

**NAME: A.V.Prasad reddy**

**REG.NO.: 192110462**

**CODE: CSA0734**

**EXPERIMENT: 17**

**AIM:** To implement of server –client using TCP socket pocket programming.

**ALGORITHM:**

1. Create a socket: Create a TCP socket to listen for incoming client connections.
2. Bind the socket to an IP address and port number: Specify the IP address and port number for the server to listen on.
3. Listen for incoming connections: Use the listen() function to start listening for incoming connections from clients.
4. Accept incoming connections: Use the accept() function to accept incoming client connections.
5. Receive data from the client: Use the recv() function to receive data from the client.
6. Process the data: Process the received data as required by the application.
7. Send response to the client: Use the send() function to send a response back to the client.
8. Close the connection: Use the close() function to close the connection with the client.
9. Repeat steps 4 to 8 as required to handle multiple clients.

**PROCEDURE:**

Client Side:

1. Create a socket: Create a TCP socket to connect to the server.
2. Connect to the server: Use the connect() function to connect to the server using the server's IP address and port number.
3. Send data to the server: Use the send() function to send data to the server.
4. Receive response from the server: Use the recv() function to receive a response from the server.
5. Process the response: Process the received response as required by the application.
6. Close the connection: Use the close() function to close the connection with the server.
7. Exit the program: Exit the program as required.

## OUTPUT:

The image displays a Wireshark network traffic capture. The top pane shows a list of captured packets, with packet 321 selected. The middle pane shows the details of the selected packet, which is an HTTP GET request. The bottom pane shows the raw packet data in hexadecimal and ASCII.

**Packets List:**

No.	Time	Source	Destination	Protocol	Length	Info
664.102.091677	172.20.10.3	172.20.10.3	172.20.10.3	TCP	54	50520 → 443 [RST, ACK] Seq=2 Ack=32 Win=0 Len=0
665.102.194744	52.131.199.126	172.20.10.3	172.20.10.3	TCP	54	443 → 50512 [FIN, ACK] Seq=1 Ack=2 Win=2049 Len=0
666.102.194860	172.20.10.3	52.231.199.126	52.231.199.126	TCP	54	50512 → 443 [ACK] Seq=2 Ack=2 Win=514 Len=0
670.108.038058	172.20.10.3	104.65.79.9	104.65.79.9	TCP	54	50535 → 80 [FIN, ACK] Seq=228 Ack=264 Win=131840 Len=0
671.108.038279	2409:4070:4d1f:5b0c::	2001:1900:2381:d0c::	2001:1900:2381:d0c::	TCP	74	50525 → 80 [FIN, ACK] Seq=288 Ack=337 Win=131072 Len=0
672.108.038353	2409:4070:4d1f:5b0c::	2001:1900:2381:200f::	2001:1900:2381:200f::	TCP	74	50536 → 80 [FIN, ACK] Seq=569 Ack=670 Win=130560 Len=0
673.108.038418	2409:4070:4d1f:5b0c::	2001:1900:2381:200f::	2001:1900:2381:200f::	TCP	74	50540 → 80 [FIN, ACK] Seq=288 Ack=335 Win=131072 Len=0
674.108.038515	172.20.10.3	8.241.130.254	8.241.130.254	TCP	54	50542 → 80 [FIN, ACK] Seq=288 Ack=349 Win=131840 Len=0
675.108.235813	104.65.79.9	172.20.10.3	172.20.10.3	TCP	54	80 → 50535 [FIN, ACK] Seq=264 Ack=229 Win=64128 Len=0
676.108.235900	172.20.10.3	104.65.79.9	104.65.79.9	TCP	54	50535 → 80 [ACK] Seq=229 Ack=265 Win=131840 Len=0
677.108.238729	2001:1900:2381:d0c::	2409:4070:4d1f:5b0c::	2409:4070:4d1f:5b0c::	TCP	74	80 → 50525 [FIN, ACK] Seq=337 Ack=289 Win=45056 Len=0
678.108.238781	2409:4070:4d1f:5b0c::	2001:1900:2381:d0c::	2001:1900:2381:d0c::	TCP	74	50525 → 80 [ACK] Seq=289 Ack=338 Win=131072 Len=0
679.108.338186	2001:1900:2381:200f::	2409:4070:4d1f:5b0c::	2409:4070:4d1f:5b0c::	TCP	74	80 → 50536 [FIN, ACK] Seq=670 Ack=570 Win=45056 Len=0
680.108.338186	2001:1900:2381:200f::	2409:4070:4d1f:5b0c::	2409:4070:4d1f:5b0c::	TCP	74	80 → 50540 [FIN, ACK] Seq=335 Ack=289 Win=45056 Len=0
681.108.338186	8.241.130.254	172.20.10.3	172.20.10.3	TCP	54	80 → 50542 [FIN, ACK] Seq=349 Ack=289 Win=45056 Len=0
682.108.338277	2409:4070:4d1f:5b0c::	2001:1900:2381:200f::	2001:1900:2381:200f::	TCP	74	50536 → 80 [ACK] Seq=570 Ack=671 Win=130560 Len=0
683.108.338483	2409:4070:4d1f:5b0c::	2001:1900:2381:200f::	2001:1900:2381:200f::	TCP	74	50540 → 80 [ACK] Seq=289 Ack=336 Win=131072 Len=0
684.108.338694	172.20.10.3	8.241.130.254	8.241.130.254	TCP	54	50542 → 80 [ACK] Seq=289 Ack=350 Win=131840 Len=0

**Packet 321 Details:**

- Frame 321: 361 bytes on wire (2888 bits), 361 bytes captured (2888 bits) on interface \Device\NPF\_{3A87...}
- Ethernet II, Src: Chongqin\_3a:1d:45 (74:12:b3:3a:1d:45), Dst: f2:1f:c7:e3:65:64 (f2:1f:c7:e3:65:64)
- Internet Protocol Version 6, Src: 2409:4070:4d1f:5b0c::b02e:8899:8b1d:71fb, Dst: 2001:1900:2381:200f::1f
- Transmission Control Protocol, Src Port: 50536, Dst Port: 80, Seq: 1, Ack: 1, Len: 287
- Hypertext Transfer Protocol

**Raw Data (Hex/ASCII):**

```
0000 f2 1f c7 e3 65 64 74 12 b3 3a 1d 45 86 dd 60 0f ...edt...E...
0010 11 9a 01 33 06 40 24 09 40 70 4d 1f 5b 0c b0 2e ...3 @B[...
0020 88 99 0b 1d 71 fb 20 01 19 00 23 81 20 0f 00 00 ...q...2...
0030 00 00 00 00 01 fe c5 68 00 50 a8 3e ee 8a 22 3d ...h...P>...=
0040 18 f9 50 18 02 01 40 e3 00 00 47 45 54 20 2f 6d ...P...@...GET /m
0050 73 64 6f 77 6e 6c 6f 61 64 2f 75 70 64 61 74 65 sdownload/update
0060 2f 76 33 2f 73 74 61 74 69 63 2f 74 72 75 73 74 /v3/stat/c/trust
0070 65 64 72 2f 65 6e 2f 64 69 73 61 6c 6c 6f 77 65 edr/en/d/issallow
0080 64 63 65 72 74 73 74 6c 2e 63 61 62 3f 65 61 35 dcerts1.cab?ea5
0090 30 63 38 32 35 35 34 31 66 66 33 31 61 20 48 54 0c825541 ff31a HT
00a0 54 50 2f 31 2e 31 0d 0a 43 6f 6e 6e 65 63 74 69 TP/1.1...Connecti
00b0 6f 6e 3a 20 4b 65 65 70 2d 41 6c 69 76 65 0d 0a on: Keep-Alive-
00c0 41 63 65 70 74 3a 20 2a 2f 2a 0d 0a 49 66 2d Accept: */* -If-
00d0 4d 6f 64 69 66 69 65 64 2d 53 69 6e 63 65 3a 20 Modified-Since:
00e0 54 75 65 2c 20 31 36 20 4d 61 72 20 32 30 32 31 Tue, 16 Mar 2021
00f0 20 30 37 3a 33 33 3a 34 32 20 47 4d 54 0d 0a 49 07:33:4 2 GMT. I
0100 66 2d 4e 6f 6e 65 2d 4d 61 74 63 68 3a 20 22 30 f-None-M atch: "0
0110 38 66 35 61 62 30 33 36 31 61 64 37 31 3a 30 22 8f5ab036 1ad71:0"
0120 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 4d 69 -User-A gent: Mi
0130 63 72 6f 73 6f 66 74 2d 43 72 79 70 74 6f 41 50 crosoft- CryptoAP
0140 49 2f 31 30 2e 30 0d 0a 48 6f 73 74 3a 20 63 74 1/10.0 - Host: ct
0150 6c 64 6c 2e 77 69 6e 64 6f 77 73 75 70 64 61 74 1d1wind owsupdat
```

## RESULT:

Therefore implementation of server has been successfully done using TCP socket programming.