Suyog Joshi
CS585 Computer Networks
Homework 3
11/29/2023

Question 1)

The first IP datagram containing the first part of the segment sent to 128.119.245.12 sent by your computer via the traceroute command is 179. Yes, the segment has been fragmented across three IP datagrams i.e. Packets 179, 180, and 181.
The “Fragment. Offset” field in the IP header indicates the offset of the data in the current fragment relative to the beginning of the original unfragmented datagram. If the “Fragment Offset” is non-zero, it indicates fragmentation.
The “Flags” field in the IP header contains the “More Fragments”(MF) flag. If the MF flag is set to 1, it indicates that more fragments are coming, and this is not the last fragment. If the MF flag is 0, it means this is the last fragment.
The total length of the IP datagram is indicated in the “Total Length” field of the IP header. For Packet 179, the length is 1514 bytes.
The change between the first and second fragment are the “Fragment Offset” and the “More Fragments” flag. For packet 179, the “Fragment Offset” is 0, and the “More Fragments” flag is set to 1 but for the packet 180, the “Fragment Offset” is 1480 and the “More Fragments” flag is set to 0.
To find the size of the payload, we need to find the lengths of the IP header and payloads. Typically the length of the IP headers is 20 bytes for TCP and 8 bytes for UDP. So,
Packet 179 : 8 bytes(IP header of UDP) + 1480 bytes(payload)
Packet 180: 8 bytes(IP header of UDP) + 1480 bytes(payload)
Packet 181:8 bytes(IP header of UDP) + 2972 bytes(payload)
The “Fragment Offset” changes between first and second fragment.

Question 2)

The IPv6 address of computer making the DNS AAAA request is 2601:193:8302:4620:215c:f5ae:8b40:a27a
The IPv6 address of the destination address is 2001:558:feed::1.
The value of flow Label: 0x63ed0.
The payload data is 29 bytes.
The IPv6 DNS response to the IPv6 DNS AAAA request made in the 20th packet in this trace is in 27th packet. There is only one response to this AAAA request and we can confirm it by looking at the Answer RRs in the 27th packet inside the DNS (response).

Question 3)
The server code is provided as follows:

```
#!/usr/bin/python3
import socket

def send_file_to_client(conn, filename):

    conn.sendall(f"{filename}\n".encode('utf-8'))

    # Open the file and send its contents
    with open(filename, 'rb') as file:
        while True:
            data = file.read(1024)  # Reading data in chunks
            if not data:
                break
            conn.sendall(data)

# Server details
server_host = 'localhost'
server_port = 2002

# Create a TCP/IP socket
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as server_socket:
    # Avoid bind() exception
    server_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

    # Bind socket to the port
    server_socket.bind((server_host, server_port))

    # Listen for incoming connections
    server_socket.listen()
    print(f"Server is listening on {server_port}")

    while True:
        # Wait for a connection
```

```
    connection, client_address = server_socket.accept()
    try:
        print(f"Connection from {client_address}")
        # Send a file to the client
        send_file_to_client(connection, 'hello.txt')
        print(f"File Sent")
    finally:
        # Close connection
        connection.close()
        print("Connection closed")
```