Chapter 5

Lab 4: Write and analyze socket programs using Python

5.1 Objective

Demonstrate socket programming using Python and analyze the headers of the UDP and TCP segments in Wireshark. The client sends a lowercase text to the server. The server converts the sentence into uppercase and returns it to the client.

- In part 1, show the exchange of text messages between a client and a server using UDP sockets.
- In part 2, show the exchange of text messages between a client and a server using TCP sockets.

5.2 Procedure

1. Students will write the UDP client program in one computer and the UDP server program in another computer. The IP address and the port number used in the server program must also be provided in the UDP client program

   UDP client program

   ```
   from socket import *
   serv_addr = "IP address of server"
   serv_port = 8000
   client_sock = socket(AF_INET,SOCK_DGRAM)
   msg = input("Enter the text message: ")
   client_sock.sendto(msg.encode(),(serv_addr,serv_port))
   mod_msg,s = client_sock.recvfrom(2048)
   print("From Server: ",mod_msg.decode())
   ```

   UDP server program

   ```
   from socket import *
   serv_addr = "IP address of server"
   serv_port = 8000
   serv_sock = socket(AF_INET,SOCK_DGRAM)
   serv_sock.bind((serv_addr,serv_port))
   print(("The server is ready to receive"))
   while 1:
           msg,client_addr  = serv_sock.recvfrom(2048)
           print(("Got message from",client_addr))
           mod_msg = msg.upper()
           serv_sock.sendto(mod_msg,client_addr)
   ```

2. Run the UDP server program. Run Wireshark on the Ethernet connection. Run the UDP client program. Save the packet capture file for analysis.

3. Students will write the TCP client program in one computer and the TCP server program in another computer. The IP address and the port number used in the server program must also be provided in the TCP client program

   TCP client program

   ```
   from socket import *
   serv_addr = "IP address of server"
   serv_port = 8000
   client_sock = socket(AF_INET, SOCK_STREAM)
   client_sock.connect((serv_addr, serv_port))
   msg = input("Enter the text message: ")
   client_sock.send(msg.encode())
   mod_msg  = client_sock.revc(2048)
   print("From Server: ", mod_msg.decode())
   client_sock.close()
   ```

   TCP server program

   ```
   from socket import *
   serv_addr = "IP address of server"
   serv_port = 8000
   serv_sock = socket(AF_INET, SOCK_STREAM)
   serv_sock.bind((serv_addr, serv_port))
   serv_sock.listen(1)
   print(("The server is ready to receive"))
   while 1:
       conn_sock, client_addr  = serv_sock.accept()
       print(("Got connection from", client_addr))
       msg  = conn_sock.revc(2048)
       mod_msg = msg.upper()
       conn_sock.send(mod_msg)
       conn_sock.close()
   ```

4. Run the TCP server program. Run Wireshark on the Ethernet connection. Run the TCP client program. Save the packet capture file.

5.3 Analysis

1. What is the client port number in the UDP segment? What is the value of the Length field?
2. What is the value of the checksum in the UDP segment sent by the client? What is the value of the checksum in the UDP segment sent by the server?
3. What are the TCP segments observed for opening the TCP connection? What are the flags set in these segments?
4. What is the source port number in the TCP segment sent by the client?
5. What is the value of the Header Length field?
6. What are the TCP segments observed for closing the TCP connection? What are the flags set in these segments?
7. What are the flags set in the remaining TCP segments which carry the data?
8. What is the round trip time observed by the client during the TCP connection?