

**TRIVIAL FILE TRANSFER PROTOCOL**

**TFTP\_SRS\_Document-v0.1**

**1. INTRODUCTION………………………………………………………………………4**

1.1PURPOSE…………………………………………………………………………….…4

1.2 SCOPE…………………………………………………………………………………..4

1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS…………………………….4

**2. BRIEF DESCRIPTION OF REQUIREMENTS……………………………………...4**

2.1. TFTP\_01………………………………………………………………………………..4

2.2. TFTP\_02………………………………………………………………………………..4

2.3. TFTP\_03………………………………………………………………………………..5

2.4. TFTP\_04………………………………………………………………………………..5

2.5. TFTP\_05………………………………………………………………………………..5

2.6. TFTP\_06………………………………………………………………………………..5

2.7. TFTP\_07………………………………………………………………………………..6

2.8. TFTP\_08………………………………………………………………………………..6

2.9. TFTP\_09………………………………………………………………………………..7

2.10. TFTP\_10……………………………………………………………………….…..….7

2.11. TFTP\_11……………………………………………………………………..…….….7

2.12. TFTP\_12………………………………………………………………………………7

2.13. TFTP\_13………………………………………………………………………………8

**3. REQUIREMENTS………………………………………………………………………8**

3.1. FUNCTIONALITY…………………………………………………………………….8

3.1.1. FEATURE TO UPLOAD AND DOWNLOAD FILE FROM/TO THE SERVER….8

3.1.2. FEATURE TO HANDLE MULTIPLE CLIENTS REQUEST……………………….9

3.1.3. APPROPRIATE ERROR HANDLING………………………………………………9

3.1.4. DATA SECURITY……………………………………………………………………9

3.1.5. RETRANSMISSION OF THE LOST PACKET……………………………………..9

3.2. USABILITY…………………………………………………………………………….9

3.3. RELIABILITY AND AVAILABILITY…………………………………………….….9

3.4. SECURITY………………………………………………………………………….…..9

3.5. SUPPORTABILITY……………………………………………………………….……9

3.6. DESIGN CONSTRAINTS………………………………………………………….…..10

3.7. USE CASE DIAGRAM…………………………………………………………………10

3.8. ON-LINE USER DOCUMENTATION AND HELP SYSTEM REQUIREMENTS…..10

3.9. PURCHASED COMPONENTS…………………………………………………….…..10

3.10. INTERFACE…………………………………………………………………….……..10

3.11. LICENSING REQUIREMENTS………………………………………………………10

3.12. LEGAL, COPYRIGHT AND OTHER NOTICES…………………………………….10

3.13. APPLICABLE STANDARDS…………………………………………………………10

**4. SUPPORTING SYSTEM………………………………………………………………..11**

**1.Introduction: -**

The introduction of the software Requirements Specification (SRS) provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations, references, and overview of the SRS. The aim of this document is to gather, analyse and give an in-depth insight into the complete TFTP by defining all the requirements in detail. The intended audience includes developers, testers, project managers and the client. The detailed requirement of TFTP is provided in this document.

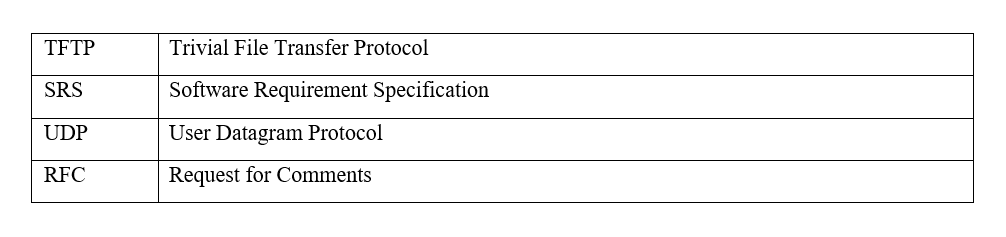
**1.1 Purpose: -**

Purpose of this document is to describe the requirement to provide file transferring through the UDP (network) between server and client. It aims at uploading or downloading a file from the TFTP server according to the request of the TFTP client.

**1.2 Scope: -**

This project provides implementation of TFTP Service (UDP based) involving a client and a server. The server handles the client's request and performs several operations (downloading and uploading of the file). Server may optionally handle multiple client’s requests simultaneously.

**1.3 Definitions, Acronyms, and Abbreviations: -**



**2. Brief description on requirements: -**

**2.1 TFTP\_01**   **The implementation will follow RFC 1350 (i.e., ftp using UDP)**

This (RFC 1350) Trivial File Transfer Protocol. As mentioned, TFTP is designed to be implemented on top of the Datagram protocol (UDP).  Since Datagram is implemented on the Internet protocol, packets will have an Internet header, a Datagram header, and a TFTP header.

**2.2 TFTP\_02 The client and server shall be configurable application which can run over network**

An application that runs on the client side and server side, accessing the server for information is called a client/server application by two-way communication. So, client and server shall configure applications which can run over the network.

**2.3 TFTP\_03** **Client should be able to download a file from the server.**

In this requirement, if a client has requested a server for file or any data server will ACK with priority of the data which sends Acknowledgement message and send the file to the Client.

**2.4 TFTP\_04** **Server should be able to provide upload file to client**

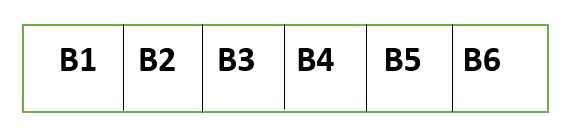
Server-side code, the server cannot send a file to client if the client didn't request it. It also provides to upload file to server which reduce issues, such as performance, and effort, and speed up your uploads.

**2.5 TFTP\_05** **Data transfer will be done in chunks of block size of 512 bytes. A data packet of less than 512 bytes signals termination of a transfer.**

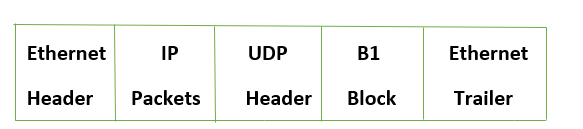
The client sends block number, containing 176 bytes of data. It waits for an acknowledgment before terminating the logical connection. The server receives a block number and sends an acknowledgment for it. Since this data message had fewer than 512 bytes, the transfer was done.

**2.6 TFTP\_06 - Each data packet contains one block of data, and must be acknowledged by an acknowledgement packet before the next packet can be sent**

If the client wants to download the data or a file from the server, at first the server divides the data into several blocks and each block is encapsulated with the UDP datagram and IP packet before sending.



Server divides the data into several blocks



        Encapsulation of block

Now, when the TFTP client receives block 1 [B1] it will generate an acknowledgment that the block is received.



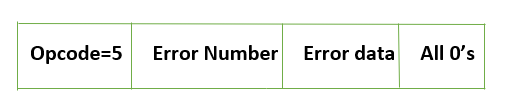
    Acknowledgment block

**2.7 TFTP\_07 – If a packet gets lost in the network, the intended recipient will timeout and may retransmit his last packet (which may be data or an acknowledgment), thus causing the sender of the lost packet to retransmit- that lost packet.**

If a data packet gets lost in the network, then the intended server or receiver will timeout (usually waits for 5 seconds before presuming that the transmitted packet has been lost) and sends an acknowledgment, thus causing the sender to retransmit that lost packet.

We can also specify a different timeout period in seconds.

**2.8 TFTP\_08 – Should handle error cases namely file not found, access violation and packet loss.**



      Error message format

Error Number for various error cases,

0 – Not defined

1 – File not found

2 – Access Violation

3 – Disk full or quota exceeded

4 – Illegal operation

5 – Unknown port Number

6 – File already exists

7 – No such user

**2.9 TFTP\_09 – File transfers will be restricted to a single configured directory on server**

For the purpose of security, the file transfer between multiple directories is restricted and the files outside the directory are not accessible.

**2.10 TFTP\_10- Server should maintain a log of all upload requests from client**

Server should listen to the client request and save the log of each request from the client. If the client raises a message as any issues, the server should notice the issues and display it in .log format.

**2.11 TFTP\_11- Server will be an iterative server to handle multiple clients**

UDP basic server-client model, one server attends only one client at a particular time, but the thing is server should be iterative.so we can achieve this iterative server process using **select () system call** method.

**2.12 TFTP\_12-Standard error codes to be used to indicate specific error cases as in RFC.**

RFC is one among the HTTP resources and specifications.

|  |  |  |
| --- | --- | --- |
| **Specification** | **Title** | **Status** |
|  |  |  |
| RFC 7233 | Hypertext Transfer Protocol (HTTP/1.1): Range Requests | Proposed Standard |

The standard error codes like error no are always present to indicate the errors .

**2.13 TFTP\_13-"Should include debug log messages with at least 4 levels (FATA, INFO, WARNING, DEBUG)**

Logging is essential to understand the behaviour of the application and to debug unexpected issues or for simply tracking events.

The program should satisfy the below log messages which is to specify that if the program runs without any errors.

**fatal:** Severe errors that cause premature termination. Expect these to be immediately visible on a status console.

**warn**:  
Use of deprecated APIs, poor use of API, 'almost' errors, other runtime situations that are undesirable or unexpected, but not necessarily "wrong". Expect these to be immediately visible on a status console.

**Info**:   
Interesting runtime events (start-up/shutdown). Expect these to be immediately visible on a console, so be conservative and keep to a minimum.

**debug**:  
Detailed information on the flow through the system. Expect these to be written to logs only.

**3.**    **Specific Requirements**

 The specific requirements are –

**3.1 Functionality**

**3.1.1Feature to download and upload files from/to the server**

  The TFTP server allows the client to download files from the server and upload the files to the server.

**3.1.2 Feature to handle multiple client’s request**

  The system shall allow the server to handle multiple client’s requests.

**3.1.3 Appropriate Error Handling**

  For all search fails or other system fails proper Error Messages namely file not found, access violation and packet loss etc. will be displayed.

This will include Log messages.

**3.1.4 Data Security**

The system shall allow the server on the remote system to set the restrictions on which files users can retrieve, as well as restrictions on storing files.

**3.1.6 Re-transmission of the lost packet**

If a data packet gets lost in the network, then the intended server or receiver will timeout (usually waits for 5 seconds before presuming that the transmitted packet has been lost) and sends an acknowledgment, thus causing the sender to retransmit that lost packet.

**3.1.4 Data Security**The system shall allow the server on the remote system to set the restrictions on which files users can retrieve, as well as restrictions on storing file**3.1.5 Re-transmission of the lost packet**If a data packet gets lost in the network, then the intended server or receiver will timeout (usually waits for 5 seconds before presuming that the transmitted packet has been lost) and sends an acknowledgment, thus causing the sender to retransmit that lost packet.

**3.2 Usability**

The system is user-friendly by avoiding users from knowing the Linux Command for file transfer between server and users.

Based on the menu – driven or console.

**3.3 Reliability & Availability**

  The server is available when the user is requested for service.

The server is available 24/7 and provides the services to users when requested.

The system has a very low failure rate.

**3.4 Security**

The TFTP servers provides an option whereby only files in a specific directory can be accessed.

**3.5 Supportability**

  The system is easy to maintain.

**3.6 Design Constraints**

  The system is built using only C++ language.

**3.7 Use Case Diagram**

Diagram

Description automatically generated

**3.8 On-line User Documentation and Help System Requirements**

  Internet Connection, Desktop with Linux Terminal.

**3.9 Purchased Components**

NA

**3.10 Interface**

File System Interface

**3.11 Licensing Requirements**

  NA.

**3.12 Legal, Copyright, and Other Notices**

All rights reserved.

**3.13 Applicable Standards**

  It shall be as per the industry standard.

**4.Supporting Information**

  1.Use case diagram