ECEN 602 Assignment 3 Test Cases

Following are the use cases tested:

1. Transfer binary file of 2048 bytes

The clients connects to the server with port number as 1234 and requests for 2048bin file. The server sends data in UDP format in terms of blocks each having 512 bytes.

Server

```
guest@gautham:-/Networks/ECEN-602-Network_Programming/Programming_Assignment_3/bin$ is 2048bin server guest@gautham:-/Networks/ECEN-602-Network_Programming/Programming_Assignment_3/bin$ ./server 127.0.0.1 1234 TFFP server is running now. Listenting on: 1234 Got a new connection from 127.0.0.1':55551! File 2048bin in mode octet will be read by 127.0.0.1:55551! Data transfer done successfully for 127.0.0.1:55551!
```

```
server * client * guest@gautham:-/clients diff 2048bin -/Networks/ECEN-602-Network_Programming/Programming_Assignment_3/bin/2048bin guest@gautham:-/clients
```

2. Transfer binary file of 2047 bytes

The clients connects to the server with port number as 1234 and requests for 2047bin file. The server sends data in UDP format in terms of blocks each having 512 bytes.

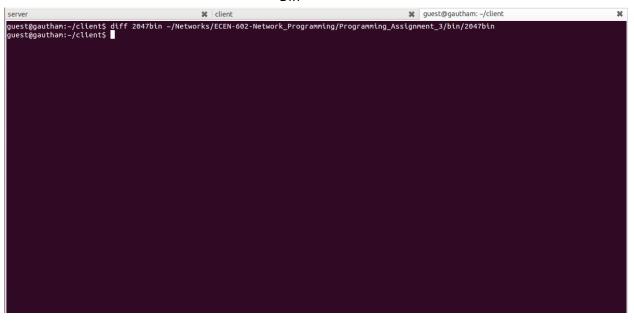
Server

```
guest@gauthan:-/Networks/ECEN-602-Network_Programming/Programming_Assignment_3/bin$ is
2048bin server
guest@gauthan:-/Networks/ECEN-602-Network_Programming/Programming_Assignment_3/bin$ is
2048bin server
guest@gauthan:-/Networks/ECEN-602-Network_Programming/Programming_Assignment_3/bin$ ./server 127.0.0.1 1234
TFIP server is running now. Listening on: 1234
Got a new connection from 127.0.0.1':59551!
Pile 2048bin in mode octet will be read by 127.0.0.1:59551!
Got a new connection from 127.0.0.1':59538!
File 2047bin in mode octet will be read by 127.0.0.1:59538!
Data transfer done successfully for 127.0.0.1:59538!
Data transfer done successfully for 127.0.0.1:59538!
```

Client

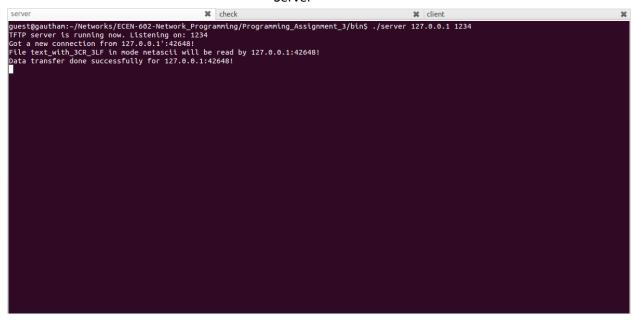
```
Server & client & guest@gautham:-/client & fftp>
fftp>
fftp>
fftp>
fftp-
```

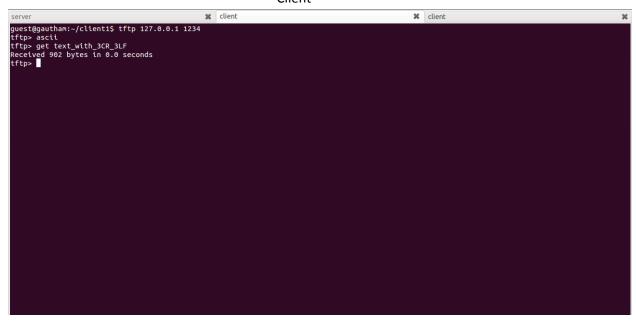
Diff



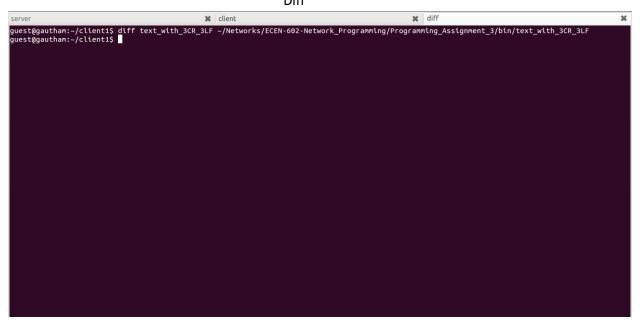
3. Transfer of ascii file that includes 3 CRs

Server



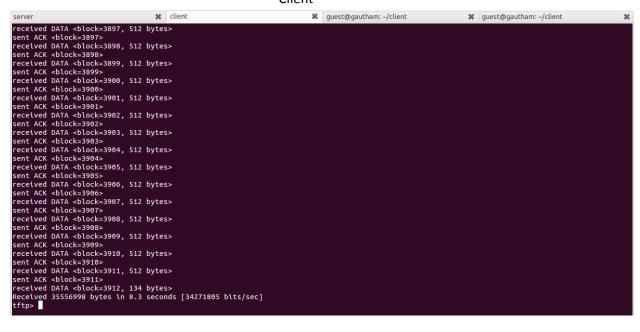


Diff



4. Transfer of 34MB file and check if wrap around occurs Client asks for 34MB file to the server. Since server can send maximum of 65535 blocks, it wraps around sending from the first block.

Server



Diff

```
server % | client % | guest@gautham:-/client % | guest@gautham:-/client % | guest@gautham:-/client | %
```

Wrap around

5. Error message when file is not present

Server sends an error code to the client when the file requested is not present.

Server

```
server % client % guest@gautham:-/client % gue
```

6. Connect to TFTP server with 3 clients

3 clients connect to TFTP server and request 34MB file simultaneously.

Server

```
server % client % client % client % client % client % guest@gautham:-/Net... % guest@gautham:-/Networks/ECEN-602-Network Programming/Programming_Assignment_3/bin$ ./server 127.0.0.1 1234

IFTP server is running now. Listening on: 1234

Got a new connection from 127.0.0.1':35533!

File 34_MB_file3.txt in mode netascti will be read by 127.0.0.1:35533!

Got a new connection from 127.0.0.1':48783!

Got a new connection from 127.0.0.1':48783!

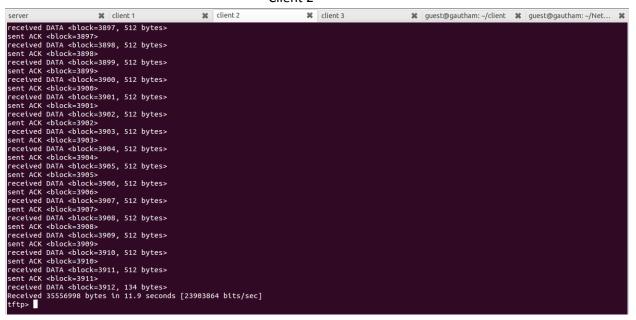
File 34_MB_file2.txt in mode netascti will be read by 127.0.0.1:48783!

Data transfer done successfully for 127.0.0.1:52831!

Data transfer done successfully for 127.0.0.1:52831!

Data transfer done successfully for 127.0.0.1:48783!
```

Client 2



Terminate TFTP client in the middle of transfer
 When the client is terminated in between the transfer, the server waits for 10 seconds and displays transfer session has failed, indicating the client connection is lost.

Server

```
guest@gautham:-/Networks/ECEN-602-Network, Programming/Programming_Assignment_3/bin$ ./server 127.0.0.1 1234
TFFP server is running now. Listenting on: 1234
Got a new connection from 127.0.0.1':55412!
File 34.MB, file.txt in mode netascit will be read by 127.0.0.1:55412!
Out a transfer done successfully for 127.0.0.1':57652!
File and MB, file.txt in mode netascit will be read by 127.0.0.1:59062!
File 34.MB, file.txt in mode netascit will be read by 127.0.0.1:59062!
no response session from 127.0.0.1:59062!
Transfer session has failed
```

8. WRQ Bonus feature

Clients request WRQ to the server. The server sends ack to that command. Later, the client starts sending data to the server.

Server

```
guest@gautham:-/client1$ ftp

ftp: connect 127.0.0.1 1234

ftp: verbose
Verbose node on.

ftp: ftp: race
Packet tracing on.

ftp: put.txt
putting put.txt to 127.0.0.1:put.txt [netascil]
sent MRQ file=put.txt, node=netascis
received Ack oblock=0;
sent DATA oblock=1; S12 bytes>
received Ack oblock=2; S12 bytes>
received Ack oblock=3; S12 bytes>
received Ack oblock=4; S12 bytes>
received Ack oblock=5; S12 bytes>
received Ack oblock=5; S12 bytes>
received Ack oblock=6; S12 bytes>
received Ack oblock=7; S12 bytes>
received Ack oblock=8; S12 bytes>
received Ack oblock=9; S12 bytes>
received Ack oblock=8; S12 bytes>
received Ack oblock=9; S12 bytes>
r
```

9. 2 Put, 1 Get

In this test case, two clients send data to the server and third client receives data from server simultaneously.

Server

```
guest@gautham:-/Networks/ECEN-602-Network_Programming/Programming_Assignment_3/bin$ ./server 127.0.0.1 1234
TFTP server is running now. Listening on: 1234
Got a new connection from 127.0.0.1'.45316!
File 34_MB_put_file.txt in mode netascit will be read by 127.0.0.1:45316!
Got a new connection from 127.0.0.1':3436!
File 34_MB_put_file.txt in mode netascit will be written by 127.0.0.1:34536!
Got a new connection from 127.0.0.1':51058!
Joata transfer done successfully for 127.0.0.1:51058!
Data transfer done successfully for 127.0.0.1:45316!
Data transfer done successfully for 127.0.0.1:4536!

I a server a server done successfully for 127.0.0.1:34536!

I a server a server done successfully for 127.0.0.1:34536!

I a server a server done successfully for 127.0.0.1:34536!
```



Client 2