

Année universitaire	2016/2017	Année	4A
Département	Informatique		
Matière	ARAR		
Enseignant	C.GERTOSIO		
Intitulé TD/TP :	TP n°1		

TP

Etude d'un logiciel de Transfert de Fichiers basé sur le protocole TFTP

Le TP est à réaliser par groupe de 4 étudiants.

L'analyse fera l'objet d'un premier compte-rendu corrigé et noté.

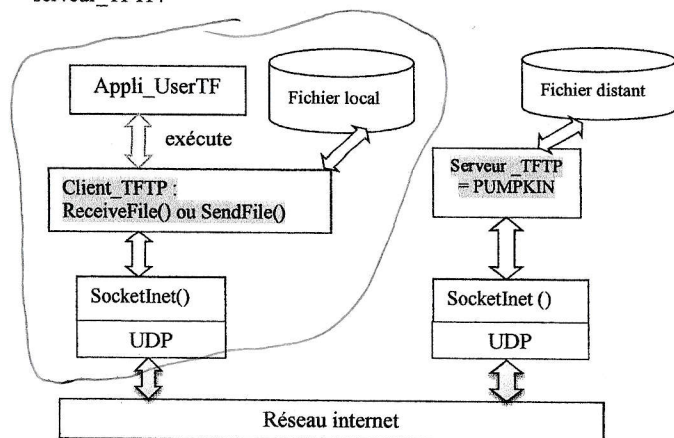
Une démonstration vous sera demandée. Elle sera notée.

Le compte-rendu final répondra aux questions 2 et 4.

On désire mettre en place un logiciel de Transfert de Fichiers (TF) fonctionnant en mode client/serveur entre des machines reliées par un réseau internet. On choisit d'implanter le protocole TFTP (Trivial File Transfer Protocol) décrit par le document Request For Comment (RFC) n°1350 pour assurer les échanges entre les programmes que nous appellerons client_TFTP et serveur_TFTP.

Le serveur_TFTP sera le logiciel PUMPKIN (logiciel libre), qui est conforme à la norme TFTP.

Le Client_TFTP sera construit sous forme de 2 méthodes SendFile() et ReceiveFile() qui seront exécutables par application (Appli_UserTF) pour échanger des fichiers avec le serveur_TFTP.



ReceiveFile () permettra de demander l'envoi d'un fichier par le serveur TFTP en fournissant les paramètres suivants:

- Adresse IP et port du serveur_TFTP,
- Nom_fichier_distant : nom du fichier à recevoir depuis le serveur_TFTP,

- Nom_fichier_local : nom local du fichier reçu.

CrRv est un paramètre retourné par la méthode ReceiveFile().

- CrRv=0 si le transfert s'est bien déroulé.
- CrRv >0 signalera une erreur de transfert intervenue sur le serveur,
- CrRv<0 signalera une erreur locale (impossible d'accéder au répertoire pour créer un fichier, le fichier existe déjà...).

Les valeurs des codes erreurs CrRv sont à définir.

SendFile() permettra au client d'envoyer un fichier sur le serveur en fournissant les paramètres suivants:

- Adresse IP et port du serveur_TFTP,
- Nom_fichier_local : nom local du fichier à envoyer.

CrEm est un paramètre retourné par la méthode SendFile().

- CrEm= 0 : si l'envoi du fichier s'est bien passé,
- CrEm>0 signalera une erreur de transfert intervenue sur le serveur,
- CrEm<0 signalera une erreur locale (fichier local non trouvé par exemple).

Les valeurs des codes erreurs CrEm sont à définir.

Le travail à réaliser.

A partir du document de référence RFC 1350 spécifiant le protocole TFTP (lien en annexe), on vous demande de :

- Donner un exemple d'échange de fichiers sous forme de deux scénarii:
 - **Réception_fichier()** : réception réussie d'un fichier de 1000 octets malgré une erreur de transfert : 1 bloc ACK perdu.
 - **Envoi_fichier()** : envoi réussi d'un fichier de 500 octets, octets malgré une erreur de transfert : 1 bloc DATA perdu.

Les scenarios feront apparaitre les niveaux User_TF, client_TFTP/serveur_TFTP, socket_Inet et UDP.

- Définir les algorithmes des primitives SendFile() et ReceiveFile() en pseudocode. Vous respecterez les règles rappelées dans le document n°2 de l'annexe.
- Développer les 2 méthodes ReceiveFile() et Sendfile() ainsi que appli_User_TF simplifié, qui permettra d'exécuter les 2 méthodes en saisissant l'adresse IP de la machine distante ainsi que le nom du fichier à transférer. Une démonstration vous sera demandée.
- Créer le manuel utilisateur qui expliquera le service offert par TF à une application qui utiliserait client_TFTP pour échanger des fichiers avec serveur_TFTP. Vous veillerez à bien définir les valeurs possibles des codes erreurs CrEm et CrRv pris en charge par votre programme.

Annexes

- Norme RFC 1350-<http://abcdrfc.free.fr/>
- Pseudo code standard: http://users.csc.calpoly.edu/~jdalbey/SWE/pdl_std.html