

Assignment 1

The Trivial File Transfer Protocol (TFTP) is an Internet software utility for transferring files that is simpler to use than the File Transfer Protocol (FTP) but less capable. It is used where user authentication and directory visibility are not required. For example, it is used by Cisco routers and switches to transfer images of the operating system from/to the devices.

NOTE: TFTP's port is 69 (a system/OS port). This is below 1024 and therefore you cannot bind a Socket to it unless you have administrative access rights on the machine you are working. To avoid any complications you can use any port > 1024 to bind server sockets.

Task 1: Implementation of the Trivial File Transfer Protocol (TFTP)

For this task you need to implement (in Java) the Trivial File Transfer Protocol (TFTP) as specified in [RFC 1350](#). You will submit source code for a client and server application that 'speak' the TFTP protocol. You will build your protocol on top of UDP. Compared to the specifications in the RFC, you will implement a slightly simplified version:

- **Support for octet mode only.** The files should be transferred as a raw sequence of bytes. Do not read, write or transfer files as characters.
- **Support** only for error handling when the server is unable to satisfy the request because the file cannot be found.
- **No support for error handling when data duplication occurs.**

The client and server applications should be simple Java console applications. The server should operate (i.e. read and write files) in the directory where it is started from. The server should support simultaneous file transfers to and from multiple clients. The client should just read command line arguments (or have a very simple console-based menu - e.g. "press 1 to store file, press 2 to retrieve file") and execute user commands (i.e. reading or writing a file).

Hint: the simplest way to implement timeouts is by calling the `setSoTimeout()` method on the `DatagramSocket` objects (assuming that you are using blocking I/O). If the timeout expires, a `java.net.SocketTimeoutException` is raised, though the `DatagramSocket` is still valid.

Task 2: Implementation of an even simpler version of TFTP on top of TCP

For this task you will use TCP sockets to implement a protocol that operates like TFTP (i.e. supports only read and write operations). Given that TCP supports in-order, reliable data transport, you should not implement the relevant mechanisms described in RFC 1350 (ACKs, retransmissions). The client and server applications should be equally simple, as in Task 1. The server must be able to handle multiple file transfers.