

Réseaux – TD 4

Programmation UDP en C: Serveur, Client

Février 2007

Exercice 1 Serveur, l'addition !

1. Réalisez en C un serveur pouvant recevoir des paquets UDP dont le contenu sera une ligne de texte de la forme :

```
123+456
```

Pour chacun de ces paquets reçus, le serveur devra renvoyer au client le résultat de l'addition, toujours sous forme d'un texte.

Indications :

- Les fonctions C à considérer sont `socket`, `bind`, `recvfrom`, et `sendto`, documentés au travers de leurs pages `man` respectives.
- Sur nivose, pensez à ajouter `-lsocket -lnsl -lresolv` à la ligne de commande de `gcc`.
- Pour tester votre serveur, vous pouvez utiliser l'utilitaire `nc` (pour `netcat`) en mode UDP :

```
nc -u <machine> <port>
```

Sur nivose, `nc` n'est pas installé en standard, il se trouve dans `~letouzey/bin`

2. Supposons que chaque calcul d'addition prenne en fait quelques secondes. Simuler cela avec un `sleep`. Comment faire pour que le serveur soit malgré tout prêt en permanence à recevoir des paquets ?
3. On souhaite maintenant que le client envoie au serveur les deux entiers à additionner via deux paquets UDP distincts. Est-il possible de faire un serveur qui gère convenablement ce scénario ?

Exercice 2 Interrogation DNS minimaliste.

1. Il est possible de former une requête DNS vide sous la forme d'un paquet UDP contenant 12 octets :
 - les deux premiers octets forment un entier 16-bit servant de numéro d'identification que l'on pourra ici choisir arbitrairement
 - les dix octets suivants sont remplis de zéros.

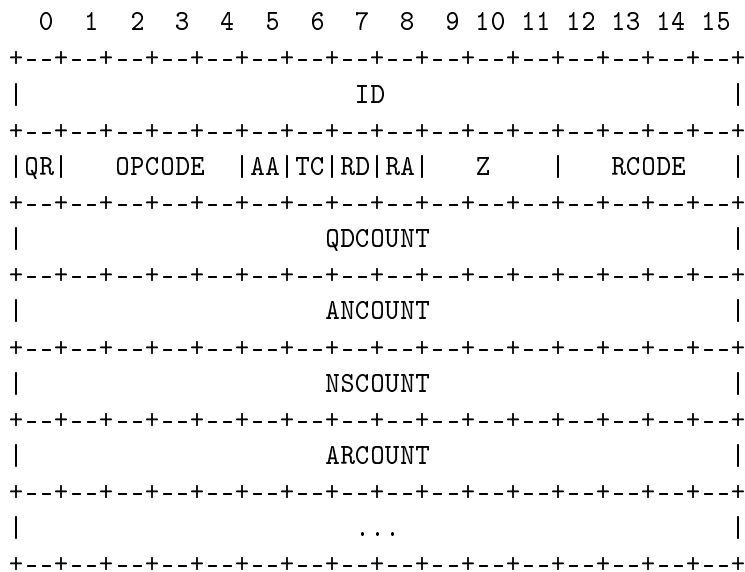
Ecrivez en C un programme permettant via l'envoi d'un tel paquet DNS de tester la présence d'un service DNS sur un serveur. Affichez et interprétez les 12 premiers octets de la réponse éventuelle en fonction du schéma de la page suivante.

Rappel : le serveur DNS sur le réseau de l'UFR est `sweet-smoke`, et le numéro de port du protocole DNS est 53.

2. En l'absence de réponse, il est impossible de savoir si l'on est face à une machine sans serveur DNS ou bien en présence d'un réseau saturé ou défectueux. Quelle stratégie de ré-émission peut-on alors adopter ? **Attention à ne pas surcharger le serveur DNS de requêtes : vérifiez auprès de votre chargé de TP avant toute expérimentation.**

Annexe : Allure d'un paquet DNS.

Comme décrit dans la RFC 1035, paragraphe 4.1.1, un message DNS a la forme suivante :



- ID est un entier qui est copié de la requête à la réponse qui permet de simplifier l'appariement requête/réponse.
- QR vaut 0 pour une requête, 1 pour une réponse.
- OPCODE vaut 0 (les autres valeurs sont obsolètes).
- AA (autorité) vaut 0 dans une réponse qui provient d'un cache.
- TC vaut 1 si la réponse est tronquée (limite de 512 octets).
- RD (recursion desired) vaut 1 si la requête demande au serveur de consulter d'autres serveurs.
- RA (recursion available) sert au serveur pour indiquer qu'il est capable de consulter d'autres serveurs.
- RCODE est un code d'erreur ; il vaut :
 - 0 no error ;
 - 1 format error ;
 - 2 server failure ;
 - 3 name error (le domaine n'existe pas) ;
 - 4 not implemented ;
 - 5 refused.