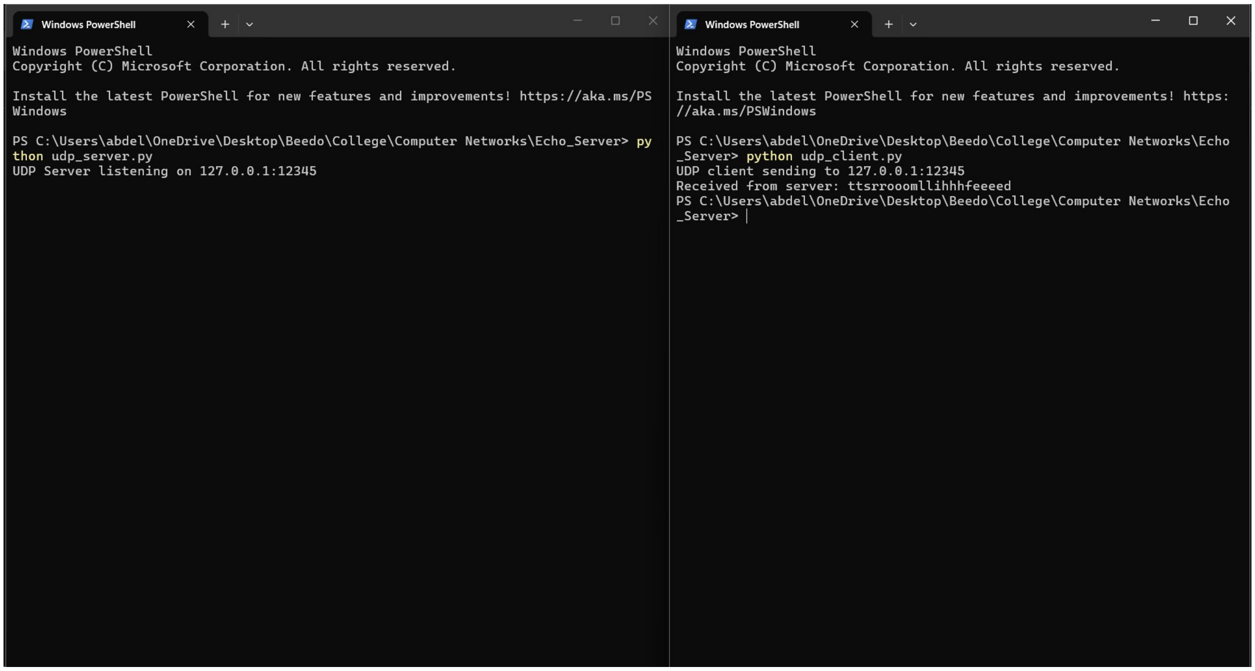
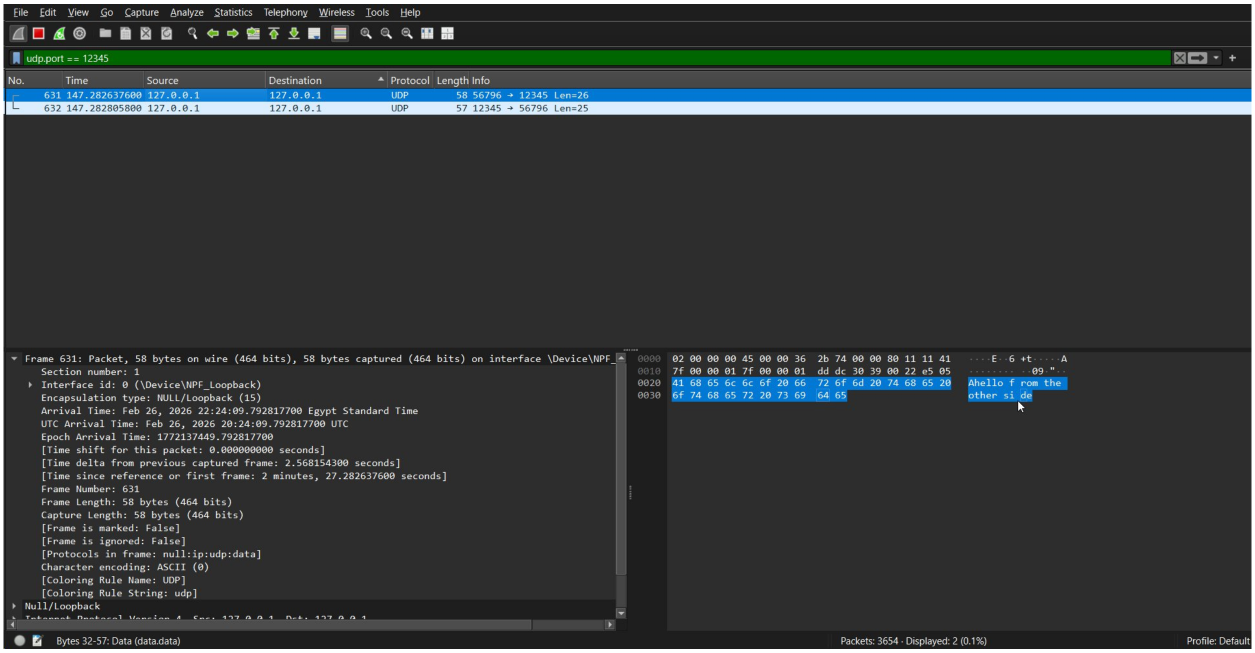
# Lab 1

## UDP Connection



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PS
Windows

PS C:\Users\abdel\OneDrive\Desktop\Beedo\College\Computer Networks\Echo_Server> py
thon udp_server.py
UDP Server listening on 127.0.0.1:12345
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https:
//aka.ms/PSWindows

PS C:\Users\abdel\OneDrive\Desktop\Beedo\College\Computer Networks\Echo
_Server> python udp_client.py
UDP client sending to 127.0.0.1:12345
Received from server: ttsrrooomllihhhfeeeed
PS C:\Users\abdel\OneDrive\Desktop\Beedo\College\Computer Networks\Echo
_Server> |
```



File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

udp.port == 12345

```
No.   Time          Source     Destination  Protocol Length Info
  631 147.282637600 127.0.0.1  127.0.0.1    UDP          58 56796 → 12345 Len=26
  632 147.282805800 127.0.0.1  127.0.0.1    UDP          57 12345 → 56796 Len=25
```

```
▾ Frame 631: Packet, 58 bytes on wire (464 bits), 58 bytes captured (464 bits) on interface \Device\NPF_
    Section number: 1
  ▸ Interface id: 0 (\Device\NPF_Loopback)
    Encapsulation type: NULL/Loopback (15)
    Arrival Time: Feb 26, 2026 22:24:09.792817700 Egypt Standard Time
    UTC Arrival Time: Feb 26, 2026 20:24:09.792817700 UTC
    Epoch Arrival Time: 1772137449.792817700
    [Time shift for this packet: 0.000000000 seconds]
    [Time delta from previous captured frame: 2.568154300 seconds]
    [Time since reference or first frame: 2 minutes, 27.282637600 seconds]
    Frame Number: 631
    Frame Length: 58 bytes (464 bits)
    Capture Length: 58 bytes (464 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: null:ip:udp:data]
    Character encoding: ASCII (0)
    [Coloring Rule Name: UDP]
    [Coloring Rule String: udp]
▸ Null/Loopback
▸ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
```

```
0000  02 00 00 00 45 00 00 36  2b 74 00 00 80 11 11 41   ····E··6 +t·····A
0010  7f 00 00 01 7f 00 00 01  dd dc 30 39 00 22 e5 05   ········ ··09·"··
0020  41 68 65 6c 6c 6f 20 66  72 6f 6d 20 74 68 65 20   Ahello f rom the
0030  6f 74 68 65 72 20 73 69  64 65                     other si de
```

Bytes 32-57: Data (data.data)

Packets: 3654 · Displayed: 2 (0.1%)

Profile: Default

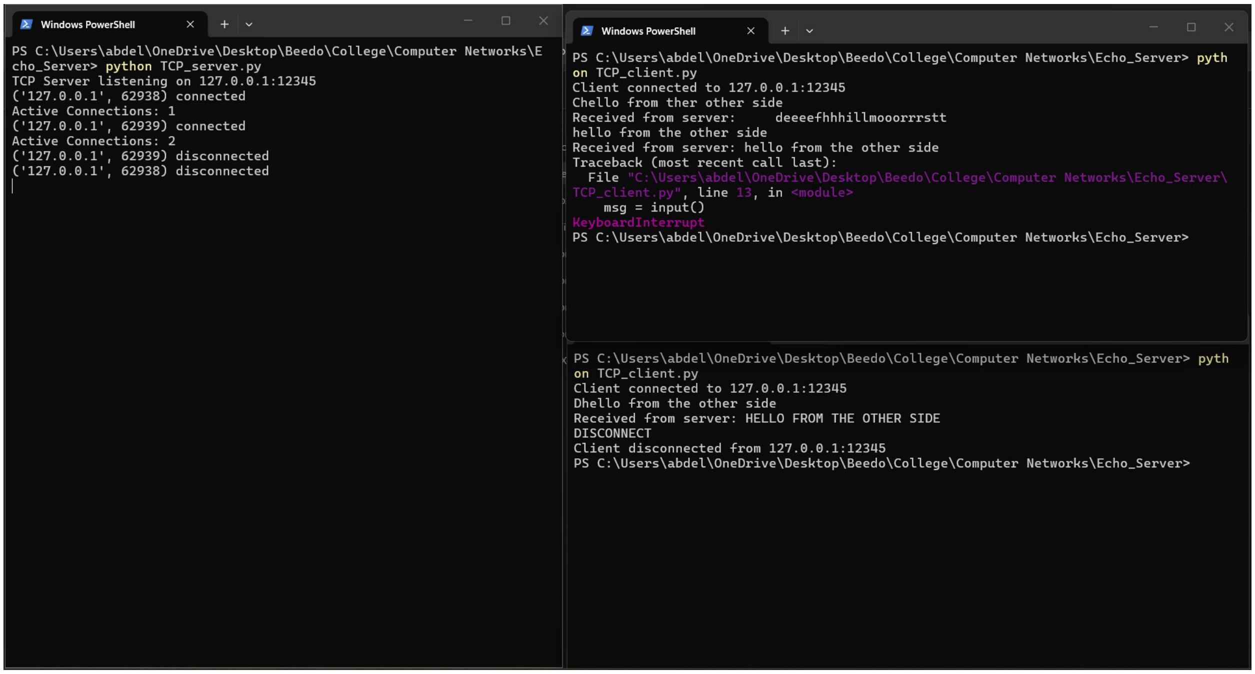
## Server code

```python
import socket

IP_address = "127.0.0.1"
port = 12345

s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
s.bind((IP_address, port))

print(f"UDP Server listening on {IP_address}:{port}")

while True:
    data, addr = s.recvfrom(1024)

    msg = data.decode('utf-8').strip()

    if not msg:
        continue    # Skip empty messages to prevent crashes on msg[0]

    # Used "".join() to convert the sorted list back into a string
    if msg[0] == 'A':
        new_msg = "".join(sorted(msg[1:], reverse=True))
    elif msg[0] == 'C':
        new_msg = "".join(sorted(msg[1:]))
    elif msg[0] == 'D':
        new_msg = msg[1:].upper()
    else:
        new_msg = msg

    s.sendto(new_msg.encode('utf-8'), addr)
```

## Client code

```
import socket

IP_address = "127.0.0.1"

port = 12345

msg = "Ahello from the other side"

s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

print(f"UDP client sending to {IP_address}:{port}")

s.sendto(msg.encode('utf-8'), (IP_address, port))

data, server_addr = s.recvfrom(1024)

print(f"Received from server: {data.decode('utf-8')}")

s.close()
```

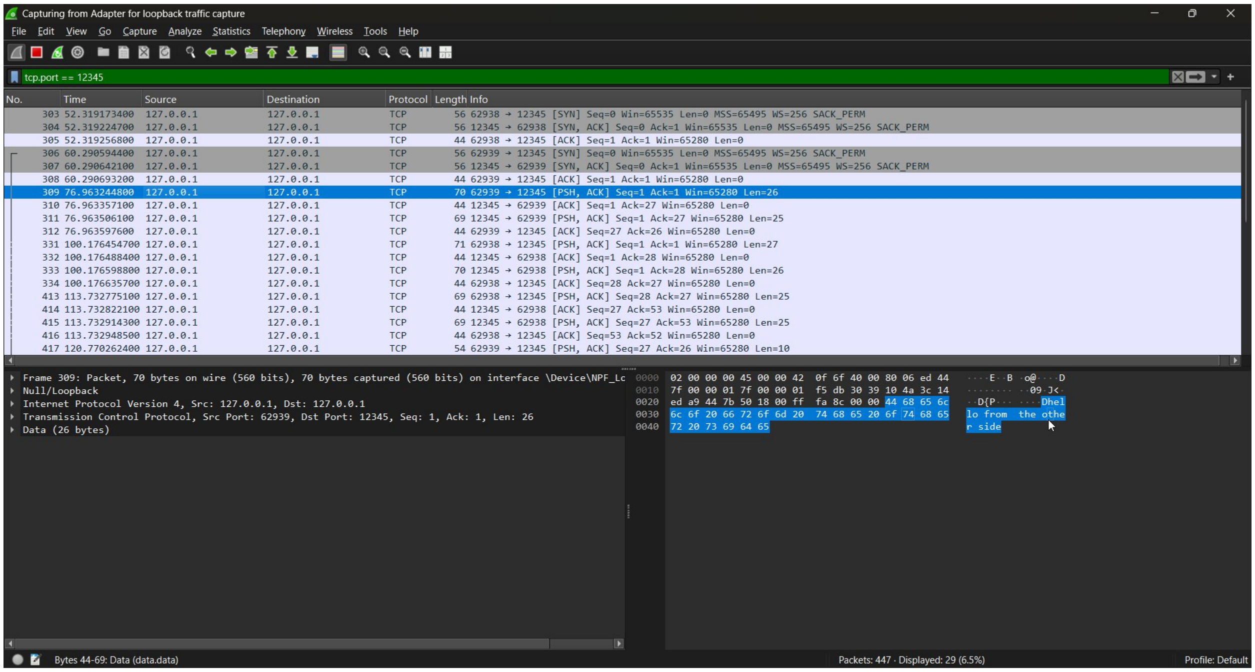
# TCP Connection



```
PS C:\Users\abdel\OneDrive\Desktop\Beedo\College\Computer Networks\E
cho_Server> python TCP_server.py
TCP Server listening on 127.0.0.1:12345
('127.0.0.1', 62938) connected
Active Connections: 1
('127.0.0.1', 62939) connected
Active Connections: 2
('127.0.0.1', 62939) disconnected
('127.0.0.1', 62938) disconnected
```

```
PS C:\Users\abdel\OneDrive\Desktop\Beedo\College\Computer Networks\Echo_Server> pyth
on TCP_client.py
Client connected to 127.0.0.1:12345
Chello from ther other side
Received from server:     deeeefhhhillmooorrrstt
hello from the other side
Received from server: hello from the other side
Traceback (most recent call last):
  File "C:\Users\abdel\OneDrive\Desktop\Beedo\College\Computer Networks\Echo_Server\
TCP_client.py", line 13, in <module>
    msg = input()
KeyboardInterrupt
PS C:\Users\abdel\OneDrive\Desktop\Beedo\College\Computer Networks\Echo_Server>
```

```
PS C:\Users\abdel\OneDrive\Desktop\Beedo\College\Computer Networks\Echo_Server> pyth
on TCP_client.py
Client connected to 127.0.0.1:12345
Dhello from the other side
Received from server: HELLO FROM THE OTHER SIDE
DISCONNECT
Client disconnected from 127.0.0.1:12345
PS C:\Users\abdel\OneDrive\Desktop\Beedo\College\Computer Networks\Echo_Server>
```


Capturing from Adapter for loopback traffic capture
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.port == 12345

```
No.  Time          Source     Destination  Protocol Length Info
 303 52.319173400  127.0.0.1  127.0.0.1    TCP      56 62938 → 12345 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
 304 52.319224700  127.0.0.1  127.0.0.1    TCP      56 12345 → 62938 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
 305 52.319256800  127.0.0.1  127.0.0.1    TCP      44 62938 → 12345 [ACK] Seq=1 Ack=1 Win=65280 Len=0
 306 60.290594400  127.0.0.1  127.0.0.1    TCP      56 62939 → 12345 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
 307 60.290642100  127.0.0.1  127.0.0.1    TCP      56 12345 → 62939 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
 308 60.290693200  127.0.0.1  127.0.0.1    TCP      44 62939 → 12345 [ACK] Seq=1 Ack=1 Win=65280 Len=0
 309 76.963244800  127.0.0.1  127.0.0.1    TCP      70 62939 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=65280 Len=26
 310 76.963357100  127.0.0.1  127.0.0.1    TCP      44 12345 → 62939 [ACK] Seq=1 Ack=27 Win=65280 Len=0
 311 76.963506100  127.0.0.1  127.0.0.1    TCP      69 12345 → 62939 [PSH, ACK] Seq=1 Ack=27 Win=65280 Len=25
 312 76.963597600  127.0.0.1  127.0.0.1    TCP      44 62939 → 12345 [ACK] Seq=27 Ack=26 Win=65280 Len=0
 331 100.176454700 127.0.0.1  127.0.0.1    TCP      71 62938 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=65280 Len=27
 332 100.176488400 127.0.0.1  127.0.0.1    TCP      44 12345 → 62938 [ACK] Seq=1 Ack=28 Win=65280 Len=0
 333 100.176598800 127.0.0.1  127.0.0.1    TCP      70 12345 → 62938 [PSH, ACK] Seq=1 Ack=28 Win=65280 Len=26
 334 100.176635700 127.0.0.1  127.0.0.1    TCP      44 62938 → 12345 [ACK] Seq=28 Ack=27 Win=65280 Len=0
 413 113.732775100 127.0.0.1  127.0.0.1    TCP      69 62938 → 12345 [PSH, ACK] Seq=28 Ack=27 Win=65280 Len=25
 414 113.732822100 127.0.0.1  127.0.0.1    TCP      44 12345 → 62938 [ACK] Seq=27 Ack=53 Win=65280 Len=0
 415 113.732914300 127.0.0.1  127.0.0.1    TCP      69 12345 → 62938 [PSH, ACK] Seq=27 Ack=53 Win=65280 Len=25
 416 113.732948500 127.0.0.1  127.0.0.1    TCP      44 62938 → 12345 [ACK] Seq=53 Ack=52 Win=65280 Len=0
 417 120.770262400 127.0.0.1  127.0.0.1    TCP      54 62939 → 12345 [PSH, ACK] Seq=27 Ack=26 Win=65280 Len=10
```

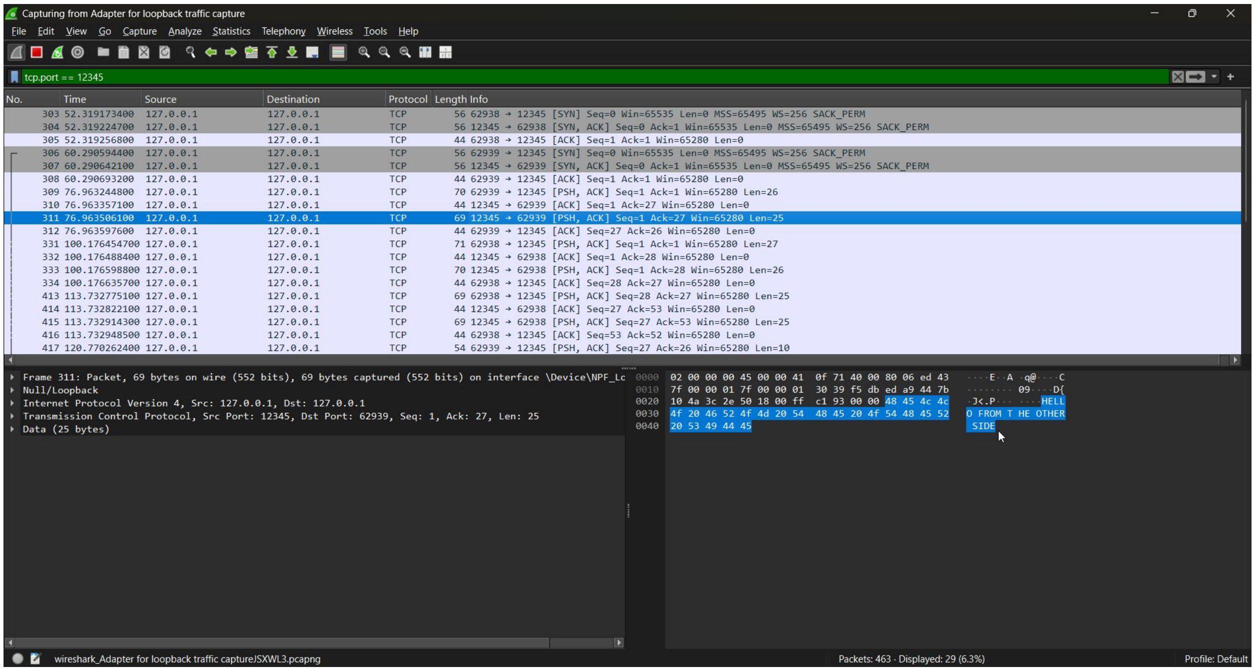
```
Frame 309: Packet, 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface \Device\NPF_Lc
Null/Loopback
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 62939, Dst Port: 12345, Seq: 1, Ack: 1, Len: 26
Data (26 bytes)
```

```
0000  02 00 00 00 45 00 00 42  0f 6f 40 00 80 06 ed 44   ····E··B ·o@····D
0010  7f 00 00 01 7f 00 00 01  f5 db 30 39 10 4a 3c 14   ········ ··09·J<·
0020  ed a9 44 7b 50 18 00 ff  fa 8c 00 00 44 68 65 6c   ··D{P··· ····Dhel
0030  6c 6f 20 66 72 6f 6d 20  74 68 65 20 6f 74 68 65   lo from  the othe
0040  72 20 73 69 64 65                                  r side
```

Bytes 44-69: Data (data.data)

Packets: 447 · Displayed: 29 (6.5%)

Profile: Default


Capturing from Adapter for loopback traffic capture
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.port == 12345

| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 303 | 52.319173400 | 127.0.0.1 | 127.0.0.1 | TCP | 56 | 62938 → 12345 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM |
| 304 | 52.319224700 | 127.0.0.1 | 127.0.0.1 | TCP | 56 | 12345 → 62938 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM |
| 305 | 52.319256800 | 127.0.0.1 | 127.0.0.1 | TCP | 44 | 62938 → 12345 [ACK] Seq=1 Ack=1 Win=65280 Len=0 |
| 306 | 60.290594400 | 127.0.0.1 | 127.0.0.1 | TCP | 56 | 62939 → 12345 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM |
| 307 | 60.290642100 | 127.0.0.1 | 127.0.0.1 | TCP | 56 | 12345 → 62939 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM |
| 308 | 60.290693200 | 127.0.0.1 | 127.0.0.1 | TCP | 44 | 62939 → 12345 [ACK] Seq=1 Ack=1 Win=65280 Len=0 |
| 309 | 76.963244800 | 127.0.0.1 | 127.0.0.1 | TCP | 70 | 62939 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=65280 Len=26 |
| 310 | 76.963357100 | 127.0.0.1 | 127.0.0.1 | TCP | 44 | 12345 → 62939 [ACK] Seq=1 Ack=27 Win=65280 Len=0 |
| 311 | 76.963506100 | 127.0.0.1 | 127.0.0.1 | TCP | 69 | 12345 → 62939 [PSH, ACK] Seq=1 Ack=27 Win=65280 Len=25 |
| 312 | 76.963597600 | 127.0.0.1 | 127.0.0.1 | TCP | 44 | 62939 → 12345 [ACK] Seq=27 Ack=26 Win=65280 Len=0 |
| 331 | 100.176454700 | 127.0.0.1 | 127.0.0.1 | TCP | 71 | 62938 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=65280 Len=27 |
| 332 | 100.176488400 | 127.0.0.1 | 127.0.0.1 | TCP | 44 | 12345 → 62938 [ACK] Seq=1 Ack=28 Win=65280 Len=0 |
| 333 | 100.176598800 | 127.0.0.1 | 127.0.0.1 | TCP | 70 | 12345 → 62938 [PSH, ACK] Seq=1 Ack=28 Win=65280 Len=26 |
| 334 | 100.176635700 | 127.0.0.1 | 127.0.0.1 | TCP | 44 | 62938 → 12345 [ACK] Seq=28 Ack=27 Win=65280 Len=0 |
| 413 | 113.732775100 | 127.0.0.1 | 127.0.0.1 | TCP | 69 | 62938 → 12345 [PSH, ACK] Seq=28 Ack=27 Win=65280 Len=25 |
| 414 | 113.732822100 | 127.0.0.1 | 127.0.0.1 | TCP | 44 | 12345 → 62938 [ACK] Seq=27 Ack=53 Win=65280 Len=0 |
| 415 | 113.732914300 | 127.0.0.1 | 127.0.0.1 | TCP | 69 | 12345 → 62938 [PSH, ACK] Seq=27 Ack=53 Win=65280 Len=25 |
| 416 | 113.732948500 | 127.0.0.1 | 127.0.0.1 | TCP | 44 | 62938 → 12345 [ACK] Seq=53 Ack=52 Win=65280 Len=0 |
| 417 | 120.770262400 | 127.0.0.1 | 127.0.0.1 | TCP | 54 | 62939 → 12345 [PSH, ACK] Seq=27 Ack=26 Win=65280 Len=10 |

```
Frame 311: Packet, 69 bytes on wire (552 bits), 69 bytes captured (552 bits) on interface \Device\NPF_Lc
Null/Loopback
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 12345, Dst Port: 62939, Seq: 1, Ack: 27, Len: 25
Data (25 bytes)
```

```
0000  02 00 00 00 45 00 00 41  0f 71 40 00 80 06 ed 43   ····E··A ·q@····C
0010  7f 00 00 01 7f 00 00 01  30 39 f5 db ed a9 44 7b   ········ 09····D{
0020  10 4a 3c 2e 50 18 00 ff  c1 93 00 00 48 45 4c 4c   ·J<.P··· ····HELL
0030  4f 20 46 52 4f 4d 20 54  48 45 20 4f 54 48 45 52   O FROM T HE OTHER
0040  20 53 49 44 45                                      SIDE
```

wireshark_Adapter for loopback traffic captureJSXWL3.pcapng

Packets: 463 · Displayed: 29 (6.3%)

Profile: Default

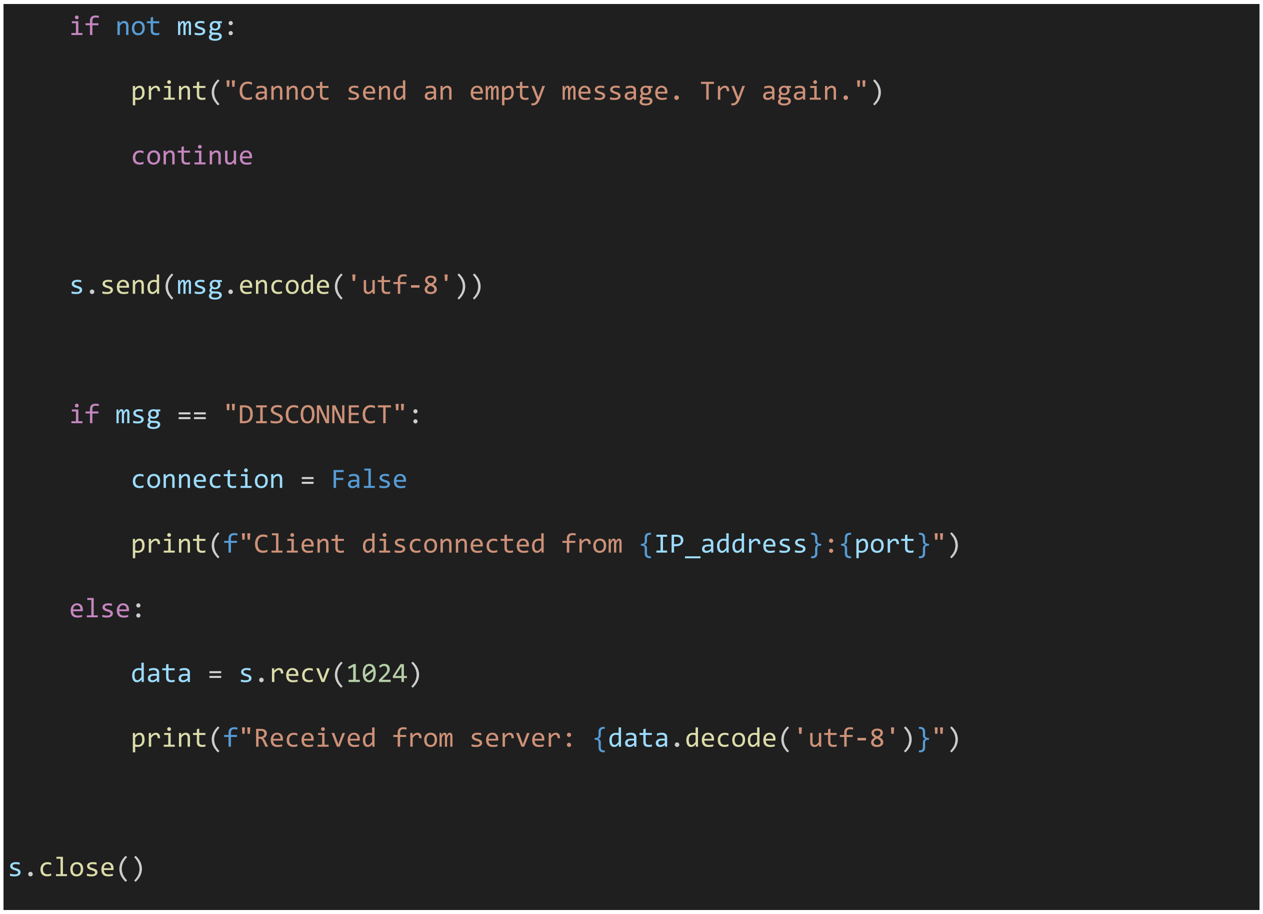
## Server code

```
import socket

import threading



def handle_client(conn, addr):

    print(f"{addr} connected")


    connection = True


    while connection:

        try:

            msg = conn.recv(1024).decode('utf-8')


            if msg == "DISCONNECT":

                connection = False
                print(f"{addr} disconnected")

            # In TCP empty msg means connection failed

            elif not msg:

                connection = False

                print(f"{addr} disconnected")

            else:

                if msg[0] == 'A':

                    new_msg = "".join(sorted(msg[1:], reverse=True))

                elif msg[0] == 'C':

                    new_msg = "".join(sorted(msg[1:]))

                elif msg[0] == 'D':

                    new_msg = msg[1:].upper()

                else:

                    new_msg = msg


                conn.send(new_msg.encode('utf-8'))

        except ConnectionResetError:

            connection = False

            print(f"{addr} disconnected")


    conn.close()


IP_address = "127.0.0.1"

port = 12345


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

s.bind((IP_address, port))
```

```
s.listen()

print(f"TCP Server listening on {IP_address}:{port}")

while True:
    conn, addr = s.accept()
    thread = threading.Thread(target=handle_client, args=(conn, addr))
    thread.start()
    print(f"Active Connections: {threading.active_count() - 1}")
```

## Client code

```
import socket

IP_address = "127.0.0.1"
port = 12345

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((IP_address, port))
print(f"Client connected to {IP_address}:{port}")

connection = True

while connection:
    msg = input()
```



```
    if not msg:
        print("Cannot send an empty message. Try again.")
        continue

    s.send(msg.encode('utf-8'))

    if msg == "DISCONNECT":
        connection = False
        print(f"Client disconnected from {IP_address}:{port}")
    else:
        data = s.recv(1024)
        print(f"Received from server: {data.decode('utf-8')}")

s.close()
```