OSI IMPLEMENTATION

AIM: To implement /simulate OSI model

ALGORITHM: Message from Application layer is passed to Presentation layer where it is encrypted/compressed and then passed to session layer which starts a unique session with destination and then transport layer helps in segmentation and network layer further divides segment to packets and routes them and Data link layer handles flow and error control, finally physical layer converts the message to form that can be transported through medium. In destination the reverse happens.

CODE:

```
#include <stdio.h>
void main()
{
printf(" in the session...");

char msg[20]=" how are you?";
printf(" In Application layer in message is ...
```

```
printf("\n Presentation layer \n message is
encoded, encrypted and compressed to ASCII
format \n");

printf("\n Session layer: \n Session is created
and session id is 123 and ensures
synchronization \n");

printf("\n Transport layer \n: TCP is used for
multiplexing and port number is 80 & message
is segmented to 3 segments \n 0 12
\n how are you? \n");

printf("\n Network layer \n: Segments are given
IP address of source as 192.168.1.1 and
destination as 192.168.1.2 and are sent
to destination using IP protocol and router
routes the segment \n");

printf("\n Data Link Layer: \n the packets are
further broken down into frames Flow and
error control is done. LLC uses WAN protocol
PPP protocol to transport the frames and
MAC addresses of source is 00-20-18-60
07-71 \n");
```

```c
printf("Physical Layer: \n Frames are converted
to binary 0 and 1 and are transported
in coaxial cable as a signal.");
printf("\n\n Destination \n");
printf("\n Physical layer:\n Signal is received
and converted to binary 0's and 1's \n");
printf("\n Data link Layer:\n Bits are
converted to frames on destination whose
MAC address is 00-26-18-10-07-72 \n");
printf("\n Network layer:\n frames from
data link layer are assembled to packets
on destination machine whose IP address
is 192.168.12 using IP protocol \n");
printf("\n Transport layer: The packets are
assembled to 3 segments 'how' , 'are' and
'you?' and is reordered to form correct
message \n");
printf("\n Session layer\n Synchronized session
whose ID is 123 is destroyed \n");
```

```
printf ("\n Presentation layer:\n message is
decoded, decrypted and decomposed from
ASCII format to machine
understandable code\n");
printf ("\n Application layer:\n message is
does\n HTTP ensures received message is
given to corresponding application\n");
```

O/P

IN SOURCE

Application layer
message is : how are you?
Protocol is Hyper Text Transfer Protocol Http

Presentation layer
message is created, encrypted and converted to
ascii format

Session layer
Session is created and session id is
123 and ensures synchronization

Transport layer

Transmission control protocol is used for
multiplexing and port number is 80.
Message is segmented into 3 segments
0 1 2

how are you?
message in sent in sequence how are
you?

Flow control: 1 segment at a time

Network layer:

Segments are given IP address of source as 192.168.1.1 and destination as 192.168.1.2 and are sent to destination using IP protocol and routers route the segment.

Data link layer:

The packets are further broken down into frames. Flow and error control is done. LLC uses WAN protocol PPP to transport the frames and MAC address of source is 00:20-18-18-02-71

Physical layer:

Frames are converted to binary 0 and 1 and are transported via coaxial cable as a signal. Transmission mode is duplex.

In Production

Physical layer: Signal is received and converted to binary 0s and 1s.

Data link layer: Bits are converted to frames on destination whose MAC address is 00-20-18-C0-02-72.

Network layer: frames from ... are assembled to packets on destination machine whose IP address is 192-168-1-2

Transport layer: the packets are assembled to form 3 segments "how" "are" "you" and is ordered to form correct message.

Session layer:
Synchronized session whose Detentors is 123 is displayed

Presentation layer,

message is decoded, decrypted, and

decompressed from ASCII, send to machine

understandable code.

Application layer

message is: how are you?

HTTP ensures received message is given

to corresponding application.