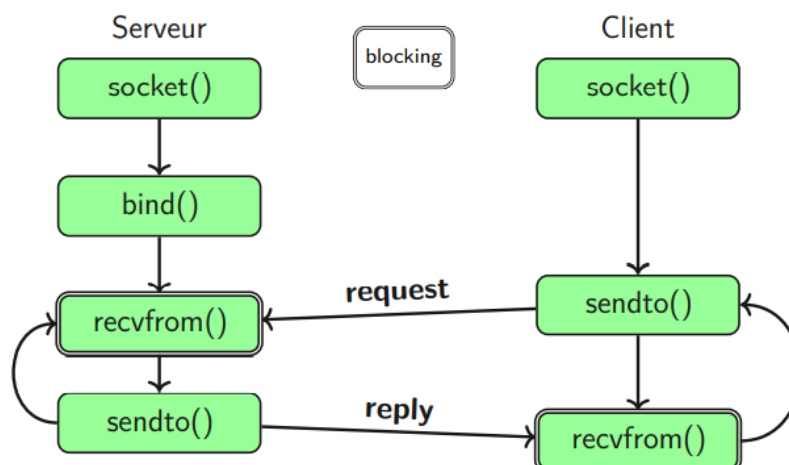


TP de synthese TFTP

Le but de ce TP était de créer un programme client qui devait être capable d'échanger des fichiers à l'aide du protocole TFTP.

Nous avons été jusqu'à la réception d'un fichier d'un seul paquet de données dans le gettftp. Ainsi, nous n'avons pas eu l'occasion de faire la partie spécifique au puttftp. Le code est identique à celui du gettftp. Pour cela nous avons utilisé le serveur téléchargeable sur moodle disponible à l'adresse 127.0.0.1 et sur le port 1069.

TFTP utilise le protocole UDP (adresse IPv4), c'est pourquoi lors de la création du socket nous utilisons les macros spécifiques à l'UDP et aux adresses Ipv4: AF_INET pour le domain, SOCK_DGRAM pour le type et IPPROTO_UDP pour le protocole.



Pour l'envoi des données avec `sendto()`, on implémente la fonction requête. Sur Wireshark, on observe que la requête a bien été envoyée ci-dessous. La source comme la destination est l'adresse IP qui correspond au serveur TFTP. On voit que 15 octets sont bien envoyés vers le port 1069.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.1	UDP	59	38183 → 1069 Len=15
2	0.000015674	127.0.0.1	127.0.0.1	ICMP	87	Destination unreachable (Port unreachable)

2 bytes	string	1 byte	string	1 byte
-----	-----	-----	-----	-----
Opcode	Filename	0	Mode	0
-----	-----	-----	-----	-----

Figure 5-1: RRQ/WRQ packet

La taille du paquet est 15 octets car on a :

- 2 octets : Opcode
- 6 octets : alt256
- 1 octet : 0
- 5 octets : "octet"
- 1 octet : 0

Cependant le port n'est pas accessible c'est pour cela que nous n'avons pas obtenu d'autres traces Wireshark. Le code correspondant au socket doit contenir une faille.

Pour le code de réception que nous avons implémenté, le programme n'affichait pas le nombre d'octets reçus par le client en réponse de sa requête au serveur.

L'acquittement lui-aussi n'a pas pu être testé avec Wireshark pour les mêmes raisons.

Ce que nous nous attendions à voir est une réception d'un paquet de longueur 516 (DAT) en passant toujours par le port 1069. 512 correspond à la longueur maximale de DAT en mode octets à laquelle on ajoute 4 octets : 2 pour Opcode et 2 pour Block# qui renvoie le numéro du paquet reçu.

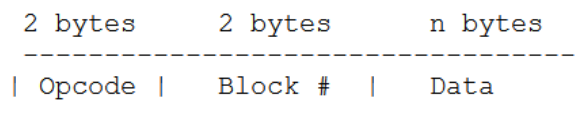


Figure 5-2: DATA packet

Le paquet acquittement renvoyé par le client au serveur pour lui signifier la bonne réception du paquet DATA aurait dû être de longueur 4 octets.

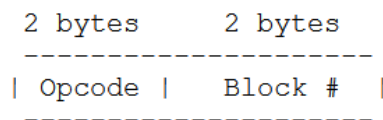


Figure 5-3: ACK packet