Shubham Chemate
Roll No. 31118

Assigment 08

Title: File transfer using UDP

Problem statement:

Write a program using UDP sockets to enable file transfer (script, text, audio & video file) between two machines.

Demonstrate the packet captured using wireshark packet analyser tool.

Prerequisite:

- Transport layer, rules & protocols (TCP/UDP)
- Socket programming
- Wireshark packet Analyzer Tool

Learning Objectives

- students will be able to understand socket programming
- Students will be able to design networking applications using UDP protocol.

S/W & H/W requrements:

- Windows 10 64-bit PC
- Wireshark packet analyzer tool
- 8GB RAM, 512 GBSSD, i5-8th gen, Intel 4-core processor

Theory:

Server-client file transfer using UDP client.

Server:

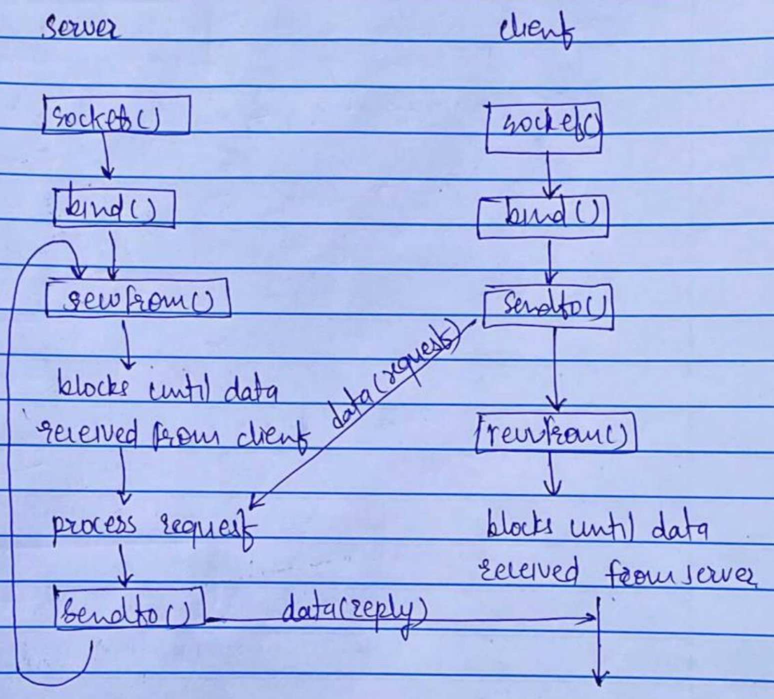
- Include appropriate header files

- create UDP socket using socket() system call
- fill in socket address structure (with server information)
- Specify the port where service will be defined to be used by client
- Bind the address & port using bind() system call
- Recieve a file name of text, audio / video. from client using recvfrom() system call.
- Send file to client using sendto() system call
- close the server socket
- stop.

Client:

- Include appropriate header files
- Create UDP socket
- fill in socket address structure with server information
- specify port of server where it is providing service
- Send file name to server using sendto() system call
- Recieve a file from server using recvfrom() system call
- close the client socket.
- Stop.

Socket functions for UDP client-server in connectionless scenario.



Test cases:

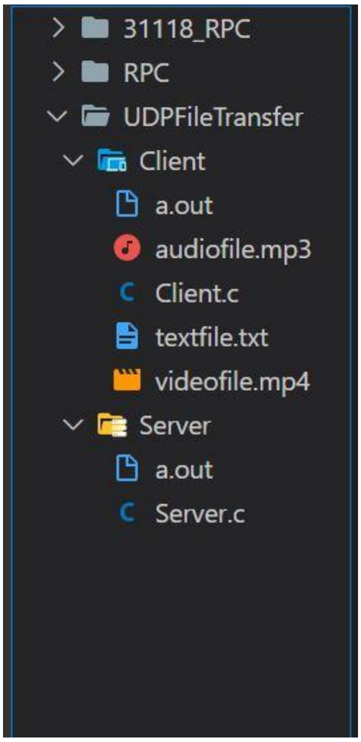
Run wireshark tool

Run client-server program

send & receive file.

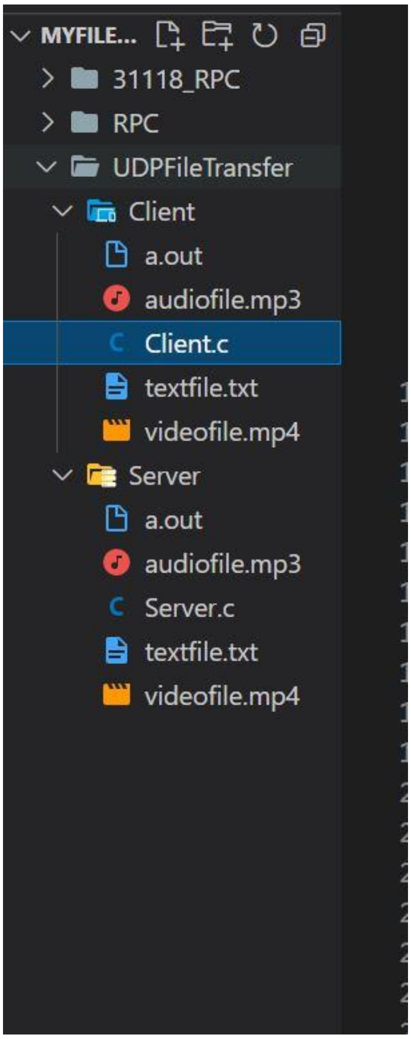
capture UDP packet in wireshark.

Conclusion: Successfully implemented & performed file transfer using UDP socket programming.



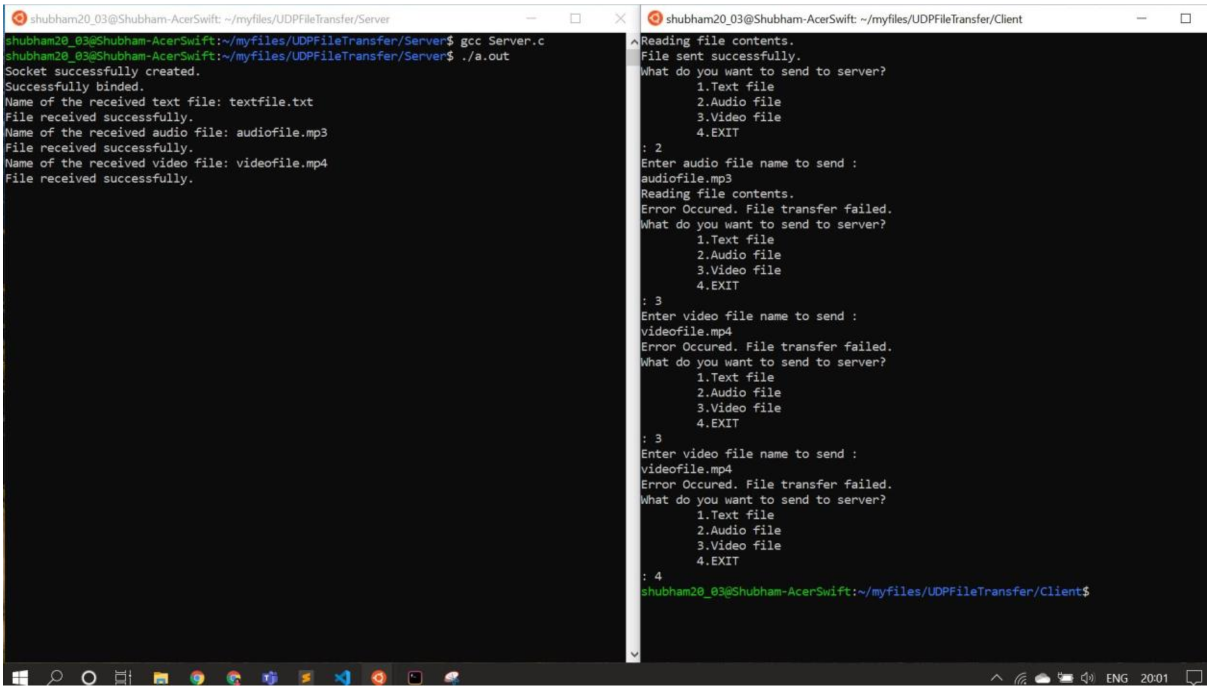
```
> 31118_RPC
> RPC
v UDPFileTransfer
  v Client
      a.out
      audiofile.mp3
      Client.c
      textfile.txt
      videofile.mp4
  v Server
      a.out
      Server.c
```

Fig: Show files in Client and Server Directory before transfer



```
v MYFILE...
  > 31118_RPC
  > RPC
  v UDPFileTransfer
    v Client
        a.out
        audiofile.mp3
        Client.c
        textfile.txt
        videofile.mp4
    v Server
        a.out
        audiofile.mp3
        Server.c
        textfile.txt
        videofile.mp4
```

Fig: Show files in Client and Server Directory after file transfer


shubham20_03@Shubham-AcerSwift: ~/myfiles/UDPFileTransfer/Server
shubham20_03@Shubham-AcerSwift: ~/myfiles/UDPFileTransfer/Client
ENG
20:01

```
shubham20_03@Shubham-AcerSwift:~/myfiles/UDPFileTransfer/Server$ gcc Server.c
shubham20_03@Shubham-AcerSwift:~/myfiles/UDPFileTransfer/Server$ ./a.out
Socket successfully created.
Successfully binded.
Name of the received text file: textfile.txt
File received successfully.
Name of the received audio file: audiofile.mp3
File received successfully.
Name of the received video file: videofile.mp4
File received successfully.
```

```
Reading file contents.
File sent successfully.
What do you want to send to server?
        1.Text file
        2.Audio file
        3.Video file
        4.EXIT
: 2
Enter audio file name to send :
audiofile.mp3
Reading file contents.
Error Occured. File transfer failed.
What do you want to send to server?
        1.Text file
        2.Audio file
        3.Video file
        4.EXIT
: 3
Enter video file name to send :
videofile.mp4
Error Occured. File transfer failed.
What do you want to send to server?
        1.Text file
        2.Audio file
        3.Video file
        4.EXIT
: 3
Enter video file name to send :
videofile.mp4
Error Occured. File transfer failed.
What do you want to send to server?
        1.Text file
        2.Audio file
        3.Video file
        4.EXIT
: 4
shubham20_03@Shubham-AcerSwift:~/myfiles/UDPFileTransfer/Client$
```

Fig: Show file transfer process.