'atténuer l'impact de ce problème.

d) Que risque-t-il de se passer si une large quantité de ressources (bande passante) se libère (par le départ d'utilisateurs partageant le goulot d'étranglement par exemple) ? Quelle est la raison de cette limitation ?

e) La version westwood de TCP (TCPW) a été proposée comme solution aux problèmes ci-dessus. Après une perte de paquet, au lieu de diviser `ccwnd` par deux comme le fait TCP Reno, TCPW met à jour `ccwnd` et `ssthresh` en se basant sur une estimation sur de la bande passante "disponible". Expliquer comment cette approche permet-elle d'améliorer les performances de TCP.

f) Une amélioration de TCPW propose que pendant le mode “congestion avoidance” `ssthresh` soit mise à jour en fonction de la bande passante disponible, avec une remontée exponentielle de `cwnd` jusqu’à la nouvelle valeur de `ssthresh`. Expliquer comment cette technique permet-elle d’améliorer encore plus les performances dans le cas des liens haut débit hétérogènes et avec un large BDP.

Question 5 : Transmission (3 points)

Un réseau linéaire est composé d’une source, d’une destination et de trois routeurs intermédiaires $S - R_1 - R_2 - R_3 - D$. Les liens entre S et R_1 , R_1 et R_2 , R_2 , R_3 et D ont une capacité $C_1 = 10$ Mbps et un délai de propagation de d sec. Le lien $R_2 - R_3$ a une capacité $C_2 = 1$ Mbps et un délai de propagation de d sec. Le routeur R_1 a un tampon mémoire de taille $B_1 > 0$ paquets. Le routeur R_2 a un tampon mémoire de taille $B_2 = K.C_1/C_2$ paquets ($1 < K < 10$). La source émet en permanence des paquets de taille L numérotés à partir du numéro 1. Le mode de transmission est le “Store and Forward” (i.e. un paquet est reçu en entier par un routeur avant de commencer de le transmettre sur une ligne de sortie. Par ailleurs, tant qu’un paquet n’est pas transmis en entier, l’espace mémoire associé ne peut être libéré). Aucun algorithme de contrôle de flux n’est appliqué. Quel est le numéro du premier paquet jeté ? Au niveau de quel routeur aura lieu cette perte ? (Il n’est pas nécessaire d’utiliser tous les paramètres cités ci-dessus).

Question 1: SWP

$$S = 2d + (L + l)/C$$

$E(X) = pS + (1 - p)[T + E(Y)]$ (Y étant le temps entre la deuxième transmission du paquet n et la première transmission du paquet $n + 1$). On a $E(Y) = E(X)$. Donc $E(X) = S + T \frac{1-p}{p}$.

$$\text{Efficacité} = \frac{1/E(X)}{C/L} = \frac{L}{CE(X)} = \frac{L}{C(S + T \frac{1-p}{p})}$$

Question 2: CIDR

Partie a)

Préfixe	Prochain Nœud
C2.0.0.0/8	Q
C3.0.0.0/8	R
C1.A3.0.0/16	PA
C1.B0.0.0/12	PB

Table 1: Table de P

Préfixe	Prochain Nœud
C1.0.0.0/8	P
C3.0.0.0/8	R
C2.0A.10.0/20	QA
C2.B0.0.0/16	QB

Table 2: Table de Q

Préfixe	Prochain Nœud
C1.0.0.0/8	P
C2.0.0.0/8	Q

Table 3: Table de R

Partie b)

Pour le routeur P (voir Table 4), on pourra agréger les deux premières entrées en un préfix plus court C2.0.0.0/7 (car le next hop est Q dans le deux cas). La table de Q ne change pas par rapport à la partie a (Table 2). Pour le routeur R (Table 5) pas d'agrégation de préfix possible.

Partie c)

A vous de voir...

Préfixe	Prochain Nœud
C2.0.0.0/8	Q
C3.0.0.0/8	Q
C1.A3.0.0/16	PA
C1.B0.0.0/12	PB

Table 4: Table de P

Préfixe	Prochain Nœud
C1.0.0.0/8	Q
C2.0.0.0/8	Q

Table 5: Table de R

Question 3: Débit TCP

a. Si la fenêtre revient une taille w après une perte, c'est qu'elle a atteint une taille $2w$ depuis la dernière perte. Le nombre de paquets émis entre les deux pertes est donc $Y = w + (w + 1) + \dots + 2w \approx 3w^2/2$. D'où $w \approx \sqrt{\frac{2}{3p}}$. Le débit est de $Y/w = \sqrt{\frac{3}{2p}}$ paquets par RTT.

b. La probabilité d'envoyer k paquets sans perte avant une perte est $(1 - p)^k p$. Le nombre moyen de paquets envoyés entre deux pertes est donc $\sum_{k=0}^{\infty} (1 - p)^k p k = \frac{1-p}{p} \approx \frac{1}{p}$.

De manière similaire au a., on a $Y_i = W_i + W_i + 1 + \dots + W_i + X_i - 1 = W_i X_i + X_i(X_i - 1)/2 \approx W_i X_i + X_i^2/2$. Comme $D_i = Y_i/X_i$, on trouve $D_i = W_i + X_i/2$.

$W_{i+1} = (W_i + X_i - 1)/2 \approx (W_i + X_i)/2$ d'où $E[W] = E[X]$. Donc $E[D] = 3E[X]/2$. En fait, l'indépendance de X et W implique la variance nulle car on peut écrire $Y_i = X_i(W_i + X_i/2) = X_i(W_{i+1} + W_i/2)$, d'où $E[Y] = E[X](3E[W]/2) = E[X](E[W] + E[X]/2)$ et donc $E[X^2] = E[X]^2$. Comme $1/p = E[Y] = 3E[X]^2/2$, on trouve $E[X] = \sqrt{\frac{2}{3p}}$, d'où $E[D] = \sqrt{\frac{3}{2p}}$. Au premier ordre d'approximation, ça se passe comme en a.

c. $(1 - p)d$. Exigeons donc un débit estimé de TCP de l'ordre de $(1 - p)d$, soit $\sqrt{\frac{3}{2p}} > d$ au premier ordre, et finalement $p < \frac{3}{2d^2}$. (d est exprimé en paquets par RTT.) La fiabilité du lien doit augmenter proportionnellement au carré du débit.

Question 4: TCPW

a) Dans un environnement hétérogène haut débit beaucoup de pertes peuvent être dues à des erreurs de transmission (bruit et interférence sur les liens sans fil) et non à une congestion et pourtant elles produiront une réduction de `cnwnd` et donc du débit TCP.

b) Si `ssthresh` est très élevé par rapport au produit bande passante délai du réseau, l'augmentation exponentielle de `cnwnd` génèrera beaucoup trop de paquets et résultera en des pertes multiples au goulot détrangement. Si `ssthresh` est choisi trop petit alors la connexion quittera le mode slow start et passera en mode augmentation linéaire (donc lente) "prématurément" ce qui donnera aussi des performances médiocres lors au début de la connexion surtout dans le cas d'un large BDP.

c) TCP s'adapte à la variation de la bande passante disponible en divisant `cnwnd` et `ssthresh` par deux lors d'une indication de congestion. L'augmentation de la fenêtre après en mode linéaire permet

de détecter (lentement) la libération de ressources et de s'adapter.

d) Comme la récupération se fait en augmentant `wnd` d'un MSS tous les RTTs, alors il faut mettre longtemps avant de récupérer. Ceci est dû à l'augmentation linéaire de `wnd`.

e) En ajustant `wnd` à une valeur qui correspond à ce que le réseau peut écouler, on évite l'approche conservatrice de TCP Reno (basée sur l'hypothèse qu'une congestion est due à un "autre" utilisateur qui vient partager les ressources et qui impose donc de diviser `wnd` par deux). Donc récupération plus rapide car la réponse est adaptée à la sévérité de la congestion.

f) En ajustant `ssthresh` en fonction de la bande passante disponible, on ajuste notre "agressivité" pour récupérer les ressources libérées en fonction de la quantité de ressources libérées. On grimpe donc `wnd` exponentiellement au nouveau `ssthresh` puis on passe en mode linéaire.

Question 5: Gestion de tampons

Après la réception complète du premier paquet dans R_2 , sa transmission vers R_3 commence. La transmission du premier paquet par R_2 se termine juste au moment de la fin de la réception du onzième paquet (Le paquet onze a pu être reçu par R_2 car $K > 1$). La transmission du deuxième paquet par R_2 se termine à la réception du 21ième paquet. A la fin de la réception du B_2 ième paquet par R_2 , le paquet numéro K est en cours d'émission. Son émission se termine juste après la réception du B_2+1 ième paquet. Il reste alors $K - 1$ places dans le tampon (grâce aux places libérées par les paquets déjà transmis). On pourra donc stocker les paquets jusqu'à $B_2 + K$. C'est le $B_2 + K + 1$ ième paquet qui sera jeté car $K < 10$ (il n'y a donc pas le temps de transmettre un paquet complet).