Computer Network Lab

CSE-325

Assignment - 6

Submitted by -
Gyanendra Shukla
CSE 1
191112040

To implement Client-Server application using UDP

I made an echo client and server using UDP protocol. The server socket is started with `SOCK_DGRAM` argument to start a UDP server. The client socket is started with `SOCK_DGRAM` argument to start a UDP client. I take in input in the client application in the `run` function. If the input is `exit`, I close the client socket and exit. In the server program, if I recieve an empty message, I close the server socket and exit.

UDP Server

```
import socket
import datetime

class UDPServer:
    def __init__(self, addr: str, port: int) -> None:
        """
        Starting a UDP server with SOCK_DGRAM on the given address and port.
        """
        self.socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
        self.socket.bind((addr, port))
        print(f"\nServer started on {addr}:{port}\n")


    def run(self) -> None:
        """
        Listen for incoming messages and echo them back to the client.
        """
        while True:
            data, addr = self.socket.recvfrom(1024)
            print(f"Got: {data.decode()}")

            if data:
                d = f"{datetime.datetime.now()} : {data.decode()}"
```

```
                self.socket.sendto(d.encode(), addr)
            else:
                self.socket.sendto("".encode(), addr)
                self.socket.close()
                break


if __name__ == '__main__':
    server = UDPServer("127.0.0.1", 1234)
    server.run()
```

UDP Client

```
import socket

class UDPClient:
    def __init__(self) -> None:
        self.sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

    def send(self, message) -> None:
        try:
            sent = self.sock.sendto(message, ("localhost", 1234))
            data, server = self.sock.recvfrom(1024)
            print(data.decode())
        except Exception as e:
            print(e)

    def close(self) -> None:
        self.sock.close()

    def run(self) -> None:
        while True:
            message = input('> ')
            if (message=='exit'):
                self.close()
                break
            self.send(message.encode())

if __name__ == '__main__':
    client = UDPClient()
    client.run()
```

Output

```
In D:\Books\sem 6\networks\labs\lab6 python .\server.py

Server started on 127.0.0.1:1234

Got: hello
Got: this is a message
Got: from
Got: 191112040
Got: Gyanendra
Got: It is over UDP
Got: This is Computer Network Lab assignment 6.
Got:
In D:\Books\sem 6\networks\labs\lab6 3m 50.125s
```

Fig: UDP Server. It takes in command, adds timestamp and echoes back to the client.

```
In D:\Books\sem 6\networks\labs\lab6 python .\client.py
> hello
2022-02-20 22:27:49.090828 : hello
> this is a message
2022-02-20 22:27:53.198262 : this is a message
> from
2022-02-20 22:27:54.763092 : from
> 191112040
2022-02-20 22:27:57.187089 : 191112040
> Gyanendra
2022-02-20 22:27:59.485370 : Gyanendra
> It is over UDP
2022-02-20 22:28:03.317134 : It is over UDP
> This is Computer Network Lab assignment 6.
2022-02-20 22:29:08.233964 : This is Computer Network Lab assignment 6.
>

> exit
In D:\Books\sem 6\networks\labs\lab6 3m 47.683s
```

Fig: UDP Client. It sends command to server and prints the output from server.