TP2 Réseaux : TCP/IP

Question 1) options détecté sur les paquets TCP :

maximum segment size : Donne la taille maximum et optimum du paquet de donnée TCP a envoyé.

No operation

Window scale : Donne le multiplicateur du champ Window dans le paquet TCP. Cela est lié a l’acroissement de la taille de la mémoire par rapport à quand la norme a été créé. Les buffers étant beaucoup plus grand maintenant, le nombre d’octets disponible ne rentrerait pas dans le champ Window. On multiplie donc cette valeur par cette option.

TCP sack permitted : Permet d’autoriser ou non de ne pas acquitter certain paquet TCP (d’où l’option “sack”). Cela se fait apparemment sur des paquets triviaux.

Timestamp : Donne le temps processeur au moment de l’envoi du paquet et le stock dans ce champ. En le comparant avec le temps processeur au moment de sa réception, cela permet de calculer avec précision le RTT entre les deux machines

Question 2)

Question 3) retarder les acquittements permet d’acquitter plusieurs paquets d’un coup en fonction de la quantité de paquets arrivées. Plus le retard est grand plus on acquitte de paquets d’un coup. Mais un délai trop important risque de faire croire à l’autre machine que l’on n’a pas reçu le paquet et elle le renverra, il faut donc que le timer d’acquittement de réémission soit synchronisé

Question 4) Le timer de retransmission évolue en fonction du RTT en faisant la moyenne des RTT depuis le début de la connexion (environs.) Cela permet donc de s’ajuster en fonction de la connexion. Par exemple, si le RTT commence à s’accroitre, le timer aussi (à moindre rythme cependant puisque c’est une moyenne), en particulier à chaque réémission. Enfin, cette mécanique évite d’avoir un timer fixe qui ne prend pas en compte le délai, et qui, par exemple, pour un réseau très distant (ou instable), passerait son temps à réémettre ce qui polluerai le réseau.

Question 5)

On voit que l’expéditeur envois par rafale, attend les acquittements du destinataire, et reprend. La rafale semble dépendre de la taille du buffer d’émission puisque le nombre d’octet envoyé avant attente semble être équivalent à la taille du buffer.

Cette rafale correspond à la taille de la fenêtre d’anticipation, qui est donc la fenêtre d’octet dans laquelle l’expéditeur peut, en cas d’erreur, réémettre immédiatement la série d’octet perdu ou corrompu. C’est donc la fenêtre de récupération d’erreur pour ainsi résumé. L’expéditeur ne remplacera pas son buffer tant que tous les octets envoyés n’ont pas été acquittés.

Enfin, plus le buffer d’émission est important, plus cette courbe est raide (en ne prenant pas en compte celui de réception au vue de sa taille dans l’exercice). Cela signifie plus simplement que plus le buffer d’émission est gros, plus l’émetteur peut envoyer avant de devoir s’arrêter pour attendre acquittement et remplacer son buffer. Donc si le buffer de réception est suffisamment gros, on pourrait en théorie expédié l’entièreté du message en une seul rafale, augmentant donc énormément notre débit (toujours selon le débit physique du câble bien sûr). De plus, des rafales plus importantes signifie moins d’échange d’acquittements et donc moins d’aller-retour sur le réseau (intéressant pour des réseaux avec une forte latence).

Question 6)

Un problème est survenu lors des expériences avec un buffer plus petit que la taille du MSS, et donne lieu à des intervalles d’émissions illogique fixe de 5 secondes entre chaque émission, peu importe qu’il y ai eu un acquittement avant ou pas. M.Sicard a suggéré une erreur lors de la release de la récente version de TCP. Nos courbes et réponse ne sont donc pas du tout corrélable au TP.

Nous pouvons cependant en déduire qu’une taille de buffer plus petite force plus d’envois et engendre donc des courbes beaucoup plus espacé que un buffer important.

Question 7)