Année Universitaire 2009-2010 session 1 d'automne

Parcours : CSB5 Licence 3 STS Informatique **UE :** INF157
Épreuve : Examen Utilisation des réseaux
Date : 16 décembre 2009 **Heure :** 11h00 **Durée :** 1h30
Modalités : Épreuve sans document, avec calculatrice
Épreuve de : Mr Samuel Thibault 2 **pages**

Exercice 1. Questions de cours

Q1.1 Citez (sans détailler) les différences et points communs entre TCP et UDP.

TCP et UDP sont tous deux des protocoles situés entre IP et l'application, fournissant des fonctionalités de couche transport*. Ils multiplexent tous deux les services à l'aide de numéros de ports et assurent tous deux l'intégrité des données à l'aide d'un* checksum *de 16 bits.*

TCP a cependant l'avantage de garantir une délivrance de données dans l'ordre et sans perte, ainsi qu'une gestion de flux, à l'aide d'une notion de connexion.

Q1.2 Pourquoi préfère-t-on utiliser UDP pour le protocole DNS, alors que l'on préfère utiliser TCP pour le protocole HTTP ?

Les requêtes et les réponses DNS, étant petites, tiennent typiquement dans un seul paquet IP, il n'y a donc pas besoin de garantir d'ordre de délivrance ou de gestion de flux, puisqu'il n'y a qu'un paquet à transmettre. UDP convient donc assez bien. Si le paquet est perdu, la couche application devra simplement réemettre elle-même.

Les requêtes HTTP sont également souvent courtes (tant que l'on ne demande pas à envoyer un fichier), mais les réponses sont typiquement longues : elles peuvent être une page html bien sûr, mais aussi une image par exemple, que l'on veut être sûr de récupérer entièrement et dans l'ordre. Il est donc préférable d'utiliser TCP.

Q1.3 Dans un article récent de `PCWorld.fr` à propos des « DNS menteurs », on peut lire : « Il s'agit d'[...]éviter à un internaute de tomber sur une erreur 404 affichée par son navigateur Internet si le domaine auquel il essaye de se connecter n'existe pas. ». Expliquez pourquoi l'auteur de la phrase n'a pas l'air de comprendre correctement l'interaction entre DNS et HTTP.

Si le domaine n'existait vraiment pas, le navigateur n'aurait pas pu obtenir d'erreur 404, puisqu'une erreur 404 n'est rapportée par le serveur qu'à l'intérieur d'une connexion HTTP*. Or pour établir une connexion HTTP, il faut établir une connexion TCP, qui nécessite de connaître d'abord l'adresse IP du serveur, ce qui nécessite donc que le domaine existe pour fournir une telle adresse IP.*

Q1.4 Un collègue a utilisé un *webmail* pour m'envoyer par *e-mail* un fichier texte. Dans celui-ci les mots apparaissent bizarrement, par exemple « protocole rÃ©seau ». Que s'est-il passé ? Qu'aurait-il dû se passer ?

« rÃ©seau » ressemble typiquement à un mot encodé en UTF-8 mais affiché en utilisant l'encodage latin1. Le webmail *du collègue aurait dû ajouter un en-tête précisant l'encodage du fichier, pour que le logiciel réceptionnant ce mail sache quel codage utiliser pour le lire.*

Exercice 2. Trace

Voici la configuration d'une machine :

```
$ /sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 00:21:70:b4:36:49
          inet adr:193.50.110.76  Bcast:193.50.110.255  Masque:255.255.255.0
          adr inet6: fe80::221:70ff:feb4:3649/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:78693 errors:0 dropped:0 overruns:0 frame:0
          TX packets:51449 errors:0 dropped:0 overruns:0 carrier:0
```

```
          collisions:0 lg file transmission:1000
          RX bytes:44994757 (42.9 MiB)  TX bytes:14060107 (13.4 MiB)
          Interruption:18

$ /sbin/route
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic Metric Ref    Use Iface
193.50.110.0    *               255.255.255.0   U     0      0        0 eth0
default         193.50.110.254  0.0.0.0         UG    0      0        0 eth0

$ /usr/sbin/arp
Address                  HWtype  HWaddress           Flags Mask            Iface

$ cat /etc/resolv.conf
nameserver 193.50.111.150

$ cat /etc/services
echo 7/udp
echo 7/tcp
ssh 22/tcp
telnet 23/tcp
smtp 25/tcp
domain 53/udp
www 80/tcp

$ cat /etc/protocols
icmp 1 ICMP
tcp 6 TCP
udp 17 UDP
```

Q2.1 On tape la commande `ping -c 1 www.google.fr` (`-c 1` indique à `ping` de n'envoyer qu'un seul paquet `ICMP echo request`). Donnez la liste de tous les paquets qui sont alors émis et reçus par cette machine en précisant les protocoles, services, adresses et ports MACs, IPs, TCP et UDP éventuellement utilisés.

Pour envoyer un paquet ICMP, il faut d'abord obtenir l'adresse IP du serveur, en utilisant une résolution DNS à l'aide du serveur `193.50.111.150`*. Puisque d'après la table de routage celui-ci n'est pas dans le réseau directement accessible (*`193.50.110.0/255.255.255.0`*), il faut passer par la passerelle* `193.50.110.254`*. Puisque la table ARP est vide, il faut d'abord obtenir son adresse MAC par le protocole ARP. La liste complète est donc :*

Résolution ARP pour atteindre la passerelle :

- *Envoi d'une requête ARP depuis l'adresse MAC de la machine* `00:21:70:b4:36:49` *vers toutes les machines (*`ff:ff:ff:ff:ff:ff`*), demandant l'adresse MAC correspondant à l'adresse IP* `193.50.110.254` *de la passerelle.*
- *Réception de la réponse ARP de la passerelle, depuis son adresse MAC (notée* `uu:vv:ww:xx:yy:zz`*) vers* `00:21:70:b4:36:49`*.*

Résolution DNS pour obtenir l'IP du serveur :

- *Envoi de la requête DNS dans un paquet UDP (protocole IP 17) de port destination* `53` *et d'un port source noté* `n`*, depuis l'adresse MAC* `00:21:70:b4:36:49` *et l'adresse IP* `193.50.110.76` *vers l'adresse MAC* `uu:vv:ww:xx:yy:zz` *(la passerelle) et l'adresse IP* `193.50.111.150` *(le serveur DNS).*
- *Réception de la réponse DNS dans un paquet UDP (protocole IP 17) de port destination* `n` *et de port source* `53`*, depuis l'adresse MAC* `uu:vv:ww:xx:yy:zz` *(la passerelle) et l'adresse IP* `193.50.111.150` *(le serveur DNS) vers l'adresse MAC* `00:21:70:b4:36:49` *et l'adresse IP* `193.50.110.76`*. La réponsse contient l'adresse IP de* `www.google.fr`*, notée* `a.b.c.d`*.*

ping-pong ICMP avec le serveur :

- *Envoi de la requête ICMP echo request (protocole IP 1) depuis l'adresse MAC* `00:21:70:b4:36:49` *et l'adresse IP* `193.50.110.76` *vers l'adresse MAC* `uu:vv:ww:xx:yy:zz` *(la passerelle) et l'adresse IP* `a.b.c.d`.
- *Réception de la réponse ICMP (protocole IP 1) depuis l'adresse MAC* `uu:vv:ww:xx:yy:zz` *(la passerelle) et l'adresse IP* `a.b.c.d`. *vers l'adresse MAC* `00:21:70:b4:36:49` *et l'adresse IP* `193.50.110.76`

Exercice 3. Calculs de débits et latence

On veut sauvegarder les *homes* CREMI des étudiants (1 To de données) à l'université de Pau, au cas où tout le campus de Bordeaux brûlerait. Considérons deux solutions :

- Emmener un disque dur en voiture à Pau (environ 3 h de trajet par l'autoroute).
- Transférer par Internet via Renater (on supposera atteindre un débit en IP de 2 Gbps).

Q3.1 Quelle est la solution la plus rapide ? (on pourra arrondir grossièrement les calculs pour simplifier) Même question dans le cas du cluster *decrypthon* de la DRIMM, pour lequel il y a cette fois-ci 3 To de données à sauvegarder.

Il faut 3 h (environ 10 000 s) pour transférer 1 To, cela représente donc un débit de 100 Mo/. Renater, avec 2 Gbps, fournit 250 Mo/s. Renater est plus rapide.

Avec 3 To de données à sauvegarder, soit trois fois plus, le débit de la voiture est alors triplé et atteint 300 Mo/s, ce qui devient plus rapide que Renater.

Q3.2 Un disque dur occupe à peu près $400\,cm^3$. En utilisant une grande quantité de disques de 2 To et une camionnette de $12\,m^3$, quel débit peut-on théoriquement atteindre entre Bordeaux et Pau ? Quelle est la latence ?

$12\,m^3$ *(=* $12\,000\,000\,cm^3$*) peuvent contenir* 30 000 *disques, soit donc* $60\,000\,To$*. Avec un trajet de 3 h (environ 10 000 s), cela représente donc un débit d'environ* $6\,To/s$! La latence reste tout de même 3 h !

Exercice 4. Serveur web et sécurité

Q4.1 Pourquoi est-il important de faire attention aux détails du chemin fourni par le client ? Que pourrait-il se passer ?

Le serveur doit faire attention à ne pas envoyer de fichier qui n'était pas censé être lisible sur Internet. Il faut par exemple faire attention à ne pas "sortir" du répertoire où sont déposés les documents et atteindre `/etc/passwd` *par exemple, via* `/../` *ou via des liens symboliques par exemple. Il faut aussi faire attention aux débordements de tampon lors du traitement du chemin.*

Exercice 5. Sendfile

L'appel système `sendfile` a le prototype suivant :

```
ssize_t sendfile(int out_fd, int in_fd, off_t *offset, size_t count);
```

Elle transfère `count` octets depuis la position `*offset` du descripteur de fichier `in_fd` vers le descripteur de fichier `out_fd`. Il s'agit donc en fait d'une combinaison d'un `read` suivi d'un `write`.

Q5.1 Pourquoi est-ce intéressant d'utiliser `sendfile` plutôt qu'appeler `read` puis `write` ?

Tout d'abord, utiliser un seul appel système plutôt que deux limite le coût des appels systèmes. De plus, on économise aussi a priori *des copies puisque le noyau peut alors directement envoyer les données (qu'il vient par exemple de lire depuis le disque dur) vers le réseau, sans passer par un tampon en espace utilisateur. Enfin, l'application peut éviter d'utiliser une boucle for (nécessaire lorsque l'on passe par un tampon de taille limitée) et laisse le noyau se débrouiller pour effectuer l'envoi complet du fichier, en un seul appel système.*

Le noyau peut ainsi à loisir optimiser l'interaction entre le flux de données depuis le disque dur et le flux de données vers le réseau.

Exercice 6. Protocole FTP

Le protocole FTP (*File Transfer Protocol*) permet d'envoyer/recevoir des fichiers vers/depuis un serveur. Le fonctionnement par défaut (appelé *actif*) pour transférer un fichier est que le client établit une première connexion TCP vers le port 21 du serveur, ouvre en écoute un port TCP local, et transmet au serveur, via la première connexion, le numéro de ce port. Le serveur établit alors une connexion vers ce port et l'utilise pour effectuer le transfert.

Q6.1 L'adresse IP de ma machine est `192.168.0.1`, pourquoi est-ce *a priori* un problème pour que ce protocole puisse fonctionner ? Quel élément de mon réseau doit être attentif et que doit-il faire ?

Le problème est que `192.168.0.1` *est une adresse privée, typiquement une adresse distribuée par les* box*es du commerce. Celle-ci doit être traduite automatiquement en une adresse IP publique par le routeur du réseau avant que les paquets ne parviennent sur Internet (technique de NAT ou* masquerading*). Lorsque le serveur essaie de se connecter sur le port TCP indiqué, c'est ce routeur qui reçoit la connexion, mais ne sait* a priori *pas à quoi cela correspond. Il faut donc qu'il soit attentif au protocole FTP, et lorsqu'il voit un client transmettre un numéro de port, effectue (en plus de la traduction d'adresse IP) une traduction du numéro de port d'écoute FTP, pour savoir ensuite rediriger la connexion TCP (que le serveur établit) vers la machine qui a ouvert ce port.*