Course Name: Networks Laboratory

Course Code: IT 3095, (Cr: 1)

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

Deemed to be University

School of Computer Engineering

BHUBANESWAR-751024

Course Objective: The objective of a computer network laboratory is to provide students with hands-on experience with computer networking concepts and technologies. This includes learning about the different layers of the OSI model, how to configure and troubleshoot networks, and how to design and implement network applications. Specifically, the objectives of a computer network laboratory may include:

- Understanding the fundamentals of computer networking. This includes learning about the different types of networks, the different protocols used to communicate over networks, and the different ways to secure networks.
- Gaining hands-on experience with network devices. This includes learning how to configure and troubleshoot routers, switches, and firewalls.
- Designing and implementing network applications. This includes learning how to develop applications that can be used to share data and resources over a network.
- Troubleshooting network problems. This includes learning how to identify and resolve common network problems.

In addition to these specific objectives, a computer network laboratory can also help students to develop the following skills:

- Problem-solving skills. Students will need to be able to identify and troubleshoot network problems.
- Communication skills. Students will need to be able to communicate effectively with other students and with network engineers.
- Teamwork skills. Many network projects require teamwork, so students will need to be able to work effectively with others.

By providing students with hands-on experience with computer networking concepts and technologies, a computer network laboratory can help them to develop the skills they need to be successful in the IT industry.

Here are some specific examples of experiments that can be conducted in a computer network laboratory:

- Details of Socket programming APIs for TCP/IP stack using C.
- Configuring a router to route traffic between two networks.
- Troubleshooting a network problem by using Wireshark to capture network traffic.
- Designing and implementing a file sharing application that can be used to share files over a network.
- Building a virtual network using a network simulator.

These are just a few examples of the many experiments that can be conducted in a computer network laboratory. The specific experiments that are conducted will depend on the course syllabus and the interests of the students. The list of experiments are given here as a reference.

List of experiments

Lab 1:

- **Coverage**
  1. Discuss what is networking and its significance in computer network. Discuss the components (i.e., h/w and s/w) required for data communication in a Computer Network. (Show the h/w components like Network Interface Card (NIC), Network Cable, RG-45 Connector, Hub, Switch, Router etc.)
  2. Highlight the importance of socket programming as a s/w for data communication and the basic fundamentals required for doing socket programming using C.
  3. Review of function, pointer, structure, structure with in a structure, pointer to structure, and command line argument concept using C programming Language.
  4. What is little endian and big endian. Discuss the significance of endianness in computer network.
- **Assignments**
  1. Write a C program to swap the content of 2 variables entered through the command line using function and pointer.
  2. Write a C program to assign values to each member of the following structure. Pass the populated structure to a function Using call-by-value and another function using call-by-address and print the value of each member of the structure.

```
struct student_info{
int roll_no;
char name [50];
float CGPA;
struct dob age;
};
```

  3. Write a C program to extract each byte from a given number and store them in separate character variables and print the content of those variables.
  4. Write a C Program to enter a number and store the number across the following structure and print the content of each member of the structure. Then aggregate each member of the structure to form the original number and print the same.

```
struct pkt{
char ch1;
char ch2[2];
char ch3;
};
```

  5. Write a C program to check whether the Host machine is in Little Endian or Big Endian. Enter a number, print the content of each byte location and Convert the Endianness of the same i.e. Little to Big Endian and vice-versa.

Lab 2:

- **Coverage**
  1. Basics of Socket Programming.
  2. Details of Connection less Socket programming APIs for TCP/IP stack using C.
- **Assignments**
  1. Write a sender and receiver program in C by passing the IP address and the port number of each other through the command line arguments using connection less socket. Both of them will exchange messages with each other continuously. If any one of them will receive the "exit" message from the other end then both of them will close the connection. (Assume both the client and server are running with in the same host)

Lab 3:

- **Coverage**
  Details of Connection Oriented Socket programming APIs for TCP/IP stack using C.
- **Assignments**
  Write a connection-oriented client and server program in C using command line arguments. At the server side, pass the port number (to whom the server will bind to) in the command line. At the client side, pass the IP address and the port number of the server (to whom the client will connect to) as command line argument and carry out the following tasks.
  - After establishment of connection print the IP Address and port number of the client to whom the server is connected now.
  - Then exchange messages.
  - After message exchange is over then the client sends a "close" message to the server to tear down the connection.

Lab 4:

- **Coverage**
  1. Demonstrate the packet Analyzer tool (Wireshark) to analyze the details of a packet which is captured during packet transmission in the network.
  2. Discuss the overview of file transfer over a computer network.
- **Assignments**
  1. Analyze the packets using Wireshark, that are captured by running both client and server with in the same and different host.
  2. Write a connection-oriented client and server program in C using command line arguments. Do the file transfer from the server as follows.
     - Server first sends the list of files present in the current directory at it's own end.
     - After receiving the same, client send the name of a file it wants to download from the server.
     - Finally, after receiving the same server uploads the file to the client.

- After sending the file, server closes the client connection at its own end.

Lab 5:

- **Coverage**
  1. Discuss how to design a stop-and-wait protocol on top of connectionless sockets.
- **Assignments**
  1. Implement stop-and-wait protocol on top of UDP sockets using C program.
  2. Implement Go-back-N protocol on top of UDP sockets using C program.

Lab 6:

- **Coverage**
  1. Discuss how to fragment and reassemble packets at the source and destination host respectively.
- **Assignments**
  1. Write a connection-oriented client and server socket program using C where the client will suppose to send a large buffer to the server. Keeping fragmentation and reassembly into account the following functionality needs to be supported.

Fragmentation functionality

- First client needs to find the MTU size of its own and display the same.
- Based on the MTU size it will decide whether to fragment the data or not and display the message accordingly.
- In case of fragmentation, first find the number of fragments required and display the same.
- Prepare each fragment with the following information and display the same.
  1. Id (Identification number)
  2. flag (More fragments flag)
  3. offset (Fragmentation offset)
  4. HL (Header Length)
  5. TL (Total Length)
  6. payload (Data)
- Send all the fragments to the server.

Reassembly functionality

- First server needs to find whether the received datagram is a fragmented one or not and display the same.
- Continue receiving all the fragments belonging to a fraction of the same data till the last fragment and display all the fragments data with their details.
- Then follow the below steps to start the reassembly procedure
  1. first find out the first fragment.
  2. Calculate the offset of the next fragment and check whether the next fragment is available or not.

3. If available, repeat step-2 till flag value 0 or else display a appropriate error.

- Finally display the assembled datagram.

Lab 7 :

- **Coverage**
  1. What is I/O multiplexing and why it is required?
  2. Discuss different types of I/O multiplexing.
  3. Discuss how to design a concurrent chart server using fork ().
- **Assignments**
  1. Design a connection oriented concurrent chart server using fork () in C where the server will serve multiple chart clients simultaneously. When the chart server receives a "logout" message from a particular client then it terminates the respective connection with that chart client.

Lab 8 :

- **Coverage**
  1. Details of I/O multiplexing using select () API.
  2. Discuss how to design a concurrent chart server using select ().

- **Assignments**
  1. Design a connection oriented concurrent chart server using select () in C where the server will serve multiple chart clients simultaneously. When the chart server receives a "logout" message from a particular client then it terminates the respective connection with that chart client.

Lab 9 :

- **Network Configuration using to network simulator**
  1. Introduction to network simulator tool (Cisco Packet Tracer /NS2/NS3) and its applications.
  2. Demonstration of how routing works using simulator tool?

- **Assignments**
  1. Simulate routing of packets in a LAN with in the same subnet.
  2. Simulate routing of packets in a LAN with different subnets.

Lab 10:

- **Coverage**
  1. Comparison and analysis of existing protocols using network simulator.

- **Assignments**
  1. Compare and analyze different routing protocol using network simulator.