TP Réseau n°4 - TCP/UDP

© 2011 tv <tvaira@free.fr> - v.1.0 - le 11 décembre 2011

Table des matières

*Un compte-rendu au format texte (**UTF-8**) devra être rédigé et envoyé à l'adresse **tvaira@free.fr***
*La convention de nommage pour les compte-rendus est la suivante : **tp-4-nom.txt***

Manipulations

Objectifs

- principe des modes connecté (TCP) et non connecté (UDP)
- notion de port

Remarque : il est conseillé de consulter la FAQ Netkit en cas de besoin.

Mise en situation

1. **Solution n°1** : Vous devez disposer d'un PC possédant une distribution Linux (sur une partition spécifique, sur une clé USB bootable, sur un Live CD ou encore à l'aide d'un logiciel de virtualisation du type *VMware* ou *VirtualBox*). Le logiciel de virtualisation **Netkit** doit être installé sur la machine Linux ainsi que le programme `uml_dump`. Évidemment, le logiciel `wireshark` doit être installé sur votre système.

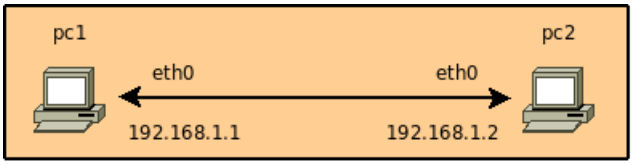
   - **Site de NetKit :** www.netkit.org
   - **Site pour `uml_dump` :** kartoch.msi.unilim.fr

2. **Solution n°2** : utilisez un **Live CD/DVD/USB Netkit**. Vous pouvez aussi utiliser l'image ISO à l'aide d'un logiciel de virtualisation du type *VMware* ou *VirtualBox*.

   - **Site du Netkit live DVD/USB :** tocai.dia.uniroma3.it
   - **Site du Netkit4TIC live DVD :** tocai.dia.uniroma3.it
   - **Site du Live CD Raizo :** www.utec-tic.org

Installation du TP

La configuration est la suivante :



 © 2011 tv <tvaira@free.fr>

Travail demandé

Séquence 1 : TCP

TCP (*Transmission Control Protocol*) est un protocole de transport fiable, en mode connecté (**RFC 793**) qui assure la transmission des données de bout en bout (d'un processus à un autre processus). C'est un protocole de la couche **Transport**.

Le protocole **TCP** utilise les **numéros de port** (une valeur codée sur 16 bits) comme technique d'adressage des bouts d'une communication.

`Netcat` est un utilitaire qui permet de réaliser des communications client ou serveur en TCP (ou UDP). `Telnet` est un utilitaire qui permet de réaliser des communications client en TCP. `Telnet` et `netcat` sont donc des programmes capables d'ouvrir une *socket* TCP sur un port et de dialoguer en mode texte (ASCII). Dans ce cas, la couche Application peut être considérée comme "vide" au niveau protocole.

Question 1. Exécuter, sur le poste **pc1**, `netcat` en mode **serveur TCP** sur le **port 5000**. Donner la commande exacte.

Question 2. Utiliser, sur le poste **pc2**, `telnet` ou `netcat` comme **client TCP**. Donner la commande qui permet de se connecter et d'échanger des données avec le serveur lancé précedemment.

Question 3. Exécuter simultanément, sur un des postes, deux programmes serveur sur le même port. Que se passe-t-il ?

Activer une capture `wireshark` sur le domaine A.

Question 4. Lancer un serveur TCP sur le port 5000 sur la machine **pc1**. Puis en utilisant la commande ci-dessous sur **pc2** et en observant l'échange avec `wireshark`, en déduire la valeur du **MSS** (*Maximum Segment Size*).

```
pc2 :~# cat /etc/passwd | netcat 192.168.1.1
```

Question 5. Lancer un serveur TCP sur le port 5000 sur la machine **pc1**. Puis en utilisant la commande ci-dessous sur **pc2** et en observant l'échange avec `wireshark`, répondre aux questions suivantes :

a) Quelles sont les valeurs des fenêtres annoncées par le client et le serveur au départ de l'échange ?

b) Cela représente combien de segments transmissibles par anticipation ?

c) Combien de trames ont-elles été échangées dans cet échange ?

d) Combien de trames ont-elles transportées des données dans cet échange ?

e) Calculer le rendement en % pour cet échange (total octets de données / total octets transmis) ?

```
pc2 :~# cat /etc/services | netcat 192.168.1.1 5000
```

 © 2011 tv <tvaira@free.fr>

Question 6. Lancer un serveur TCP sur le port 5000 sur la machine **pc1**. Puis en utilisant la commande ci-dessous sur **pc2** et en observant l'échange avec `wireshark`, compléter le diagramme des échanges TCP ci-dessous en précisant le **rôle de chaque échange**.

a) Indiquer les numéros de port du client et du serveur.

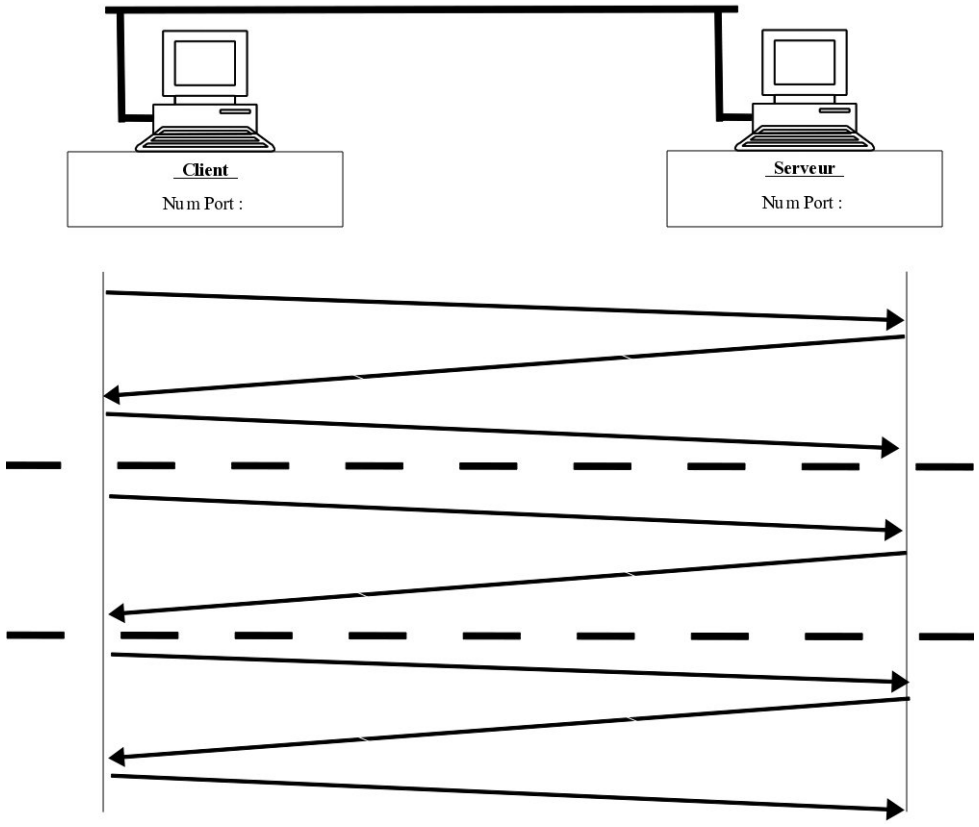
b) Indiquer sur chaque flèche les *flags* (SYN, ACK, ...) ainsi que que les numéros de séquence et d'acquittement échangés (donner les valeurs réelles).

c) Indiquer les nombres d'octets de données échangées par les deux programmes client/serveur.

d) Vérifier les nombres d'octets de données échangées par les deux programmes client/serveur à partir des numéros de séquence et d'acquittement capturés dans l'échange.

e) Quel comptage est assuré par les numéros de séquence et d'acquittement de chaque côté de la communication ?

```
pc2 :~# netcat 192.168.1.1 5000
client ---hello world !---> serveur
client <-------ok--------- serveur
client : Ctrl-C
```



 © 2011 tv <tvaira@free.fr>

Question 7. Exécuter un **serveur TCP** avec `netcat` sur un des ports compris entre 5000 et 5005 sur le poste **pc1**. Puis à partir de **pc2**, détecter les ports ouverts acceptant des connexions **TCP** dans la plage **5000-5005**, puis dans la plage **7-13**, en utilisant l'outil `netcat` puis `nmap`. Tester avec `netcat`. Tester et donner la commande avec `nmap`.

```
pc2 :~# netcat -vv -z pc1 7-13
```

Question 8. En vous aidant de `wireshark`, comment `nmap` procède-t-il pour détecter les ports ouverts ? En vous aidant des pages **man**, `nmap` propose-t-il plusieurs techniques pour détecter les ports ouverts ?

Question 9. Est-ce que les ports non-ouverts ont répondu à `nmap` ? Si oui, qui a envoyé quoi ?

Question 10. Comment le serveur connaît-il le port utilisé par le client ?

Question 11. Comment le client connaît-il le port utilisé par le serveur ?

 © 2011 tv <tvaira@free.fr>

Séquence 2 : UDP

UDP (*User Datagram Protocol*) est un protocole souvent décrit comme étant non-fiable, en mode non-connecté (**RFC 768**), mais plus rapide que TCP. Il assure lui aussi la transmission des données de bout en bout (d'un processus à un autre processus). C'est un protocole de la couche **Transport**.

Le protocole **UDP** utilise les **numéros de port** (une valeur codée sur 16 bits) comme technique d'adressage des bouts d'une communication.

`Netcat` est un utilitaire qui permet de réaliser des communications client ou serveur en UDP (ou TCP). `Netcat` est donc un programme capable d'ouvrir une *socket* UDP sur un port et de dialoguer en mode texte (ASCII). Dans ce cas, la couche Application peut être considérée comme "vide" au niveau protocole.

Question 12. Exécuter, sur le poste **pc1**, `netcat` en mode **serveur UDP** sur le **port 5000**. Donner la commande exacte.

Question 13. Utiliser, sur le poste **pc2**, `netcat` comme **client UDP**. Donner la commande qui permet de se connecter et d'échanger des données avec le serveur lancé précedemment.

Question 14. Pourquoi `telnet` ne peut-il pas être utilisé ici ?

Activer une capture `wireshark` sur le domaine A.

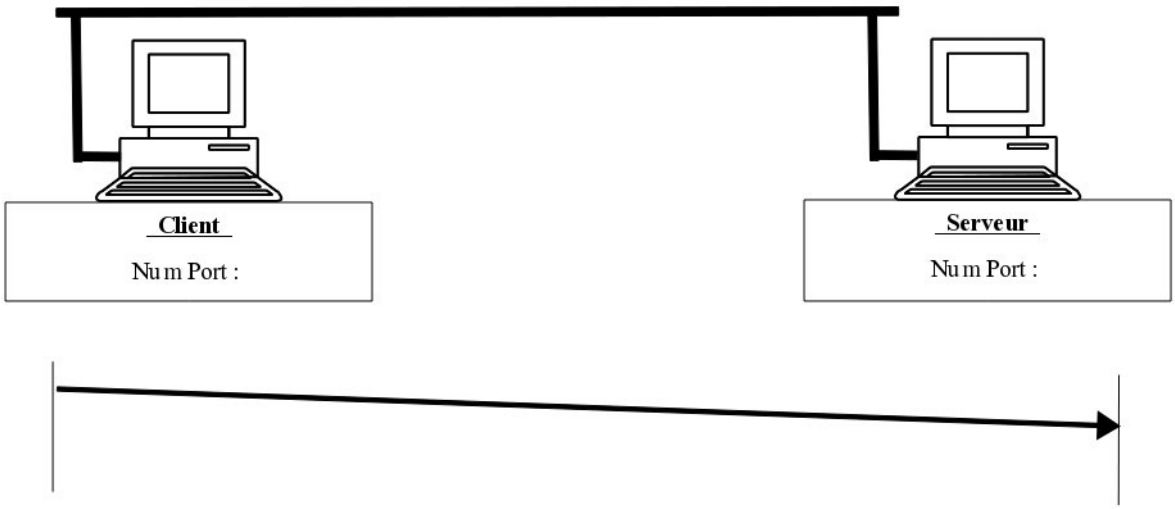
Question 15. Compléter le diagramme des échanges UDP ci-dessous en précisant le **rôle de chaque échange**.

a) Indiquer les numéros de port du client et du serveur.

b) Indiquer les données échangées par les deux programmes client/serveur.

c) Comment UDP fait-il pour acquitter les données reçues ?

client $\xrightarrow{hello\ world\ !}$ serveur
client $\xleftarrow{ok}$ serveur



Question 16. Lancer un serveur UDP sur le port 5000 sur la machine **pc1**. Puis en utilisant la commande ci-dessous sur **pc2**, relever l'échange des datagrammes UDP en indiquant la taille des données contenus dans chacun des datagrammes.

a) Quelle est la valeur maximale des DATAS dans les datagrammes UDP de votre échange ?

b) Est-ce que chacun des datagrammes de l'échange est envoyé dans une seule trame ?

c) Si non, quelle technique est utilisée pour acheminer chaque datagramme ?

d) Qu'a envoyé le serveur ? Pourquoi ?

e) Combien de trames ont-elles été échangées dans cet échange ?

f) Combien de trames ont-elles transportées des données dans cet échange ?

g) Calculer le rendement pour cet échange (octets de données / octets total transmis) ?

```
pc2 :~# cat /etc/services | netcat -u 192.168.1.1 5000
```

Question 17. Est-il possible d'exécuter un serveur TCP et un serveur UDP sur le même port ? Tester.

Question 18. Comment `nmap` procède-t-il pour détecter des ports ouverts en UDP ?

Question 19. Comment le serveur connaît-il le port utilisé par le client ?

Question 20. Comment le client connaît-il le port utilisé par le serveur ?

 © 2011 tv <tvaira@free.fr>