



# **CSE 336 - Computer Networks**

## **Lab 1**

**Submitted by:**

Ahmed Hani Ahmed 7387

**Due on:**

March 1<sup>st</sup>, 2024

# Python Socket Server

## Problem statement:

You are required to create a server using Python's socket server module that counts the number of words and characters in a string sent by a client. The server will also perform specific operations based on the first character of the string.

## Test cases used:

- Test 1
  - o Input: "Wpython Socket Server"
  - o Expected Output: "The number of words is 3"
- Test 2
  - o Input: "LpythonSocketServer"
  - o Expected Output: "The number of lowercase letters is 16"
- Test 3
  - o Input: "UPYTHONSOCKETSERVER"
  - o Expected Output: "The number of uppercase letters is 18"
- Test 4
  - o Input: "R1234567890"
  - o Expected Output: "The number of numeric characters is 10"
- Test 5
  - o Input: "TpythonSocketServer123"
  - o Expected Output: "The total number of characters is 22"
- Test 6
  - o Input: "pythonSocketServer123"
  - o Expected Output: "pythonSocketServer123"

## Outputs:

### Server side

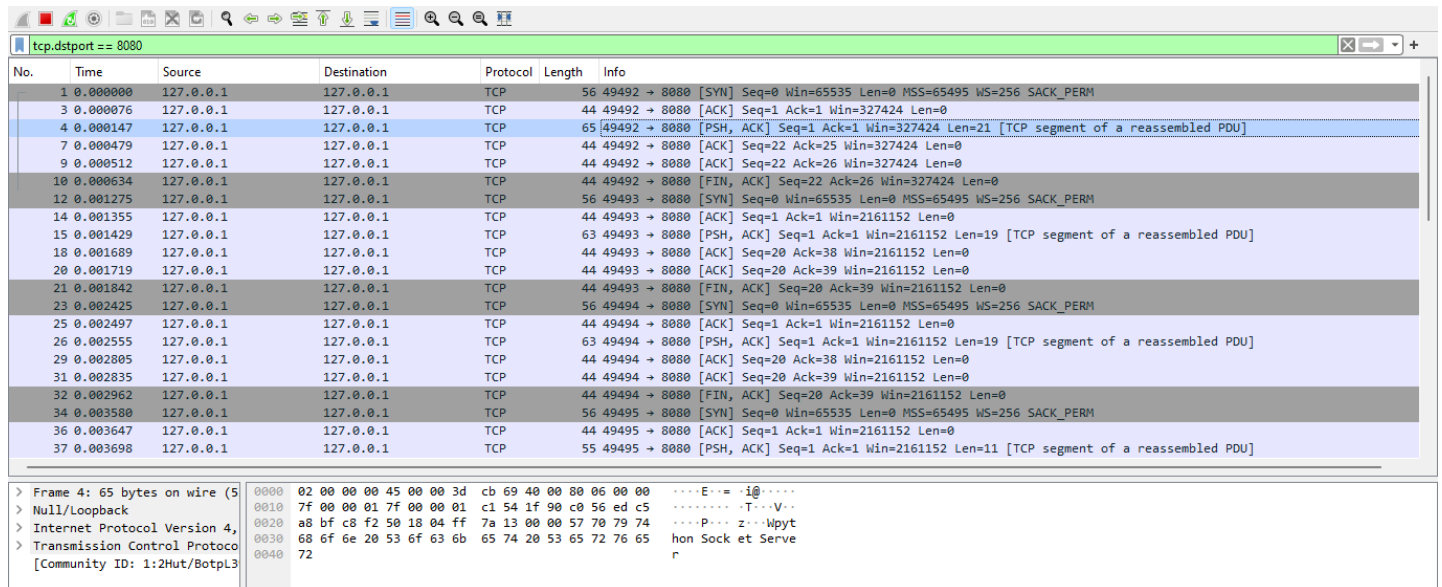
```
PS G:\eng\Term 8\Computer Networks\Labs\lab1> python server.py
recieved message: "python Socket Server" with command letter: "W"
recieved message: "pythonSocketServer" with command letter: "L"
recieved message: "PYTHONSOCKETSERVER" with command letter: "U"
recieved message: "1234567890" with command letter: "R"
recieved message: "pythonSocketServer123" with command letter: "T"
recieved message: "ythonSocketServer123" with command letter: "p"
█
```

### Client side

```
PS G:\eng\Term 8\Computer Networks\Labs\lab1> python client.py
Sending message: Wpython Socket Server
Server response: The number of words is 3
Sending message: LpythonSocketServer
Server response: The number of lowercase letters is 16
Sending message: UPYTHONSOCKETSERVER
Server response: The number of uppercase letters is 18
Sending message: R1234567890
Server response: The number of numeric characters is 10
Sending message: TpythonSocketServer123
Server response: 21
Sending message: pythonSocketServer123
Server response: pythonSocketServer123
PS G:\eng\Term 8\Computer Networks\Labs\lab1> █
```

# Using wireshark:

Using wireshark to sniff packets going from server to client and vice-versa I applied a filter on the port I used which is 8080 and viewed the packets



The image shows a Wireshark packet capture interface. The top bar displays the filter 'tcp.dstport == 8080'. The packet list pane shows 37 captured packets, all of which are TCP segments. The packet details pane shows the structure of packet 4, which is a SYN packet from 127.0.0.1 to 127.0.0.1 on port 8080. The packet bytes pane shows the raw data of the packet, including the Ethernet II header, Internet Protocol Version 4 header, and Transmission Control Protocol header.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	127.0.0.1	127.0.0.1	TCP	56	49492 → 8080 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
3	0.000076	127.0.0.1	127.0.0.1	TCP	44	49492 → 8080 [ACK] Seq=1 Ack=1 Win=327424 Len=0
4	0.000147	127.0.0.1	127.0.0.1	TCP	65	49492 → 8080 [PSH, ACK] Seq=1 Ack=1 Win=327424 Len=21 [TCP segment of a reassembled PDU]
7	0.000479	127.0.0.1	127.0.0.1	TCP	44	49492 → 8080 [ACK] Seq=22 Ack=25 Win=327424 Len=0
9	0.000512	127.0.0.1	127.0.0.1	TCP	44	49492 → 8080 [ACK] Seq=22 Ack=26 Win=327424 Len=0
10	0.000634	127.0.0.1	127.0.0.1	TCP	44	49492 → 8080 [FIN, ACK] Seq=22 Ack=26 Win=327424 Len=0
12	0.001275	127.0.0.1	127.0.0.1	TCP	56	49493 → 8080 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
14	0.001355	127.0.0.1	127.0.0.1	TCP	44	49493 → 8080 [ACK] Seq=1 Ack=1 Win=2161152 Len=0
15	0.001429	127.0.0.1	127.0.0.1	TCP	63	49493 → 8080 [PSH, ACK] Seq=1 Ack=1 Win=2161152 Len=19 [TCP segment of a reassembled PDU]
18	0.001689	127.0.0.1	127.0.0.1	TCP	44	49493 → 8080 [ACK] Seq=20 Ack=38 Win=2161152 Len=0
20	0.001719	127.0.0.1	127.0.0.1	TCP	44	49493 → 8080 [ACK] Seq=20 Ack=39 Win=2161152 Len=0
21	0.001842	127.0.0.1	127.0.0.1	TCP	44	49493 → 8080 [FIN, ACK] Seq=20 Ack=39 Win=2161152 Len=0
23	0.002425	127.0.0.1	127.0.0.1	TCP	56	49494 → 8080 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
25	0.002497	127.0.0.1	127.0.0.1	TCP	44	49494 → 8080 [ACK] Seq=1 Ack=1 Win=2161152 Len=0
26	0.002555	127.0.0.1	127.0.0.1	TCP	63	49494 → 8080 [PSH, ACK] Seq=1 Ack=1 Win=2161152 Len=19 [TCP segment of a reassembled PDU]
29	0.002805	127.0.0.1	127.0.0.1	TCP	44	49494 → 8080 [ACK] Seq=20 Ack=38 Win=2161152 Len=0
31	0.002835	127.0.0.1	127.0.0.1	TCP	44	49494 → 8080 [ACK] Seq=20 Ack=39 Win=2161152 Len=0
32	0.002962	127.0.0.1	127.0.0.1	TCP	44	49494 → 8080 [FIN, ACK] Seq=20 Ack=39 Win=2161152 Len=0
34	0.003580	127.0.0.1	127.0.0.1	TCP	56	49495 → 8080 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
36	0.003647	127.0.0.1	127.0.0.1	TCP	44	49495 → 8080 [ACK] Seq=1 Ack=1 Win=2161152 Len=0
37	0.003698	127.0.0.1	127.0.0.1	TCP	55	49495 → 8080 [PSH, ACK] Seq=1 Ack=1 Win=2161152 Len=11 [TCP segment of a reassembled PDU]

> Frame 4: 65 bytes on wire (51840 bits) captured on interface eth0  
> Null/Loopback  
> Internet Protocol Version 4, Src: 127.0.0.1, Destination: 127.0.0.1  
> Transmission Control Protocol, Seq: 1, Win: 0, Len: 0  
[Community ID: 1:2Hut/BotpL3]