ASSIGNMENT – 1

Name: Foram Patel

Roll No.: 83

Subject : Advanced Networking

Course: M.Sc.(CS - 5)

Q1_CLIENT

```
import socket

def start_client():
try:
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client_socket.connect(('localhost', 12345))
msg = input("Enter a Message : ")
client_socket.send(msg.encode())
data = client_socket.recv(1024).decode()
print(f"Received from server: {data}")

except Exception as e:
print(f"Error: {e}")

finally:
client_socket.close()
```

```
start_client()
```

A2Q1_SERVER


```
import socket

def start_server():
try:
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.bind(('localhost', 12345))
server_socket.listen(1)
print("Server is listening on port 12345...")
conn, addr = server_socket.accept()
print(f"Connected by {addr}")
data = conn.recv(1024).decode()
print(f"Received from client: {data}")
conn.send(data.encode())

except Exception as e:
print(f"Error: {e}")

finally:
conn.close()
server_socket.close()

start_server()
```

A2Q2_CLIENT


```
import socket

def start_client():
try:
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client_socket.connect(('localhost', 12345))
msg = input("Enter A Message : ")
client_socket.send(msg.encode())
data = client_socket.recv(1024).decode()
print(f"Received from server: {data}")

except Exception as e:
print(f"Error: {e}")

finally:
client_socket.close()

start_client()
```

A2Q2_SERVER


```
import socket
```

```
def start_server():

try:

server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

server_socket.bind(('localhost', 12345))

server_socket.listen(1)

print("Server is listening on port 12345...")

conn, addr = server_socket.accept()

data = conn.recv(1024).decode()

print(f"Received from client: {data}")

response = data.upper()

conn.send(response.encode())


except Exception as e:

print(f"Error: {e}")


finally:

conn.close()

server_socket.close()


start_server()
```

A2Q3_CLIENT

```
---------------------------------------------------------------------------------------------------------------------------------
---

import socket

def start_client():
try:
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client_socket.connect(('localhost', 12345))
msg = input("Enter a message: ")
client_socket.send(msg.encode())
data = client_socket.recv(1024).decode()
print(f"Received from server: Length of string is {data}")

except Exception as e:
print(f"Error: {e}")
finally:
client_socket.close()

start_client()
```

A2Q3_SERVER

```
---------------------------------------------------------------------------------------------------------------------------------
---

import socket

def start_server():
```

```
try:
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.bind(('localhost', 12345))
server_socket.listen(1)
print("Server is listening on port 12345...")
conn, addr = server_socket.accept()
data = conn.recv(1024).decode()
print(f"Received from client: {data}")
length = str(len(data))
conn.send(length.encode())

except Exception as e:
print(f"Error: {e}")

finally:
conn.close()
server_socket.close()

start_server()
```

A2Q4_CLIENT

```
----------------------------------------------------------------------------------------------------------------------------------------
---
```

```
import socket

def xor_encrypt_decrypt(data, key):
return bytes([b ^ key for b in data])
```

```
encryption_key = 123

def start_client():
try:
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client_socket.connect(('localhost', 12345))
msg = input("Enter a message: ")
encrypted_msg = xor_encrypt_decrypt(msg.encode(), encryption_key)
client_socket.send(encrypted_msg)
data = client_socket.recv(1024)
decrypted_response = xor_encrypt_decrypt(data, encryption_key).decode()
print(f"Decrypted response from server: {decrypted_response}")

except Exception as e:
print(f"Error: {e}")
finally:
client_socket.close()
```

```
start_client()
```

A2Q4_SERVER


```
import socket

def xor_encrypt_decrypt(data, key):
return bytes([b ^ key for b in data])

encryption_key = 123

def start_server():
try:
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.bind(('localhost', 12345))
server_socket.listen(1)
print("Server is listening on port 12345...")
conn, addr = server_socket.accept()
encrypted_data = conn.recv(1024)
decrypted_data = xor_encrypt_decrypt(encrypted_data, encryption_key).decode()
print(f"Decrypted message from client: {decrypted_data}")
response = xor_encrypt_decrypt("Message received".encode(), encryption_key)
conn.send(response)
```

```
except Exception as e:
print(f"Error: {e}")

finally:
conn.close()
server_socket.close()

start_server()
```

A2Q5_CLIENT

```
----------------------------------------------------------------------------------------------------------------------------------
---

import socket

def start_client():
try:
client_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
client_socket.sendto("START".encode(), ('localhost', 12345))
print("Sent file data to server.")
data, _ = client_socket.recvfrom(1024)
print(f"Server response: {data.decode()}")

except Exception as e:
```

```
print(f"Error: {e}")

finally:
client_socket.close()

start_client()
```

A2Q5_SERVER

```
-----------------------------------------------------------------------------------------------------------------------------------
---
```

```
import socket

def start_server():
try:
# Create the UDP socket
server_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
server_socket.bind(('localhost', 12345))
print("Server is listening for file transfer...")

while True:
# Wait for data from client
data, client_address = server_socket.recvfrom(1024)
message = data.decode()
```

```
if message == "START":

print("Received file data from client.")

server_socket.sendto("ACK: File received".encode(), client_address)

break


except Exception as e:

print(f"Error: {e}")


finally:

server_socket.close()


start_server()
```

A2Q6_CLIENT

```
import socket


def start_client():

try:

client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

client_socket.connect(('localhost', 12345))

msg = input("Enter a message : ")
```

```
client_socket.send(msg.encode())
data = client_socket.recv(1024).decode()
print(f"Hash value received from server: {data}")

except Exception as e:
print(f"Error: {e}")

finally:
client_socket.close()

start_client()
```

A2Q6_SERVER

```
----------------------------------------------------------------------------------------------------------------------------------
---

import socket
import hashlib

def start_server():
try:
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.bind(('localhost', 12345))
server_socket.listen(1)
print("Server is listening on port 12345...")
```

```
conn, addr = server_socket.accept()
data = conn.recv(1024).decode()
print(f"Received from client: {data}")
hash_value = hashlib.sha256(data.encode()).hexdigest()
conn.send(hash_value.encode())

except Exception as e:
print(f"Error: {e}")

finally:
conn.close()
server_socket.close()

start_server()
```


ASSIGNMENT – 2

DNS(query and response)

```
C:\Users\an>ping google.com

Pinging google.com [2404:6800:4009:814::200e] with 32 bytes of data:
Reply from 2404:6800:4009:814::200e: time=45ms
Reply from 2404:6800:4009:814::200e: time=47ms
Reply from 2404:6800:4009:814::200e: time=157ms
Reply from 2404:6800:4009:814::200e: time=215ms

Ping statistics for 2404:6800:4009:814::200e:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 45ms, Maximum = 215ms, Average = 116ms
```

--w:

```
Frame 23: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface
\Device\NPF_{0B0982C6-A1E0-4094-9FA3-5EC7A41B2326}, id 0
Ethernet II, Src: Intel_b4:b5:f4 (a0:--:--:--:--:--:f4), Dst: OppoMobileTe_76:ff:df (84:6f:ce:76:ff:df)
Internet Protocol Version 4, Src: 192.---.---.--0, Dst: 192.168.43.1
User Datagram Protocol, Src Port: 54821, Dst Port: 53
Domain Name System (query)
Transaction ID: 0xf540
Flags: 0x0100 Standard query
0... .... .... .... = Response: Message is a query
.000 0... .... .... = Opcode: Standard query (0)
.... ..0. .... .... = Truncated: Message is not truncated
.... ...1 .... .... = Recursion desired: Do query recursively
.... .... .0.. .... = Z: reserved (0)
.... .... ...0 .... = Non-authenticated data: Unacceptable
Questions: 1
Answer RRs: 0
Authority RRs: 0
Additional RRs: 0
Queries
google.com: type A, class IN
[Response In: 25]
```

---HTTP:

```
Frame 3839: 186 bytes on wire (1488 bits), 186 bytes captured (1488 bits) on interface
\Device\NPF_{0B0982C6-A1E0-4094-9FA3-5EC7A41B2326}, id 0
Ethernet II, Src: Intel_b4:b5:f4 (a0:--:--:--:--:--:f4), Dst: OppoMobileTe_76:ff:df (84:6f:ce:76:ff:df)
Internet Protocol Version 6, Src: 2--5:---:----:----:----:----:----:---7, Dst: 2405:200:1609:1731::312c:7009
Transmission Control Protocol, Src Port: 50379, Dst Port: 80, Seq: 1, Ack: 1, Len: 112
Hypertext Transfer Protocol
GET /connecttest.txt HTTP/1.1\r\n
Request Method: GET
Request URI: /connecttest.txt
Request Version: HTTP/1.1
Connection: Close\r\n
User-Agent: Microsoft NCSI\r\n
Host: ipv6.msftconnecttest.com\r\n
\r\n
[Response in frame: 3845]
[Full request URI: http://ipv6.msftconnecttest.com/connecttest.txt]
```

--Ip and TCP:

```
Frame 737: 1424 bytes on wire (11392 bits), 1424 bytes captured (11392 bits) on interface
\Device\NPF_{0B0982C6-A1E0-4094-9FA3-5EC7A41B2326}, id 0
Ethernet II, Src: Intel_b4:b5:f4 (a0:--:--:--:--:--:f4), Dst: OppoMobileTe_76:ff:df (84:6f:ce:76:ff:df)
Internet Protocol Version 4, Src: 192.---.---.--0, Dst: 52.168.117.170
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 1410
Identification: 0x2770 (10096)
010. .... = Flags: 0x2, Don't fragment
```

```
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 128
Protocol: TCP (6)
Header Checksum: 0x374d [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.---.---.--0
Destination Address: 52.168.117.170
[Stream index: 8]
Transmission Control Protocol, Src Port: 50415, Dst Port: 443, Seq: 5595, Ack: 7104, Len: 1370
Source Port: 50415
Destination Port: 443
[Stream index: 13]
[Stream Packet Number: 40]
[Conversation completeness: Incomplete, DATA (15)]
[TCP Segment Len: 1370]
Sequence Number: 5595 (relative sequence number)
Sequence Number (raw): 2973686452
[Next Sequence Number: 6965 (relative sequence number)]
Acknowledgment Number: 7104 (relative ack number)
Acknowledgment number (raw): 1289265097
0101 .... = Header Length: 20 bytes (5)
Flags: 0x010 (ACK)
Window: 254
[Calculated window size: 65024]
[Window size scaling factor: 256]
Checksum: 0xe6f8 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
[Timestamps]
[SEQ/ACK analysis]
TCP payload (1370 bytes)
```

```
[Reassembled PDU in frame: 738]

TCP segment data (1370 bytes)


-------------------------------------------------------------ICMP (ping)

C:\Users\an>ping -n 10 google.com


Pinging google.com [2404:6800:4009:82c::200e] with 32 bytes of data:

Reply from 2404:6800:4009:82c::200e: time=86ms

Reply from 2404:6800:4009:82c::200e: time=47ms

Reply from 2404:6800:4009:82c::200e: time=51ms

Reply from 2404:6800:4009:82c::200e: time=255ms

Reply from 2404:6800:4009:82c::200e: time=262ms

Reply from 2404:6800:4009:82c::200e: time=72ms

Reply from 2404:6800:4009:82c::200e: time=81ms

Reply from 2404:6800:4009:82c::200e: time=77ms

Reply from 2404:6800:4009:82c::200e: time=330ms

Reply from 2404:6800:4009:82c::200e: time=70ms


Ping statistics for 2404:6800:4009:82c::200e:

Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 47ms, Maximum = 330ms, Average = 133ms


------------------------------------------------------------ARP

Frame 6: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface

\Device\NPF_{0B0982C6-A1E0-4094-9FA3-5EC7A41B2326}, id 0

Ethernet II, Src: Intel_b4:b5:f4 (a0:--:--:--:--:f4), Dst: OppoMobileTe_76:ff:df (84:6f:ce:76:ff:df)

Address Resolution Protocol (reply)

Hardware type: Ethernet (1)
```

Protocol type: IPv4 (0x0800)

Hardware size: 6

Protocol size: 4

Opcode: reply (2)

Sender MAC address: Intel_b4:b5:f4 (a0:--:--:--:--:f4)

Sender IP address: 192.---.---.--0

Target MAC address: OppoMobileTe_76:ff:df (84:6f:ce:76:ff:df)

Target IP address: 192.168.43.1