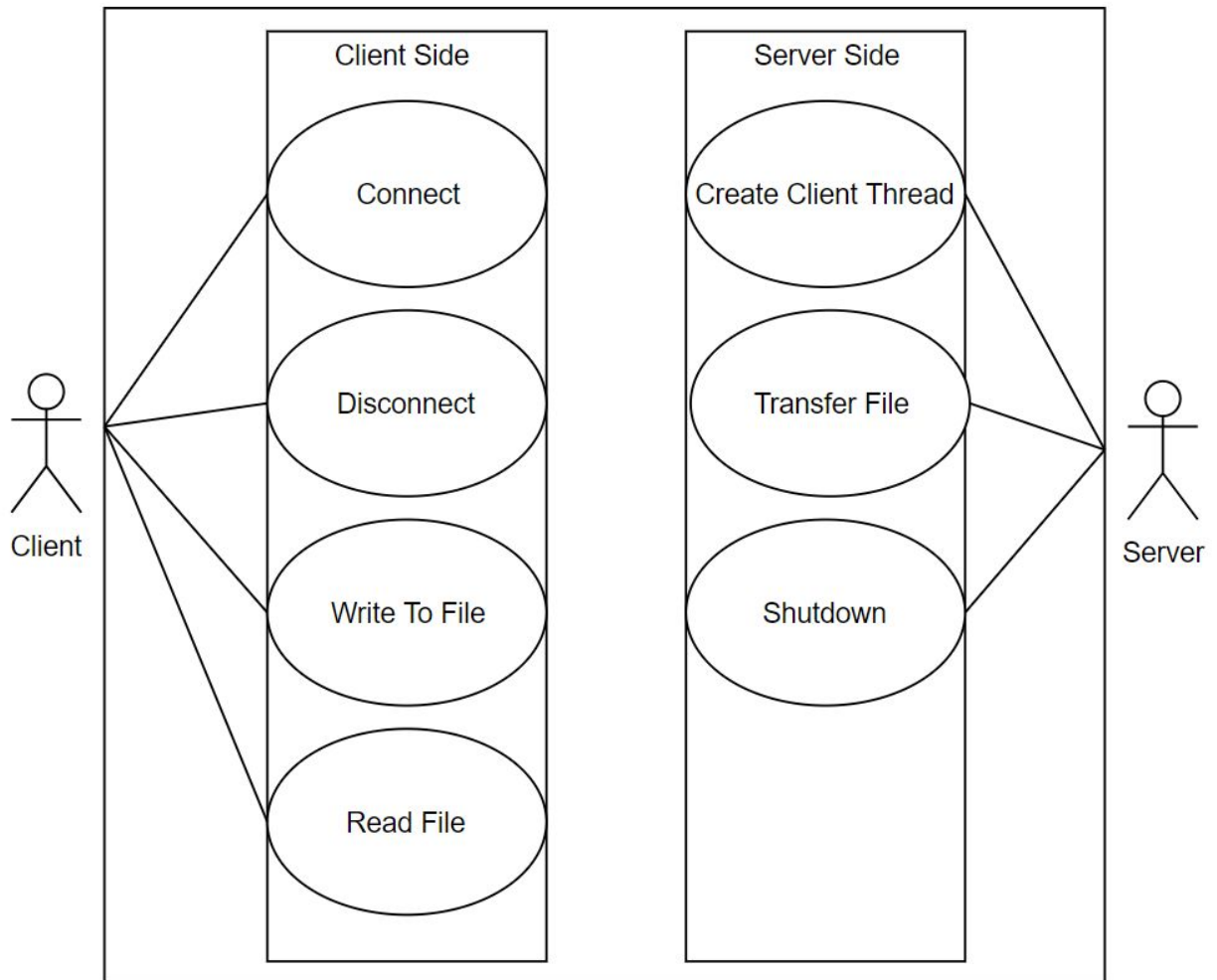


Use Case Diagram



Use Cases

Read File

Description: User requests to read file from server

Actor: User/client

Precondition:

512 byte array must be available for transferring

User must have established a connection to the server

Assumptions: File must exist

Main Sequence:

1. Client sends a read request with file name to the server
2. Server checks validity of request

3. If request is valid, server sends back the data packet
4. If client receives a data packet and it is a valid packet, client sends ACK packet (corresponding to the data packet block number received) to server
5. If server receives the ACK packet, server sends the next data packet
6. Repeat step 4 and 5 until the entire file is read

Post Condition: Entire file must be read in

Alternate Sequence:

- 3a. If invalid request or illegal TFTP operation, send error packet with error code 4 to client, then disconnect from client.
- 4a. If packet received is an illegal operation, send error packet with error code 4 to host, then disconnect from server.
- 4b. If source TID of packet is unknown, discard the packet, and send an ERROR packet with error code 5
- 5a. If source TID of packet is unknown, host discards the packet, and sends an ERROR packet with error code 5
- 5b. If packet is an illegal operation, send error packet with error code 4 to host, then disconnect connection.
- 5c. If server does not receive ACK packet during transfer after certain time, the most recent data packet is resent.
- 5d. If server receives duplicate ACK packet, it is ignored.

Write to File

Description: Client requests file for reading from server

Actor: User/Client

Precondition:

User is connected to Server

512 byte array must be available for transferring

Main Sequence:

1. Client sends a write request to the server
2. Server checks validity of request
3. If valid, server sends an ACK packet to the client (with block number 0)
4. User sends a data packet (with up to 512 bytes of data) to the server
5. If the data packet is received by the server and is a valid packet, the server sends an ACK packet back to the client
6. If the client receives an ACK packet, client sends the next data packet to the server.
7. Repeat steps 5 and 6 until the entire file is written

Post Conditions: The entire file must be written to server

Alternate Sequence:

- 3a. If invalid request or illegal TFTP operation, send error packet with error code 4 to client, then disconnect from client

- 4a. If packet received is an illegal operation, send error packet with error code 4 to host, then disconnect from server.
- 4b. If source TID of packet is unknown, discard the packet, and send an ERROR packet with error code
- 6a. If source TID of packet is unknown, host discards the packet, and sends an ERROR packet with error code 5
- 6b. If packet is an illegal operation, send error packet with error code 4 to host, then disconnect connection.
- 6c. If client does not receive ACK packet during transfer after certain time, the most recent data packet is resent.
- 6d. If client receives duplicate ACK packet, it is ignored

Delay Packet

Description: Error Simulator delays the transfer of a specified packet by a set time

Actor: Error Simulator with secondary actors users - server and client

Precondition:

Client and Server are transferring files through the Error Simulator; and
Packet to delay and time to delay is provided by the user.

Main Sequence:

- 1. Error Simulator receives packet during transfer
- 2. Error Simulator checks if packet is the specified packet to delay
- 3. If it is the specified packet, Error Simulator sleeps for the given amount of time (specified delay time)
- 4. Error Simulator continues transfer of packets between server and client

Post Conditions: Transfer continues normally, system recovers from delayed packet

Lose Packet

Description: Error Simulator loses a specified packet during transfer

Actor: Error Simulator with secondary actors users - server and client

Precondition

Client and Server are transferring files through the Error Simulator; and
Packet to be lost is provided by the user.

Assumptions:

Main Sequence:

- 1. Error Simulator receives packet during transfer
- 2. Error Simulator checks if packet is the desired packet to lose.
- 3. If it is the desired packet, Error Simulator skips transfer. In other words, the Error Simulator does not send the packet.
- 4. Error Simulator waits for the next packet
- 5. Either the client or server re-sends the most recent data packet after timeout
- 6. System continues transfer of packets between the client and the server

Post Conditions: Transfer continues normally, system recovers from lost packet

Duplicate Packet

Description: Error Simulator duplicates a specified packet and transfers both to the corresponding actor

Actor: Error Simulator with secondary actors users - server and client

Precondition:

Client and Server are transferring files through the Error Simulator; and
Packet to be duplicated is provided by the user.

Main Sequence:

1. Error Simulator receives packet during transfer
2. Error Simulator checks if packet is the desired packet to duplicate
3. If it is the desired packet, Error Simulator creates a duplicate packet
4. Error Simulator passes both packets to corresponding Actor (one after the other)
5. Either Client or Server receives duplicate packets
6. If duplicate packets received by the secondary actor are data packets, the duplicated (second) data packet received is discarded and the ACK packet is resent.
7. System continues transfer of packets between the client and the server

Alternate Sequence:

- 6a. If duplicate packets received by the secondary actor are ACK packets, the duplicate packet (second packet received) is discarded, no packets are sent back.

Post Conditions: Transfer continues normally, duplicate packet is handled

Corrupt a Packet

Description: Error Simulator corrupts a packet by creating an invalid packet or a valid packet with a different TID

Actor: Error Simulator with secondary actors users - server and client

Precondition:

Client and Server are transferring files through the Error Simulator;
Type of error simulation is provided; and
Packet to be corrupted is provided by the user.

Main Sequence:

1. Error Simulator receives packet during transfer
2. Error Simulator checks if packet is the desired packet to corrupt
3. If it is the desired packet and error code 4 simulation is selected, the Error Simulator modifies the received packet to one that contains an invalid opcode.
4. Error Simulator passes the invalid packet to corresponding secondary actor
5. Either Client or Server receives the invalid packet
6. The secondary actor that receives the packet, discards the packet, and creates an error packet to send back to the Error Simulator.

7. If the error code 4 simulation mode was selected, both secondary actors terminate the connection. One of the secondary actors terminates after receiving the error packet.

Alternate Sequence:

3a. If it is the desired packet and error code 5 simulation is selected, the Error Simulator

creates a new client thread that sends a packet to the same port of the current server port.

7a. If the error code 5 simulation mode was selected, the transfer is continued normally after the other secondary actor receives the error packet.

Post Conditions: Transfer is stopped (if error code 4 was simulated) or continues to operate normally (if error code 5 was simulated).