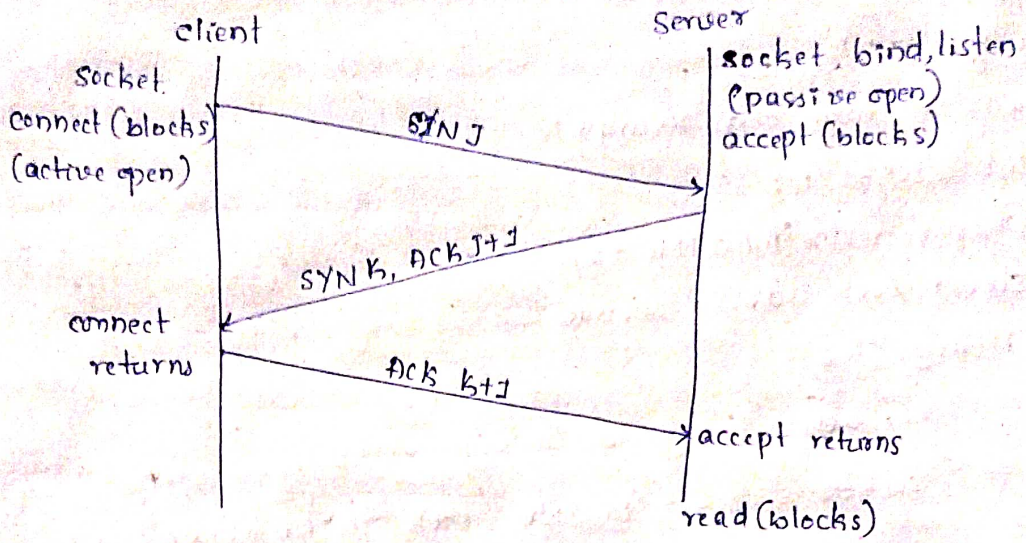
1] Write a program to implement Tcp daytime client

Tcp daytime client :

```
#include "unp.h"
int main (int argc, char **argv)
{
    int sockfd, n;
    char recvline[MAXLINE + 1];
    struct sockaddr_in servaddr;
    if(argc != 2)
        err_quit ("usage: a.out <IPaddress>");
    if ((sockfd = socket (AF_INET, sock_STREAM,0)) <0)
        err_sys ("socket error");
    bzero (&servaddr, sizeof(servaddr));
    servaddr.sin_family = AF_INET;
    servaddr.sin_port = htons (13);  //daytime server
    if (inet_pton (AF_INET, argv[1], &servaddr.sin_addr) <=0).
        err_quit ("inet_pton error for %s", argv[1]);
    if (connect (sockfd, (SA *) &servaddr, sizeof(servaddr)) <0)
        err_sys ("connect error");
    while((n = read (sockfd, recvline, MAXLINE)) >0) {
        recvline[n] =0;      //null terminate
        if (fputs (recvline, stdout) ==EOF)
            err_sys ("fputs error");
    }
    if(n<0)
        err_sys("read error");
    exit(0);
}
```

TCP connection establishment & termination

Three way handshake



The following scenario occurs when a TCP connection is established.

1] The server must be prepared to accept an incoming connection. This is normally done by calling socket, bind and listen and is called passive open.

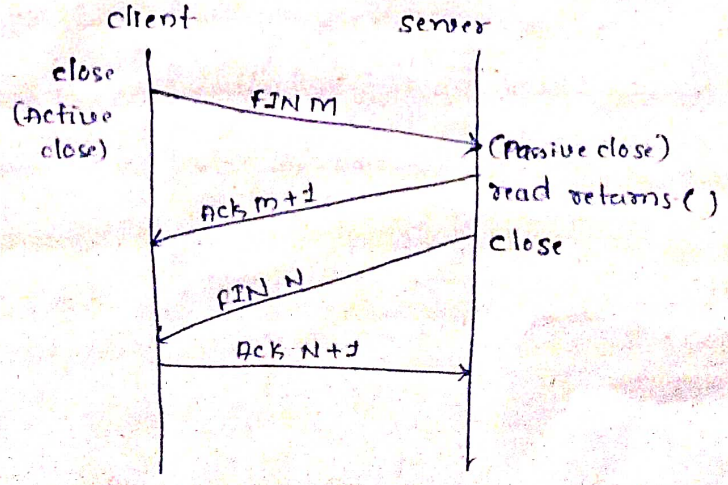
2] The client issues an active open by calling connect. This causes the client TCP to send a "synchronize" (SYN) segment, which tells the server the client's initial sequence number for the data that the client will send on the connection. Normally, there no data sent with the SYN; it just contains an IP ~~address~~ header, a TCP header, and possible TCP options.

3] The server must acknowledge (ACK) the client's SYN & the server must also send its own SYN containing the intial sequence number ~~for~~ data that the server will send on the connection. The server sends its SYN & the ACK of the client's SYN in a single segment.

4] The client must acknowledge the server's SYN.

→ The minimum number of packets required for this exchange is three hence, this is called TCP's three-way handshake.

TCP connection Termination



→ While it takes three segments to establish a connection, it takes four to terminate a connection.

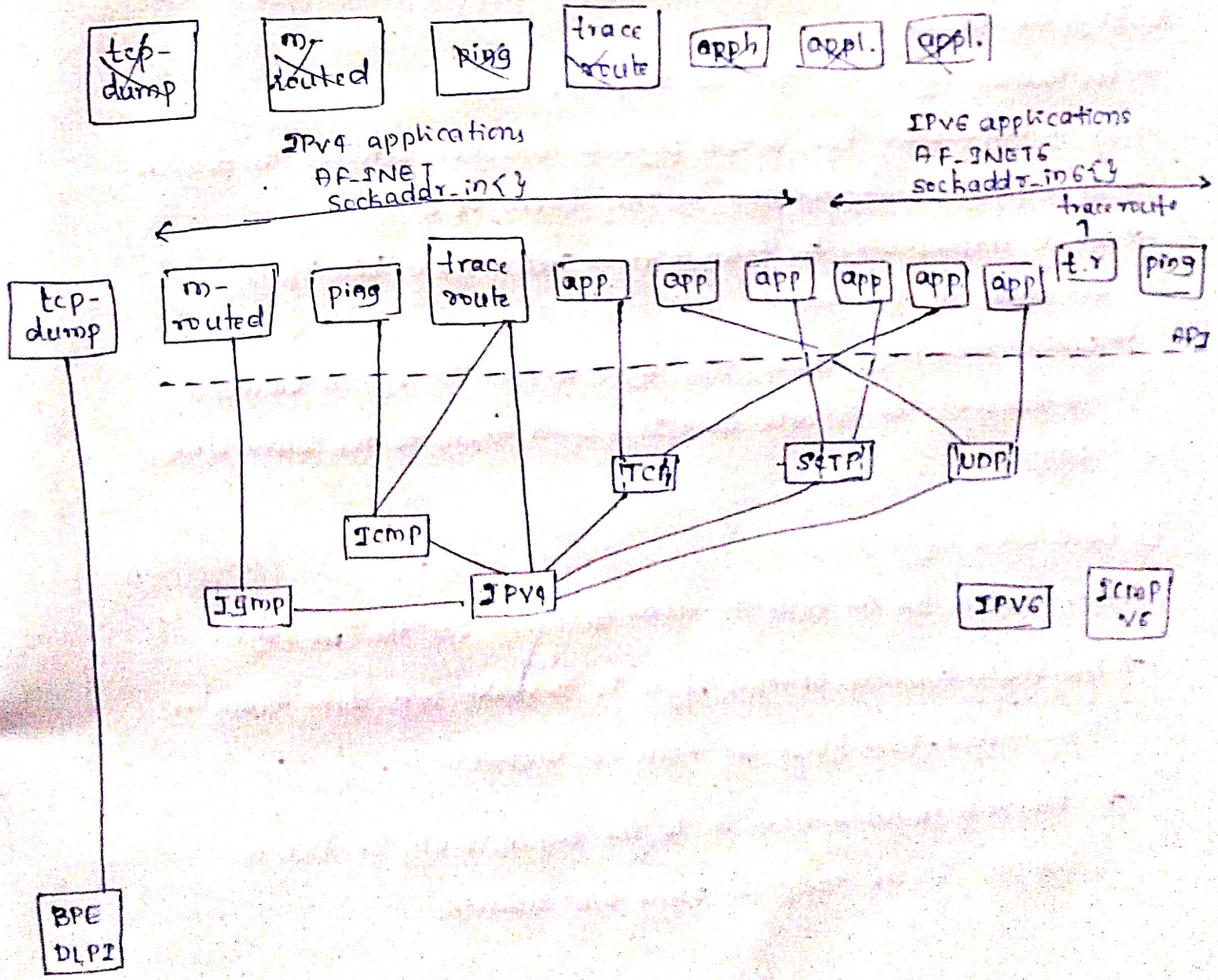
1] one application calls close first, and we say that this end performs the active close. This end's TCP sends a FIN segment, which means it is finished sending data.

2] The other end that recieves the FIN performs the passive close. The recieved FIN is acknowledged by TCP. The reciept of the FIN is also passed to the application as an end-of-file, since the recippt of the FIN means the application will not recieve any additional data on the connection.

3] Sometime later, the application that recieved the end-of-file will close its socket. This causes its TCP to send a FIN.

4] The TCP on the system that recieves this final FIN acknowledges the FIN

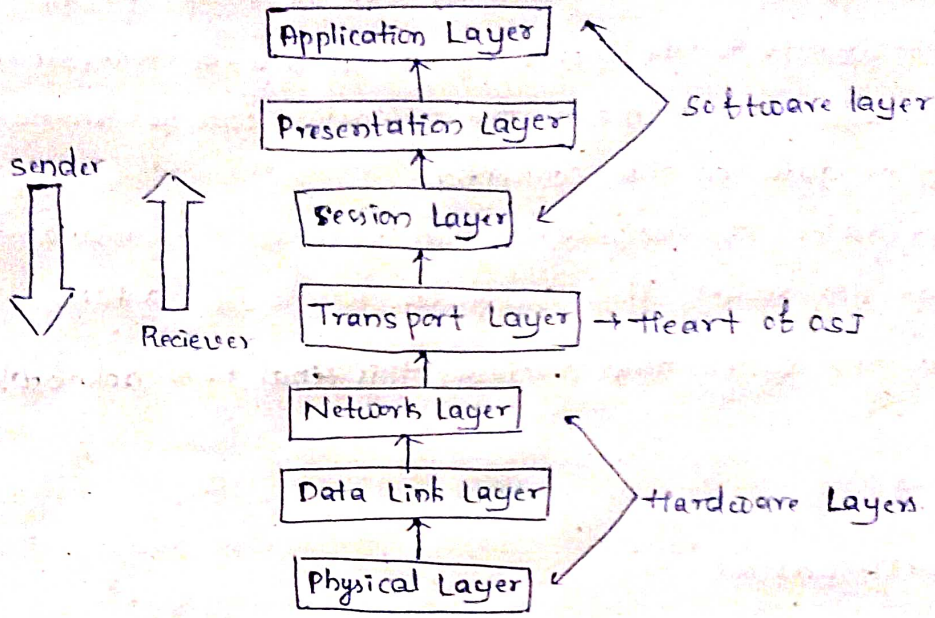
④ Explain SNTP protocol.



Layers of OSI model

OSI stands for Open Systems Interconnection. It has been developed by ISO - 'International organization for standardization' in the year 1984.

→ It is a 7 layer architecture with each layer having specific functionality to perform. All these 7 layers work collaboratively to transmit the data from one person to another across the globe.



1] Physical Layer :

→ The lowest layer of the OSI reference model

→ It is responsible for actual physical connection between the devices

→ It contains information in the form of bits

→ It is responsible for transmitting individual bits from one node to the next.

→ While recieving data, this layer will get the signal recieved & convert it into 0s & 1s & send them to the Data Link Layer.

2] Data Link Layer :

→ Responsible for the node-to-node delivery of the message

→ The main function of this layer is to make sure data transfer is error-free from one-node-to another

→ When a packet arrives it is the responsibility of DLL to transmit it to the Host using MAC address

3] Network Layer :

→ The network layer works for the transmission of data from one host to the other located in different networks.

→ It also takes care of packet routing i.e. selection of the shortest path to transmit the packet, from the no. of routes available

→ The sender's & reciever's IP addresses are placed in the header by the network layer

4] Transport Layer :

→ The transport layer provides services to the application layer & takes service from the network layer

→ The data in the transport layer is referred to as segments.

→ It is responsible for the end-to-end delivery of the complete message

→ The transport layer also provides the acknowledgement of the successful data transmission & retransmits the data if an error is found.

5] Session Layer :

→ This layer is responsible for the establishment of connection, maintenance of ~~sessio~~ sessions, authentication & also ensures security.

6] Presentation Layer :

→ This layer is also called as Translation layer. The data from the application layer is extracted here & manupulated as per the required format to transmit over the network.

7] Application layer :

→ It is the top layer of the OSI model which is implemented by the network applications

→ These applications produce the data, which has to be transfered over the network.

→ This layer also serves as a window for the application services to access the network and for displaying the recieved information to the user.

Ex : Browsers, skype, Messenger etc