## VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi



A Mini-Project Report on

### “OSI MODEL”

**Computer Graphics & Visualization Laboratory with Mini Project 15CSL68**

Submitted by

### Pawan S Nadig (4VV16CS073)

**Pooja G (4VV16CS075)**

Under the Guidance of

### Pavan Kumar S P Gururaj H L

**Assistant Professor Assistant Professor**



**Department of Computer Science & Engineering**

**VIDYAVARDHAKA COLLEGE OF ENGINEERING MYSURU-570 002**

**2018-19**

**Vidyavardhaka College of Engineering**

Department of Computer Science and Engineering

Mysuru, Karnataka 570 002

**(Affiliated to Visvesvaraya Technological University, Belagavi)**



# Certificate

This is to certify that the Mini-project work entitled **“OSI Model”**, is a bonafide work carried out by **Pawan S Nadig (4VV16CS073) and Pooja G (4VV16CS075)** having completed the Computer Graphics & Visualization with Mini-project (15CSL68) during the year 2018-2019.

|  |  |
| --- | --- |
| **Guide**  Dept. CS & E,  VVCE, Mysuru | **Dr. Ravikumar V** Professor & Head, Dept. CS & E, |
| **Dr. B Sadashive Gowda**  Principal, VVCE | |

### External Viva

Name of the Examiners Signature with date

1.

2.

**ABSTRACT**

The main aim of this project is to demonstrate the working of Open Systems Interconnection (OSI) model. OSI model acts as a reference tool for understanding data communications between any two networked systems. This model splits the communications processes into seven layers.  Layers 1-4 are considered the lower layers, and mostly concern themselves with moving data around. Layers 5-7, the upper layers, contain application-level data. This model works on the basic principle: "pass it on." Each layer both performs specific functions to support the layers above it and offers services to the layers below it using the openGL functions.

## TABLE OF CONTENTS

**Chapter No. Chapter Name Page No.**

1. **INTRODUCTION 1** 
   1. Computer graphics 1
   2. OpenGL Interface 2
   3. Project Description 3
2. **METHODOLOGY 4**
   1. Functions in OpenGL 4

2.2. User defined functions6

1. **RESULTS 7**
   1. Snapshots 7

**4. CONCLUSION 13**

**REFERENCES 14**

**LIST OF FIGURES**

**FIGURE NO. FIGURE NAME PAGE NO.**

1.2.1. Graphics Pipeline 2

3.1.1. Outlook of OSI Model 7

3.1.2 Display of Menu 8

3.1.3. Functioning of Application Layer 8

3.1.4. Functioning of Presentation Layer 9

3.1.5. Functioning of Session Layer 9

3.1.6. Functioning of Transport Layer 10

3.1.7. Functioning of Network Layer 10

3.1.8. Functioning of Data-Link Layer 11

3.1.9. Functioning of Physical Layer 11

3.2.0. Packet at Destination 12

**Chapter 1**

**INTRODUCTION**

Computer graphics are pictures and films created using computers. Usually, the term refers to computer-generated image data created with the help of specialized graphical hardware and software.

* 1. **Computer Graphics**

Computer graphics is made up of number of pixels. Pixel is the smallest graphical picture or unit represented on the [computer](http://ecomputernotes.com/fundamental/introduction-to-computer/what-is-computer) screen. Basically, there are two types of computer graphics namely. Interactive Computer Graphics involves a two- way communication between computer and user. Here the observer is given some control over the image by providing him with an input device for example the video game controller of the ping pong game [10]. This helps him to signal his request to the computer. In non-interactive computer graphics otherwise known as passive computer graphics. It is the computer graphics in which user does not have any kind of control over the image. Image is merely the product of static stored program and will work according to the instructions given in the program linearly. The image is totally under the control of program instructions not under the user. Example: screen savers. Computer graphics started with the display of data on hardcopy plotters and cathode ray tube screens soon after the introduction of computers themselves. It has grown to include the creation, storage, and manipulation of models and images of objects. These models come from a diverse and expanding set of fields, and include physical, mathematical, engineering, architectural, and even conceptual structures, natural phenomena, and so on. Computer graphics today is largely interactive. Computers and Graphics is dedicated to disseminate information on research and applications of computer graphics (CG) techniques. [8]

### OpenGL Interface

OpenGL is an application program interface (API) which makes available to the programmer a small set of geometric primitives - points, lines, polygons, images, and bitmaps. OpenGL provides a set of commands that allow the specification of geometric objects in two or three dimensions, using the provided primitives, together with commands that control how these objects are rendered. [3]. OpenGL offers different coordinate system and frames. It also offers translation, rotation and scaling of objects. Most of our applications will be designed to access OpenGL directly through functions in three libraries [2]. They are:

* + - **Main GL:** Library has names that begin with the letter gl and are stored in a library usually referred to as GL.
    - **OpenGL Utility Library (GLU):** This library uses only GL functions but contains code for creating common objects and simplifying viewing.
    - **OpenGL Utility Toolkit (GLUT):** provides minimum functionality that Should be accepted in any modern windowing system.

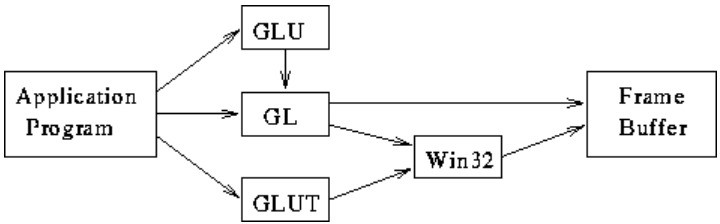


Fig 1.2.1. Graphics pipeline

Figure 1.2.1 shows the organization of the libraries for an X Window System environment. For this window system, GLUT will use GLX and the X libraries. The application program, however, can use only GLUT functions and thus can be recompiled with the GLUT library for other window systems.

### Project Overview

The open system interconnection model, better known as the OSI model, is a network map that was originally developed as a universal standard for creating networks. But instead of serving as a model with agreed-upon protocols that would be used worldwide, the OSI model has become a teaching tool that shows how different tasks within a network should be handled in order to promote error-free data transmission [1]. These jobs are split into seven layers, each of which depends on the function’s “handed-off” from other layers. As a result, the OSI model also provides a guide for troubleshooting network problems by tracking them down to a specific layer. Here we’ll take a look at the layers of the OSI model and what functions they perform within a network. Instead, the committee decided to establish a common reference model for which others could then develop detailed interfaces, which in turn could become standard. OSI was officially adopted as an international standard by the International Organization of Standards (ISO).

The animations are used to explain basic communication protocol principles that shows the communication process along all the essential coordinates: space, time and contents [9]. Along each axis, an appropriate level of abstraction is chosen. Along the space dimension, both communication entities and communication channels are easily distinguishable. Entities are usually shown as boxes of different shapes and colors that depend on their respective roles and importance’s. Messages are presented as dynamic objects that are created when the messages are transmitted, is shown as propagating through the channel, and it can be removed from the scene when they are consumed by receiving entities [7]. Usually their contents are denoted either with their abstract syntax or, even more abstractly, with only their message types. Either simple text or colors, and possibly combination of both, can be used for this purpose. Sometimes, however, the essential structure of the messages can even be shown, if this is considered important by the animation designer.[5]

# Chapter 2

**METHODOLOGY**

# This chapter includes various methodologies used in computer graphics. It explains the syntax of various functions implemented in our project.

### Functions in OpenGL

* + - **void glClear(glEnum mode);**

Clears buffers like color and depth.

### void glTranslatef(TYPE x, TYPE y, TYPE z);

Alters the current matrix by displacement of (x, y, z), TYPE is either GLfloat or GLdouble.[4].

### void glutSwapBuffers();

Swaps the front and back buffers.

### void glMatrixMode(GLenum mode);

Specifies which matrix will be affected by subsequent transformations

### void glLoadIdentity( );

Sets the current transformation matrix to identity matrix.

### void glPushMatrix(); void glPopMatrix();

Pushes to and pops from the matrix stack corresponding to the current matrix mode.

### void glutBitmapCharacter(void \*font, int character);

It renders the character in the named bitmap font.

### void glRasterPos2D(GLfloat x, GLfloat y, GLfloat z);

It is used to position pixel and bitmap operations.

### void glOrtho2D(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top);

Specifies the co-ordinates for clipping planes.

### void glutInit(int \*argc, char \*\*argv)

Initializes GLUT; the arguments from main are passed in and can be used by the application [6][8].

### void glutInitDisplayMode(unsigned int mode);

Requests a display with the properties in the mode; the value of mode is determined by the logical OR of options including the color model (GLUT\_RGB, GLUT\_INDEX) and buffering (GLUT\_SINGLE, GLUT\_DOUBLE).

### void glutCreateWindow(char \*title);

Creates a window on display; the string title can be used to label the window. The return value provides a reference to the window that can be used when there are multiple windows.

### void glutMainLoop();

Causes the program to enter an event-processing loop.

### void glClearColor(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha);

Specifies the red,green,blue and alpha values used by glClear to clear the color buffers.Values specified by glClearColor lies in the range 0 to 1.

### int glutCreateMenu(int val);

The callback function for the menu is called when a menu entry from the menu is selected. The value passed to the callback is determined by the value for the selected menu entry.

### void glutPostRedisplay(void);

glutPostRedisplay() essentially sets a flag so that on the next iteration of the mainloop, your registered display() function is called.

|  |  |  |
| --- | --- | --- |
|  |  |  |

### 2.2.User Defined Functions

* + - **void drawtext(char s, int x, int y);**

This function is used to display the characters ‘L’ and ‘R’ within each node.

### void drawtext2(char s1[], int x, int y);

This function is used to display sentences.

### void node(int x, int y, int z);

This function creates the nodes at the specified position.This uses primitives like GL\_LINE\_LOOP, GL\_LINES.

### void render();

This function is used to display the nodes by using node function.

### void mov1();

Translates nodes in direction of y- axis

### void mov2();

Translates nodes in direction of y- axis

### void menu(int id);

This function creates a new pop up menu and returns a unique small integer.

# Chapter 3

**RESULTS**

This chapter explains the working of OSI Model. The output is shown by using snapshots.

### Snapshots

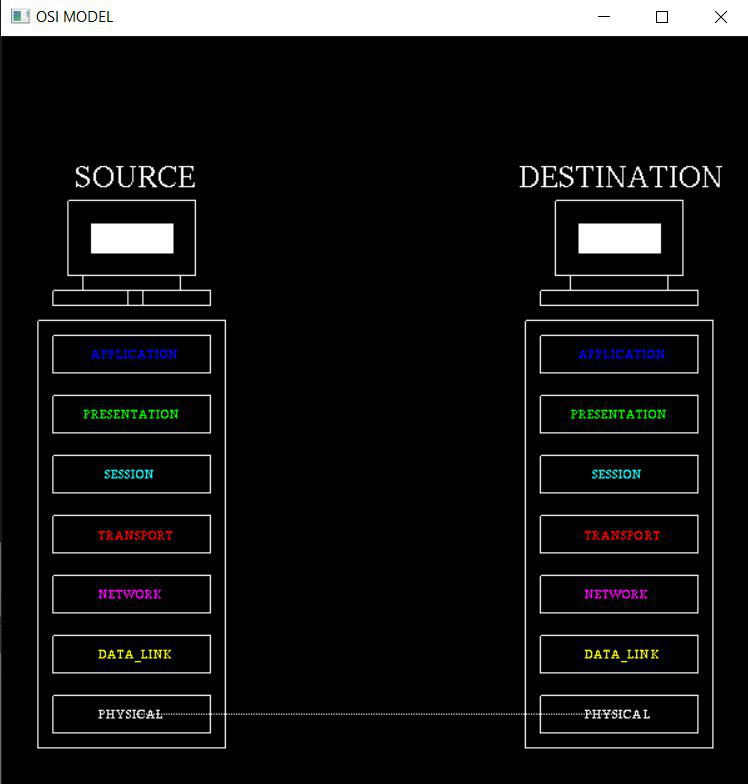


Fig 3.1.1. Outlook of OSI model

This figure shows the structure of OSI model. It mainly consists of seven layers at both source and destination side.

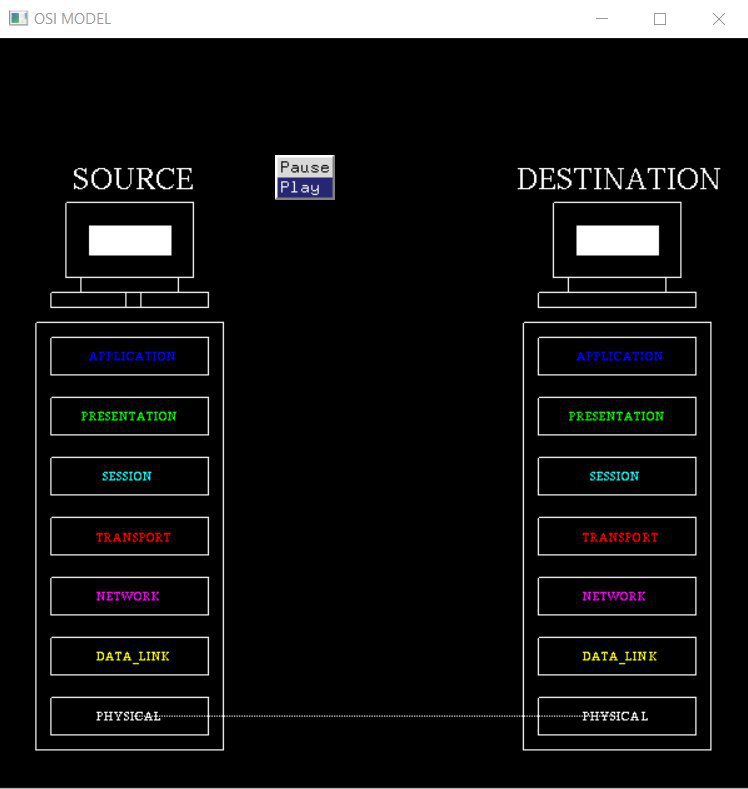


Fig 3.1.2. Display of menu

The above snapshot shows the display of menu with two options play and pause.

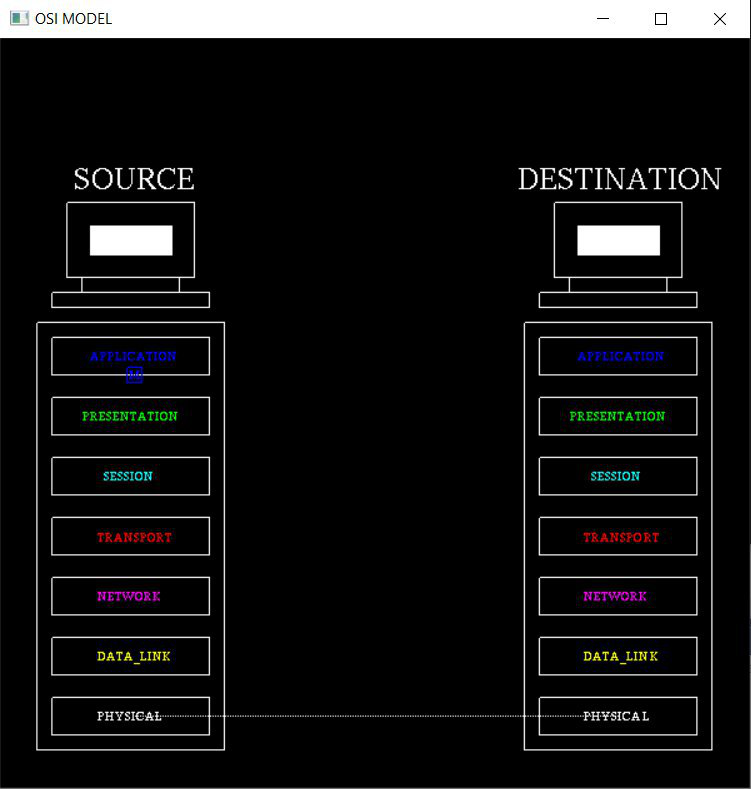


Fig 3.1.3. Functioning of Application Layer

The above snapshot shows the data generated from the application layer, later is transferred to the below layer.

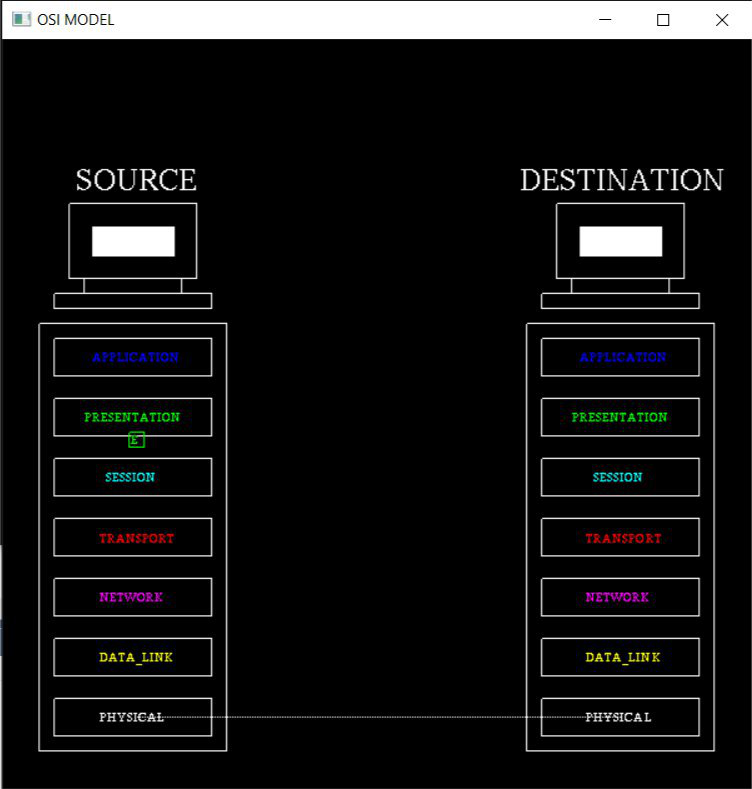


Fig 3.1.4. Functioning of Presentation Layer

The above snapshot shows the encryption of message at presentation layer

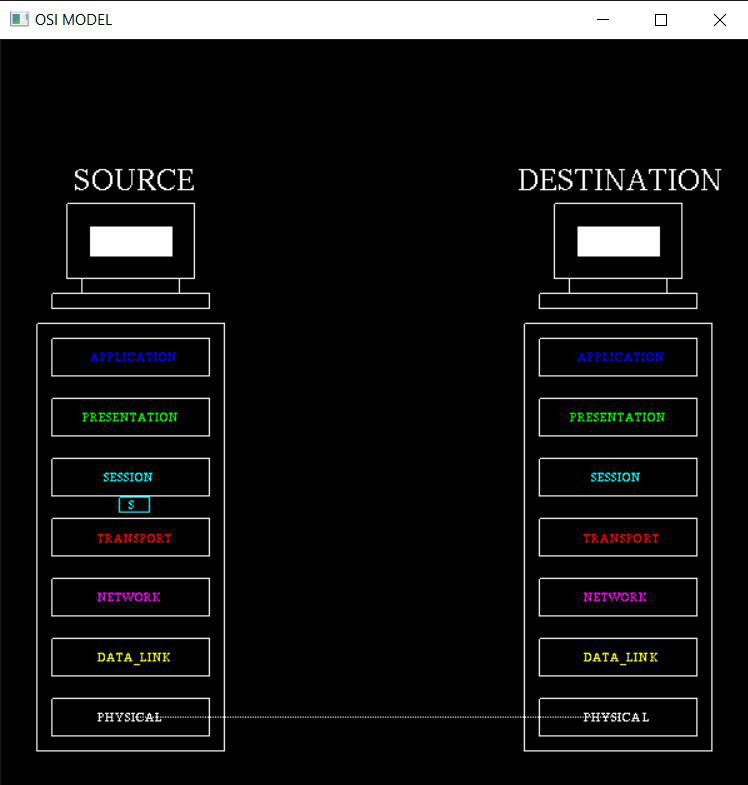


Fig 3.1.5. Functioning of Session Layer

The above snapshot shows the working of session layer which mainly ensures the security of data.

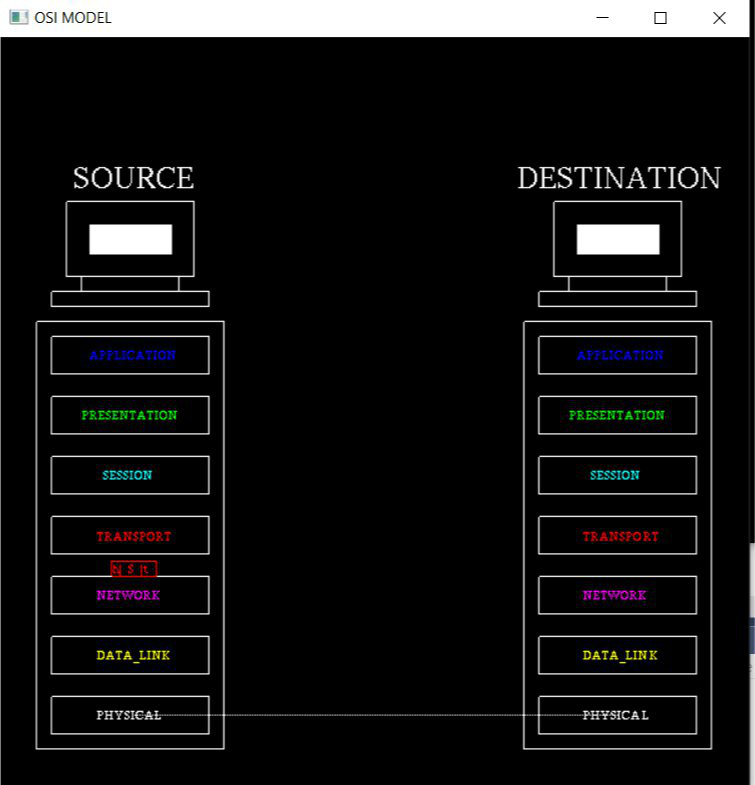


Fig 3.1.6. Functioning of Transport Layer

The above snapshot shows how the data is being converted into segments.

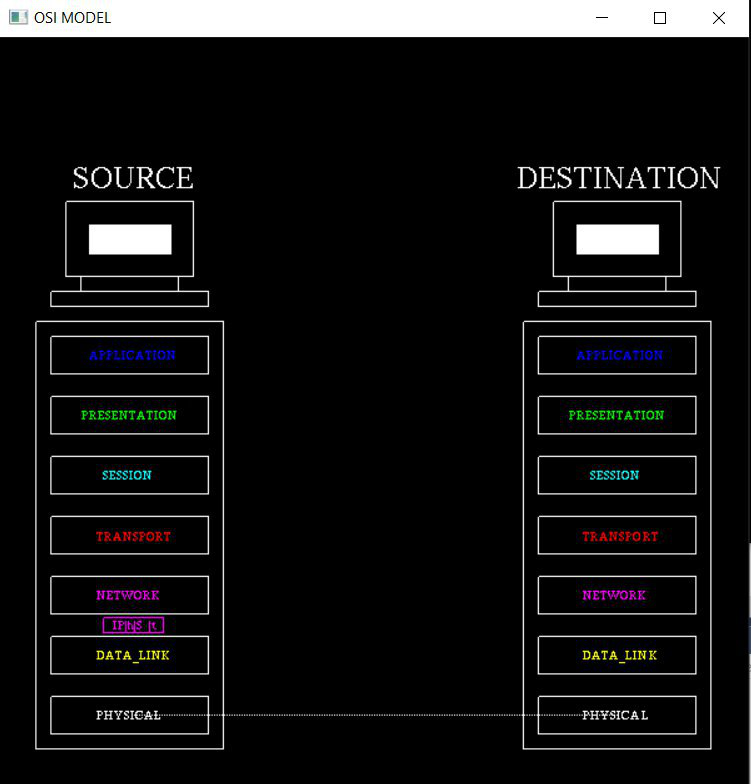


Fig 3.1.7. Functioning of Network Layer

The above snapshot shows how the network layer takes care of packet routing.

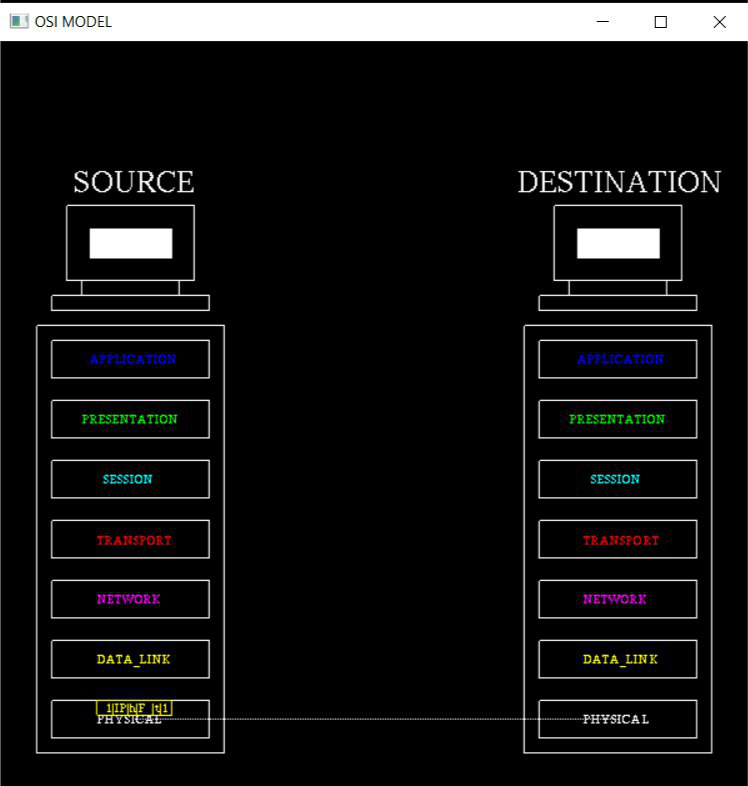


Fig 3.1.8. Functioning of Data\_Link Layer

The above snapshot shows how the data is converted in to frames and it is being transferred to the below layer.

# 

Fig 3.1.9. Functioning of Data\_Link Layer

The above snapshot shows the bit transmission of data from physical layer of source host to the physical layer of the destination host.

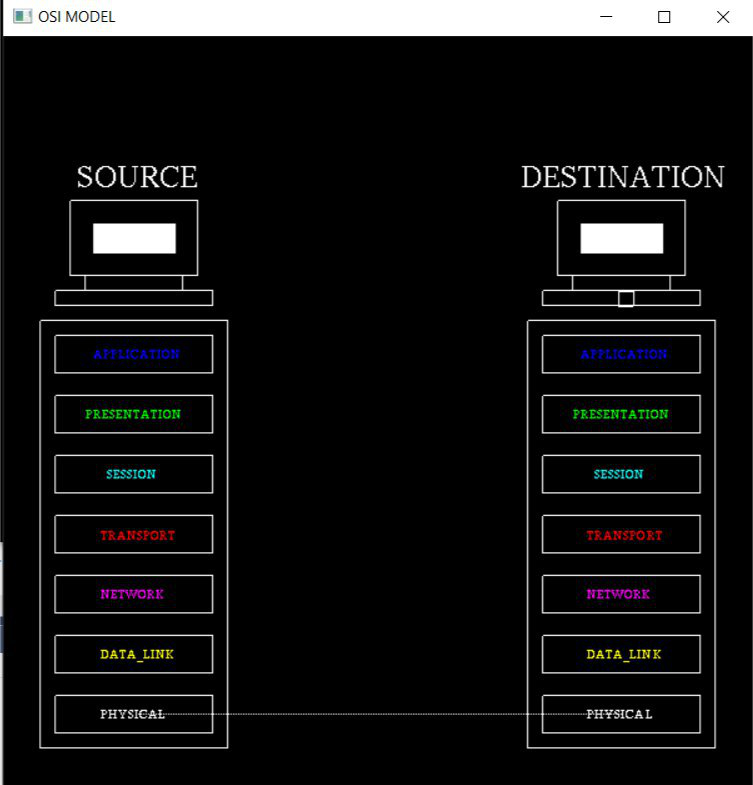


Fig 3.2.0. Packet at Destination

The above snapshot shows the packet at destination. It is received after the reverse process at each layer.

# Chapter 4

**CONCLUSION**

The idea we have implemented in our project is working well. Creation of this project helped us a lot in learning the new concepts and a proper usage of it. This project would be helpful in websites which hosts information about the animation of OSI model in a manner to help the students to learn about the functioning OSI model in a better way. The program is user friendly as the only skill required in executing this program is the knowledge of graphics.

**REFERENCES**

1. Vibhu Chinmay, Rishabh garg, " A Review paper on OSI model – A Seven

Layered Architecture of OSI Model”, ISSN: 2349-6002, 2015.

1. Nakhoon Baek1 and Kwan-Hee Yoo2, “Providing Direct3D Features over the

Desktop OpenGL”, October, 2011

1. Hui Zhao, “Implementing Surfaces in OpenGL”, 2006.
2. Woo,Neider, Davis, OpenGL Programming guide, MA, Reading: Addison-Wesley, Developers Press, 2005.
3. Johan Persson, “A study on the use of OpenGL in window systems”, MCS- 2004.
4. Woo, Neider, Davis, OpenGL Reference Manual, MA, Reading:Addison-Wesley, Developers Press, 2003.
5. Drago Hercog, “Animation-Based Explanation of Basic Data Communication

Principles’, SI-1000, July 21–25, 2003.

1. Sven Maerivoet Transport & Mobility Leuven, “Advanced Computer Graphics

using OpenGL”, January 2001.

1. Schroeder,Martin,Lorensen, The visualization tool kit: An Object- Oriented

approach to 3D graphics, New Jersey,Upper Sadle River:2001.

1. M. Roth, K. Tanaka, C. Weissman, W. Yerazunis, “Computer Vision for

Interactive Computer Graphics”, TR99-02 January 1999.