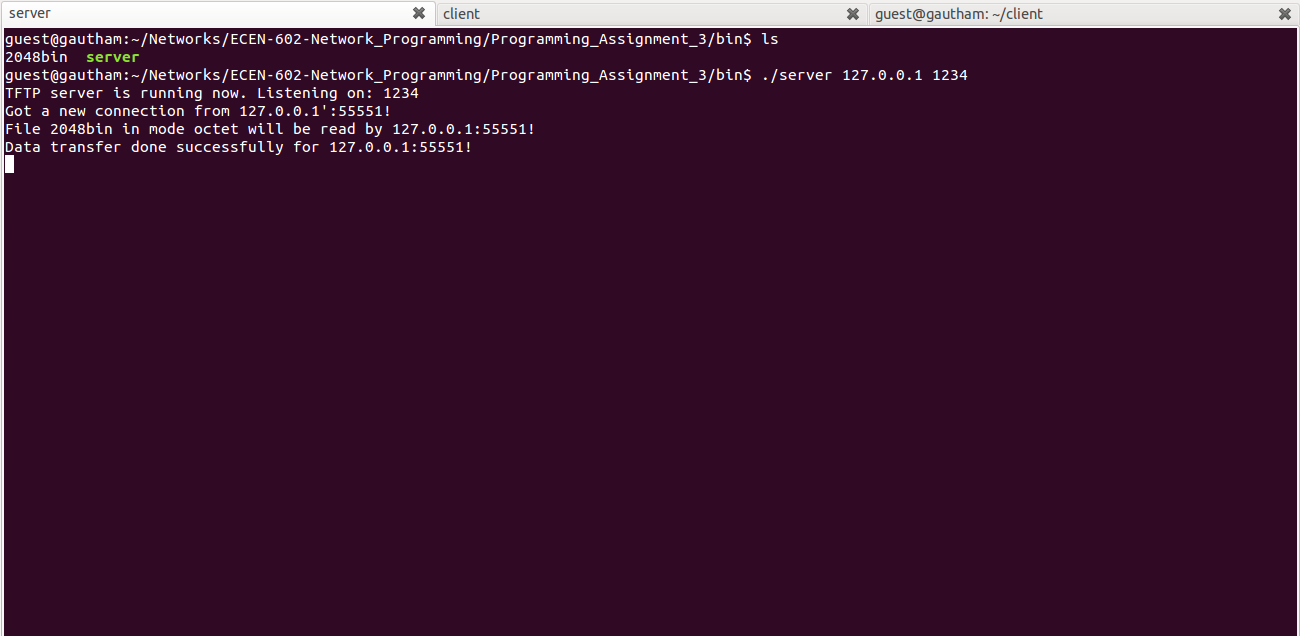
**ECEN 602 Assignment 3 Test Cases**

Following are the use cases tested:

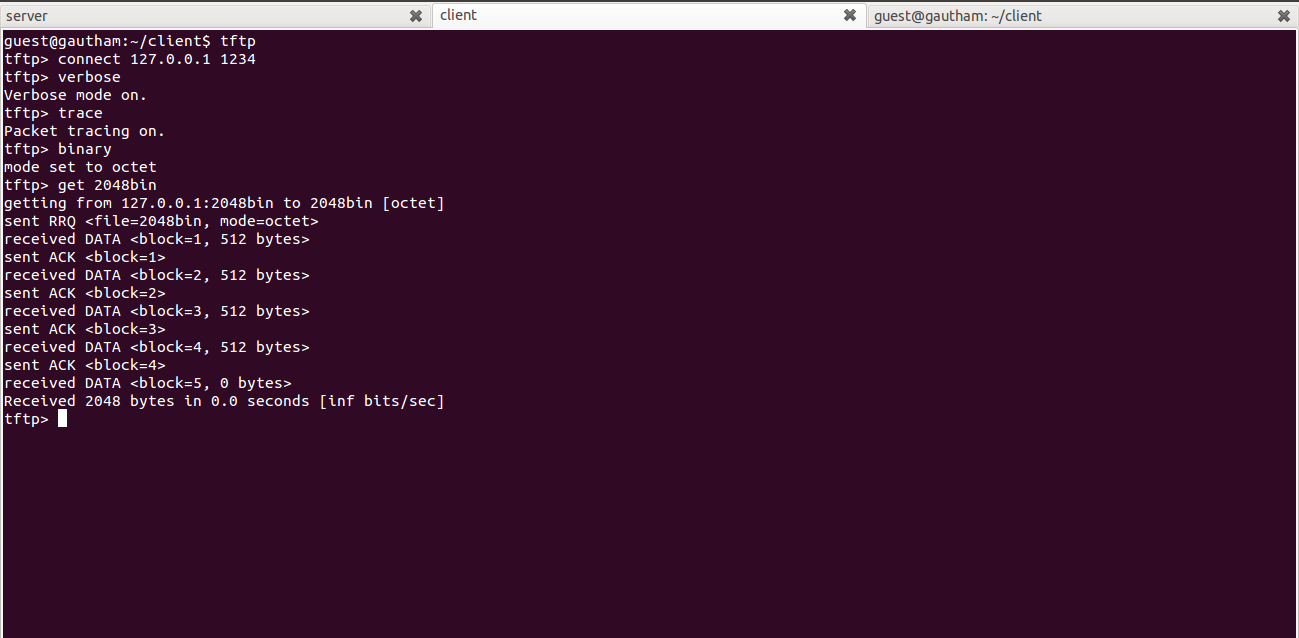
1. Transfer binary file of 2048 bytes

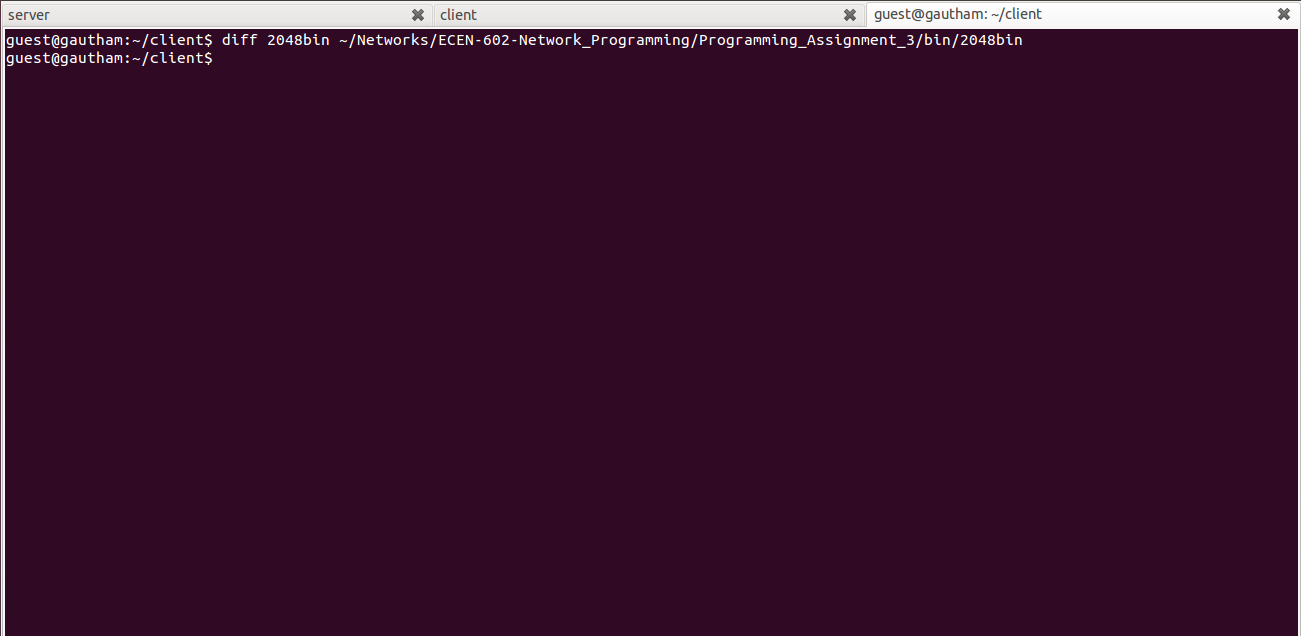
The clients connects to the server with port number as 1234 and requests for 2048bin file. The server sends data in UDP format in terms of blocks each having 512 bytes.

Server



Client

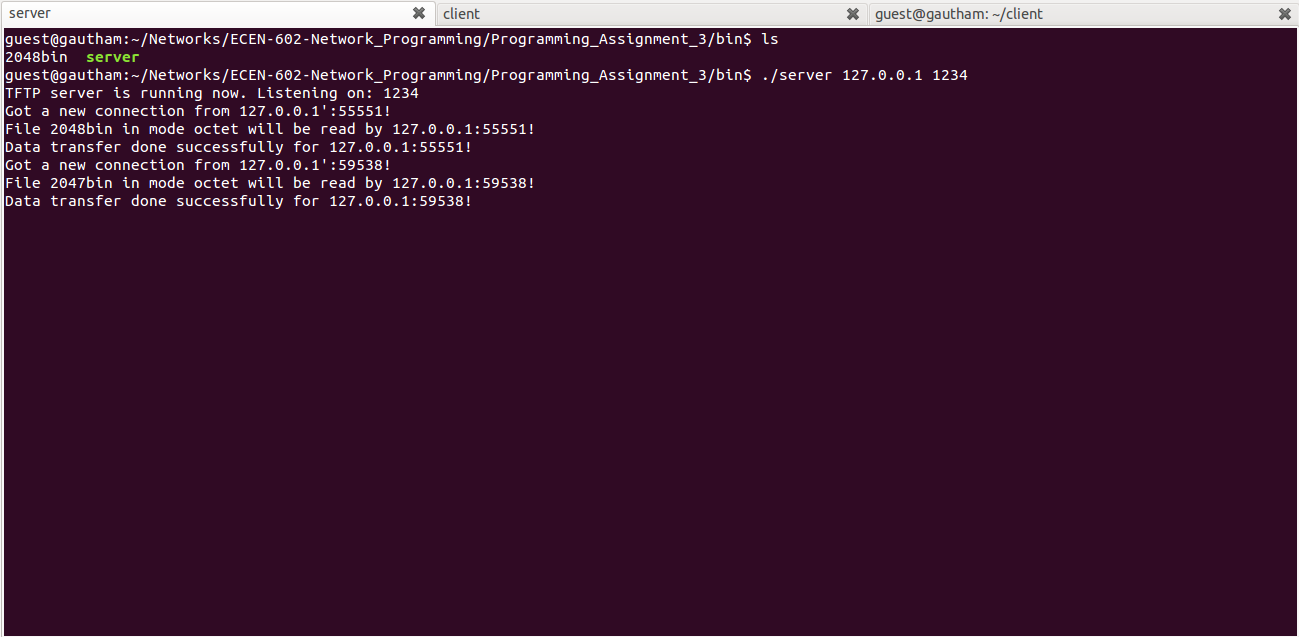


diff

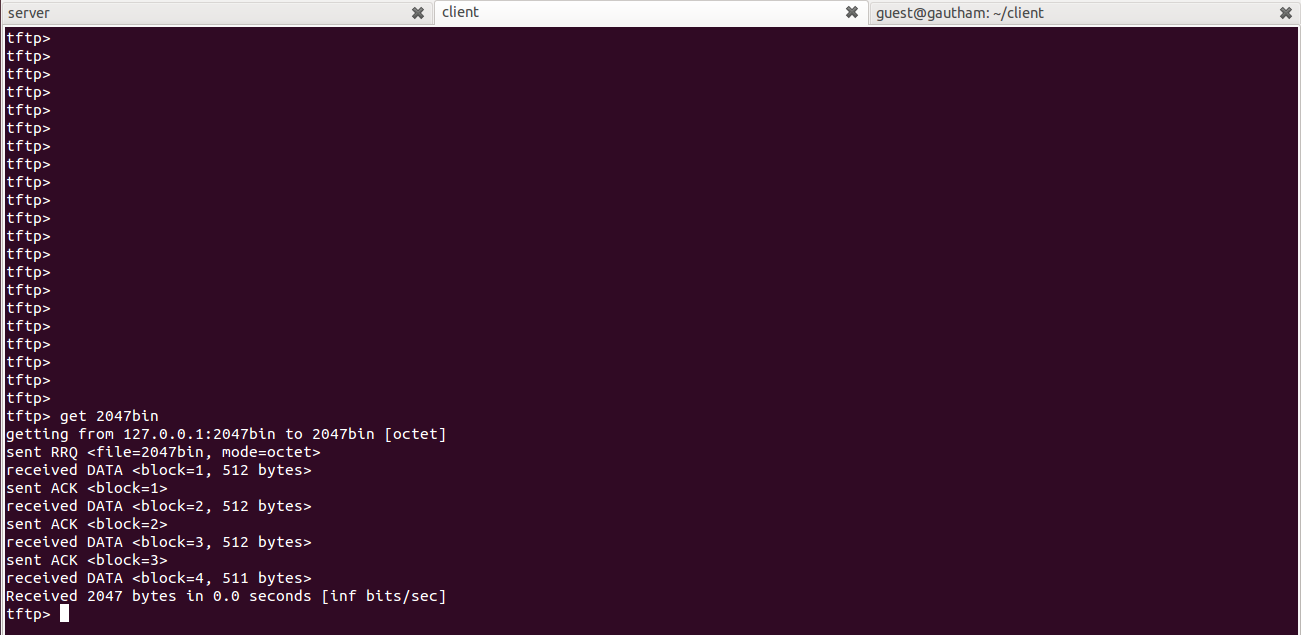
1. Transfer binary file of 2047 bytes

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.

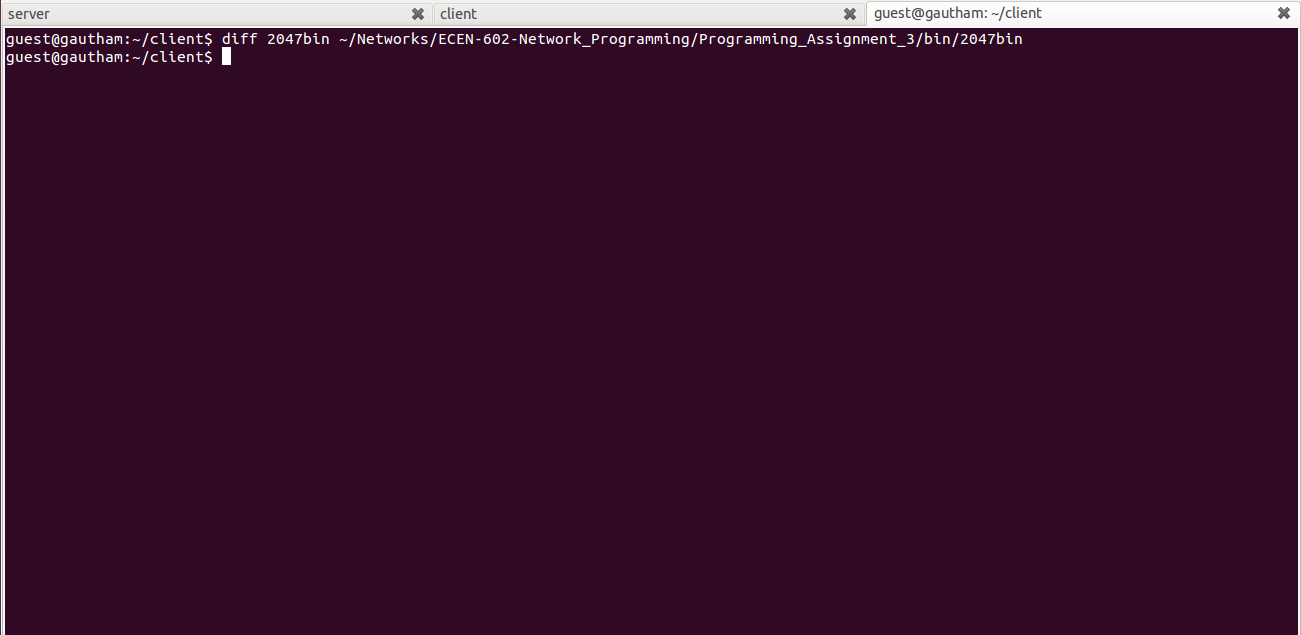
Server



Client

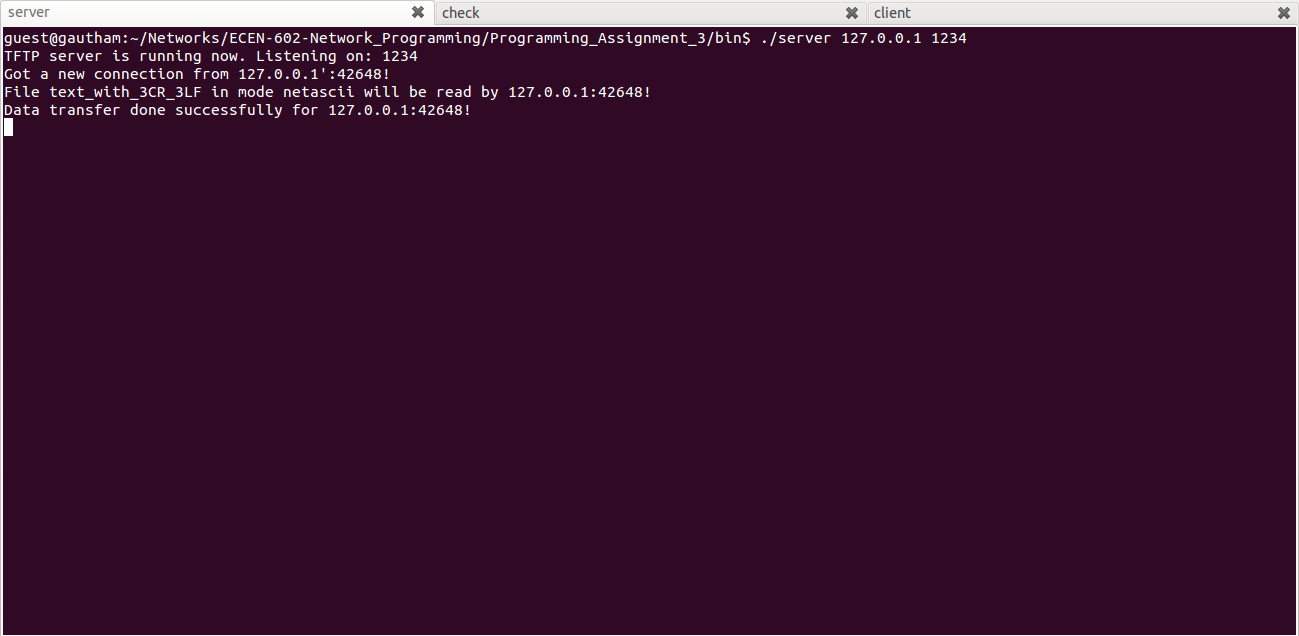


Diff



1. Transfer of ascii file that includes 3 CRs

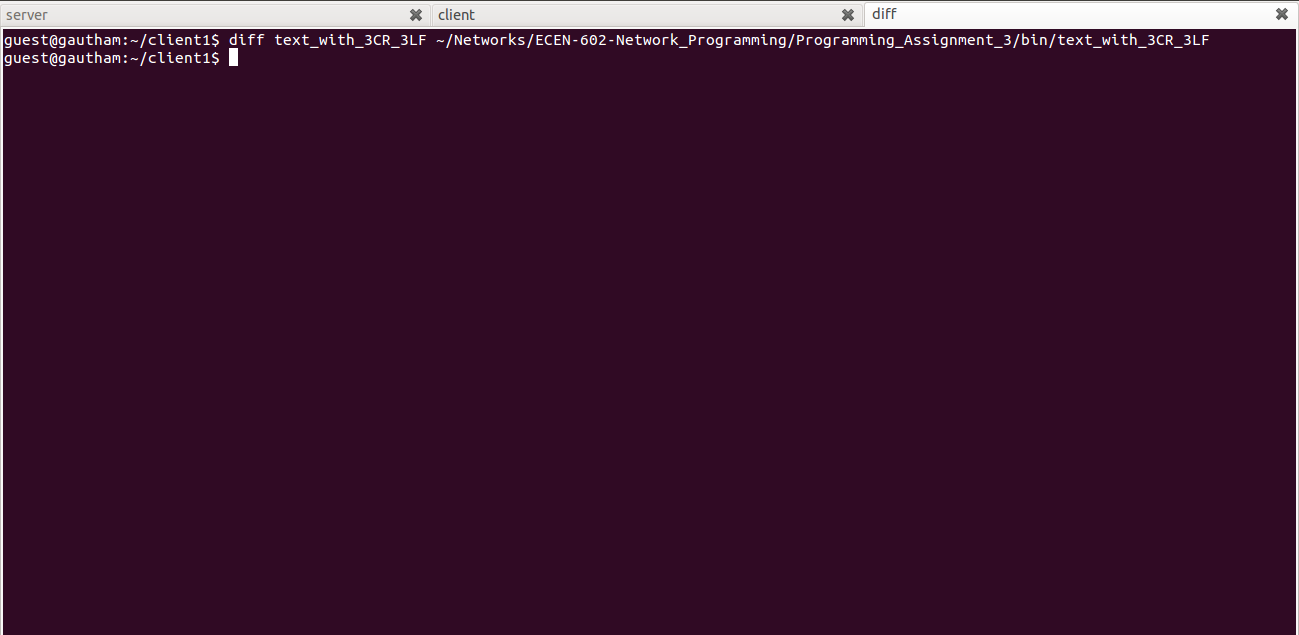
Server



Client



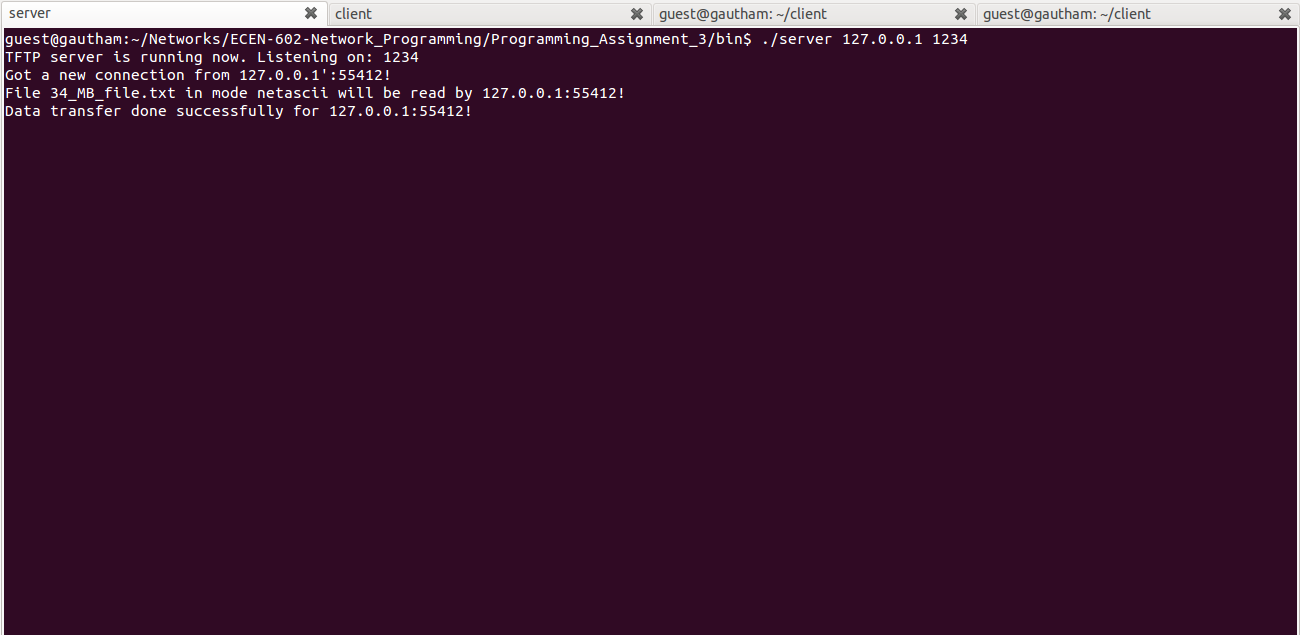
Diff



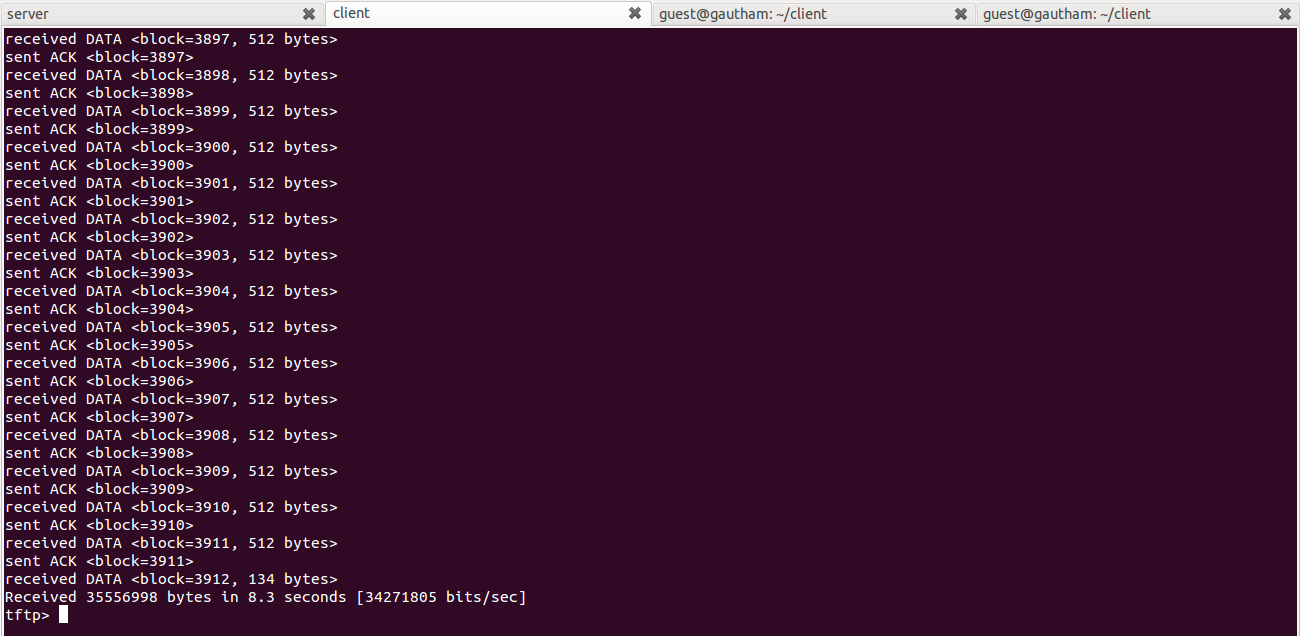
1. Transfer of 34MB file and check if wrap around occurs

Client asks for 34MB file to the server. Since server can send maximum of 65535 blocks, it wraps around sending from the first block.

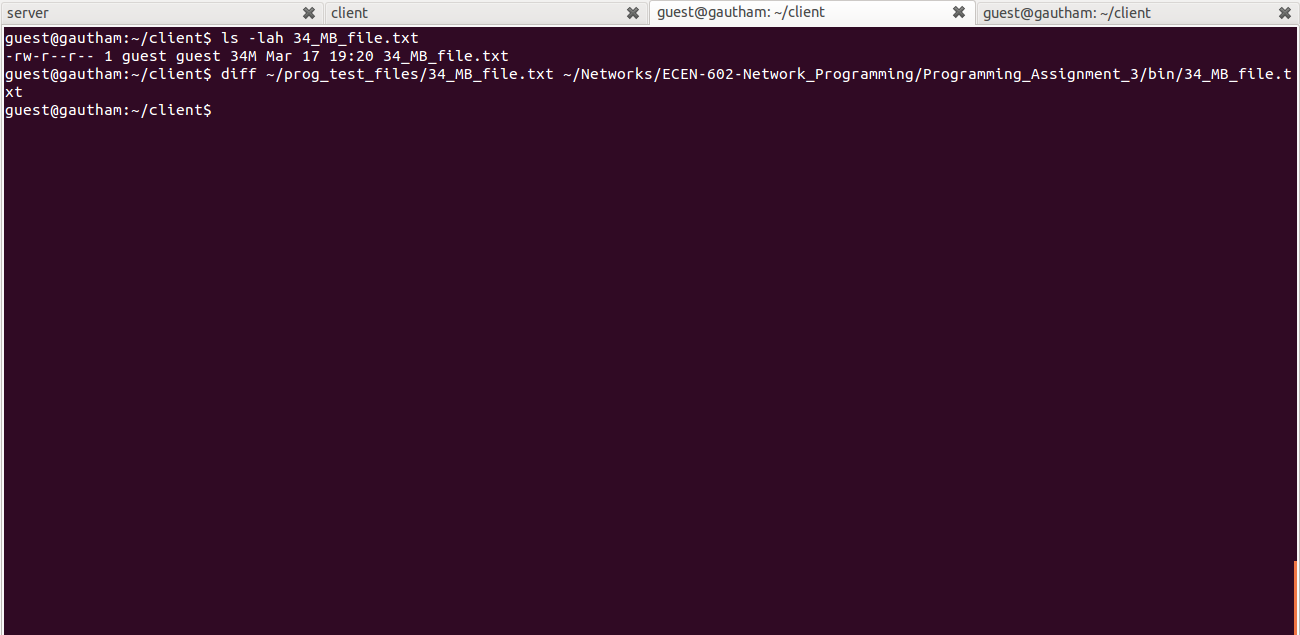
Server



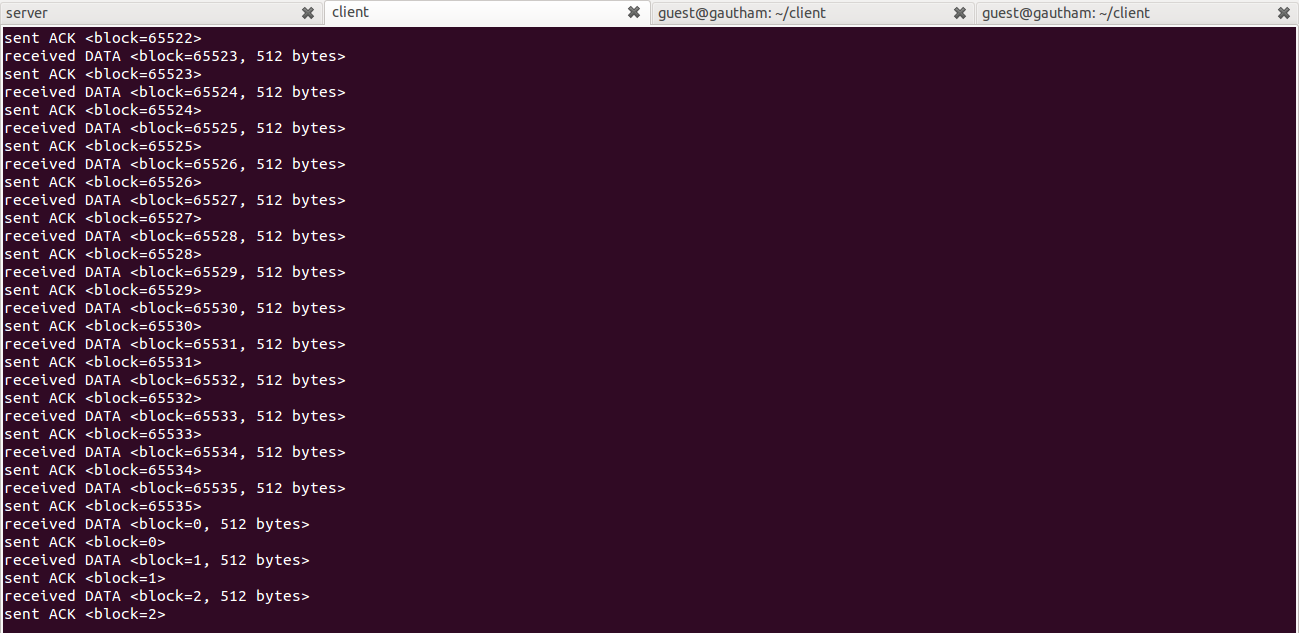
Client



Diff



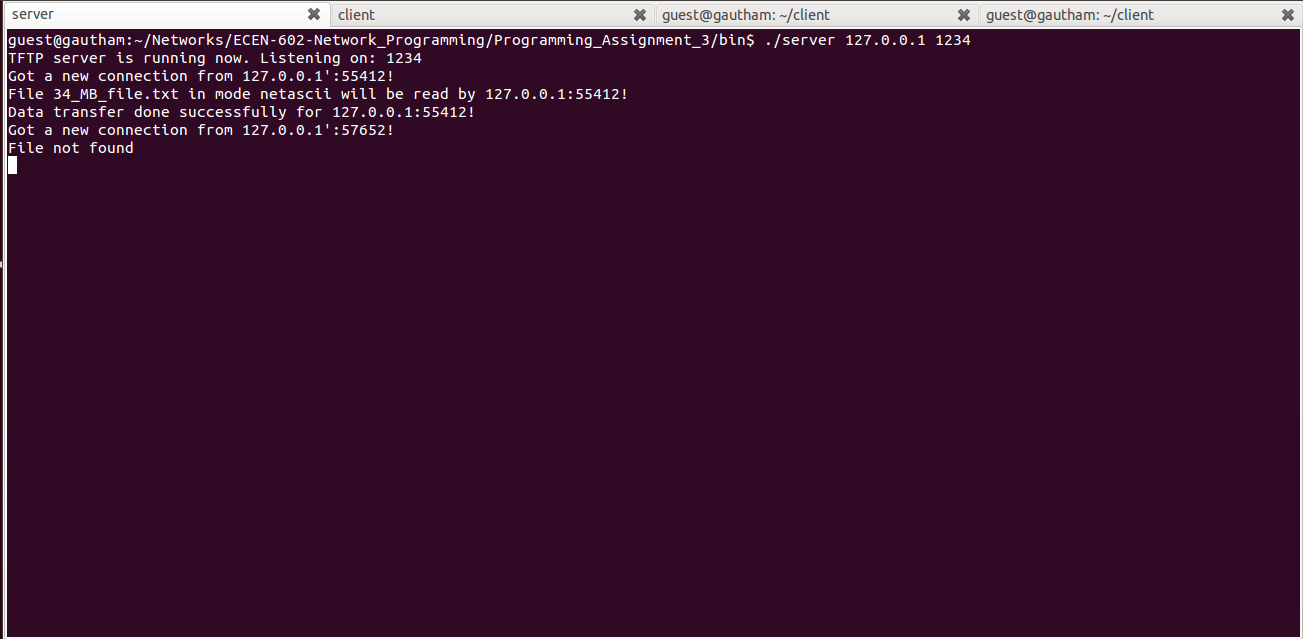
Wrap around



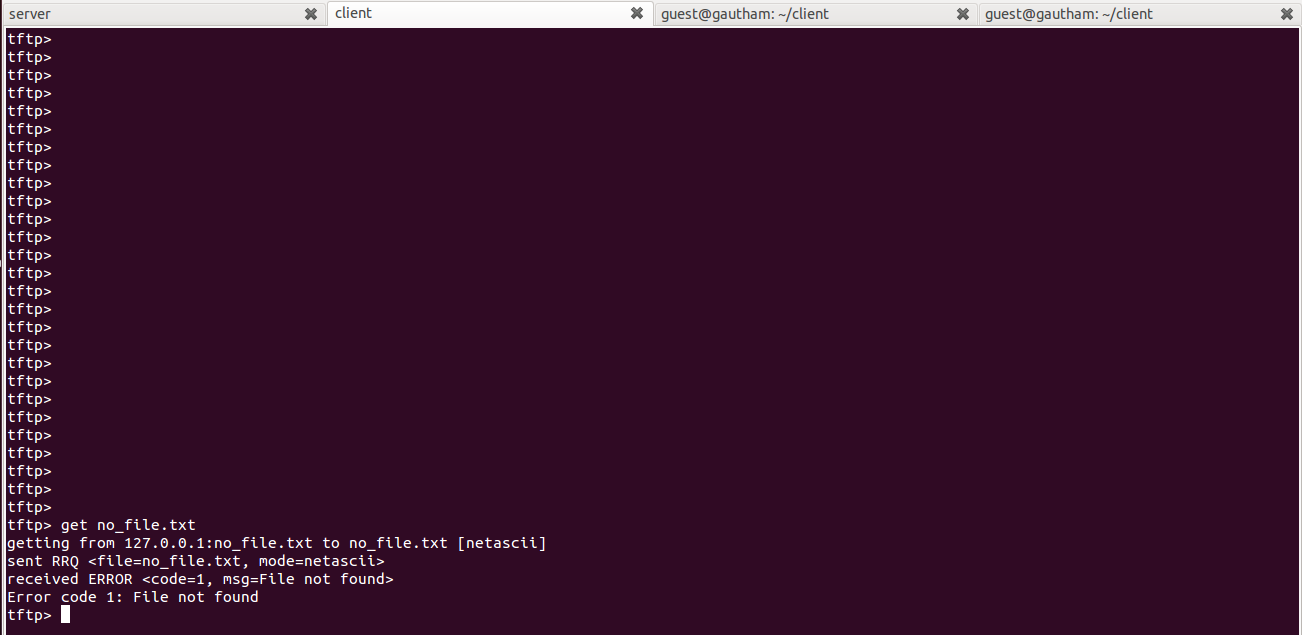
1. Error message when file is not present

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

Server



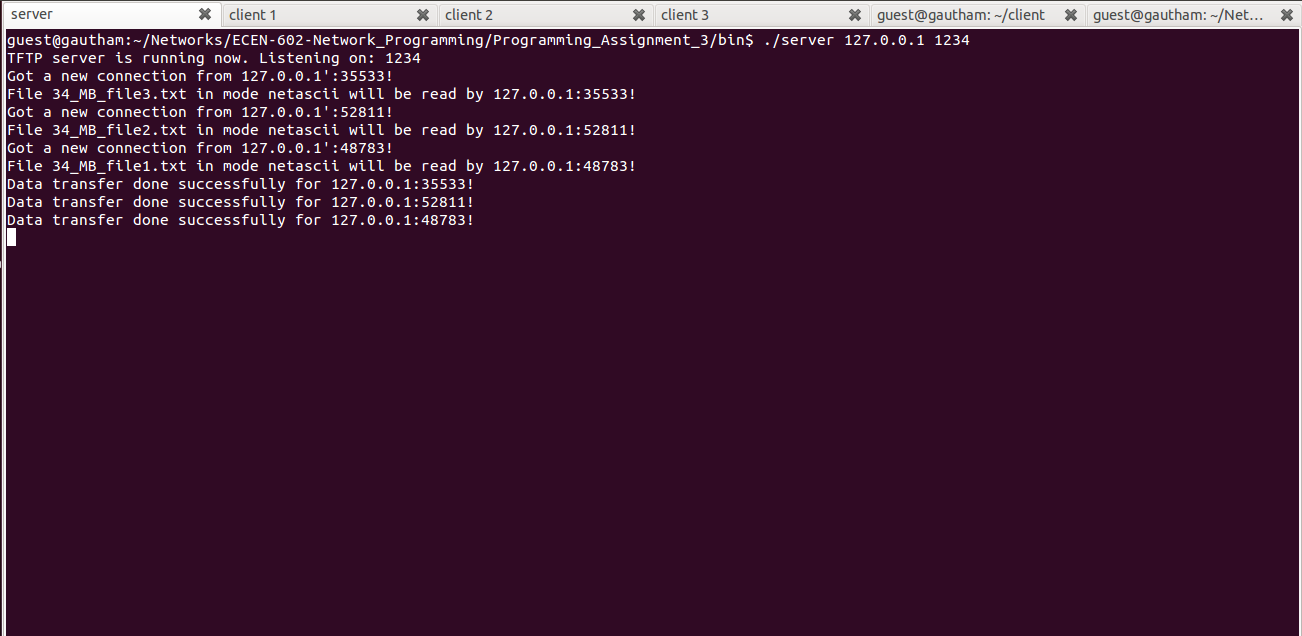
Client



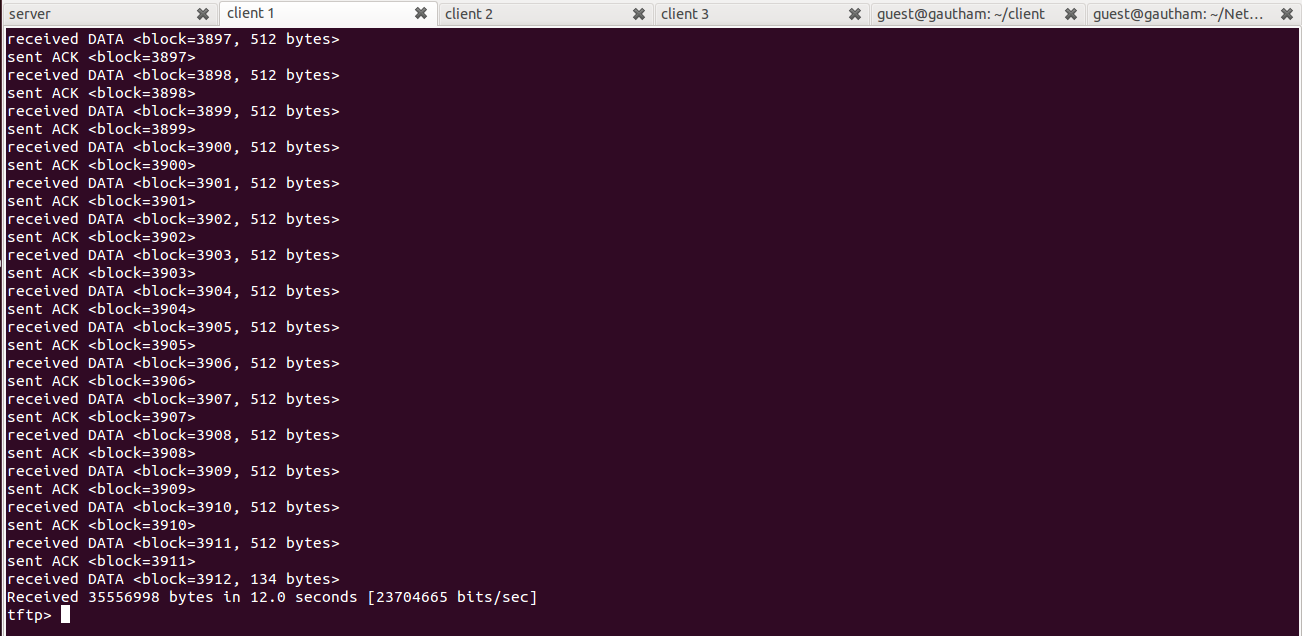
1. Connect to TFTP server with 3 clients

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

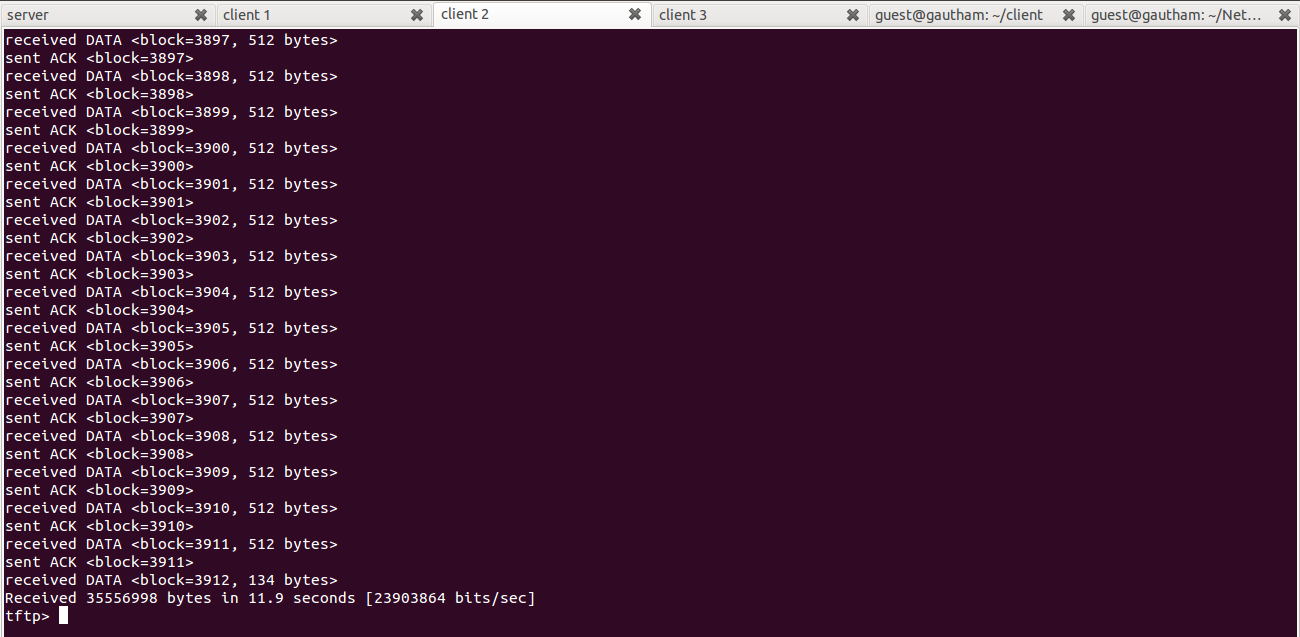
Server



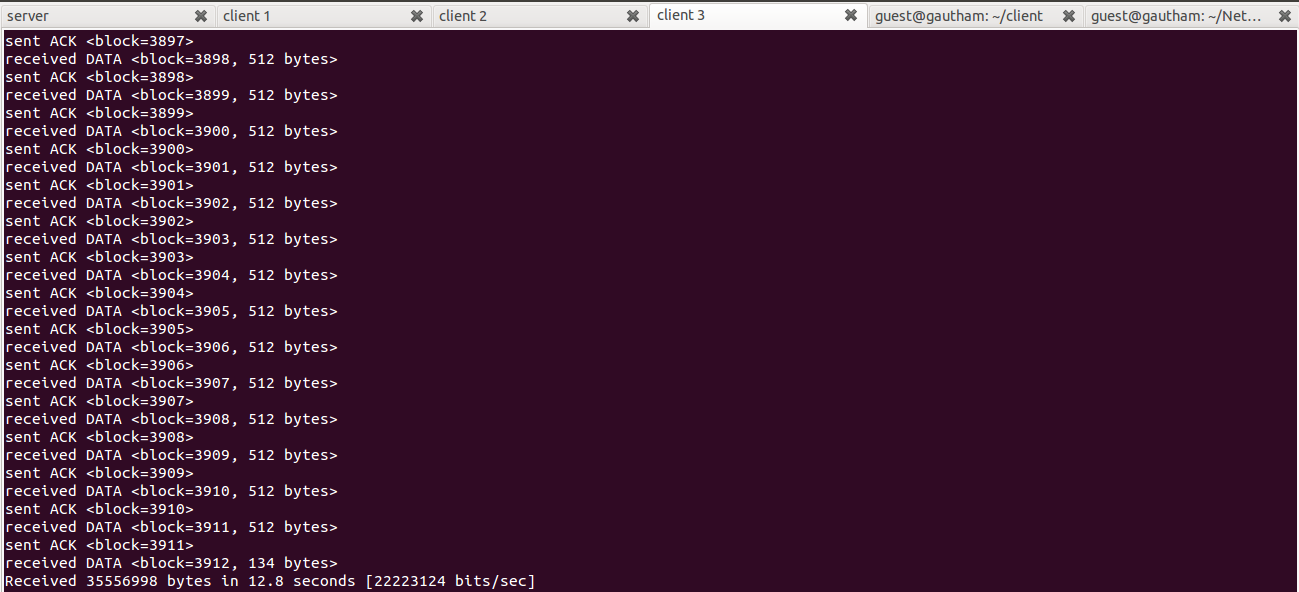
Client 1



Client 2



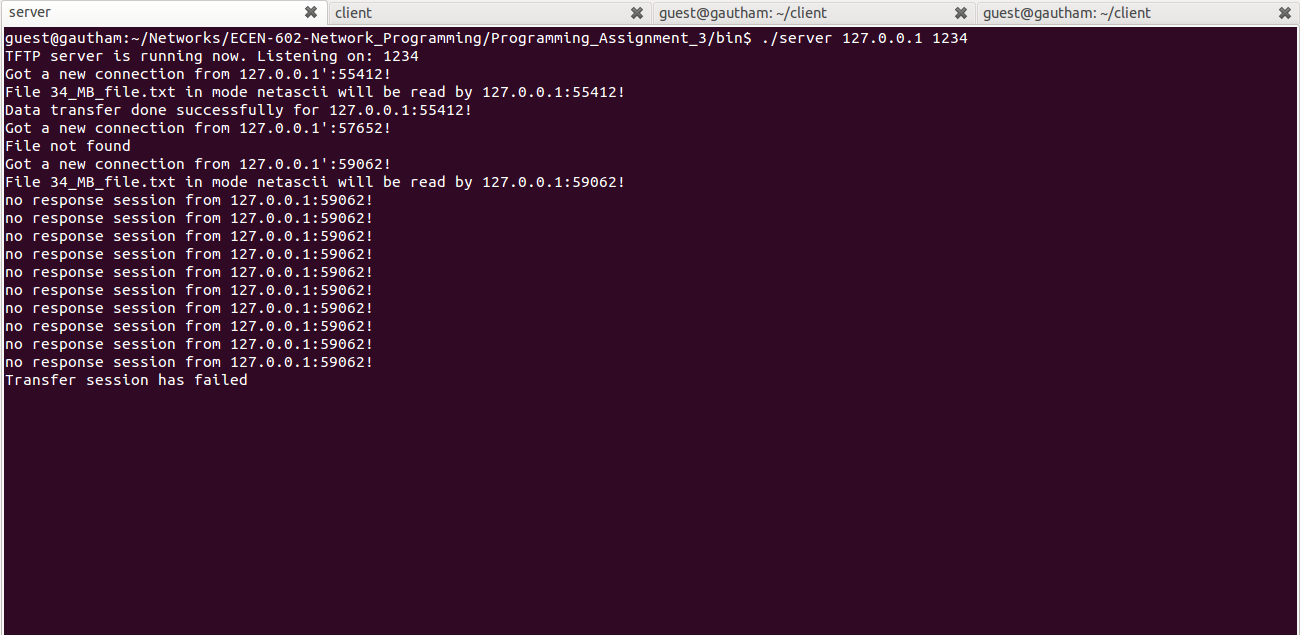
Client 3



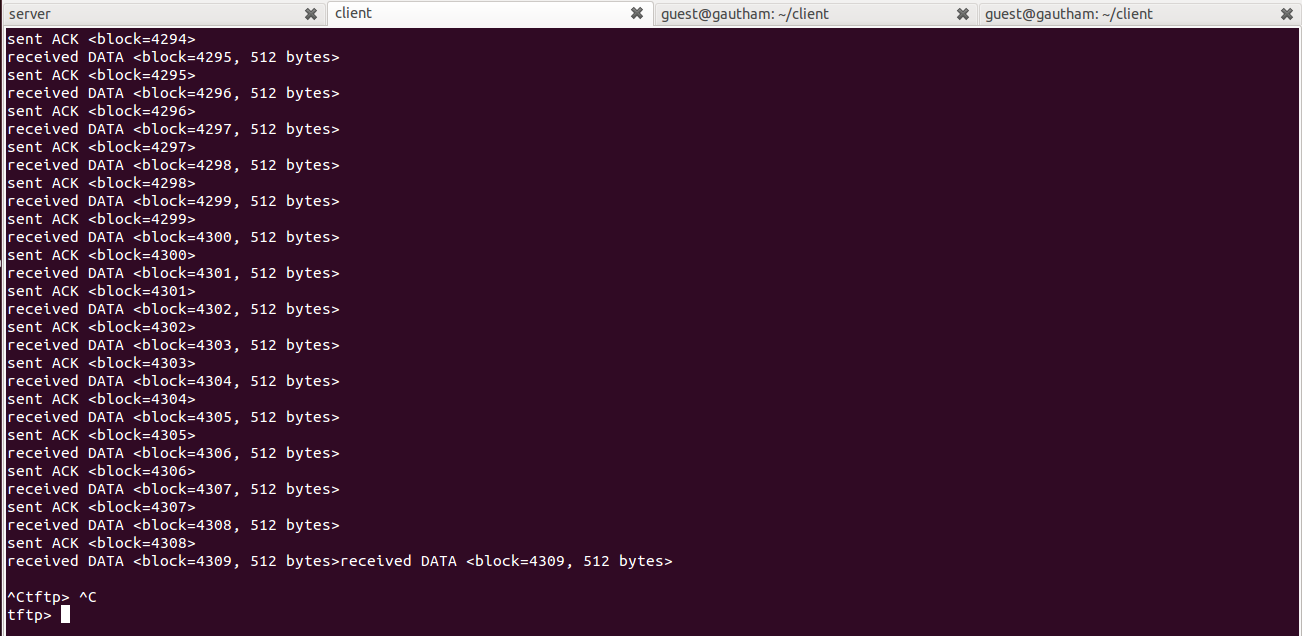
1. Terminate TFTP client in the middle of transfer

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



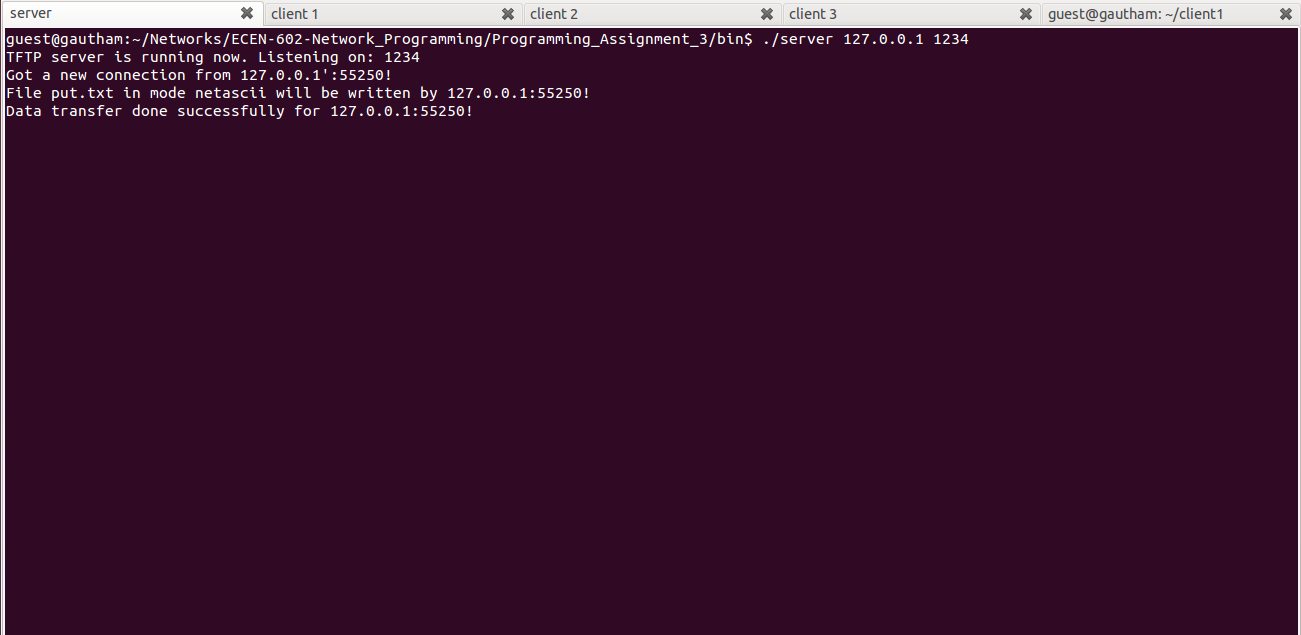
Client



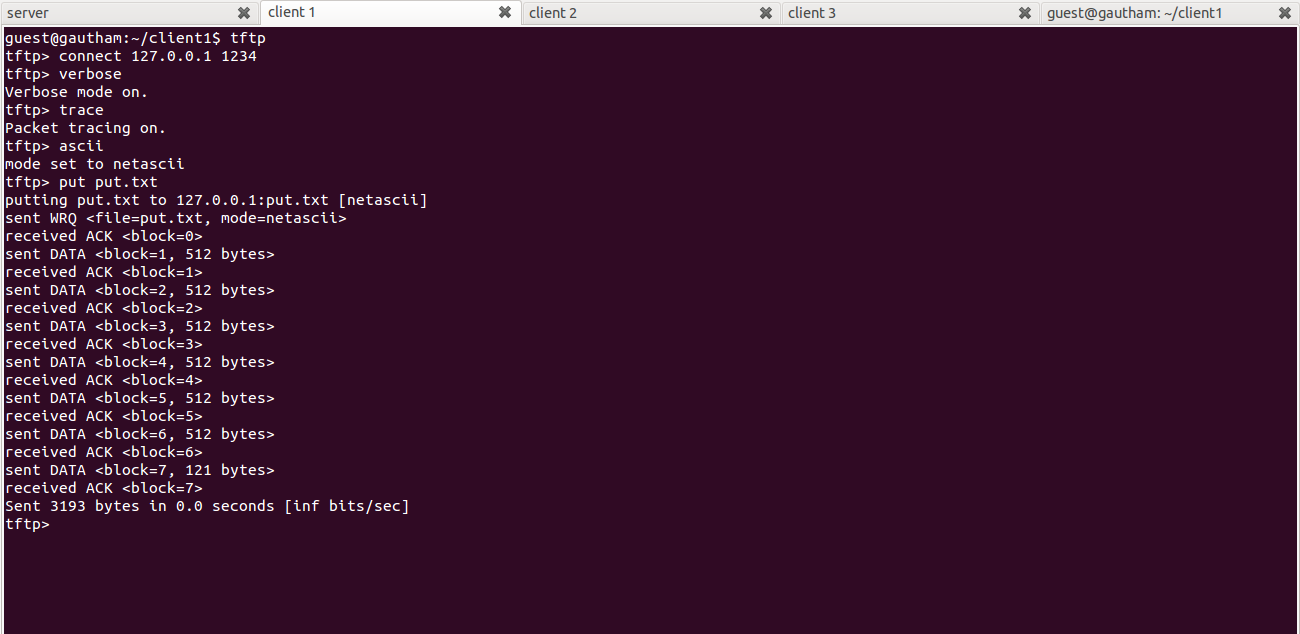
1. WRQ Bonus feature

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

Server



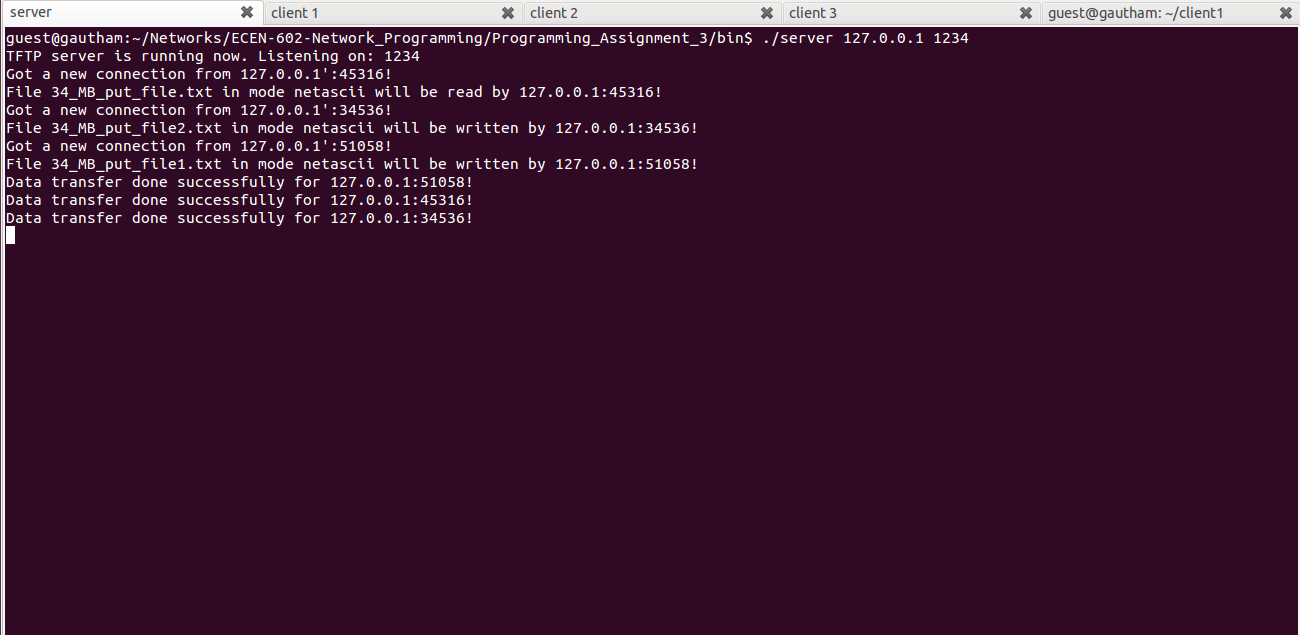
Client



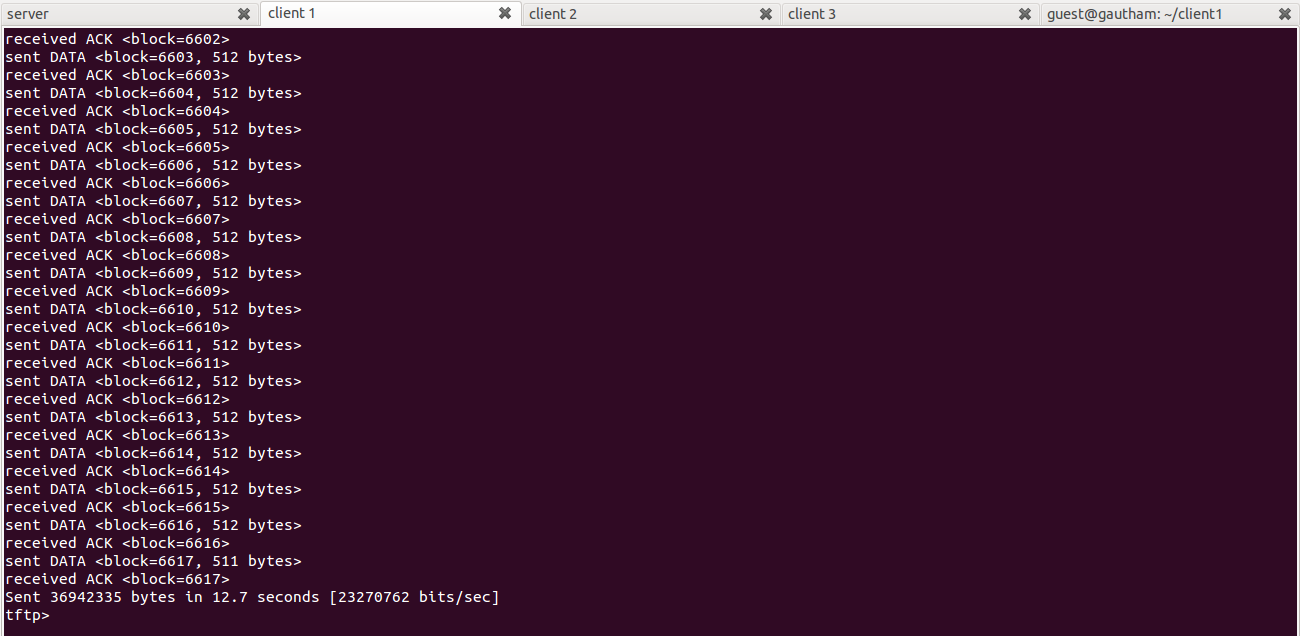
1. 2 Put, 1 Get

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

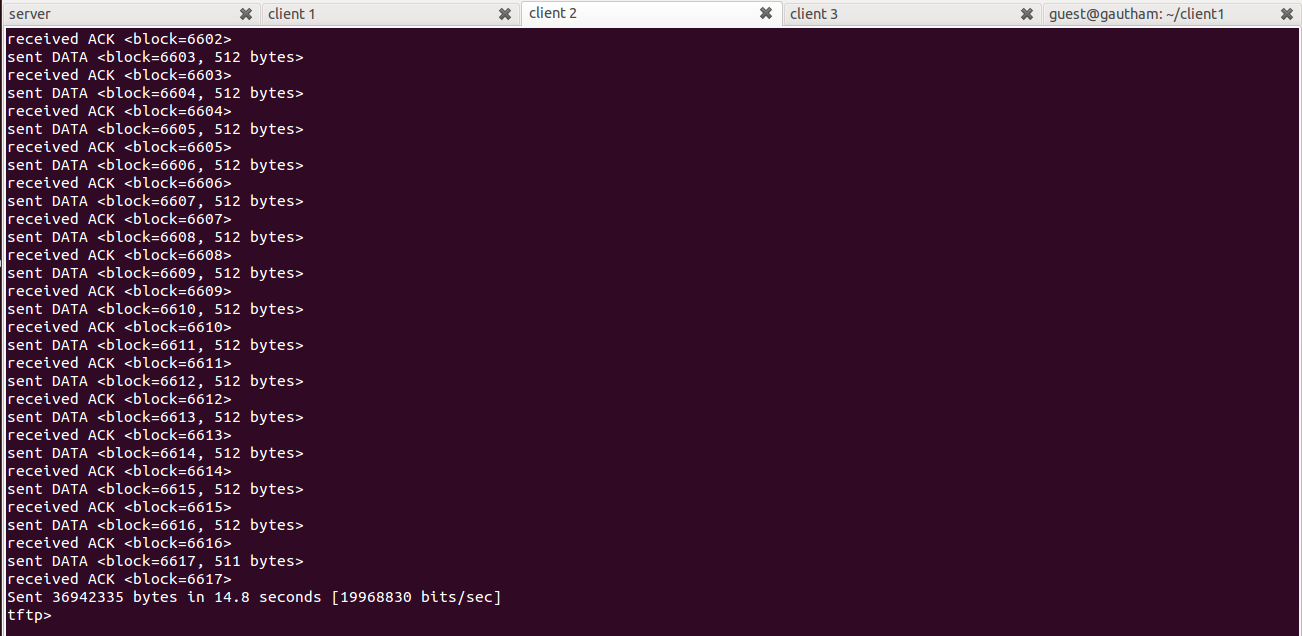
Server



Client1



Client 2



Client 3

