

NAME: A.V.Prasad reddy

REG.NO.: 192110462

CODE: CSA0734

EXPERIMENT:18

AIM: To implement of server-client using UDP socket programming.

ALGORITHM:

Server Side:

1. Create a socket: Create a UDP socket to listen for incoming client datagrams.
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. Receive datagrams from clients: Use the `recvfrom()` function to receive datagrams from clients.
4. Process the data: Process the received datagram as required by the application.
5. Send response to the client: Use the `sendto()` function to send a response back to the client, specifying the client's IP address and port number.
6. Repeat steps 3 to 5 as required to handle multiple clients.

PROCEDURE:

Client Side:

1. Create a socket: Create a UDP socket to send datagrams to the server.
2. Send data to the server: Use the `sendto()` function to send datagrams to the server, specifying the server's IP address and port number.
3. Receive response from the server: Use the `recvfrom()` function to receive a response from the server.
4. Process the response: Process the received response as required by the application.
5. Close the socket: Use the `close()` function to close the socket.
6. 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 the first few being DNS queries and responses. The middle pane shows the details of the selected packet (No. 334), which is a DNS query response. The bottom pane shows the raw packet data in hexadecimal and ASCII.

Packet List:

No.	Source	Destination	Protocol	Length	Info
552	fe80::f01f:c7ff:fee...	ff02::fb	MDNS	106	Standard query 0x0000 PTR _companion-link_tcp.local, "QM" question
553	172.20.10.3	224.0.0.251	MDNS	119	Standard query 0x0000 PTR _apple-mobdev_tcp.local, "QM" question PTR 51b6059e_sub_apple-mobdev2_tcp.local, "QM" question
554	172.20.10.3	224.0.0.251	MDNS	511	Standard query response 0x0000 PTR, cache flush LAPTOP-R3A570T1.local PTR, cache flush LAPTOP-R3A570T1.local PTR, cache flush LAP...
555	fe80::199:6450:ef07...	fe80::f01f:c7ff:fee...	DNS	109	Standard query 0x34bc A context-enroll.ccs.mcafee.com
556	fe80::199:6450:ef07...	fe80::f01f:c7ff:fee...	DNS	109	Standard query response 0x34bc A context-enroll.ccs.mcafee.com
557	fe80::199:6450:ef07...	fe80::f01f:c7ff:fee...	DNS	109	Standard query 0x0811 AAAA context-enroll.ccs.mcafee.com
558	fe80::199:6450:ef07...	fe80::f01f:c7ff:fee...	DNS	109	Standard query response 0x0811 AAAA context-enroll.ccs.mcafee.com CNAME context-enroll.ausmcafeeccs.mcafee.com A 44.237.230.37 A 52...
559	172.20.10.3	172.20.10.1	DNS	89	Standard query 0x0811 AAAA context-enroll.ccs.mcafee.com
560	fe80::f01f:c7ff:fee...	fe80::199:6450:ef07...	DNS	232	Standard query response 0x0811 AAAA context-enroll.ccs.mcafee.com CNAME context-enroll.ausmcafeeccs.mcafee.com SOA ns-781.awsdns...
561	172.20.10.3	172.20.10.3	DNS	212	Standard query response 0x0811 AAAA context-enroll.ccs.mcafee.com CNAME context-enroll.ausmcafeeccs.mcafee.com SOA ns-781.awsdns...
562	fe80::199:6450:ef07...	fe80::f01f:c7ff:fee...	DNS	98	Standard query 0x869f A edge.microsoft.com
563	fe80::199:6450:ef07...	fe80::f01f:c7ff:fee...	DNS	98	Standard query 0x3f4d HTTPS edge.microsoft.com
564	fe80::199:6450:ef07...	fe80::f01f:c7ff:fee...	DNS	201	Standard query response 0x869f A edge.microsoft.com CNAME edge-microsoft-com.dual-a-0036.a-msedge.net CNAME dual-a-0036.a-msedge...
565	fe80::199:6450:ef07...	fe80::f01f:c7ff:fee...	DNS	98	Standard query 0x3f4d HTTPS edge.microsoft.com
566	fe80::199:6450:ef07...	fe80::f01f:c7ff:fee...	DNS	225	Standard query response 0x6757 AAAA edge-microsoft-com.dual-a-0036.a-msedge.net CNAME dual-a-0036.a-msedge...
567	fe80::199:6450:ef07...	fe80::f01f:c7ff:fee...	DNS	216	Standard query response 0x3f4d HTTPS edge-microsoft-com.dual-a-0036.a-msedge.net CNAME dual-a-0036.a-msedge...
568	172.20.10.3	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
569	172.20.10.3	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1

Packet Details:

Frame 3: 534 bytes on wire (4272 bits), 534 bytes captured (4272 bits) on interface \Device\NPF_{3AB714...}

Ethernet II, Src: Chongqin_3a:1d:45 (74:12:b3:3a:1d:45), Dst: IPv4multicast_fb (01:00:5e:00:00:fb)

Internet Protocol Version 4, Src: 172.20.10.3, Dst: 224.0.0.251

User Datagram Protocol, Src Port: 5353, Dst Port: 5353

Multicast Domain Name System (response)

Raw Data:

```
0000  01 00 5e 00 00 fb 74 12 b3 3a 1d 45 08 00 45 00  ...t...E...E...
0010  02 08 d6 86 00 00 ff 11 4c 4b ac 14 0a 03 e0 00  ...LK.....
0020  00 fb 14 e9 14 e9 01 f4 9c 24 00 00 04 00 00 00  ...$.....
0030  00 08 00 00 00 06 01 33 02 31 30 02 32 30 03 31  ...-3 10 20 1
0040  37 32 07 69 6e 2d 61 64 64 72 04 61 72 70 61 00  72-in-ad dr-arp
0050  00 0c 00 01 00 00 00 78 00 17 0f 4c 41 50 54 4f  ...x...LAPTO
0060  50 2d 52 33 41 35 37 4f 54 49 05 0c 6f 63 61 6c  P-R3A570 T1 local
0070  00 01 42 01 45 01 36 01 44 01 42 01 38 01 42 01  B E 6 D B 8 8
0080  46 01 31 01 32 01 42 01 44 01 36 01 44 01 38 01  F 1 2 8 D 6 D 8
0090  33 01 43 01 30 01 42 01 35 01 46 01 31 01 44 01  3 C 0 8 5 F 1 D
00a0  34 01 30 01 37 01 30 01 34 01 39 01 30 01 34 01  4 0 7 0 4 9 0 4
00b0  32 03 69 70 36 c0 20 00 0c 00 01 00 00 78 00 2  ip6: .....x
00c0  02 c0 30 01 42 01 46 01 31 01 37 01 44 01 31 01  0 B F 1 7 D 1
00d0  42 01 38 01 39 01 39 01 38 01 38 01 45 01 32 01  8 8 9 9 8 8 E 2
00e0  30 01 42 c0 67 00 0c 00 01 00 00 00 78 00 02 c0  0 8 g: .....x...
00f0  30 01 36 01 37 01 37 01 35 01 37 01 30 01 46 01  0 6 7 7 5 7 0 F
0100  45 01 30 01 35 01 34 01 36 01 39 01 39 01 31 01  C 0 5 4 6 9 9 1
0110  30 01 30 01 30 01 30 01 30 01 30 01 30 01 30 01  0 0 0 0 0 0 0 0
0120  30 01 30 01 30 01 30 01 30 01 30 01 38 01 45 01  0 0 0 0 0 0 8 E
0130  46 c0 87 00 0c 00 01 00 00 78 00 02 c0 30 c0  F: .....x...0
0140  30 00 01 80 01 00 00 00 78 00 04 ac 14 0a 03 c0  0: .....x...0
0150  30 00 1c 80 01 00 00 00 78 00 10 24 09 40 70 4d  0: .....x...$ 8gM
```

RESULT:

Therefore implementation of server—client using UDP socket programming.