Couche application

Réseaux — C/TD 2

Exercice 1 : *Questions du cours*

- Un site de commerce électronique aimerais avoir une trace (enregistrer) des achats de ses clients. Est-il possible de le faire avec HTTP ? Décrire une implémentation qui utilise les cookies.
- Est-il possible que les serveurs de maile et Web d'une organisation aient le même alias (p.e. `worldcomp.com`) ? Si non, expliquer. Si oui, indiquer les entrées RR DNS.

Problème 2 : *Applications multi-protocole*

Un utilisateur veut retrouver le document Web à l'URL A. L'adresse IP du serveur spécifié par A n'est pas initialement connu. Le document recherché contient une référence vers une image GIF qui se trouve sur le même serveur et une référence vers une adresse maile. Quels sont les protocoles transport et application utilisés pour répondre à la demande de l'utilisateur ?

Problème 3 : *Temps de réponse*

Supposons que dans votre navigateur, vous cliquez sur un lien pour obtenir un document Web. Supposons que l'adresse IP du serveur qui détient ce document n'est pas *cachée* (mémorisée) sur votre hôte, donc un accès DNS est nécessaire pour l'obtenir. Supposons que pour cela, n serveur DNS doivent être visités par la requête avant d'obtenir l'adresse IP ; chaque visite prend un RTT de respectivement RTT_1, ..., RTT_n.

- Le document recherché contient un seul objet, la page Web avec un petit texte en HTML. Soit RTT_0 le RTT entre votre hôte et le serveur qui détient le document. Si la transmission de l'objet est négligeable, combien de temps s'est écoulé entre le moment où vous avez cliqué sur le lien et le moment où vous obtenez la page ?
- Le document recherché fait référence à trois petits objets sur le même serveur (que le document). Si on néglige les temps de transmission, combien de temps est nécessaire pour afficher ces objets (dans le document initial) dans chacun des cas suivants :
  1. HTTP non-persistent sans connexions TCP en parallèle,
  2. HTTP non-persistent avec connexions parallèles et
  3. HTTP persistent avec pipelining.

Problème 4 : *Performances*

On désire calculer l'*utilisation effective* du réseau par les protocole application. Pour cela, on considère le transfert de deux documents, un de 1Ko et l'autre de 5Mo avec les protocoles SMTP et HTTP.

1. Calculer le temps nécessaire pour transmettre ce fichier sur des liens ayant un largeur de bande de respectivement 8 kbps, 40 kbps, 800 kbps, 10 Mbps, 800 Mbps.
2. Calculer la durée du transfert de chaque document par SMTP et HTTP sur les liens suivants :
   - une liaison GSM (téléphone portable), ldb 8 Kpbs, RTT 600 ms,

- une liaison téléphonique, ldb 40 kbps, RTT 200 ms,
- une liaison DSL, ldb 800 kbps, RTT 100 ms,
- un réseau local, ldb 10 Mbps, RTT 0.2 ms,
- une liaison satellite, ldb 800 Mbps, RTT 500 ms,
- une liaison avec l'ENS Mars, ldb 800 Kbps, RTT 10 min.

On n'oubliera pas de prendre en compte le temps nécessaire pour l'établissement d'une connexion TCP, mais on omettra celui nécessaire à sa fermeture.

Remarque : En fait, TCP suppose un RTT qui ne dépasse pas 128 s ; la connexion interplanétaire devra donc utiliser un autre protocole de transport. Le RTT avec Mars varie de 3.3 min à 11 min si la Terre est Mars sont en opposition et environ 45 min lors d'une conjonction.

3. Dans chacun des cas précédents, calculer l'utilisation effective du réseau, soit le rapport entre le temps calculé à la question (1) et le temps calculé à la question (2).

   Proposer des solutions au problème mis en évidence (des améliorations au niveau de l'implémentation, des changements dans les protocoles application ou même dans le protocole de transport).

Problème 5 : *Hashing dans le caching*

Ce problème s'intéresse au hachage des URL dans le cache coopératif Web. Soit une institution ayant trois caches, nommés respectivement cache-0, cache-1 et cache-2. Soit un schéma de hachage qui effectue les opérations suivantes :

1. il extrait uniquement le nom du serveur de l'URL à chercher/stocker,
2. tous les caractères non alphabétiques sont ignorés,
3. à chaque caractère alphabétique il lui associe sa position dans l'alphabet,
4. ces numéros sont additionnés modulo 3 pour obtenir le code de hachage,
5. si ce code est 0, la requête (la mémorisation) est faire dans le cache-0, etc.

Par exemple, pour l'URL `http://aaa.bbb.com/...`, on calcule $1+1+1+2+2+2+3+15+13$ $[3] = 1$, donc la requête est envoyée vers le cache-1.

- Calculez les code hash pour les URLs ayant les serveurs suivants : `www.liafa.jussieu.fr`, `www.eurecom.fr`, `www.ens-chachan.fr`.
- Supposons que les caches sont initialement vides. Pendant la matinée, les requêtes des utilisateurs incluent les URL `www.ens-chachan.fr/di/stic.html`, `www.ens-chachan.fr/lsv/index.html`, `www.liafa.jussieu.fr/index.html`. Pendant l'après-midi, il y a 5 nouvelles requêtes vers chacune de ces URL.

  Quels sont les serveurs qui satisferons chacune de ces 18 requêtes ?