Experiment 7

TCP Socket:

Server:

```
import socket

HOST = '127.0.0.1'
PORT = 1024

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
   s.bind((HOST, PORT))
   s.listen()
   conn, addr = s.accept()
   with conn:
       print(f'Connected to: {addr}')

       while True:
           data = conn.recv(4)
           if not data:
               break

           print(f'Received data: {data}')
```

Client:

```
import socket

HOST = '127.0.0.1'
PORT = 1024

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
   s.connect((HOST, PORT))
   s.sendall(b'ABCDEFGHIJKLMNOPQRSTUVWXYZ')
```

```
comp@comp:~$ python3 socket_tcp_server.py
Connected to: ('127.0.0.1', 50670)
Received data: b'ABCD'
Received data: b'EFGH'
Received data: b'IJKL'
Received data: b'MNOP'
Received data: b'QRST'
Received data: b'UVWX'
Received data: b'YZ'
```

UDP Socket:

Server:

```python
import socket

HOST = '127.0.0.1'
PORT = 1024

with socket.socket(socket.AF_INET, socket.SOCK_DGRAM) as s:
   s.bind((HOST, PORT))

   while True:
       data, addr = s.recvfrom(32)
       if not data:
           break

       print(f'Received data: {data}')
```

Client:

```python
import socket

HOST = '127.0.0.1'
PORT = 1024

with socket.socket(socket.AF_INET, socket.SOCK_DGRAM) as s:
   s.sendto(b'ABCDEFGHIJKLMNOPQRSTUVWXYZ', (HOST, PORT))
```

```
comp@comp:~$ python3 socket_udp_server.py
Received data: b'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
```

Sending file using sockets:
Server:

```
import socket

HOST = '127.0.0.1'
PORT = 1024

write_to = open('receieved.txt', 'w')

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
   s.bind((HOST, PORT))
   s.listen()
   conn, addr = s.accept()
   with conn:
       print(f'Connected to: {addr}')

       while True:
           data = conn.recv(1024)
           if not data:
               break

           write_to.write(data.decode())
```

Client:

```
import socket

HOST = '127.0.0.1'
PORT = 1024

to_send = open('sample.txt', 'r')
data = to_send.read()

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
   s.connect((HOST, PORT))
   s.sendall(data.encode())
```