

TCP PROJECT REPORT

Run the program:

```
shihao@LAPTOP-73HD8DLJ: ~/tcp_project
shihao@LAPTOP-73HD8DLJ: ~/tcp_project$ ./server
```

```
shihao@LAPTOP-73HD8DLJ: ~/tcp_project
shihao@LAPTOP-73HD8DLJ: ~/tcp_project$ ./client -h 127.0.0.1 -p 8080
```

RESULT: (3 client)

```

shihao@LAPTOP-73HD8DLU:~/tcp.project
182 |     fprintf(stderr, "pthread_create: %s\n", strerror(rc));
server.c:182:46: warning: format ‘%s’ expects argument of type ‘char *’, but argument 3
has type ‘int’ [-Wformat=] 182 |         fprintf(stderr, "pthread_create: %s\n", strerror
+rc);
|           |
|           +-----+
|           int
|           %
|           %

make: *** [Makefile-7: server] Error 1
shihao@LAPTOP-73HD8DLU:~/tcp.project$ code .
shihao@LAPTOP-73HD8DLU:~/tcp.project$ make
gcc -Wall -Wextra -O2 server.c -o server
shihao@LAPTOP-73HD8DLU:~/tcp.project$ ./server
server: waiting for connections...
server: got connection from 127.0.0.1
server: got connection from 127.0.0.1
server: got connection from 127.0.0.1

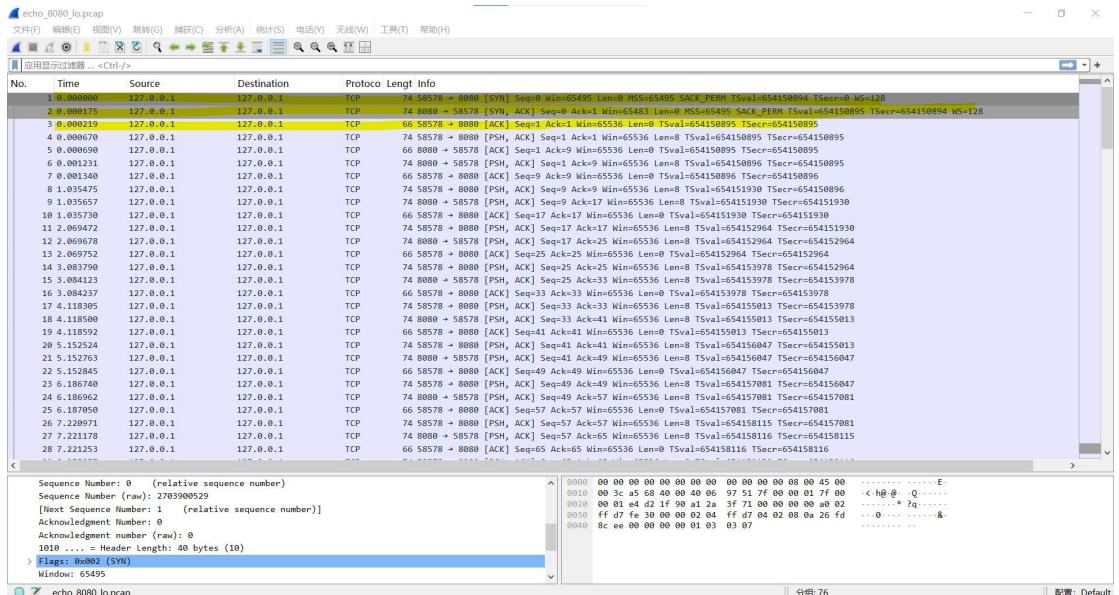
client: recv:count=196
client: sent: count=197
client: recv:count=197
client: sent: count=198
client: recv:count=198
client: sent: count=199
client: recv:count=199
client: sent: count=200
client: recv:count=200
client: sent: count=201
client: recv:count=201
client: sent: count=202
client: recv:count=202
client: sent: count=203
client: recv:count=203
client: sent: count=204
client: recv:count=204
client: sent: count=205
client: recv:count=205
client: sent: count=206
client: recv:count=206
client: sent: count=207
client: recv:count=207

client: sent: count=31
client: recv:count=31
client: sent: count=32
client: recv:count=32
client: sent: count=33
client: recv:count=33
client: sent: count=34
client: recv:count=34
client: sent: count=35
client: recv:count=35
client: sent: count=36
client: recv:count=36
client: sent: count=37
client: recv:count=37
client: sent: count=38
client: recv:count=38
client: sent: count=39
client: recv:count=39
client: sent: count=40
client: recv:count=40
client: sent: count=41
client: recv:count=41

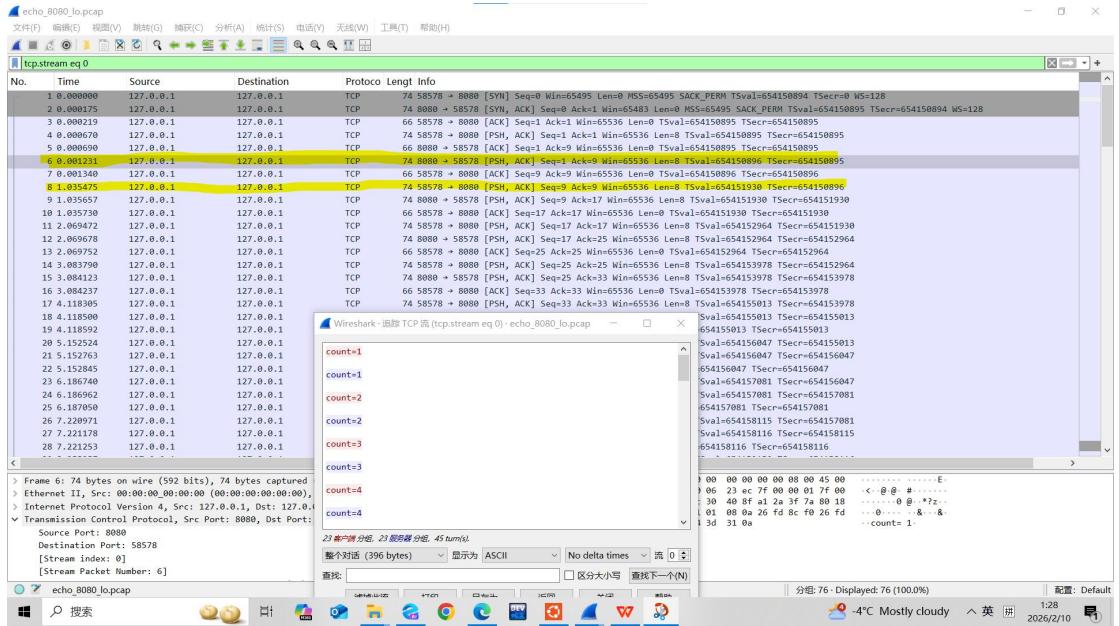
client: recv:count=14
client: sent: count=15
client: recv:count=15
client: sent: count=16
client: recv:count=16
client: sent: count=17
client: recv:count=17
client: sent: count=18
client: recv:count=18
client: sent: count=19
client: recv:count=19
client: sent: count=20
client: recv:count=20
client: sent: count=21
client: recv:count=21
client: sent: count=22
client: recv:count=22
client: sent: count=23
client: recv:count=23
client: sent: count=24
client: recv:count=24

```

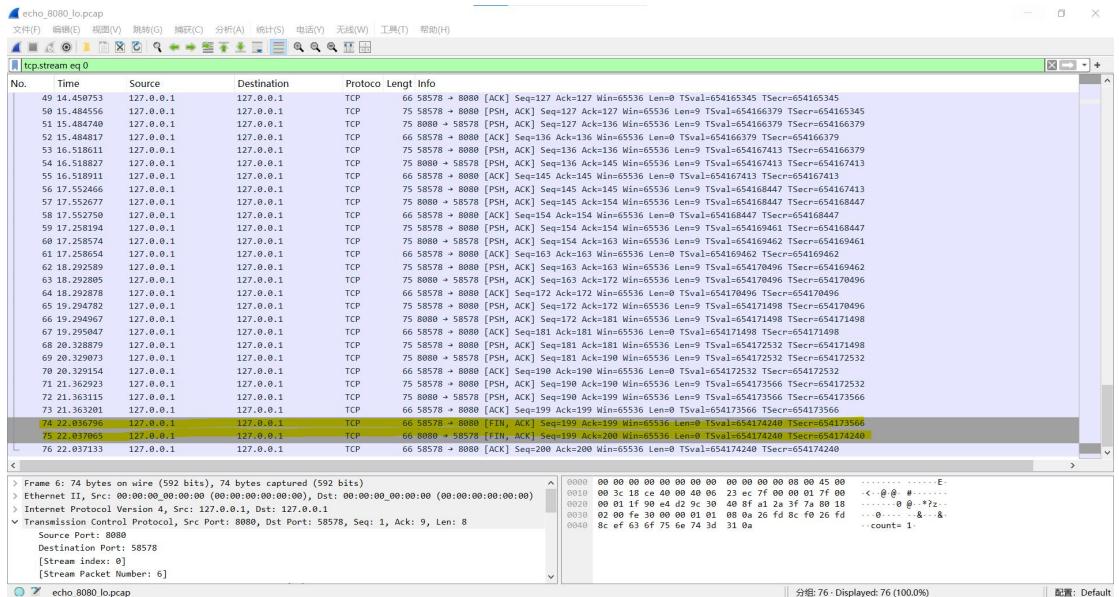
packet capture in Wireshark:



TCP 3-way handshake: the client initiates with SYN, the server replies SYN-ACK, and the client confirms with ACK, establishing the connection.



Echo data: The client sends application payload to the server (PSH, ACK with Len=8), and the server echoes the same payload back to the client (PSH, ACK in the opposite direction).



Graceful close (FIN): the client initiates termination with FIN, ACK, the server responds with FIN, ACK, and the final ACK completes the connection teardown. On the server side, this corresponds to recv() returning 0 (peer performed an orderly shutdown).