National Institute Of Technology Srinagar

INFORMATION TECHNOLOGY

Computer Networking

Project Report

Network Simulator

Student Name	Student ID
1. Arpunit Kour	2020BITE004
2. Syeda Azher Fatimah	2020BITE037
2. Asrar Ul Haq	2020BITE092

Lecturer in charge:

Mrs Iqra Altaf Gillani

Submission Date : 25/06/2023

1 Introduction

The network communication simulation project aims to replicate and study the process of communication between devices in a computer network. It provides a hands-on experience of how data is transmitted, processed, and received across various layers of the network protocol stack. This report provides an overview of the project, its objectives, and the key components involved.

The project consists of several components that simulate different aspects of network communication. These components include the Hub, CRC (Cyclic Redundancy Check), Physical Layer, Data Link Layer, Network Layer, Transport Layer, and Application Layer. Each component has its own set of functionalities and responsibilities in the overall communication process.

The Hub serves as a central point for connecting devices and relaying data packets. It manages the flow of data and ensures proper transmission to the intended destination. The CRC component is responsible for error detection and ensures the integrity of data during transmission.

The Physical Layer handles the transmission and reception of raw data bits over the physical medium. It deals with aspects such as signaling, voltage levels, and physical connectors. The Data Link Layer provides framing, addressing, and error control mechanisms to facilitate reliable data transfer between directly connected devices.

The Network Layer focuses on the logical addressing and routing of data packets across different networks. It assigns IP addresses to devices using the DHCP (Dynamic Host Configuration Protocol) and maintains an IP address table. The Transport Layer manages the end-to-end communication between source and destination devices, establishing and closing TCP (Transmission Control Protocol) connections.

Finally, the Application Layer enables communication between applications running on different devices. It allows users to exchange data and interact with network services. The project simulates the communication between devices at this layer, showcasing the exchange of messages and data between applications.

The simulation process involves configuring devices, setting up connections, transmitting data, and monitoring the flow of information. Devices are assigned IP addresses using DHCP, and TCP connections are established and closed as required. The simulation provides a realistic environment to observe how devices interact and communicate within a network.

In summary, the network communication simulation project offers a comprehensive understanding of network communication protocols and provides valuable insights into the functioning of different network layers. By simulating real-world scenarios, this project enables students and researchers to gain practical knowledge and hands-on experience in network communication. The subsequent sections of this report delve into the project's implementation details, simulation process, results, and observations, providing a comprehensive analysis of the network communication simulation.

2 Objectives and Learning Outcomes

1. Understand the fundamental concepts of network communication protocols and the layered architecture of the network protocol stack.
2. Gain practical knowledge of configuring network devices, assigning IP addresses, and establishing TCP connections.
3. Simulate the flow of data packets across different network layers and observe the behavior of devices during data transmission.
4. Implement error detection mechanisms, such as CRC, to ensure data integrity.
5. Study the role of each network layer, including the Physical Layer, Data Link Layer, Network Layer, Transport Layer, and Application Layer.
6. Explore the functionalities and responsibilities of different network components, such as the Hub, DHCP, and TCP.
7. Analyze the challenges and limitations of network communication and propose possible solutions.
8. Enhance problem-solving skills and critical thinking by troubleshooting network communication issues within the simulation.
9. Foster teamwork and collaboration by simulating communication between multiple devices and applications.

3 System Architecture:

The network communication simulation project is built upon a layered architecture, following the principles of the network protocol stack. The system architecture consists of multiple interconnected components that work together to simulate network communication and data transmission. Each component corresponds to a specific layer in the network protocol stack and performs its designated functions. The following sections describe the key components and their interactions within the system.

- Physical Layer:

  The Physical Layer represents the lowest layer in the network protocol stack. It deals with the actual transmission and reception of raw data bits over the physical medium. In the simulation, the Physical Layer is emulated by the network devices, including Device A, Device B, Device C, Device D, and Device E. These devices are responsible for transmitting and receiving data packets.

- Data Link Layer:

  The Data Link Layer is responsible for providing reliable and error-free communication between directly connected devices. It handles the framing of data packets, performs error detection using CRC , and manages MAC addresses. The Data Link Layer is simulated by the Data link layer module, which receives data from the Physical Layer and applies framing and error detection techniques.

- Network Layer:

  The Network Layer is responsible for addressing and routing data packets across multiple networks. It assigns IP (Internet Protocol) addresses to devices and manages the routing of data. In the simulation, the Network Layer is represented by the NetworkLayer class and the DHCP (Dynamic Host Configuration Protocol) module. The DHCP module assigns IP addresses to devices, while the NetworkLayer class maintains an IP address table and provides methods for IP address assignment and retrieval.

- Transport Layer:

  The Transport Layer ensures reliable and orderly delivery of data between source and destination devices. It establishes and manages end-to-end connections, assigns port numbers to applications, and handles segmentation and reassembly of data. The Transport Layer is implemented using the TransportLayer class, which maintains a port table for assigning and retrieving port numbers for devices.

- Application Layer:

  The Application Layer represents the highest layer in the network protocol stack. It provides services and protocols for network applications to exchange data. In the simulation, the Application Layer is simulated by the ApplicationLayer class, which facilitates communication between devices by sending messages to other devices at the Application Layer.

- Hub:

  The Hub component represents a network device that connects multiple devices together in a star topology. It receives data packets from the source device and forwards them to the destination device(s). The Hub is implemented as a class and supports the Star Topology Connection mode.

- Switch:

  The Switch component represents a network device that connects multiple devices together in a dedicated network connection mode. It maintains a table of connected devices and performs switching of data packets based on MAC addresses. The Switch is implemented as a class and supports the Dedicated Network Connection mode.

4 Implementation Details

The implementation of the network communication simulation project involved the development of various modules and classes to emulate the behavior of different network layers and devices. This section provides an overview of the key implementation details and techniques used in the project.

- Programming Language and Framework:

  The project was implemented using Python programming language, leveraging its simplicity and extensive libraries for networking and simulation. Python provides a rich set of features that facilitated the implementation of the network protocol stack and device behaviors.

- Object-Oriented Design:

  The project followed an object-oriented design approach, which helped in organizing the code into modular and reusable components. Each layer of the network protocol stack was represented by a separate class, encapsulating its specific functionality and interactions with other layers.

- Class Hierarchy:

  The project implemented a class hierarchy to represent the network layers and devices. The NetworkLayer, Data link layer, TransportLayer, and ApplicationLayer classes represented the respective layers of the network protocol stack. The Device class served as the base class for the network devices, with specific devices like Device A, Device B, etc., inheriting from it.

- Layered Communication:

  The communication between different layers of the network protocol stack was facilitated through method calls and data passing. Each layer received data from the layer above it, performed its designated operations, and passed the processed data to the layer below it. This layered communication ensured the proper flow of data and adherence to the protocol stack.

- Data Encapsulation:

  The project implemented the concept of data encapsulation, where data from higher layers was encapsulated within specific headers and control information as it moved down the protocol stack. For example, the Data Link Layer added frame headers and CRC checksums to the data received from the Network Layer.

- IP Address Assignment:

  The DHCP (Dynamic Host Configuration Protocol) module was responsible for assigning IP addresses to devices in the Network Layer. It maintained a pool of available IP addresses and assigned them to devices upon request. The assigned IP addresses were stored in the IP address table of the Network Layer for later retrieval.

- Port Assignment:

  The Transport Layer utilized the TransportLayer class to assign port numbers to devices. The assigned port numbers were stored in the port table, which facilitated the establishment and management of end-to-end connections.

- Hub and Switch:

  The project implemented two types of network devices: Hub and Switch. The Hub class represented a hub device that connected devices in a star topology. The Switch class represented a switch device that facilitated dedicated network connections. Both devices received data packets from the source device and forwarded them to the destination device(s) based on the specified connection mode.

- Error Handling and Validation:

  The implementation included robust error handling and data validation mechanisms. Input data, such as user choices, device data, and IP addresses, were validated to ensure their correctness and adherence to the required format. Error messages were displayed when invalid inputs or inconsistent configurations were detected.

- User Interface:

  The project utilized a text-based user interface to interact with the users. Users could input their choices, view the status and data of devices, and observe the flow of data packets through the network layers. The user interface provided a user-friendly and intuitive way to interact with the network simulation.

The implementation details of the project demonstrated a systematic approach to emulate the behavior of network layers and devices. Through the effective use of programming techniques, the project provided a realistic simulation of network communication and facilitated a deeper understanding of networking concepts and protocols.

5 Simulation Process

The simulation process of the network communication project involved the execution of various steps and interactions among the network layers and devices. This section outlines the simulation process and the flow of data through the different layers of the network protocol stack.

- Physical Layer:

  - The simulation started with the user inputting the source and destination devices for communication. - The user also provided the source device data in binary format. - The physical layer received the source device data and validated it for correctness. - The source device data was stored in the respective device object. - The user also chose the type of network connection: either a star topology or a dedicated network connection.

- Data Link Layer:

  - The data link layer received the source device data from the physical layer. - It encapsulated the data into a frame by adding frame headers and a CRC (Cyclic Redundancy Check) checksum. - The frame was then forwarded to the network layer for further processing.

- Network Layer:

  - The network layer received the frame from the data link layer. - The DHCP module in the network layer assigned IP addresses to the devices if not already assigned. - The IP addresses were stored in the IP address table of the network layer. - The IP addresses of the source and destination devices were retrieved from the table. - The network layer passed the frame to the transport layer for transmission.

- Transport Layer:

  - The transport layer received the frame from the network layer. - It assigned port numbers to the devices using the TransportLayer class. - The port numbers were stored in the port table of the transport layer. - The TCP module in the transport layer established a TCP connection between the source and destination devices based on their port numbers. - The TCP connection was established for reliable end-to-end communication. - Data packets were transmitted between the devices through the established TCP connection. - The TCP module in the transport layer also facilitated the closing of TCP connections.

- Application Layer:

  - The application layer received the data packets from the transport layer. - Each device in the application layer sent messages to other devices using the ApplicationLayer class. - The messages were sent to the respective destination devices. - The application layer facilitated communication and message exchange between devices.

6 Summarize

The simulation process for the network communication project involved several key steps and interactions among the different layers of the network protocol stack. It began with the user inputting the source and destination devices, along with the source device's data in binary format. The physical layer received and validated the data, storing it in the respective device object. The user also selected the type of network connection, either a star topology or a dedicated network connection.

The data link layer encapsulated the source device data into a frame by adding headers and a CRC checksum. The frame was then passed to the network layer, where IP addresses were assigned to devices using the DHCP module. The IP addresses were stored in the network layer's IP address table. The transport layer assigned port numbers to the devices and established TCP connections based on these ports, enabling reliable end-to-end communication. The TCP module also facilitated the closing of TCP connections when needed.

At the application layer, messages were sent between devices using the ApplicationLayer class, enabling communication and message exchange. In the case of a star topology connection, the hub device received data packets from the source device and forwarded them to the destination device(s) based on MAC addresses. In the case of a dedicated network connection, the source device's data was replaced with "0," indicating no transmission, while the destination device's data was updated with the source device's data to simulate communication between specific devices.

Throughout the simulation process, users received feedback and information regarding device states, data transmission, IP address assignments, port numbers, and connection establishment or closure. They could observe the data before and after transmission, IP address assignments, MAC address tables, and TCP connection statuses.

Overall, the simulation process effectively emulated the flow of data through the network layers and demonstrated the behavior of devices in a network communication scenario. It provided users with a practical understanding of network protocols and the interactions involved at each layer of the network protocol stack.

7 References

[1] Stack Overflow.

[2] Cisco Networking Academy

[3] Network World

[4] Computer Networking: A Top-Down Approach by James F. Kurose and Keith W. Ross