Machine Project

Start Assignment

Due 23 Jun by 23:59 **Points** 100 **Submitting** a file upload
Available 5 May at 14:15 - 23 Jun at 23:59 about 2 months

Objectives:

This project is formulated as a supplement to classroom instruction for students to demonstrate the following:

- Review and comprehend the detailed function and design of network protocols as described in Internet standards documents
- Implement a working network application that conforms to Internet standards.

Background:

The Trivial File Transfer Protocol (TFTP) is a UDP-based application layer protocol that is used to perform a simple file upload or download within a LAN setting. At present, it is commonly used to transfer network appliance operating system images and configuration files to and from a computer for backup or update purposes. Since it is UDP-based, the TFTP specification includes mechanisms that allow applications to perform simplified reliable and ordered data delivery that would have otherwise been unavailable from the transport layer. The majority of TFTP applications conform to TFTP version2 , which has its specifications documented in RFC 1350 and later extended with additional features using RFCs 2347, 2348 and 2349.

For this project, you will be programming a TFTP **client** program that complies with these protocol specifications

Requirements:

The client program may be implemented using either C, Java or Python sockets and must feature the following:

- GUI or a command line-based user interface are acceptable.
- User is allowed to specify the server IP address.
- Support for both upload and download of binary files
- When uploading:
  - The program can send any file on the computer to the TFTP server as long as the file is accessible to the user using his / her OS privileges
- When downloading:

    - The program must allow the user to provide the filename to use when saving the downloaded file to the client computer
- Proper error handling which at the minimum should include the following:
    - Timeout for unresponsive server
    - Handling of duplicate ACK
    - User prompt for file not found, access violation, and disk full errors

Support for option negotiation will merit additional points if correctly implemented

- Option to specify transfer block size
- Communicate transfer size to server when uploading

Submission Details:

The project is to be submitted in 2 phases.

Milestone 1: Program Design

1. Document showing your message format details and sequence diagrams for messages passed between client and server for file upload and download.
2. Application messages in code form implemented using your chosen programming language. (e.g. C structure definitions, Java classes with necessary attributes or Python scripts)

This should clearly reflect conformity to the protocol specs in both order, datatype and size of message fields.

Submission date: **June 9, 2022 9:00 PM** via AnimoSpace Upload

Milestone 2: Program Implementation

1. Updated document showing your message format details and sequence diagrams for messages passed between client and server for file upload, download **and error conditions**.
2. Complete program source code

Submission date: **June 23, 2020 9:00 PM** via AnimoSpace Upload

Demo date: Finals week. Exact date to be announced

NSCOM01_MP_T2_2021-2022.pdf (https://dlsu.instructure.com/courses/93392/files/9324680/download?download_frd=1) **1350 TFTP version 2.pdf** (https://dlsu.instructure.com/courses/93392/files/9324713/download?download_frd=1) **2347 TFTP Option Extension.pdf** (https://dlsu.instructure.com/courses/93392/files/9324720/download?download_frd=1) **2348 TFTP Block Size Option.pdf** (https://dlsu.instructure.com/courses/93392/files/9324728/download?

download_frd=1) **2349 TFTP Timeout and Transfer Size Option.pdf**
(https://dlsu.instructure.com/courses/93392/files/9324734/download?download_frd=1)