

**TD2 : Internet Protocol - correction**

**Objectifs :**

- Comprendre l'utilisation de TCP par HTTP
- Comprendre TCP et ses limites

---

**Partie I : Communication Web**

Dans cette première partie, on considère C, un client, et S un serveur, reliés tous les deux au travers d'Internet. Le client demande une page web constituée d'un seul élément.

**Hypothèses :**

- |   |                             |
|---|-----------------------------|
| - MTU est de 1500B partout,   | - Taille(Get) = 210B        |
| - $T_p(CS) = T_p(SC) = 50ms$  | - Taille (Elément) = 14000B |
| - Débit pour les données transportées<br>par les technologies = 1Mbit/s | - Awnd donné par C = 6000B  |
| - En-tête TCP = 20B   | - Awnd donné par S = 8760B  |
| - En-tête IP = 20B  | - RTO = 250ms               |

**1.1 Demande d'http à TCP**

Du côté de C que demande http à TCP ? Comment cela se traduit-il au niveau TCP ? On donnera les informations relatives à cela dans l'entête TCP/IP.

La couche applicative HTTP sur C demande à la couche TCP d'ouvrir une connexion avec S puis, une fois la connexion établie, d'acheminer son message GET de manière fiable jusqu'à la couche applicative http sur S (qui à son tour, demandera à TCP d'acheminer à C sa réponse).

Les informations nécessaires à ce service sont échangées à travers du point d'accès au service dénommé port en TCP. Le port source est directement connu par TCP sur C, vu que c'est grâce à ce port qu'HTTP communique avec lui. Mais il a besoin pour envoyer sa demande de connexion des informations sur la destination :

- Adresse IP destination (HTTP fera une requête auprès de son resolver DNS) pour envoyer au bon équipement dans Internet ;te
- Port destination – pour savoir avec quel service discuter sur S.

A noter la notion de socket qui est l'adresse IP, le protocole qui utilise IP (TCP ou UDP) et le port.

Enfin TCP aura bien sûr besoin des messages (ici le GET) que souhaite transmettre HTTP, pour lui c'est ce qu'il nomme le flux applicatif.

**1.2 Chronogramme de la communication**

Représenter par un chronogramme l'ensemble de la communication TCP qui entre C et S qui permet au navigateur de C d'afficher sa page web.

Par la suite vous pouvez reprendre le chronogramme avec une perte, celle du 5eme segment de donnée envoyé par S.

Commençons par calculer les temps dont nous avons besoin :

Pour les messages de signalisation de TCP, on considère qu'il n'y a qu'un en-tête de 20B TCP et un en-tête IP de 20B, pas de donnée et pas d'option (généralement c'est faux pour le SYN, on utilise aujourd'hui pas mal d'options)

$Te_{SYN} = Te_{ACK} = Te_{FIN} = 40 \times 8 \cdot 10^{-6} = 320 \mu s = 0,32ms$  (négligeable ici)

**NB :** Vous pourrez trouver le sujet du TD sur Moodle

$$Te_{GET} = (210+20+20) \times 8 \times 10^{-6} = 2 \text{ ms}$$

Il faut émettre un message http 14000B en retour alors que la MTU est de 1500B. TCP va segmenter ce flux applicatif en morceaux, appelé segment de données où chaque segment pourra avoir une taille maximale de  $1500 - 20 - 20 = 1460\text{B}$ .

Du coup cela nous donne 9 segments de 1460B et un dernier segment de 860B. Il faut bien sûr ajouter les en-têtes pour calculer le temps d'émission sur le réseau.

$$Te_{1500B} = 1500 \times 8 \times 10^{-6} = 12 \text{ ms}$$

$$Te_{900B} = 900 \times 8 \times 10^{-6} = 7,2 \text{ ms}$$

A partir de là, nous pouvons commencer à tracer notre chronogramme, on prendra soin de considérer que C ne peut recevoir que jusqu'à 4 segments vu son awnd.

Pour voir l'image en plus grand cf pdf chronogramme.

Il faut penser à représenter les fenêtres d'émissions.

On pourrait se poser la question du taux d'efficacité du protocole TCP dans ce cas ci et de comment l'améliorer.

Reprendre le chronogramme avec la perte du 5ème segment de données (numéro 4)

