

Assignment 5

Computer Networks

Vikas Gola : 2016UCS0023

October 14, 2018

Question 1 Write a brief note on the meaning of each component of both the structures `in_addr` and `sockaddr_in`.

Answer

- **struct in_addr:** This structure contains only one member which is `s_addr` of type `u_long` type for storing the 32 bits IP address which can be `netID` or `hostID`.
- **struct sockaddr_in:** This structure contains 4 members total of 16 bytes. `sin_family` that is of 2 bytes of type `short` and used to store the family type of socket e.g. `AF_INET`, `AF_UNIX`. `sin_port` that is also of 2 bytes of type `u_short` (unsigned short) stores port number of host machine e.g. 23, 9939. `sin_addr` is of 4 byte of type `in_addr` that already has been discussed in last point which used to store the IP address of host machine. `sin_zero` of 8 byte of type `char` array which may be used in future or can be used to extra info about address.

Question 2 Write a program with two functions: (1) First function takes an IPA as argument from the user on the command line (i.e. in the form of a string) and populates the structure `sockaddr_in` with the IPA in the NBO using the function `inet_aton(const char*cp, struct in_addr*inp)` (2) Second, reads the IPA from an already populated structure and prints the IPA in the string format using the counterpart function `inet_ntoa(const char*cp, struct in_addr*inp)`

Answer Both function has been implemented in program with filename `q2.c`

- (1) First function name is `void populateSockAddr(struct sockaddr_in *addr)`.
- (2) Second function name is `void printPopulatedAddr(struct sockaddr_in addr)`.

Question 3 As discussed in class, the Intel machines are little-endian machines whereas the NBO is big-endian. Using the logic discussed in the class, write a program that detects and prints whether a machine is a little-endian or a big-endian. Write another program that would - convert the same to network byte order and given an IPA and port number in network byte order, would convert the same to the host byte order.

Answer First and Second program is written in file with name q3.1.c and q3.2.c respectively

Question 4 Write a tcp client program that does not use bind() function, but relies upon the kernel to provide an IP address and port number. Since the IP address and port number are provided by the kernel, write a function which obtains the local socket name using the call viz. int getsockname(int sockfd, struct sockaddr* localaddr, socklen_t* addrlen) and displays the allocated IP address and port numbers. Look up the man pages of getsockname to learn how to use this function.

Answer Program is written in file with name q4.c

Question 5 Write a program that takes multiple command line arguments (which are machine names) and uses appropriate function to print their IP addresses. Write its counterpart that takes multiple command line arguments (which are IP addresses) and uses appropriate function to print their hostnames. Include an additional option which determines which function to invoke and which argument to take.

Answer Program is written in file with name q5.c

Question 6 Modify the TCP and UDP daytime service client illustrated in the lab to display the hostname and the port number of the server, also along with the time of the day. Recollect that the daytime client uses the daytime service running on port no 13 on the server to fetch the time of the day from the server and display the same on the standard output on the client side.

Answer Programs of daytime service with TCP are written in files with name q6.tcpclient.c and q6.tcpsrvr.c
and Programs of daytime service with UDP are written in files with name q6.udpcient.c and q6.udpsrvr.c

Question 7 Now modify the client in the problem number 6 that requests an iterative TCP server on port no 99XX (XX are the last two digits of your IIT

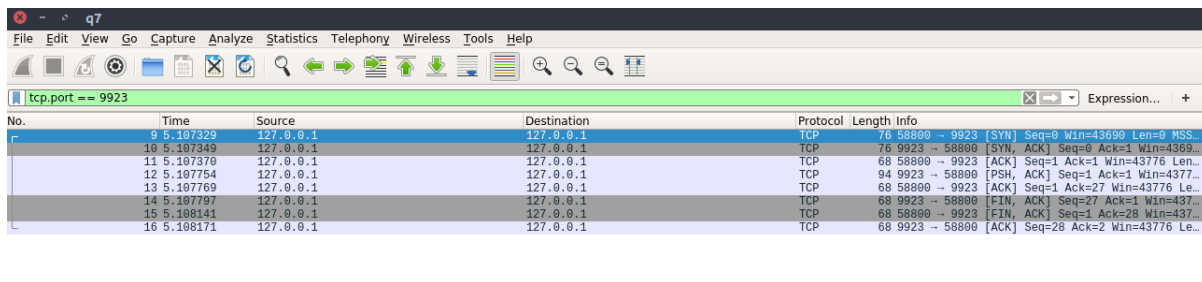
enrollement number) to send it the time of the day. The server upon receiving the client request responds with the time of the day, which the client outputs on the standard output, along with the hostname and the port number of the server. Now start the tcpdump with appropriate arguments (refer \$man tcpdump) to capture packets from or to a remote host. After you have started the TCP server and the TCP client terminate tcpdump, examine its output and make appropriate inferences based on the output.

Answer Total packets captured : 8

3 packets for 3 way handshake

1 packet for data sending

4 packets for 4 way handshake.



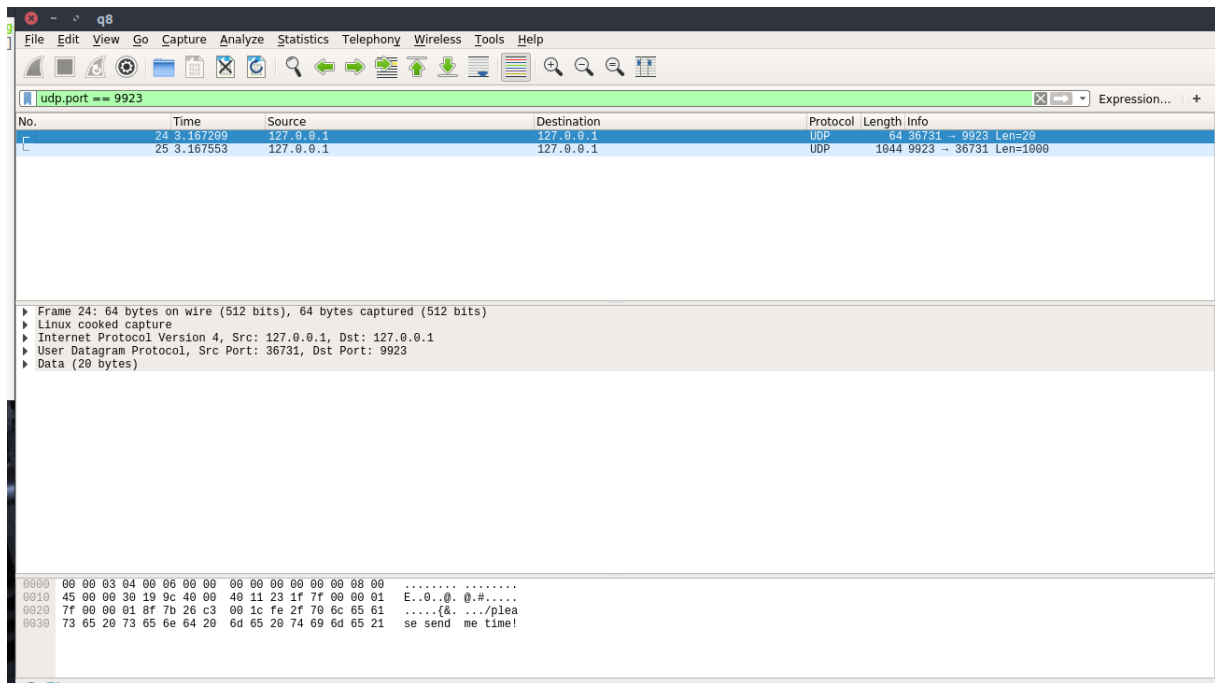
No.	Time	Source	Destination	Protocol	Length	Info
9	5.197329	127.0.0.1	127.0.0.1	TCP	76	58800 → 9923 [SYN] Seq=0 Win=43690 Len=0 MSS...
10	5.197349	127.0.0.1	127.0.0.1	TCP	76	9923 → 58800 [SYN, ACK] Seq=0 Ack=1 Win=43690...
11	5.197370	127.0.0.1	127.0.0.1	TCP	68	58800 → 9923 [ACK] Seq=1 Ack=1 Win=43776 Len=0...
12	5.197754	127.0.0.1	127.0.0.1	TCP	94	9923 → 58800 [PSH, ACK] Seq=1 Ack=1 Win=43776...
13	5.197769	127.0.0.1	127.0.0.1	TCP	68	58800 → 9923 [ACK] Seq=1 Ack=27 Win=43776 Len=0...
14	5.197797	127.0.0.1	127.0.0.1	TCP	68	9923 → 58800 [FIN, ACK] Seq=27 Ack=1 Win=437...
15	5.198141	127.0.0.1	127.0.0.1	TCP	68	58800 → 9923 [FIN, ACK] Seq=1 Ack=28 Win=437...
16	5.198171	127.0.0.1	127.0.0.1	TCP	68	9923 → 58800 [ACK] Seq=28 Ack=2 Win=43776 Le...

Question 8 Repeat the exercise in Problem no 7, but now with UDP as the underlying Transport protocol.

Answer Total packets captured : 2

1 packet for sending message "please send me time" or known as hello message from client to server.

1 packet for sending time to client from server.



Question 9 Modify the server program in problem 7 to make it work as a concurrent server.²

Answer Program filename: q9.c