Faculté des Sciences & Techniques — Université de Limoges

Master 1ère année

Protocoles & Prog. Réseau

TD n°4

TCP vs UDP

TCP

1 – Est-ce que la fragmentation et le ré-assemblage des datagrammes IP concernent TCP ?
Est-ce que cela veut dire que TCP n'a pas à se préoccuper de l'ordre d'arrivée des données ?

2 – La connexion TCP :

a. Pourquoi le schéma d'établissement d'une connexion TCP correspond à trois échanges ?
b. Pourquoi le protocole TCP structure les échanges de données en segment alors qu'il rend un service de flux d'octets ?
c. Pourquoi ne pas faire commencer l'ISN, « *Initial Sequence Number* » à 0 ?

3 – Les numéros de séquence du protocole TCP font référence au nombre d'octets transmis et non aux numéros des paquets incrémentés de 1 pour chaque paquet envoyé.

a. pourquoi utilise-t-on ce type de notation pour définir les numéros de séquence ?

b. On suppose que l'on utilise TCP sur un lien de débit 1Gbps et que l'émetteur n'est jamais bloqué par la fenêtre d'émission (servant à empêcher la congestion).
Combien faut-il de temps pour utiliser l'ensemble complet des numéros de séquence ?

c. Les règles suivantes s'appliquent sur une connexion TCP :
   - ◇ Il n'est pas autorisé d'avoir des paquets dans une même connexion ayant le même numéro de séquence qui transitent simultanément sur le réseau.
   - ◇ La durée de vie d'un paquet est prise en compte pour le calcul du débit.

   Illustration de ces règles dans un protocole ressemblant à TCP :
   - ◇ les numéros de paquets sont incrémentés de 1 pour chaque paquet ;
   - ◇ la taille maximum des paquets est de 128 ko :
   - ◇ la durée de vie maximum d'un paquet dans le réseau est de 30 secondes ;
   - ◇ les numéros de séquences sont codés sur 8 bits.

   Quel est le débit maximum que l'on peut atteindre durant une connexion ?

UDP

4 – On veut utiliser le protocole UDP dans le modèle « Client/Serveur ».

a. Quel est la nature des échanges entre le client et le serveur ?
b. Comment ces échanges sont-ils identifiés sur le client et le serveur ?
c. Comment le serveur peut-il être trouvé par le client et comment peut-il répondre au client ?

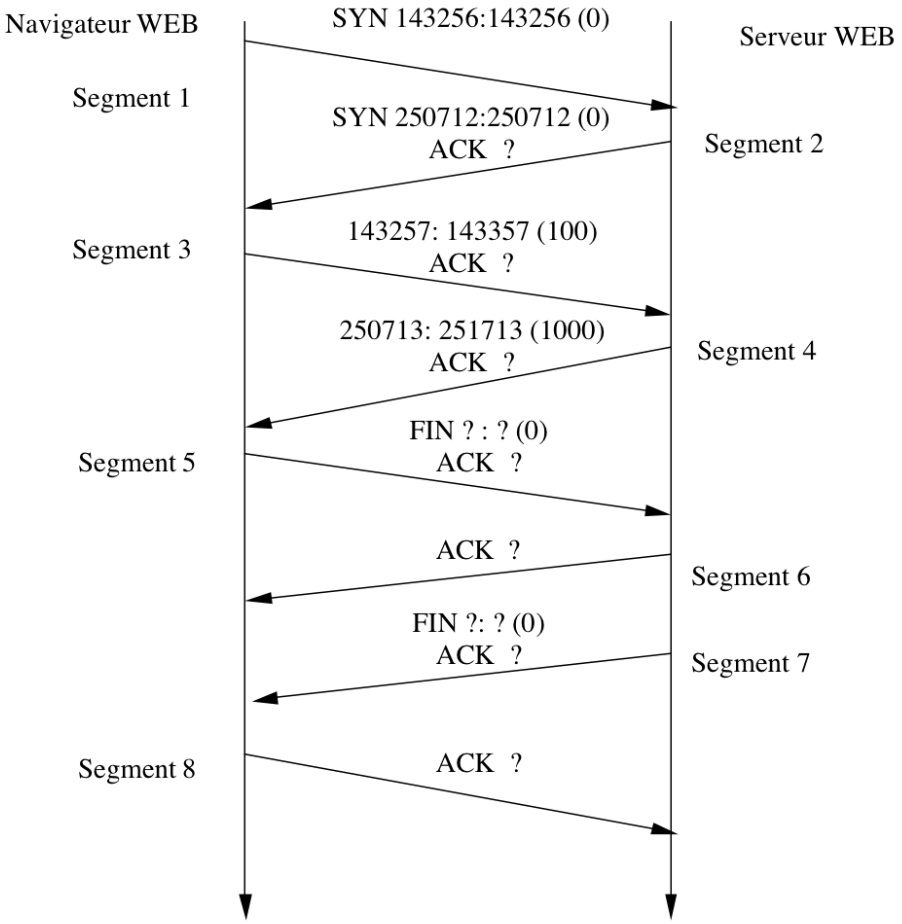
Dans le cas d'un protocole « Question/Réponse », où une question dans un paquet UDP est envoyée du client vers le serveur, et une réponse est envoyée du serveur vers le client dans un seul paquet UDP :

d. Est-ce que toutes les réponses arrivent dans le même ordre que les questions ont été envoyées ?
e. Comment ne pas mélanger les réponses ?
f. Comment tenir compte des questions et réponses perdues ?
g. Si l'on veut que le serveur gère simultanément plusieurs clients est-ce que cela fonctionne toujours ?

Vous illustrerez vos réponses de MSCs, « Message Sequence Charts ».

5 – L'échange TCP de la figure suivante correspond au transfert d'une page WEB entre un navigateur Web et un serveur Web.
On fait les hypothèses suivantes :

- ▷ que la requête à la page Web fait 100 octet ;
- ▷ que la page Web retournée fait 1000 octets.
- ▷ qu'il n'y a pas d'erreurs de transmission.



Pour chaque segment de données, différentes informations apparaissent :

- ⋆ la présence d'un ou plusieurs des différents indicateurs comme `SYN`, `FIN`, `ACK`
- ⋆ sur la chaque ligne trois nombres sont indiqués :
  - ◇ le premier nombre correspond au numéro de séquence du premier octet du segment ;
  - ◇ le second nombre correspond au numéro du premier octet du prochain segment à envoyer.
  - ◇ le nombre entre parenthèses correspond au nombre total d'octets transmis dans le segment.

Si le segment est porteur d'un acquittement positif, l'indicateur ACK est mentionné et à coté de lui doit figurer la valeur du champ acquittement du segment TCP.

Questions :

a. Complétez les numéros de séquence et les numéros d'acquittement qui manquent sur la figure (qui apparaissent sous forme de point d'interrogation) dans la trace obtenue par l'outil `tcpdump`.

b. Indiquez à quoi correspondent les différents segments numérotés de 1 à 8 ;

c. Quelle semble être la taille de la fenêtre glissante TCP ?

6 – Une communication est décrite du point de vue de la couche 5, « application », en 6 étapes :

1. connexion du client sur le serveur ;
2. envoi de 80 octets du serveur vers le client ;
3. envoi de 320 octets du client vers le serveur ;
4. envoi de 50 octets du serveur vers le client ;
5. envoi de 2300 octets du serveur vers le client ;
6. déconnexion du client, suivie de celle du serveur.

Écrivez un MSC, « *Message Sequence Chart* », à la manière de la commande `tcpdump`, en détaillant les valeurs des différents champs des segments TCP échangés lors de cette communication, avec les paramètres TCP suivants :

	MSS (octets)	Window size (octets)
client	`1000`	`1200`
serveur	`1000`	`800`