

15

Using TCP/IP sockets, write a client-server program to make client sending the file name & the server to send back the contents of the requested file if present

### \* Client TCP.py

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
serverSocket = socket(AF_INET, SOCK_STREAM)
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
sentence = input("\n Enter file name: ")
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print("\n File Server: \n")
print(filecontents)
clientSocket.close()
```

### Server TCP.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)
while 1:
    print("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
    file = open(sentence, "r")
    file = open(sentence, "r")
```

inside  
while  
loop

```
l = file.read(1024)
connectionSocket.send(l.encode())
print("\n Sent contents of " + sentence)
file.close()
connectionSocket.close()
```

O/P:

The server is ready to receive  
Sent contents of server TCP.py  
The server is ready to receive.

} Server side

Enter file name: server <sup>TCP</sup> ~~UDP~~.py  
Reply from server:

} client side

3/1/25



16. Using UDP sockets, write a client-server program to make client sending the file name & the server to send back the contents of the requested file if present.

\* client UDP.py:

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("\n Enter File name:")
clientSocket.sendto(bytes(sentence, "utf-8"),
                    (serverName, serverPort))
filecontents, serverAddress = clientSocket.recvfrom(
    2048)

print("\n Reply from server: \n")
print(filecontents.decode("utf-8"))
# for i in filecontents:
#     print(str(i), end=" ")
clientSocket.close()
clientSocket.close()
```

Server UDP.py:

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print("The server is ready to receive")
while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode("utf-8")
    file = open(sentence, "r")
    con = file.read(2048)
```

```

serverSocket.sendto(bytes, (con, "192.168.1.1"))
client Address)
print("In Sent contents of", end='')
print(sentence)
# for i in sentence:
    # print(str(i), end=" ")
file.close()

```

O/p:

→ The server is ready to receive  
 Sent contents of server UDP.py  
 The server is ready to receive

} Server Side

{ Enter file name: server UDP.py  
 Reply from server:

} client side

N  
 3/1/25

## 17 Tool Exploration - Wireshark

Wireshark is a powerful & widely used network protocol analyzer. It allows you to capture & inspect data packets travelling over a network in real-time, making it a crucial tool for studying computer networks, troubleshooting network issues & understanding protocols.

### Key Features:

1. Packet Capture: Captures live network traffic from various interfaces (ex: ethernet, wi-fi).
2. Protocol Analysis: Supports hundreds of protocols (Ex: TCP, UDP, HTTP, FTP).
3. Filtering: Offers powerful filters to isolate specific packets or traffic types.
4. Visualization: Displays packets details with hierarchical layers (ethernet, IP, TCP/UDP).

### Use Cases of Wireshark

1. Network Troubleshooting:
  - \* Diagnosing slow network speeds.
  - \* Identifying bottle necks or misconfigurations.
2. Security Analysis:
  - \* Detecting malicious traffic or intrusions.
3. Protocol Study:
  - \* Understanding packet structures and communication flow.



### Common Filters:

- \* http: show only HTTP traffic
- \* tcp.port == 80: show traffic on TCP port 80
- \* ip.addr == 192.168.1.1: show packets to or from a specific IP address
- \* ~~udp: show only UDP traffic~~

3/1/25