ENSEA
Ecole Nationale Supérieure de l'Electronique et de ses Applications

2e année – Majeure Informatique

TP Synthèse – client TFTP

C. BARÈS

Objectifs : Réaliser un client TFTP, à l'aide des RFC et de captures wireshark.

Protocole

Vous allez créez un programme client qui doit être capable d'échanger des fichiers à l'aide du protocole TFTP (Trivial File Transfert Protocol).

Ce protocole TFTP est notamment utilisé pour l'installation d'OS via le réseau (protocole PXE), le fonctionnement des postes de travail sans disque (diskless node), pour la mise à jour de firmware sur les équipements réseaux (routeurs, IoT...)

Pour cela, vous allez devoir vous appuyer sur :

- des RFC :
  - RFC1350 : TFTP v2
  - RFC2347 : TFTP Option Extension
  - RFC2348 : TFTP Blocsize Option
  - RFC2349 : TFTP timeout & tsize options
  - RFC7440 : TFTP Windowsize Option
- des captures wireshark :
  - réalisées par vos soins (+++) ;
  - disponibles sur le site de Wireshark[1] (+) ;
  - ou disponibles sur internet sur le site CloudShark (-).

Vous êtes fortement encouragés à découper votre code dans plusieurs fichiers et à utiliser un Makefile. De même, le découpage de votre programme à l'aide de fonctions, ainsi que le choix des noms de variable doit améliorer la lisibilité de votre code.

Vous pourrez tester votre client en utilisant au choix, et de manière non-exclusive :

- un serveur tftp de votre distribution (`sudo apt install atftp` sur la VM) [2] ;
- le serveur tftp fourni sur moodle [3] ;
- `netcat` et `hexdump` comme « analyseur de trame » :

```
$ nc -l -u 1069 | hexdump -C
```

- le serveur tftp situé sur la machine `srvtpinfo1.ensea.fr`, port 69.

1. https ://wiki.wireshark.org/SampleCaptures#TFTP
2. dans ce cas les fichiers du serveur se trouve dans `/srv/tftp/` et le port est le 69.
3. dans ce cas le port est le 1069, et l'adresse à préciser

Travail à réaliser

Vous allez créer 2 clients, à utiliser depuis la ligne de commande :

- un pour télécharger un fichier depuis le serveur :

```
$ gettftp host file
```

- un pour téléverser un fichier sur le serveur :

```
$ puttftp host file
```

Vous allez implémenter les fonctionnalités suivantes, à réaliser dans l'ordre :

1. Utilisation des arguments passés à la ligne de commande des programmes `gettftp` et `puttftp` pour obtenir les informations de requêtes (serveur et fichier)
2. Appel à `getaddrinfo` pour obtenir l'adresse du serveur ;
3. Réservation d'un socket de connexion vers le serveur ;
4. Pour `gettftp` :
   a) Construction d'une requête en lecture (RRQ) correctement formée, et envoie au serveur.
   b) Réception d'un fichier constitué d'un seul paquet de données (DAT) et son acquittement (ACK) ;
   c) Réception d'un fichier constitué de plusieurs paquets de données (DAT) et leurs acquittements respectifs (ACK) ;
5. Pour `puttftp` :
   a) Construction d'une requête en écriture (WRQ) correctement formée, et envoie au serveur.
   b) Envoie d'un fichier constitué d'un seul paquet de données (DAT) et réception de son acquittement (ACK) ;
   c) Envoie d'un fichier constitué de plusieurs paquets de données (DAT) et réception de leurs acquittements respectifs (ACK) ;
6. Utilisation de l'option `blocksize`,
7. Recherche du `blocksize` optimal.
8. Gestion de la pertes de paquets (et donc des acquittements) et des paquets d'erreur (ERR).

Remarques pour les différents points du cahier des charges :

1. Récupérez un pointeur vers une structure `addrinfo` à l'aide de `getaddrinfo`, en utilisant une adresse fournie via la ligne de commande (voir le TDm3) ;
2. Utilisez cette structure `addrinfo` pour réserver un socket et ouvrez une connexion vers le serveur ;
3. Pour `gettftp` :
   a) Lors de la construction de la requête en lecture (RRQ), respectez les RFC : ordre des octets, présence du caractère nul `'\0'`, choix du mode de transfert…
   b) Pour vos essais, le serveur dispose des fichiers suivant :

- des fichiers contenant que des zéros : `zerosXXX`
- des fichiers contenant que des uns : `onesXXX`
- des fichiers qui alterne 8 zéros puis 8 uns : `alt256`
- des fichiers de petites taille : `zeros256`, `ones256`, `alt256` ;
- des fichiers de taille "spéciale" : `zeros512`, `ones512` ;
- des fichiers plus grands : `zeros1024`, `ones1024`, `zeros2048`, `ones2048` ;
- un fichier `ensea.png` : vous pouvez vérifier son bon transfert grâce à sa signature PNG ;

4. Pour `puttftp`, même remarques, et faites attention aux acquittements.
5. L'option `blocksize` : voir la RFC2348. Quelle est la taille maximale de fichier transférable ?
6. Utilisez Wireshark ou réfléchissez…