Année Universitaire 2016-2017 session 1 d'automne

Parcours : IN4 Licence 2 Informatique **UE :** 4TIN401U
Épreuve : Examen réseaux
Date : 2 mai **Heure :** 9h00 **Durée :** 1h30
Modalités : Épreuve sans document, calculatrice autorisée
Épreuve de : Mr Samuel Thibault 3 **pages**

université de BORDEAUX

Collège Sciences et technologie

Exercice 1. Questions de cours

Q1.1

L'adresse MAC est l'adresse propre à la carte réseau physique, visible sur le réseau local seulement (couche liaison), elle ne change normalement pas.

L'adresse IP identifie l'ordinateur sur le réseau Internet (couche réseau), elle change quand on change de réseau, elle est souvent attribuée dynamiquement par un serveur du réseau. Il arrive qu'elle soit partagée par plusieurs ordinateurs dans le cas du masquerading (e.g. la box à la maison), auquel cas les ordinateurs ont des adresses privées, qui ne sont plus forcément uniques dans le monde.

Le port TCP identifie un service pour un serveur (de nombreux ports ont une valeur bien connue), une connexion d'application pour un client (il est alors attribué dynamiquement), permettant ainsi de multiplexer plusieurs connexions pour un même serveur ou un même client.

Q1.2

Il vaut mieux utiliser UTF-8 comme recommandé par l'IETF, car il permet d'écrire du texte dans toutes les langues utilisées dans le monde.

Q1.3

Séparer le fond et la forme permet de plus facilement travailler sur chacun des deux aspects séparément : travailler sur le contenu d'un article sans se soucier du design de la mise en forme, et inversement.

Cela économise par ailleurs des téléchargements, la .css pouvant être téléchargée une seule fois pour différentes pages d'un site ayant la même forme.

Enfin, cela permet aux lecteurs d'écran d'accéder plus facilement au contenu, pour faire un rendu adapté à des utilisateurs en situation de handicap.

Exercice 2. Questions de cours (2)

Q2.1

Rebâtir tout le World Wide web signifierait remettre en route des millions de sites web, 7 personnes ne suffiraient pas :)

Q2.2

Dans le cas de ssh, on a vu que le serveur ssh a sa propre paire de clés publique/privée, et l'utilisateur fabrique également sa propre paire de clés publique/privée. La vérification est faite par l'utilisateur (vérification de l'empreinte, mise en place de la clé dans le fichier authorized_keys). À aucun moment on n'a affaire à une clé centrale de chiffrement, la gestion des clés est complètement acentrée.

Q2.3

Dans le cas de https, on a vu qu'on utilise des autorités de certifications, qui permettent d'éviter à l'utilisateur d'avoir à vérifier les empreintes etc. Il y a donc une certaine centralisation apportée par ces autorités. Il y a cependant bien plus qu'une seule autorité, il y en a plutôt de l'ordre de quelques centaines, il faudrait alors corriger en "des clés de chiffrement chez chacune des autorités de certification".

Q2.4

La racine des noms de domaine ne contient que la partie TLD (Top-Level Domain) : .com, .net, .org, etc. et donc ne change que très peu. Ainsi il a été possible de répliquer cette racine des centaines de fois. Ainsi, même si l'arbre des noms de domaine a un centre, sa racine, celle-ci est répliquée, et donc résiste même aux attaques de grande ampleur.

Exercice 3. Adresses

Q3.1

Le réseau `140.77.128.0/23` a 23 bits dans son masque réseau, les adresses vont donc de `140.77.128.1` à `140.77.129.254` (`140.77.128.0` est réservée pour désigner le réseau et `140.77.129.255` est réservée pour le *broadcast*)

Q3.2 Les adresses `140.77.128.127`, `140.77.129.128`, et `140.77.128.129` en font partie.

Q3.3

Le réseau `2001:660:5000:128::/64` a 64bits dans son masque réseau, sur les 128 bits d'adresse il en reste donc encore 64 pour la partie machine. Il y a donc 2^64 adresses utilisables (il n'y a pas d'adresse réservée), donc environ 18 milliards de milliards.

Exercice 4. Calculs

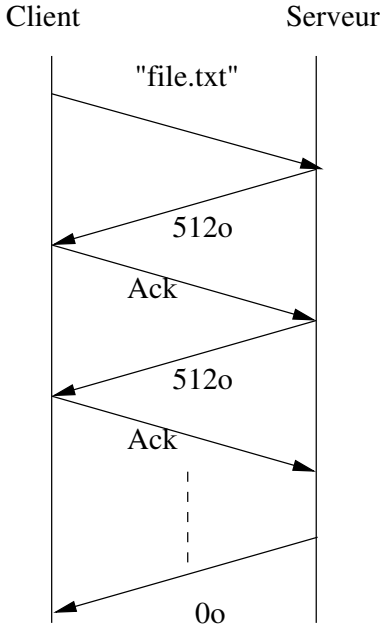
TODO

Exercice 5. Analyse de paquet

TODO

Exercice 6. TFTP

Q6.1



Q6.2

Le serveur attend, avant d'envoyer la suite du fichier, que le client acquitte les données précédemment envoyées. Ainsi, pour chaque paquet de 512 octets, on attend la latence aller/retour. Le débit est donc limité à au plus $512/latence$. Avec une latence ADSL de 50ms par exemple, on ne peut donc espérer plus que 10Ko/s...

Pour éviter le problème, il faudrait que le serveur n'attende pas les acquittements du client. Mais alors pour s'assurer que tout le fichier a été reçu dans l'ordre, il faudrait numéroter les datagrammes. On en revient à réimplémenter TCP...

Q6.3

Si l'on multiplie par 2 la taille des datagrammes, par exemple, la limite mentionnée ci-dessus devient deux fois plus grande. On est cependant limité à la taille maximale des paquets sur le réseau. La MTU est typiquement de 1500 octets, ce qui limite donc à un débit 3 fois plus rapide seulement. En exploitant la fragmentation IPv4, on peut envoyer des paquets d'une taille plus grande que la MTU, mais on reste limité par la taille du champ length, sur 16 bits, donc 65Ko au maximum, et donc un débit 130 fois plus grand seulement.