TD 10

PROTOCOLES DE TRANSPORT (PARTIE 2)

1. CONTROLE D'ERREUR AVEC TCP

Afin d'offrir un service de remise fiable à ses utilisateurs, TCP met en œuvre des mécanismes de détection d'incidents (erreurs, pertes, duplications, déséquencements) et de reprise sur ces incidents.

Exercice 1.1 | Généralités

1. Dans l'en-tête TCP, quels sont les champs qui participent au contrôle d'erreur ?
2. Sont-ils suffisants ? Si oui, pourquoi, et sinon, que manque-t-il ?

Exercice 1.2 | Espace de numérotation

Combien de temps met le numéro de séquence pour reboucler complètement :

1. sur une connexion TCP disposant (en 1998) d'un débit de 56 kbit/s ?
2. sur une connexion TCP disposant (de nos jours) d'un débit de 1 Gbit/s ?
3. Que penser des valeurs obtenues ?
4. En considérant toujours un débit de 1 Gbit/s, on utilise dans le champ d'options de TCP une estampille temporelle, codée sur 32 bits, qui s'incrémente 1000 fois pendant le temps trouvé précédemment. Combien de temps met l'estampille pour reboucler ?

Exercice 1.3 | MSS

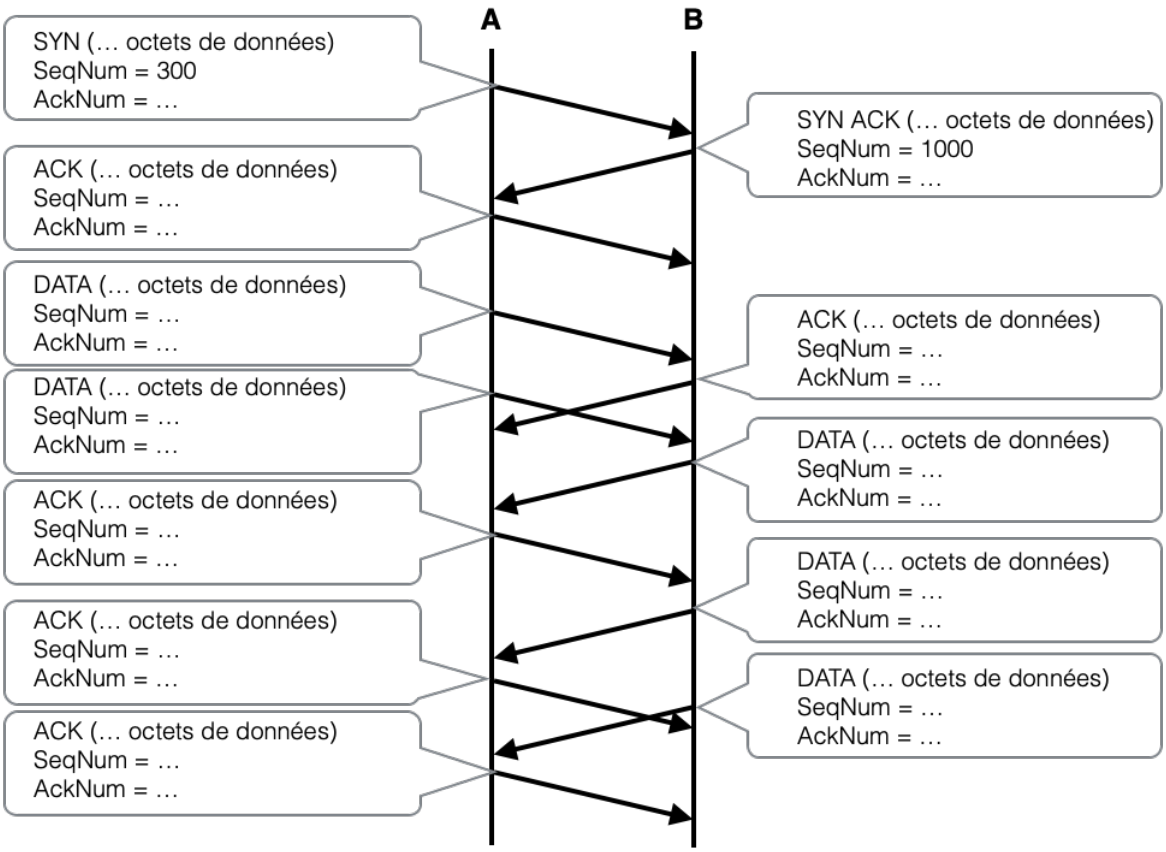
TCP est orienté-octet : la numérotation et la fenêtre portent sur des octets.

1. Quand est-ce que TCP décide de constituer et d'envoyer un segment ?

Exercice 1.4 | Numérotation des données et des acquittements

A et B doivent s'envoyer mutuellement 2000 et 3000 octets de données, sur une connexion TCP. On considère que le MSS a une valeur de 1460 octets et que tous les segments envoyés ont un champ de données de taille maximum.

1. Compléter sur le schéma suivant la taille du champ de données ainsi que les numéros de séquence (SeqNum) et d'acquittement (AckNum) manquants.



Exercice 1.5 | Retransmission des données perdues

Dans l'échange précédent, on considère que le 2ème segment de données envoyé par B a été perdu.

1. Reprendre l'échange en indiquant comment la reprise sur erreur s'effectue.

Exercice 1.6 | Dimensionnement du temporisateur de retransmission

1. Pourquoi la valeur du temporisateur de retransmission RTO (*Retransmission Time-Out*) doit-elle être ajustée dynamiquement et ce, pour chaque connexion TCP ? Comment peut-on l'estimer ?
2. Comment mesurer le délai aller-retour RTT (*Round Trip Time*) ?

TCP calcule une valeur lissée du RRT appelée le SRTT (*Smoothed Round Trip Time*) en appliquant la formule suivante après chaque nouvelle mesure du RTT :

$$SRTT = \alpha * SRTT + (1-\alpha) * RTT$$

où α est un facteur de pondération typiquement compris entre 0,8 et 0,9.

La valeur du RTO est alors estimée par :

$$RTO = \beta * SRTT$$

où β est un facteur lié à la variance du délai, généralement compris entre 1,3 et 2.

3. La valeur courante de SRTT est de 30 ms. Les acquittements des trois segments de données suivants reviennent respectivement après 26 ms, 32 ms et 24 ms. Que deviennent l'estimation du délai aller-retour (SRTT) et la durée de la temporisation de retransmission (RTO) ?

   On prendra $\alpha = 0.9$ et $\beta = 2$.

2. CONTROLE DE FLUX AVEC TCP

Afin d'assurer qu'un émetteur n'envoie pas plus de données que ce qu'est capable de mémoriser le récepteur, TCP utilise un mécanisme de fenêtre coulissante de taille variable.

Exercice 2.1 | Fenêtre d'anticipation

Une machine A envoie un segment TCP avec les valeurs suivantes : SeqNum = 1227, AckNum = 2513 et Window = 3007.

1. Quelle plage de numéros d'octets est-elle prête à recevoir ?
2. Qu'en est-il si l'option TCP *Window Scale* (option 3) est activée avec la valeur 1 ?

Exercice 2.2 | Contrôle de flux

On considère l'échange suivant dans lequel la machine A doit envoyer 1500 octets de données à la machine B, mais B a auparavant annoncé à A une fenêtre de taille 1000. A envoie donc seulement 1000 octets qui viennent remplir le buffer de réception de B. Peu de temps après, l'application de B vient lire 800 octets dans le buffer.

1. Compléter l'échange.