Computer Network

TCP Socket Programming

Ritika Thakur(2022408) Swarnima Prasad (2022525)

About TCP Server and How it is different from UDP

TCP is connection-oriented, meaning a connection must be established between the client and server before any data is transferred. This involves a handshake process. TCP ensures that all data sent is received in the correct order and without errors.

Performance Considerations:
1. TCP: Due to the reliable delivery and connection establishment, TCP is best suited for scenarios where all data must be received correctly and in order, such as web pages or file downloads.
2. UDP: Due to its lower overhead and faster communication, UDP is ideal for applications like video stream-ing, gaming, or DNS queries, where speed is critical, and some packet loss is acceptable.

Differences in Code:
1. TCP: Requires establishing a connection with accept(), uses send() and recv(), ensures reliable data transfer,
and requires closing the connection.
2. UDP: Does not establish a connection, uses recvfrom() and sendto(), is faster but unreliable, and does not require closing the connection

Server Code

```python
from socket import *
import sys

# Stream instead of datagram in TCP
serverSocket = socket(AF_INET, SOCK_STREAM)

serverPort = 6789
serverSocket.bind(('127.0.0.1', serverPort))
serverSocket.listen(1)


while True:
    print('Ready to serve...')

    connectionSocket, addr = serverSocket.accept()

    try:
        message = connectionSocket.recv(1024).decode()

        if not message:
            connectionSocket.close()
            continue

        filename = message.split()[1]
        f = open(filename[1:])
        outputdata = f.read()

        response = "HTTP/1.1 200 OK\r\n\r\n" + outputdata
        connectionSocket.send(response.encode())

        print(response)

        connectionSocket.send("\r\n".encode())
        f.close()
        connectionSocket.close()

    except IOError:
        connectionSocket.send("HTTP/1.1 404 Not Found\r\n\r\n".encode())
        connectionSocket.send("<html><body><h1>404 Not Found</h1></body></html>\r\n".encode())
        connectionSocket.close()

serverSocket.close()
sys.exit()
```

HTML Code

```html
<!DOCTYPE html>
<html Lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Hello World</title>
</head>
<body>
    <h1>This course is called Computer Networks</h1>
    <p>This course is kind of hard.</p>
</body>
</html>
```

Output for local host

```
PS C:\Users\Ritika\OneDrive\Documents\Sem - V\CN_Assignments> cd Ass4
PS C:\Users\Ritika\OneDrive\Documents\Sem - V\CN_Assignments\Ass4> python TCPServer.py
The server is ready to receive
Ready to serve...
Requested file: /HelloWorld.html
Sent response to client:
Connection closed.
Ready to serve...
```

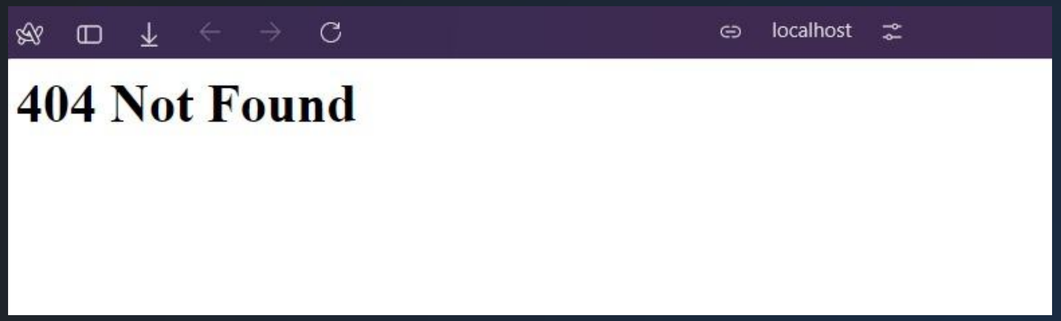
HTTP server for local host

```
PS C:\Users\Ritika\OneDrive\Documents\Sem - V\CN_Assignments> cd Ass4
PS C:\Users\Ritika\OneDrive\Documents\Sem - V\CN_Assignments\Ass4> python TCPClient.py 127.0.0.1 6789 HelloW
orld.html
Connected to server at 127.0.0.1:6789
Sent request:
GET /HelloWorld.html HTTP/1.1
Host: 127.0.0.1
 Connection: close


Received response:
HTTP/1.1 200 OK


<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Hello World</title>
</head>
<body>
    <h1>This course is called Computer Networks</h1>
    <p>This course is kind of hard.</p>
</body>
</html>

Connection closed.
PS C:\Users\Ritika\OneDrive\Documents\Sem - V\CN_Assignments\Ass4>
```

```
PS C:\Users\Ritika\OneDrive\Documents\Sem - V\CN_Assignments\Ass4> python TCPServer.py
The server is ready to receive
Ready to serve...
File not found, sending 404 response.
Ready to serve...
```

Not found error

Web page when server is hosted at local host

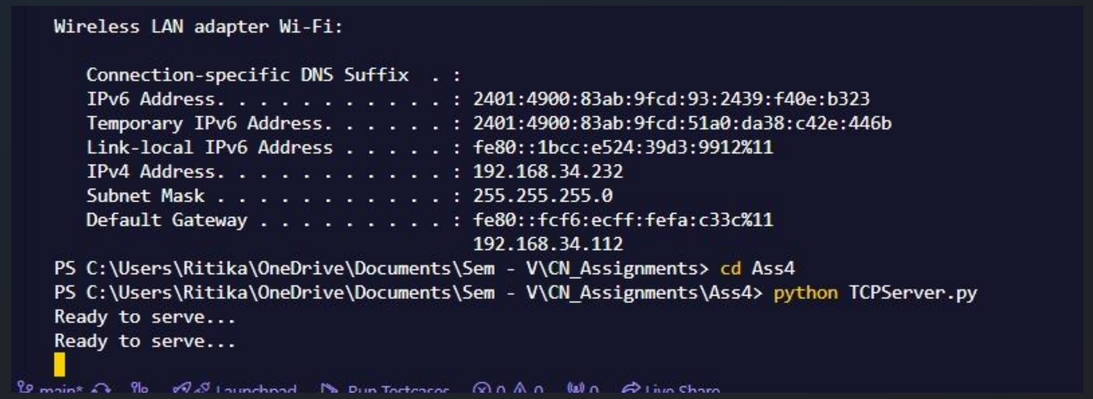


Web page when the requested html file is not present in the directory

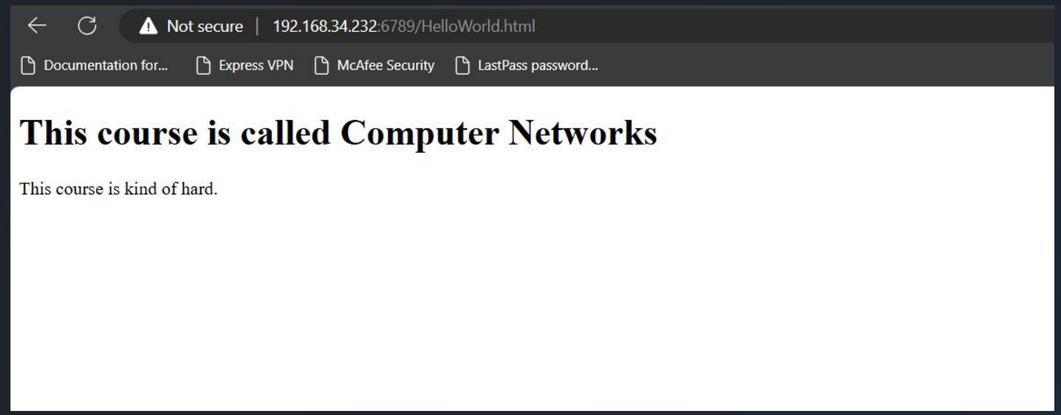
Output for Different Machines

Bind the server to the local IP address or 0.0.0.0 (which binds to all available interfaces).

```
serverSocket.bind(('0.0.0.0', serverPort))
```



Finding IP Address of server



Hosted on other machine by giving IP address of first machine

Multithreaded Server

```python
import socket
import threading

class WebServer:
    def __init__(self, host='', port=6789):
        self.host = host
        self.port = port
        self.serverSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.threads = []

        try:
            self.serverSocket.bind((self.host, self.port))
            self.serverSocket.listen(5)
            print(f"Server listening on {self.host}:{self.port}...")
        except socket.error as e:
            print(f"Socket error during binding/listening: {e}")
            self.serverSocket.close()
            raise

    def start(self):
        try:
            while True:
                try:
                    connectionSocket, addr = self.serverSocket.accept()
                    print(f"\n Connected to: {addr}")

                    thread = threading.Thread(target=self.threaded_client, args=(connectionSocket,))
                    thread.start()
                    self.threads.append(thread)

                except socket.error as e:
                    print(f"Socket error in main loop: {e}")
                    break
                except Exception as e:
                    print(f"Unexpected error in main loop: {e}")
                    break
        except KeyboardInterrupt:
            print("Server is shutting down...")
        finally:
            self.serverSocket.close()
            for thread in self.threads:
                thread.join()
            print("Server shutting down.")
```

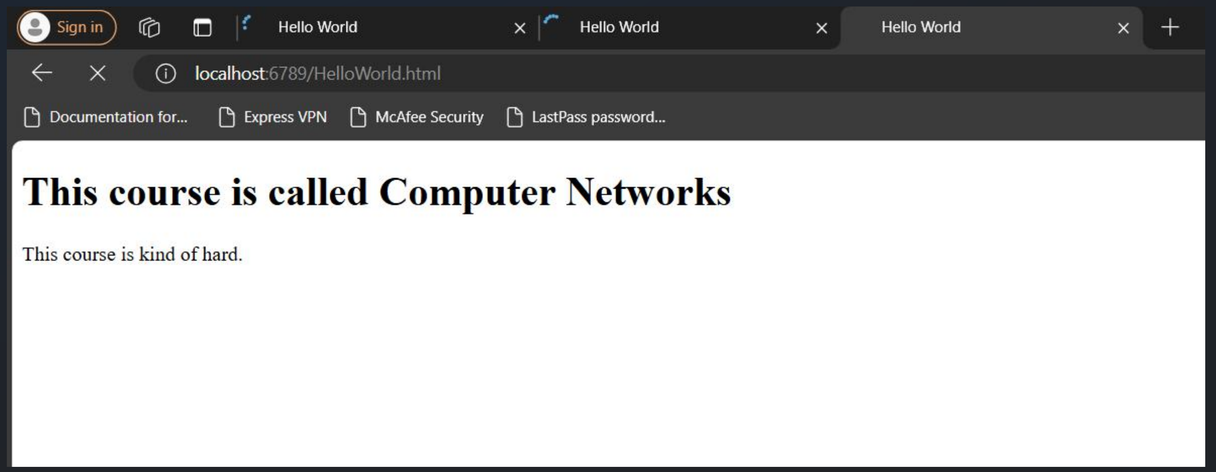
```python
f threaded_client(self, connectionSocket):
    try:
        while True:
            try:
                data = connectionSocket.recv(1024).decode()
                if not data:
                    print("No data received; closing connection.")
                    break

                print("Received request.")

                # Simple HTTP request handling
                lines = data.splitlines()
                if len(lines) > 0:
                    # Getting the requested file from the HTTP request
                    filename = lines[0].split()[1]
                    if filename == "/":
                        filename = "/HelloWorld.html"

                    # Attempting to open the requested file
                    try:
                        print(f"Attempting to open file: {filename[1:]}")
                        with open(filename[1:], 'r') as f:
                            outputdata = f.read()

                        # Sending HTTP response header and content
                        response = "HTTP/1.1 200 OK\r\nConnection: keep-alive\r\n\r\n" + outputdata
                        print("File found and response sent.")
                    except FileNotFoundError:
                        print("File not found, sending 404 response.")
                        response = "HTTP/1.1 404 Not Found\r\n\r\n<html><body><h1>404 Not Found</h1></body></html>"
                    except Exception as e:
                        print(f"Error while opening file: {e}")
                        response = "HTTP/1.1 500 Internal Server Error\r\n\r\n<html><body><h1>500 Internal Server Error</h1></body></html

                    connectionSocket.send(response.encode())

            except socket.error as e:
                print(f"Socket error in thread: {e}")
                break
            except Exception as e:
                print(f"Error in thread: {e}")
                break
    finally:
        connectionSocket.close()
        print("Connection closed.\n")
```

```python
if __name__ == "__main__":
    try:
        server = WebServer(host='0.0.0.0', port=6789)
        server.start()
    except Exception as e:
        print(f"Failed to start server: {e}")
```

Output

```
PS C:\Users\swarnima prasad\OneDrive\Desktop\CN_ass\CN_Assignments\Ass4> python multithreaded_TCPServer.py
Server listening on 127.0.0.1:6789...

 Connected to: ('127.0.0.1', 57338)
Received request.
Attempting to open file: HelloWorld.html
File found and response sent.

 Connected to: ('127.0.0.1', 57339)

 Connected to: ('127.0.0.1', 57340)
Received request.
Attempting to open file: HelloWorld.html
File found and response sent.
Received request.
Attempting to open file: HelloWorld.html
File found and response sent.
```

When multiple Tabs were opened



opening multiple tabs

```
 Connected to: ('127.0.0.1', 57672)
Received request.
Attempting to open file: HelloWorld.html
File found and response sent.
Received request.
Attempting to open file: HelloWorld.html
File found and response sent.
No data received; closing connection.
Connection closed.

No data received; closing connection.
Connection closed.

No data received; closing connection.
Connection closed.

Server is shutting down...
Server shutting down.
```

shutdown if CTRL + C is pressed and all tabs are closed

After Server Shut Down

Output across different machine

```
PS C:\Users\Ritika\OneDrive\Documents\Sem - V\CN_Assignments> cd Ass4
PS C:\Users\Ritika\OneDrive\Documents\Sem - V\CN_Assignments\Ass4> python multithreaded_TCPServer.py
Server listening on 0.0.0.0:6789...
Connected to: ('192.168.34.67', 55355)
Received request.
Attempting to open file: HelloWorld.html
File found and response sent.
No data received; closing connection.
Connection closed.

Connected to: ('192.168.34.67', 55356)
Received request.
Attempting to open file: HelloWorld.html
File found and response sent.
No data received; closing connection.
Connection closed.

Connected to: ('192.168.34.67', 55370)
Received request.
Attempting to open file: HelloWorld.html
File found and response sent.
Connected to: ('192.168.34.67', 55371)
```

Multithreaded TCP Server hosted on 192.168.34.67

Client Code

```python
import socket
import sys

class TCPClient:
    def __init__(self, host, port, filename):
        self.host = host
        self.port = port
        self.filename = filename
        self.client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    def connect(self):
        try:
            self.client_socket.connect((self.host, self.port))
            print(f"Connected to server at {self.host}:{self.port}")
        except socket.error as e:
            print(f"Error connecting to server: {e}")
            sys.exit(1)

    def create_request(self):
        request_line = f"GET /{self.filename} HTTP/1.1\r\n"
        headers = f"Host: {self.host}\r\n Connection: close\r\n\r\n"
        return request_line + headers

    def send_request(self):
        try:
            request = self.create_request()
            self.client_socket.sendall(request.encode())
            print("Sent request:")
            print(request)
        except socket.error as e:
            print(f"Error sending request: {e}")
            sys.exit(1)

    def receive_response(self):
        try:
            response = self.client_socket.recv(4096)
            if not response:
                raise ValueError("No response received or server closed connection unexpecte
            print("Received response:")
            print(response.decode())
        except socket.error as e:
            print(f"Error receiving response: {e}")
        except ValueError as ve:
            print(f"Error: {ve}")
        except Exception as e:
            print(f"Unexpected error while receiving response: {e}")

    def close(self):
        try:
            self.client_socket.close()
            print("Connection closed.")
        except socket.error as e:
            print(f"Error closing the connection: {e}")
```

```python
def main():
    if len(sys.argv) != 4:
        print("Usage: client.py <server_host> <server_port> <filename>")
        sys.exit(1)

    server_host = sys.argv[1]
    server_port = int(sys.argv[2])
    filename = sys.argv[3]

    client = TCPClient(server_host, server_port, filename)

    try:
        client.connect()
        client.send_request()
        client.receive_response()
    except Exception as e:
        print(f"Error: {e}")
    finally:
        client.close()

if __name__ == "__main__":
    main()
```

Output

TCP Server and TCP client

```
PS C:\Users\Ritika\OneDrive\Documents\Sem - V\CN_Assignments> cd Ass4
PS C:\Users\Ritika\OneDrive\Documents\Sem - V\CN_Assignments\Ass4> python TCPServer.py
The server is ready to receive
Ready to serve...
Requested file: /HelloWorld.html
Sent response to client:
Connection closed.
Ready to serve...
```

```
PS C:\Users\Ritika\OneDrive\Documents\Sem - V\CN_Assignments> cd Ass4
PS C:\Users\Ritika\OneDrive\Documents\Sem - V\CN_Assignments\Ass4> python TCPClient.py 127.0.0.1 6789 Hell
orld.html
Connected to server at 127.0.0.1:6789
Sent request:
GET /HelloWorld.html HTTP/1.1
Host: 127.0.0.1
 Connection: close


Received response:
HTTP/1.1 200 OK


<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Hello World</title>
</head>
<body>
    <h1>This course is called Computer Networks</h1>
    <p>This course is kind of hard.</p>
</body>
</html>

Connection closed.
PS C:\Users\Ritika\OneDrive\Documents\Sem - V\CN_Assignments\Ass4>
```

```
PS C:\Users\swarnima prasad\OneDrive\Desktop\CN_ass\CN_Assignments\Ass4> python TCPClient.py 127.0.0.1 6789 HelloWorld.html
Connected to server at 127.0.0.1:6789
Sent request:
GET /HelloWorld.html HTTP/1.1
Host: 127.0.0.1
Connection: close


Received response:
HTTP/1.1 200 OK


<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Hello World</title>
</head>
<body>
    <h1>We are smart!</h1>
    <p>This world needs smart people like us</p>
</body>
```

```
PS C:\Users\swarnima prasad\OneDrive\Desktop\CN_ass\CN_Assignments\Ass4> python TCPClient.py 127.0.0.1 6789 HelloWorld.html
Connected to server at 127.0.0.1:6789
Sent request:
GET /HelloWorld.html HTTP/1.1
Host: 127.0.0.1
Connection: close


Received response:
HTTP/1.1 200 OK


<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Hello World</title>
</head>
<body>
    <h1>We are smart!</h1>
    <p>This world needs smart people like us</p>
</body>
```

PROBLEMS OUTPUT DEBUG CONSOLE PORTS

TERMINAL

```
    </body>
    </html>

PS C:\Users\swarnima prasad\OneDrive\Desktop\CN_ass\CN_Assignments\Ass4> python TCPClient.py 127.0.0.1 6789 HelloWorld.html
Connected to server at 127.0.0.1:6789
Sent request:
GET /HelloWorld.html HTTP/1.1
Host: 127.0.0.1
Connection: close


Received response:
HTTP/1.1 200 OK


<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Hello World</title>
</head>
<body>
    <h1>We are smart!</h1>
    <p>This world needs smart people like us</p>
</body>
</html>

PS C:\Users\swarnima prasad\OneDrive\Desktop\CN_ass\CN_Assignments\Ass4>
```

```
PS C:\Users\swarnima prasad\OneDrive\Desktop\CN_ass\CN_Assignments\Ass4> python multithreaded_TCPServer.py
Server listening on port 6789...
Connected to: ('127.0.0.1', 62374)
Received request: GET /HelloWorld.html HTTP/1.1
Host: 127.0.0.1
Connection: close


Attempting to open file: HelloWorld.html
File found and response sent.
No data received; closing connection.
Connection closed.

Connected to: ('127.0.0.1', 62375)
Received request: GET /HelloWorld.html HTTP/1.1
Host: 127.0.0.1
Connection: close


Attempting to open file: HelloWorld.html
File found and response sent.
No data received; closing connection.
Connection closed.

Connected to: ('127.0.0.1', 62376)
Received request: GET /HelloWorld.html HTTP/1.1
Host: 127.0.0.1
Connection: close


Attempting to open file: HelloWorld.html
File found and response sent.
No data received; closing connection.
Connection closed.
```

Multithreaded server and TCP client

Output across different machine

```
PS C:\Users\swarnima prasad\OneDrive\Desktop\CN_ass\CN_Assignments\Ass4> python TCPClient.py 192.168.34.232 6789 HelloWorld.html
Connected to server at 192.168.34.232:6789
Sent request:
GET /HelloWorld.html HTTP/1.1
Host: 192.168.34.232
Connection: close


Received response:
HTTP/1.1 200 OK


<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Hello World</title>
</head>
<body>
    <h1>This course is called Computer Networks</h1>
    <p>This course is kind of hard.</p>
</body>
</html>

PS C:\Users\swarnima prasad\OneDrive\Desktop\CN_ass\CN_Assignments\Ass4>
```

TCP Client when server is hosted on different machine

```
PS C:\Users\Ritika\OneDrive\Documents\Sem - V\CN_Assignments\Ass4> python WebServer.py
Traceback (most recent call last):
  File "C:\Users\Ritika\OneDrive\Documents\Sem - V\CN_Assignments\Ass4\WebServer.py", line 8, in <module>
    serverSocket.bind(('localhost', serverPort))
OSError: [WinError 10048] Only one usage of each socket address (protocol/network address/port) is normally permitted
PS C:\Users\Ritika\OneDrive\Documents\Sem - V\CN_Assignments\Ass4>
```

When we tried to host Multithreaded TCP Server after TCP Server was already hosted on the same port

```
PS C:\Users\swarnima prasad\OneDrive\Desktop\CN_ass\CN_Assignments\Ass4> python TCPClient.py 172.20.10.4 6789 HelloWorld.html
Error: [WinError 10060] A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond
```

When the socket is binded to 'localhost' instead of 0.0.0.0 we notice that our client-server connection does not work across different machines