


# TP Synthèse – client TFTP

C. BARÈS, N. PAPAZOGLU

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

---

## Conseils généraux :

- Reprenez les instructions du 1<sup>er</sup> TP, elles sont identiques,
  - Vous devez appeler votre professeur à chaque checkpoint (reperé par le symbole ) pour qu'il valide votre avancement. S'il n'est pas disponible à ce moment précis, continuez et appelez le plus tard.
- 

---

## PROTOCOLE

---

Vous allez créer un programme client qui doit être capable d'échanger des fichiers à l'aide du protocole TFTP (Trivial File Transfer 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 Blocksize Option
  - RFC2349 : TFTP timeout & tsize options
  - RFC7440 : TFTP Window Size Option
- des captures wireshark :
  - réalisées par vos soins (+++);
  - disponibles sur le site de Wireshark<sup>1</sup> (+);
  - ou disponibles sur internet sur le site CloudShark (-).

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)<sup>2</sup>;
- le serveur tftp fourni sur moodle<sup>3</sup>;
- netcat et hexdump comme « analyseur de trame » :

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

- le serveur tftp situé sur la machine `srvtpinf01.ensea.fr`, port 69.

---

<sup>1</sup><https://wiki.wireshark.org/SampleCaptures#TFTP>

<sup>2</sup>dans ce cas les fichiers du serveur se trouvent dans `/srv/tftp/` et le port est le 69.

<sup>3</sup>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 envoi 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 envoi au serveur.
  - b) Envoi d'un fichier constitué d'un seul paquet de données (DAT) et réception de son acquittement (ACK);
  - c) Envoi 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 puttfp, 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...