School of Computer Engineering
KIIT deemed to be University
Laboratory Lesson Plan – Autumn 2024 (5th Semester)

Discipline: CS/IT/CSCE/CSSE

Course name and Code: Computer Networks Laboratory, CS39003 (L-T-P-Cr:1)

Instructor: **Dr. Pinaki Sankar Chatterjee**, **Email**: pinakifcs@kiit.ac.in

Instructor Chamber: Faculty block-F211, Campus-15

Technical Assistants Names:

Course Contents:

List of experiments:

Lab 1: Week-1: Overview of C Programming

• Coverage

1. Highlight the importance of socket programming as a s/w for data communication and the basic fundamentals required for doing socket programming using C.
2. Review of function, pointer, structure, structure with in a structure, pointer to structure, and command line argument concept using C programming Language.
3. 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: Week-2: Basics of Socket Programming

• 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: Week-3: Connection Oriented Socket Programming

• 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: Week-4: Socket Programming with Multiple clients

• 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 chat server using **fork ()** and **select()** API.

• Assignments

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

Lab 5: Week-5: Packet Analysis using Wireshark

• Coverage

1. Demonstrate the packet Analyzer tool (Wireshark) to analyze the details of a packet which is captured during packet transmission in the network.

• Assignments

1. Start the Wireshark packet sniffer and start capturing packets. Then open a terminal and execute command ***ping -s 3500 ping-ams1.online.net -c 5.*** After execution is complete stop the Wireshark capture and save the file for further analysis.

Answer the following questions based on above obersation -

a) How many total IP packets are exchanged in the communication between your host and the remote server representing **ping-ams1.online.net** ?

b) What is the size of each ping request sent from your host to remote server?

c) Make a table for each ping request packet sent from your host to remote, the respective field indicating it, if the request packet is fragmented or not. If packet is fragmented (add details of number of IP fragments and on each fragment), Time of sending each individual fragment/packet, length of the individual fragment/packet), time of receiving ping response, the respective field indicating if response packet is fragmented or not, if response packet is fragmented include the number of IP fragments, total actual length of data carried by the respective fragment in respective ping request and response.

2. Start the Wireshark again and execute command ***traceroute -q 5 ping-ams1.online.net 3500*** in the terminal. After execution is complete stop the Wireshark capture and save the file for further analysis.

3. Answer the following questions based on above observation -

a) How many hops are involved in finding the route to this **ping-ams1.online.net**

b) How many total IP packets are exchanged in the communication to get the final traceroute output of ping-ams1.online.net? How many of them are sent from client to remote machine (server/router)? How many of them are sent from the remote

machine(hop/server/router) to the local client? Tabulate this with an entry for a router/server and the client too.

c) Which fields in the IP datagram always change from one datagram to the next within this series of IP packets send by your host/client? Which fields stay constant? Which of the fields must stay constant? Which fields must change? Why?

Lab 6 : Week-6: Introduction to Network Simulators

• Coverage

• Network Configuration using to network simulator

1. Introduction to network simulator tool (Cisco Packet Tracer/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 7: Week - 7: Simulating network protocols in Network Simulators

- Cover DHCP Protocol.
- Create a network in simulator and demonstrate DHCP Protocol in that network.

Lab 8: Week-8

• 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 9: Week-9

• 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 10: Week-10 (Optional)

• Coverage

Secured end-to-end message transmission between two clients connected to the same server.

1. Discuss about connecting multiple clients to the server and exchange messages between clients in TCP/IP stack socket programming using C.

2. Discuss about any simple encryption technique (like ceasar cipher, substitute cipher, etc.) and demonstrate how to send encrypted message from one client to another.

• Assignments

Write a connection-oriented client and server program in C using command line

Arguments.

Apply appropriate encryption techniques for secure communication.

Lab 11: Week-11 (Optional)

• Coverage

1. Comparison and analysis of existing protocols using network simulator.

• Assignments

1. Compare and analyze different routing protocol using network simulator

Grading Policy:

- Lab Record + TA's Evaluation - 30 Marks
- Activities (Quiz / Assignment) - 30 Marks
- Sessional Exam - 40 Marks

Online Resources:

- Socket Programming Tutorial

https://www.youtube.com/playlist?list=PLOholRFa862SX9Ypgk3vsByrfMT_hMq-l