“TCP/IP STACK IMPLEMENTAION USING C”

A PROJECT REPORT

SUBMITTED IN THE PARTIAL FULFILMENT OF THEREQUIREMENTS
FOR THE AWARD OF DEGREEOF

MASTER OF TECHNOLOGYIN
DATA SCIENCE

Submitted By

Sayan Sarkar(2K22/SWE/18)

Under the supervision of

Prof. Ruchika Malhotra
Head of Department (Software Engineering)
Department of Software Engineering Delhi
Technological University, Delhi

DEPARTMENT OF SOFTWARE ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

June, 2024

DEPARTMENT OF SOFTWARE ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE DECLARATION

I, Sayan Sarkar, 2K22/SWE/18 student of M.Tech (SWE), hereby declare that the project entitled "***TCP/IP stack implementation using C***" which is submitted by me to Department of Software Engineering, Delhi Technological University, Shahbad Daulatpur, Delhi in partial fulfilment of requirement for the award of the degree of Masterof Technology in Data Science, has not been previously formed the basis for any fulfilment of requirement in any degree or other similar title or recognition.

This report is an authentic record of my work carried out during my degree under the guidance of Prof. Ruchika Malhotra.

Candidate signature

This is to certify that the student has incorporated all the corrections suggested by the examiners in the thesis and the statement made by the candidate is correct to the best of our Knowledge.

Signature of Supervisor

DEPARTMENT OF SOFTWARE ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the project entitled "***TCP/IP stack implementation using C***" which is submitted by Sayan Sarkar (2K22/SWE/18) to Department of Software Engineering, Delhi Technological University, Shahbad Daulatpur, Delhi in partial fulfilment of requirement for the award of the degree of Master of Technology in Data Science, is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any degree or diploma to this university or elsewhere.

Place: Delhi

Date:

Prof. Ruchika Malhotra

SUPERVISOR

HOD and Professor

Department of Software Engineering

ACKNOWLEDGEMENT

I am very thankful to **Prof. Ruchika Malhotra** (Head of Department, Professor, DTU, Department of Software Engineering) and all the faculty members of the Department of Computer Science at DTU. They all provided us with immense support and guidance for theproject. I would also like to express my gratitude to the University for providing us with thelaboratories, infrastructure, testing facilities and environment which allowed us to work without any obstructions. I would also like to appreciate the support provided to us by our lab assistants, seniors and our peer group who aided us with all the knowledge they had regarding various topics.

Sayan Sarkar

2K22/SWE/18

INDEX

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviations	Full Form
TCP	Transmission control protocol
IP	Internet Protocol
Eth	Ethernet
MAC	Media Access Control
FreeBSD	Free Berkley Software Distribution
ARP	Address resolution Protocol
MTF	Modulation Transfer Function
TTL	Time to live

ABSTRACT

Approaches that facilitate the application and transmission of data to actual circumstances with the goal of having a fast and secure connection are steadily becoming more and more popular in the current world.The knowledge gap between theory and implementation will be closed by this initiative.A graph is a simple data structure that, when applied to a network topology, allows us to see how network topologies might be created.C programming, which fills in the gaps between network topologies and fundamental data structures, is used to implement this task.This paper investigates the MAC address, interfaces, and routing protocols.This work aids in our comprehension of the TCP/IP stack and network application end-to-end architecture and design.

1.INTRODUCTION

1.1 MOTIVATION

The importance of TCP/IP is due to the fact that it serves as the foundation upon which most of the internet is built.Network nodes communicate with one another using this protocol.Data communication and Internet or inter-networking of these devices require TCP/IP to function. It harmonizes the way we interact with different kinds of technology and software..The flexibility of its applications ranges from browsing, emailing, among others.It can also be relied on for interpretation.This system can be scaled up or down depending on our requirements. A TCP/IP stack gives you a way to drive home your networking concepts practically. You are able to use what you learned about networks in theory in practical world situations through programming networks lives. Having one's TCP/IP stack allows one to think out of the box and come up with new ideas that will make him/her develop a customized network stack meeting his/her own specific needs. You can introduce new features, enhance performance or even build solutions around certain applications.Proficiency in networking and understanding how network protocols work are highly valued within IT today. When you develop your own TCP/IP stack it shows that you have specialized competence hence it should feature prominently on your resume .It proves that you have hands-on experience with low-level networking technologies

1.2 RELATED WORK

There is a lot of related work on implementing TCP/IP stack using C.
An open-source TCP/IP stack called Lightweight IP is made to be compact and effective.It is utilized with C programming languages in embedded systems.
Another open-source architecture that is utilized in environments with limited resources is uIP.It offers us a variety of protocols, including IPV6, TCP, and UDP.· FreeBSD TCP/IP Stack is a part of FreeBSD operating system it is highly efficient and optimised.

1.3 OBJECTIVE OF THE PROPOSED FRAMEWORK

The project's goal is to develop robust, automated network layer protocols. This research aims to improve the existing models and analysis in brief the functionality. The main goal is to obtain screen-pigmented analysis.This will enhance better understanding of different underlying layers and how they communicate with each other.

The proposed project's impact is twofold: This will significantly advance the integration of feature in layers and classification in networking field. (ii) This will significantly improve understanding of the relationship between different layers which are not usually visible or understandable.

2.PROPOSED FRAMEWORK

2.1 OVERVIEW OF THE PROPOSED FRAMEWORK

Using C programming language we try to implement network topologies We use graph data structure to optimise our work.

We follow 8 steps in total for implementing this model.

- Generic graph creation.
- Construction of network topology.
- Command line Integration.
- Communication Setup
- Implementing ARP
- Implementing L2 switching
- Implementing VLan based forwarding.
- Setting up L3 routing infrastructure.

2.2 GENERIC GRAPH CREATION

Generic graph creation typically involves defining a data structure to represent a graph and implementing functions to create and manipulate the graph.

1. Defining a structure to represent node in a graph.Each node has a identifier and a list of pointers.
2. Defining the graph data structure which contains which contains list of nodes in the graph.
3. Implementing functions to create node
4. Implementing weighted paths between the nodes.

2.3 CONSTRUCTION OF NETWORK TOPOLOGY

It is very important to define the network terminologies for the graph.Previously we only defined the graph data structure but now we gonna add network features to the graph.

1. Each node represents a server , which is having an IP address and a MAC address.
2. The weighted paths between the servers are the channels which connect each server.
3. There are interfaces which helps us to recognise which server are we gonna call upon.
4. Then we create API to the networking properties.
5. The node transfers data from one server to another server using the links .

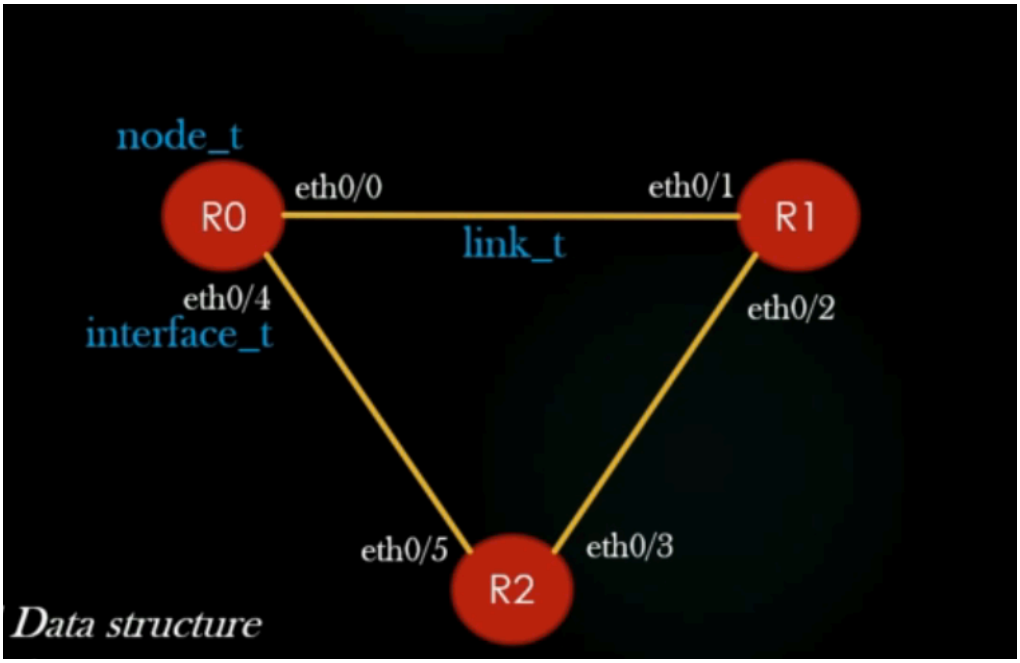


Fig 1: Network Topology

Network functionality is significantly influenced by network topology. Namely, network functionality is directly impacted by topology. Since a well-chosen and maintained network topology improves energy efficiency, selecting the best topology can help boost performance. That is to say, network functionality directly depends on the topology. Appropriate topology selection can boost performance since an appropriately selected and maintained network topology reduces energy consumption.

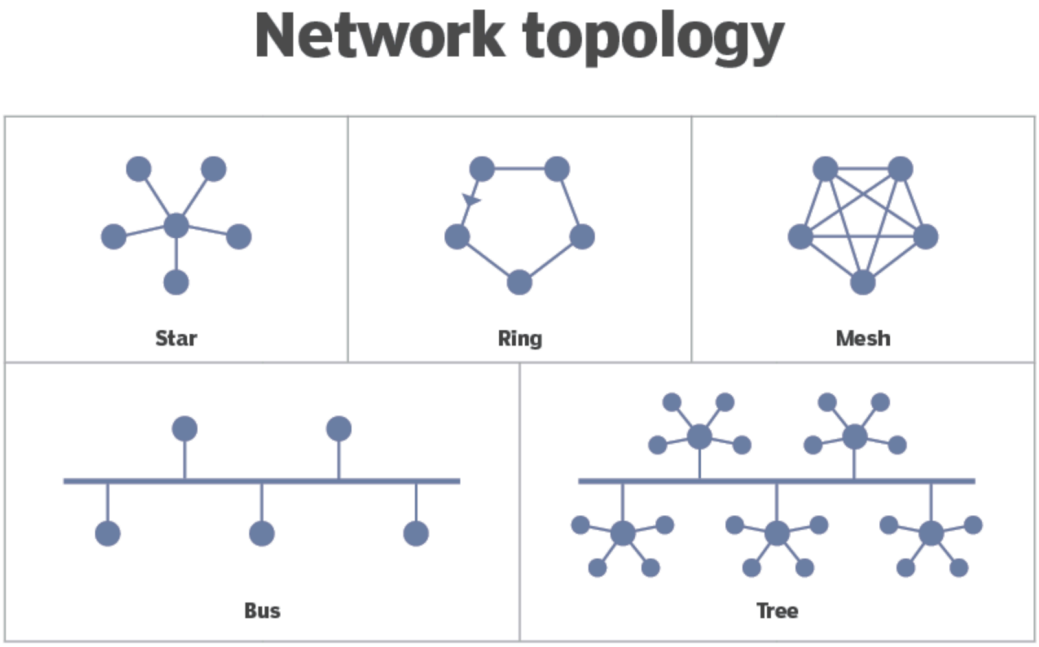


Fig 2: Network topologies

Network administrators find it simpler to identify problems, resolve problems, and assign network resources when the topology of the network is clearly specified.

2.4 COMMAND LINE INTEGRATION

Command line helps in integrating all essential features together for the graph along with the defined network terminologies.

1. We use the LibCli library for integration.
2. First we define the commands CLI will support.
3. Once the command is defined we will parse the inputs.
4. After parsing LibCli will call appropriate functions to execute the input.
5. Error handling can also be done very easily.

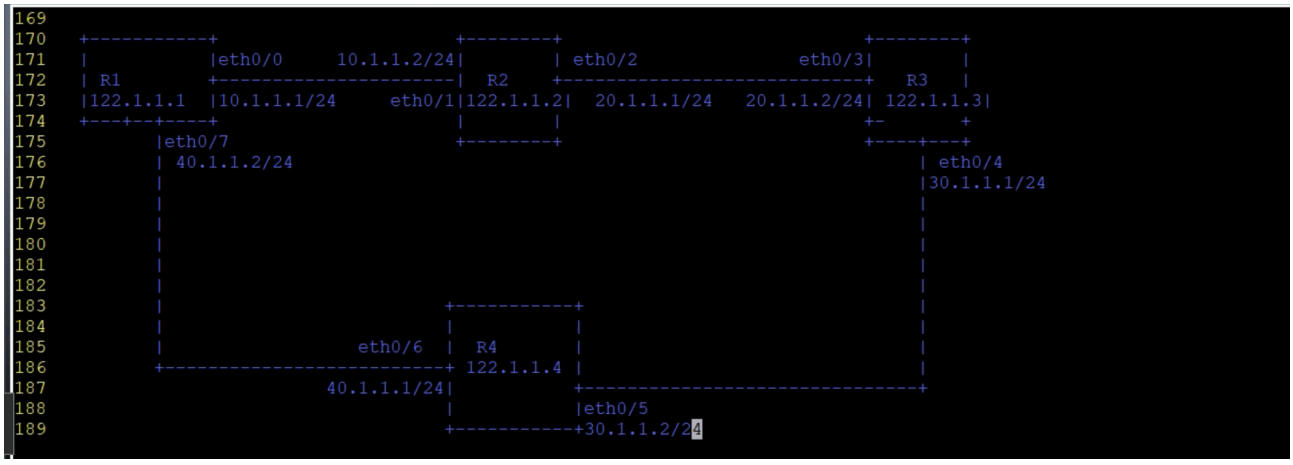


Fig 3 : The nodes topology

The text-dependent Command Line Interface (CLI) is a user interface (UI) used for computer interaction, data file management, and program execution. Character user interface, console user interface, and command line user interface are some other names for it. Command Line Interfaces (CLIs) receive keyboard commands and process them on a prompt command basis on the machine. Nowadays, the majority of suppliers employ graphical user interfaces (GUIs), which come with default operating systems like Windows, Linux, and macOS. Both graphical user interfaces and command-line interfaces are used by the majority of contemporary Unix-based systems.

2.5 COMMUNICATION SETUP

As our network graph is fully setup and we can interact with our routing device using CLI.

1. Our goal in this segment is to implement public API comm.h/comm.c.It is a setup where we can exchange packets between nodes.
2. We will describe the logical and physical view of the network.
3. We will create sockets and use this sockets to monitor threads.
4. We will start a socket monitoring thread.
5. We will start packet transmission, nodes communicate by sending destination port number with ip = 127.0.0.1
6. Then we do packet reception based on the echoed data received by the application.
7. Using auxiliary information, recipient interface can be known.
8. Actual packet without auxiliary data is received.

These variables specify how a datagram is fragmented. The Maximum Transfer Unit (MTU), which measures the length of the physical frames being carried on the network, has no bearing whatsoever on the length of an IP datagram. When a datagram exceeds the maximum transmission unit (MTU), it is divided into multiple segments with nearly same headers, each containing the maximum amount of data that may be contained in a single physical frame. The FRAGMENT OFFSET indicates the fragment's relative position inside the original datagram, and the IDENT flag is used to distinguish segments that are part of the same datagram. A fragmented datagram remains fractured until it reaches its destination.Once a datagram is fragmented it stays like that until it receives the final destination. If one or more segments are lost or erroneous the whole datagram is discarded.

2.6 IMPLEMENTING ARP

ARP (address resolution protocol) is used to map ip address to Mac address.The MAC address of the destination node is required to send data to the node in the same network.The ARP request contains the ip address of the destination device along with its own MAC address.The destination node replies with the MAC address of its own.
It operates in the data link layer which is helps in transferring the data physically. The steps followed are:-

1. We will set the interface to send and receive data.
2. The ethernet packet header format is defined.
3. Acceptance or Rejection of the packet depends on many factors including interface operating mode, interface configuration , packet contents.

Because IP and MAC addresses have different lengths and must be translated in order for the systems to recognize one another, the mapping process is crucial. IPv4 is currently the most widely used form of IP. A 32-bit number is an IP address. MAC addresses, however, are 48 bits long. The 32-bit address is translated to 48 using ARP, and vice versaOne networking paradigm is the Open Systems Interconnection (OSI) model. The OSI model, which was created in the late 1970s, uses layering to help IT teams see a specific networking system. By doing so, it is possible to ascertain which device, application, or network software is impacted by which layer, which aids in identifying the IT or engineering personnel in charge of managing that layer. Data transfer is made possible by the data link layer, commonly referred to as the MAC address, which establishes a connection between two physically linked devices when connection formation and termination are finished.On computer networks, the Network Layer, often known as the Internet Protocol (IP), is responsible for data packet forwarding via various routers.ARP performs the role in between these layers.

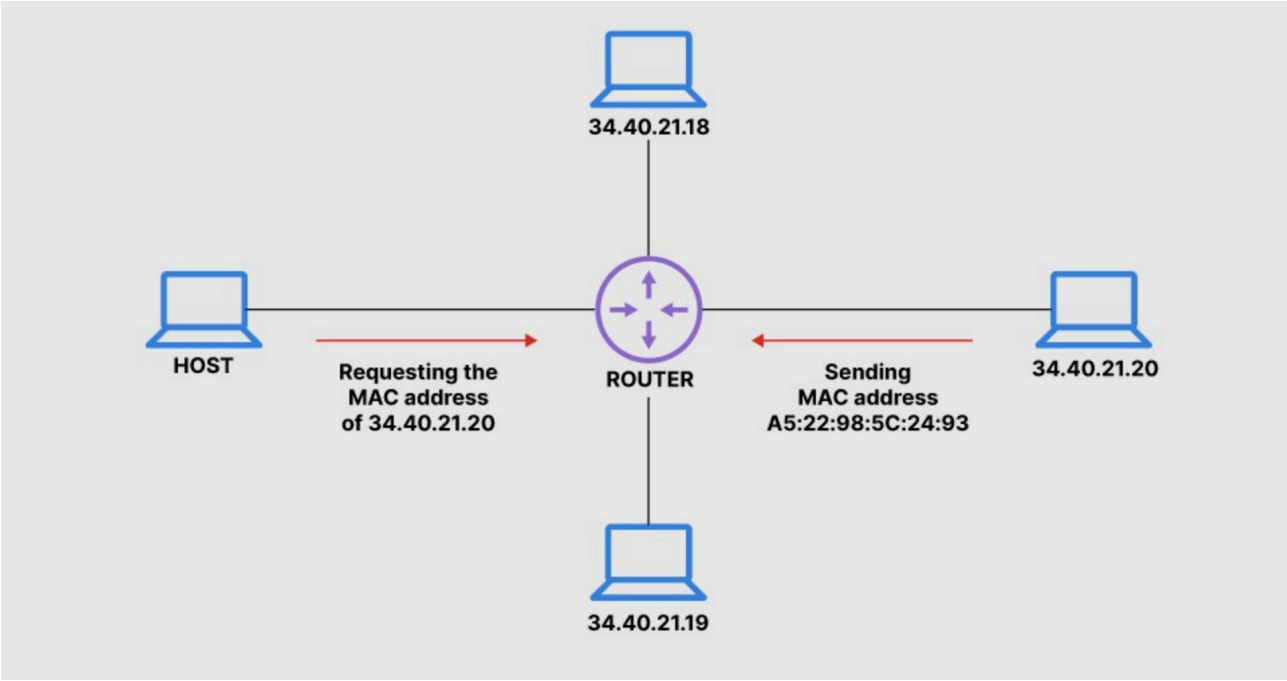


Fig 5: ARP resolution

Systems Interconnection (OSI) model. The OSI model, which was created in the late 1970s, uses layering to help IT teams see a specific networking system. By doing so, it is possible to ascertain which device, application, or network software is impacted by which layer, which aids in identifying the IT or engineering personnel in charge of managing that layer. Data transfer is made possible by the data link layer, commonly referred to as the MAC address, which establishes a connection between two physically linked devices when connection formation and termination are finished.

2.7 IMPLEMENTING L2 SWITCHING

In computer networking L2 switching is a very crucial component which operates in the data link layer .When a device sends data to another device on the same network , L2 examines which port to forward the packet based on the MAC address table in memory.

The steps which are followed are:-
1. The MAC address forwarding is initialised.
2. The L2 switches IP address is configured on its interfaces.
3. The switch inspects through all ethernet header of any frame passing through it.
4. All the address are stored in the Mac table in the L2 switch and is configured to operate in access mode.
5. We implement TRUNK access mode also.

2.8 IMPLEMENTING VLAN BASED FORWARDING

VLAN is a networking technique where we divide the physical networks into smaller virtual networks.Each VLAN is a logical group which can communicate with each other and its seems they are connected on the same physical network.
In VLAN based forwarding, network switches are used to segment the physical network not multiple virtual networks.Each VLAN is assigned with a unique ID, which can be used to identify traffic in the network.
When a switch receives a data , it examines the VLAN ID in the packet header to examine which packet it belongs .
The steps followed are:-
1. Conversion of the tagged frame to untagged frame.
2. Configuring API's to VLAN membership.
3. The frame egress table is filled with the data which is stored as cache memory in the VLAN.

One physical interface of the router is linked to a switchport in VLAN10, while another physical interface is linked to a switchport in VLAN20. The router does what all routers do, which is route IP packets between subnets. As a result, it has one interface in subnet 192.168.1.0/24 and one interface in subnet 10.1.0.0/24.
Forwarding to the main port is easily possible because the forwarding tools which is been used the nodes.

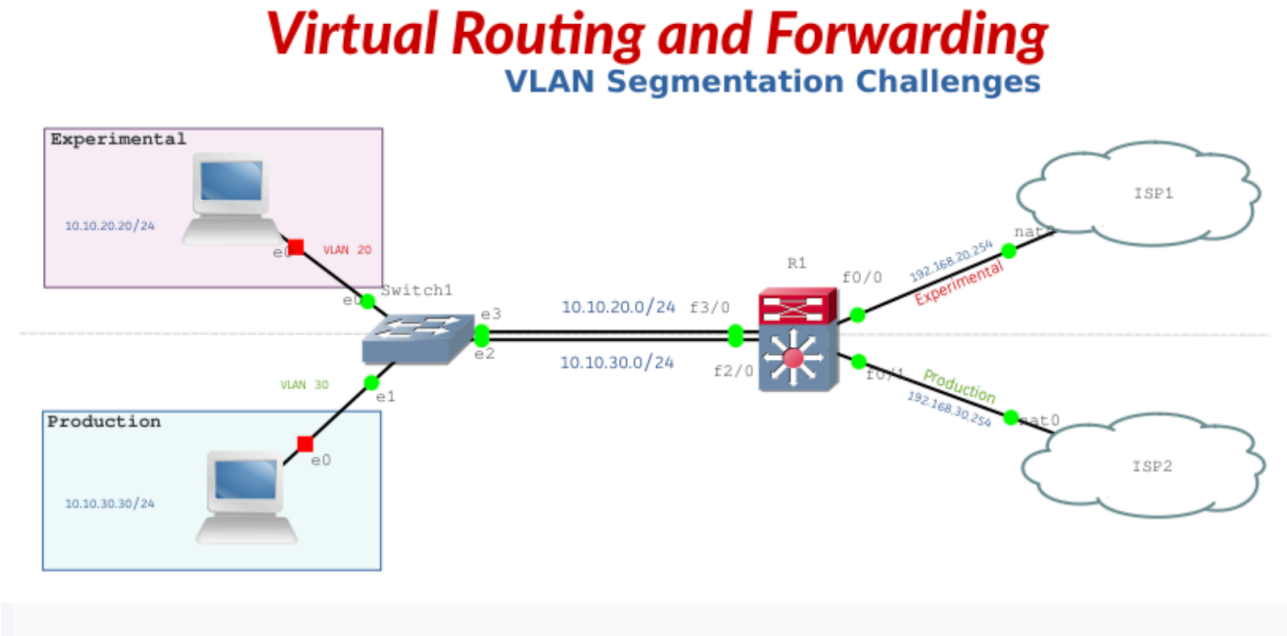


Fig 6: VLAN based forwarding

2.9 SETTING UP L3 ROUTING INFRASTRUCTURE

L3 routing infrastructure also known as Layer 3 routing infrastructure, refers to the network architecture and devices used to routing and managing traffic between different IP subnets or networks at the network layer (L3) of the OSI model.

The key components of L3 routing infrastructure include:-

1. Routers are network devices
2. L3 routing infrastructure forms the backbone of large- scale networks, allowing for efficient and secure communication between different networks.It enables the routing of packets based on IP addressing.
3. L3 route installation configuration.
4. Topology used to configure ARP's.
5. Implementing Ping as a application to test our L3 code.
6. TCP/IP stack layer interaction.
7. Routing Ping CLI's payload data transfer from L2 to L3 switching payload data transfer from L2 to L3 switching

In our TCP/IP stack implementation project using C, we have incorporated Layer 3 (L3) infrastructure to facilitate efficient routing of packets between nodes. The L3 infrastructure is crucial as it enables the determination of the best path for data packets to travel across the network, ensuring they reach their intended destinations.

Our network topology consists of four nodes, each configured with unique IP addresses and subnet masks. The use of L3 infrastructure allows us to implement dynamic and static routing protocols, which

are essential for the forwarding decisions made by each node. Static routing tables are configured at each node, specifying the next hop for each destination IP address. This ensures that packets are directed correctly through the network from source to destination.

Additionally, the L3 infrastructure supports Address Resolution Protocol (ARP) to map IP addresses to MAC addresses, essential for local network communication. ARP requests and responses ensure that nodes can discover each other's physical addresses, allowing seamless packet transmission.

The implementation of L3 infrastructure in our TCP/IP stack not only enhances the routing capabilities but also contributes to the overall robustness and efficiency of the network. It ensures that our stack can handle complex routing scenarios and provides a solid foundation for future scalability and enhancements in our project.

2.10 IMPLEMENTING LAYER 2 AND LAYER 3 ROUTING FLOWCHARTS

Layer 2 is very extensively used and is very good in functioning.The layer 2 payload transfer from L2 to L3 .It checks if there is a matching subnet or else it gets transferred to the next hop.When the packet is transferred to the next hop it is checked for the subnetting.Then it is checked for the Mac address and then finally it is routed for the next hop.The current processing node is taken for consideration.There is no if as destination is present in local subnet.Network layer needs to tell data link layer what payload it is passing down.

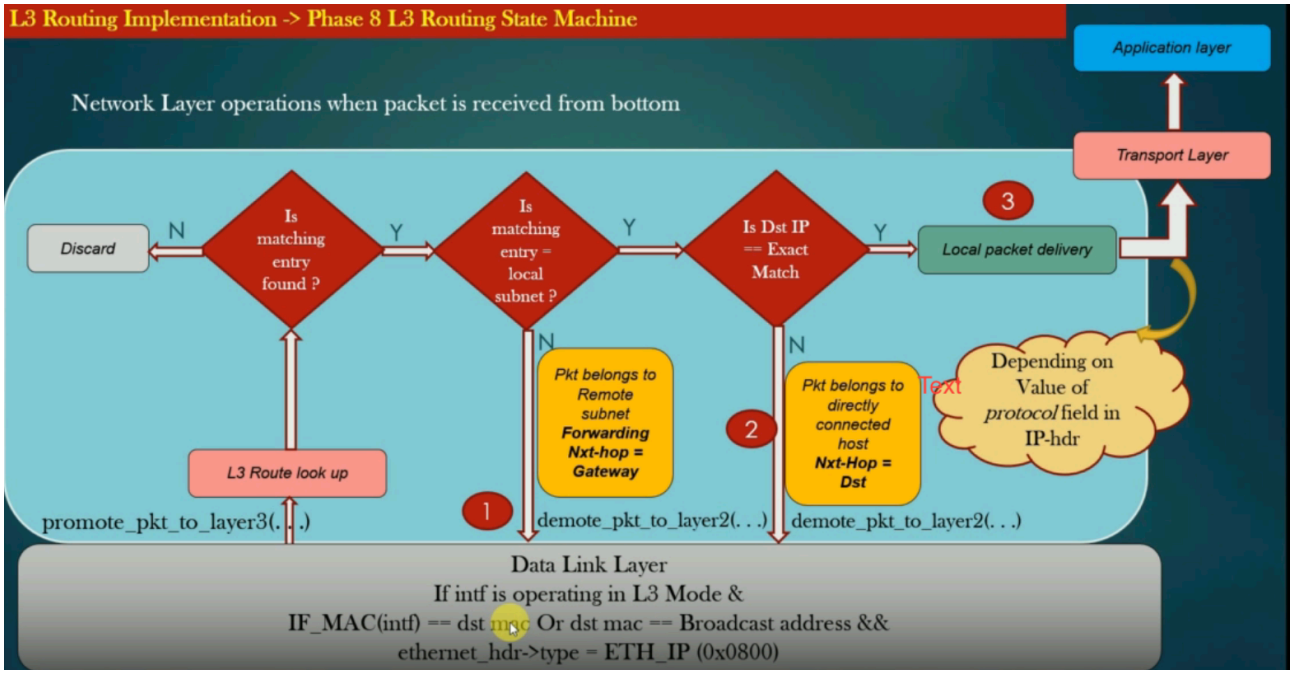


Fig 7: L3 routing state machine.

Then we create ethernet HDR and then the port is assigned a subnet.Then we resolve the arp for the next hop.Then the packet is promoted for the next step.

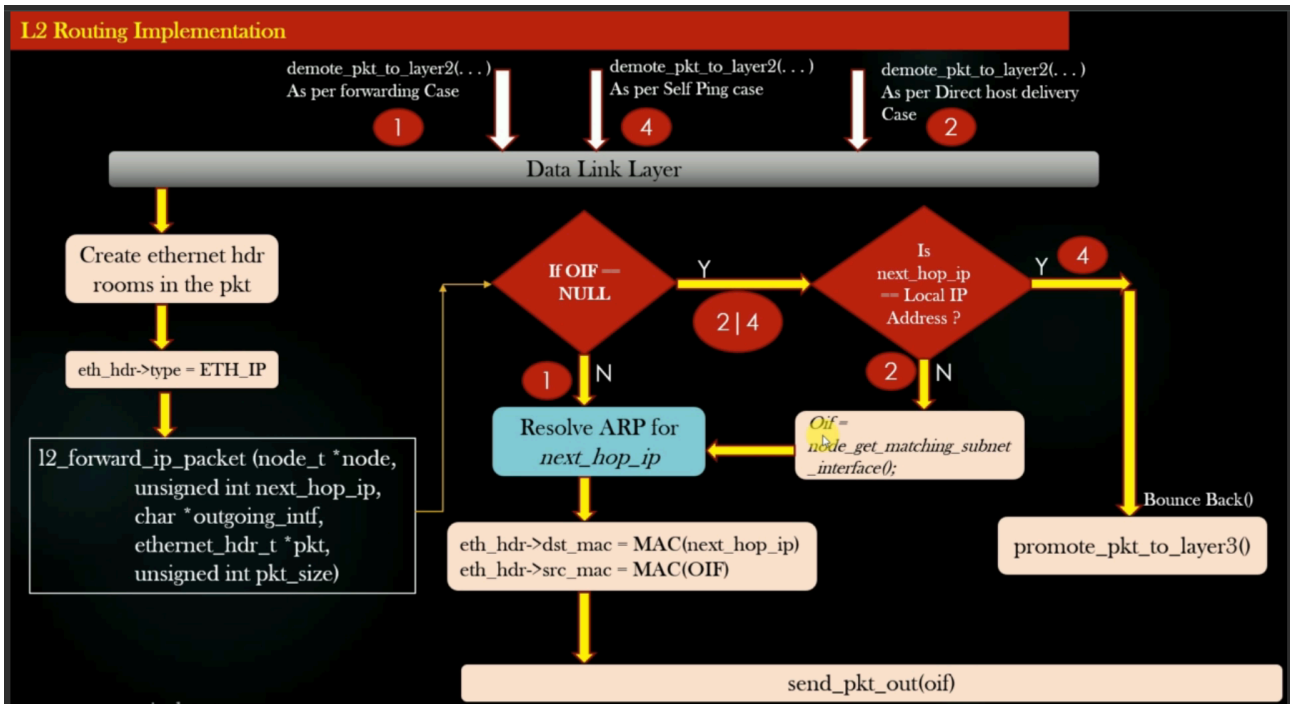


Fig 8: L3 to L2 routing

2.11 ARP ON DEMAND DESIGN

Now there is a problem ARP broadcast request on interface eth6 and wait for ARP reply.And there is a question which is associated what will a router will do until ARP request is not resolved .There are other packets in the queue and needs to be addressed by the same router.
The solution to that is very simple that we stored the temporary packet and serve the and got busy processing the packets.
1. ARP Cache:
- An ARP cache is a table that each node keeps that contains recently resolved IP-to-MAC address mappings.
- Entries in the ARP cache have a time-to-live (TTL) to ensure they are periodically refreshed or removed if no longer needed.

2. ARP Request and Reply Mechanism:
- When a node needs to send a packet to an unknown IP address, it broadcasts an ARP request packet on the local network.
- The requested IP address's owner, the target node, replies with an ARP reply that includes its MAC address.
- The initiating node updates its ARP cache with the received mapping and proceeds with packet transmission.

Workflow
1. Packet Transmission Attempt:
- A node attempts to send a packet to a destination IP address.
- It first checks its ARP cache for an existing IP-to-MAC mapping.

2. ARP Cache Miss:
   - If the mapping is ,found in the ARP cache, the node generates and broadcasts an ARP request.
   - The ARP request packet contains the sender's IP,The MAC addresses and the target IP address.
3. ARP Request Handling:
   -Every node within the local network obtains the ARP request
   - An ARP reply including the MAC address is sent by the node that has the matching destination IP address.
4. ARP Reply Processing:
   - The originating node changes its ARP cache after receiving the ARP reply, then saves the updated mapping.receives the ARP reply.
   - The node then retransmits the original packet using the resolved MAC address.
5. Cache Management:
   - ARP cache entries have a TTL to ensure stale mappings are periodically purged.
   - Nodes may also handle ARP replies unsolicitedly, updating their caches if relevant.

Advantages of ARP On-Demand Design

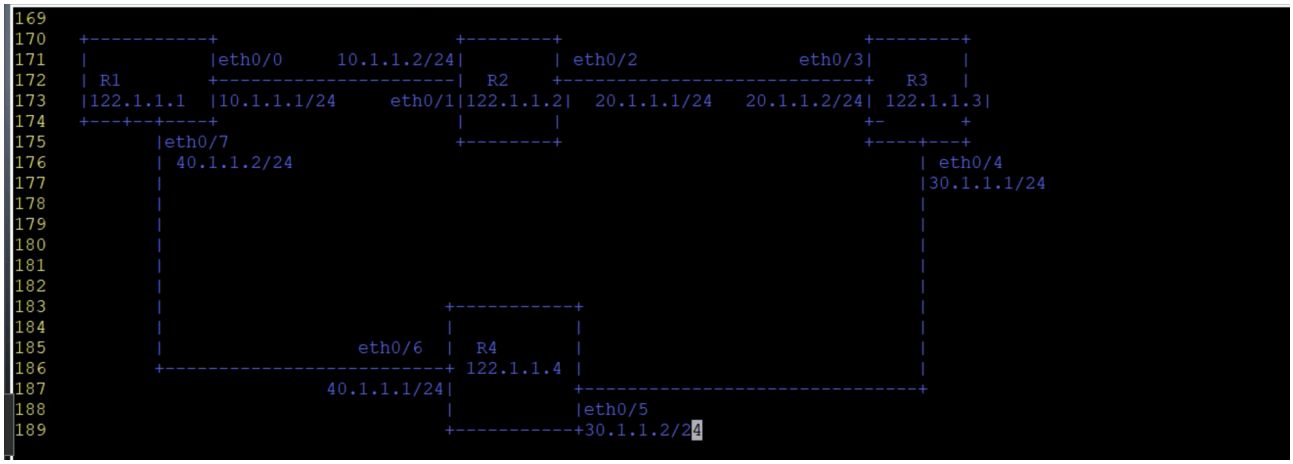
Efficiency: Reduces unnecessary ARP traffic, conserving network bandwidth and reducing processing overhead.

- Scalability: Supports larger networks by limiting the frequency and volume of ARP broadcasts.
- Resource Optimization: Maintains a lean ARP cache, storing only actively needed mappings.

Fig 9: ARP on demand

PROJECT RESULT

The implementation shows good performance under light to moderate network load. Under heavy load, buffer management and processing speed need optimization.The modular design allows for easy scalability. Adding support for IPv6 or other transport protocols can be achieved with minimal changes to existing code.
Basic error handling mechanism are in place, but more robust error correction and retransmission strategies could improve reliability, especially for TCP.Now when we run the command ping which is. connected to router R1 and we call for the subnet.I send the first packet and then it is rerouted to the next router R2 and ARP resolution is observed.The frame L2 is accepted in node R2.
Then we again do the same for all the 4 routers and we can see these are connected.



L2 router received the packet

Then I repeat the process for all the nodes.The first packet is recieved in eth0/0 with subnet 10.1.1.1/24 . Then it goes to the next node having ethernet address as eth0/2 and having subnet 20.1.1.1/24. The ARP request is fetched and we move towards the next node having address eth0/3 and subnet as 20.1.1.2/24 .The ARP request is then allowed and fetched for the next node.The next node is R4.The eth0/4 is the next node and the subnet is 40.1.1.1/24.These gives the flexibility of connecting all the nodes and the packets can easily transfer from each node.
Now, the packets are transferred to different nodes and the process is repeated and finally the whole setup is checked for ARP connection. This setup allows for seamless connectivity between all nodes, ensuring that packets can traverse the network efficiently
The ARP request is then allowed and fetched for the next node.The next node is R4.The eth0/4 is the next node and the subnet is 40.1.1.1/24.These gives the flexibility of connecting all the nodes and the packets can easily transfer from each node.

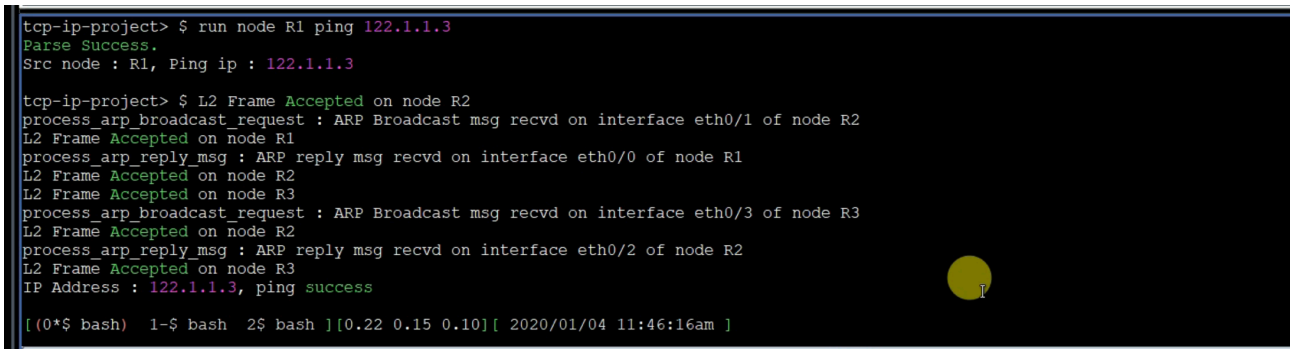


Fig 11: Connection established

CONCLUSION

In summary, the whole process of implementing a TCP/IP stack using C has been quite a complex and interesting activity that showed how network communication protocols operate. With proper design and careful coding we have come up with an exceptional stack which transmits data across diverse network configurations more efficiently than before. This project has further enhanced my understanding on some of the key networking rules like packet routing, error handling and ARP resolution among others; as such I now appreciate their significance in real life scenarios. Successful integration and testing of several protocols into our stack is a proof that what we had was not only versatile but also reliable. By simulating and resolving ARP requests, routing packets through different nodes, making sure that it was easy to just plug another node in, this assignment lays groundwork for any future improvements or scalability options. This project serves as both a practical application of theoretical knowledge as well as a useful precursor for more advanced networking projects thereby contributing to our knowledge and skills in computer networks domain.

FUTURE SCOPE

I've created a configuration that can accommodate additional nodes despite only having four.I've employed more straightforward protocols that can be expanded for increased performance and security.The utilization of additional protocols at various network tiers can improve the quality of service.It is possible to improve fault tolerance and error handling to function in a noisy environment.I've employed static routing techniques that can be expanded upon and solved dynamically.Additionally, a GUI can be implemented to improve the project's visual appeal.The error handling and fault tolerance can be increased to work in noisy environment.The routing methods which I have used are static which can be extended and dynamically can be solved.And to add GUI can be added to give better visuality to the project.

REFERENCE

1. W. R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*, 1st ed. Boston, MA, USA: Addison-Wesley, 1994.

2. W. R. Stevens, B. Fenner, and A. M. Rudoff, UNIX Network Programming, Volume 1: The Sockets Networking API, 3rd ed. Boston, MA, USA: Addison-Wesley, 2003.

3. C. M. Kozierok, The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference, 1st ed. San Francisco, CA, USA: No Starch Press, 2005.

4. . F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach*, 7th ed. Boston, MA, USA: Pearson, 2016.

5. D. E. Comer, Internetworking with TCP/IP Volume One: Principles, Protocols, and Architecture, 6th ed. Boston, MA, USA: Prentice Hall, 2013.

6. The TCP/IP Guide, "TCP/IP Overview and History," The TCP/IP Guide, 2005. [Online]. Available: http://www.tcpipguide.com/free/t_TCPIPOverviewandHistory.htm.

7. D. S. Fallows, "Socket programming in C," Linux Journal, Jan. 1995. [Online]. Available: http://www.linuxjournal.com/article/2076