

# Mini projet de conception de protocoles réseaux informatiques

## *Modélisation du protocole TFTP – Partie 1*

Trivial File Transfert Protocol (TFTP) est un protocole client/serveur extrêmement simple conçu pour être implémenté sur des équipements ne disposant que de très peu de ressources matérielles. Il est spécifié dans la RFC 1350<sup>1</sup>

On le trouve très souvent dans les mémoires flash d'équipements divers nécessitant d'effectuer des mises à jour du système ou le téléchargement d'un fichier de démarrage ou de configuration d'un équipement depuis un serveur contenant ces fichiers.

Le but de ce mini-projet sera de concevoir un client/serveur TFTP simplifié, en utilisant les méthodes vues en cours. Il sera divisé en deux étapes. Ce document explique l'objectif de la première étape.

### **1. Modélisation et analyse informelle du protocole**

Le service, l'environnement, les types de messages et leur encodage ainsi que leur règles d'échange devront être décrits informellement à partir du RFC 1350 pour ensuite être modélisés aussi informellement en utilisant l'une des notations vues en cours : machine à états utilisée<sup>2</sup> pour décrire le protocole « lockstep » ou bien langage Flowchart<sup>34</sup> utilisée pour décrire le protocole de Lynch et diagrammes temps séquence.

La modélisation devra spécifier le fonctionnement et le déroulement du service offert par le protocole du côté client et du côté serveur en se limitant au mode de transfert binaire du protocole (c'est à dire le mode « octet »). Une machine à état ou un flowchart devra spécifier le fonctionnement du service et du protocole côté client, une autre machine à état devra spécifier le côté serveur. Pour simplifier, le modèle supposera que le transfert des données utiles est unidirectionnel.

Quelques exemples d'exécution de votre modèle devront être réalisés sous la forme de diagrammes temps séquences, en particulier sur le comportement de votre modèle si il y a des pertes de paquets.

Les différents cas d'erreurs pouvant se produire devront être déroulés sur des diagramme temps séquence et démontrer que le protocole récupère bien l'erreur dans les cas présentés, sur la base des machines à état proposées. Un diagramme temps séquence ne devra dérouler qu'un seul cas d'erreur.

---

1 <https://tools.ietf.org/html/rfc1350>

2 [https://hepia.infolibre.ch/reseauxI-2019-2020/reseaux\\_et\\_app\\_cours\\_A.pdf](https://hepia.infolibre.ch/reseauxI-2019-2020/reseaux_et_app_cours_A.pdf)

3 [https://hepia.infolibre.ch/reseauxI-2019-2020/biblio/design\\_and\\_validation\\_computer\\_protocols\\_chapter2.pdf](https://hepia.infolibre.ch/reseauxI-2019-2020/biblio/design_and_validation_computer_protocols_chapter2.pdf)

4 [https://hepia.infolibre.ch/reseauxI-2019-2020/biblio/design\\_and\\_validation\\_computer\\_protocols\\_appendix\\_B.pdf](https://hepia.infolibre.ch/reseauxI-2019-2020/biblio/design_and_validation_computer_protocols_appendix_B.pdf)

### **Modalité de rendu**

L'ensemble du projet devra être réalisé en priorité en binôme. L'évaluation compte pour 50 % de la note, une deuxième partie suivra sous peu et comptera aussi pour 50 % de la note. Dans le cas où une personne se retrouverait seule, elle doit effectuer le projet en trinôme.

- Le rendu sous la forme d'un document au format pdf envoyé par mail au professeur et de l'URL d'une courte vidéo (5 min) orale de présentation du modèle. Ce document devra contenir la modélisation et l'analyse telle que décrite au point 1. Elle devra être rendue avant le **22/11/2019 23h59 pour les étudiants du jour, et avant le 8/12/2019 pour les étudiants du soir.**