

## **Assignment No:1**

**Title:** Setting up a Local Area Network (2PC/4 PC) and configuration for sharing resources. It includes preparation of cable, testing of cable using line tester, performing an Initial Switch Configuration and Initial Router Configuration, configuration machine using IP addresses, testing using PING utility and demonstrating the PING packets captured traces using Wireshark Packet Analyzer Tool.

**Objective:** To learn the network architecture.

**Outcome:** Choose the appropriate network topology and connecting media for configuring the network.

**Tools Required:** Computer, LAN Cards, RJ-45 Connectors, Switch, CAT-5 Cable, Cable tester, Crimping tool, etc.

**Theory:**

### **1. Introduction:**

A computer network is a system of interconnected devices that communicate and share resources with each other. These devices, known as nodes, can include computers, servers, printers, switches, routers, and other hardware components. The primary purpose of a computer network is to enable the exchange of data, facilitate communication, and allow multiple users to access shared resources like files, applications, and internet connections.

In a LAN, various networking devices like switches, routers, and cables play crucial roles. A switch is a device that connects multiple devices within the same network and uses MAC addresses to forward data only to the intended device. Routers, on the other hand, connect different networks and use IP addresses to route data packets between them.

### **How to Crimp a Cat 5 cable with RJ 45 Connector:**

- Strip approximately 1 inch of the cable jacket to expose the wires.
- Untwist each wire pair and straighten them.
- Arrange the wires according to the desired wiring standard (T568A or T568B).
- Recheck the wire order against the wiring diagram.

- Hold the wires tightly and cut them at a 90-degree angle, leaving about 1/2 inch from the cable jacket.
- Insert the wires into the RJ45 connector with the pins facing up. Ensure the cable jacket is slightly inside the connector.
- Place the connector in the crimping tool and squeeze firmly until fully crimped.
- For a straight-through cable, repeat the process using the same wiring pattern.
- Use a cable tester to ensure proper continuity and correct wiring.

**Cable Testing Tool:**

- A cable tester checks for continuity between the two terminals and identifies the type of crimp used, ensuring no breaks in the connection.

**Conclusion:** Hence, will have successfully set up a LAN with 2 or 4 PCs, configured essential network devices, tested connectivity, and analyzed network traffic.

## Assignment No: 2

**Title:** Write a program for error detection and correction for 7 or 8 bits ASCII codes using CRC.

**Objective:** To learn various networking protocols & layers.

**Outcome:** Apply the Error control and Flow control technique of data link layer.

### Software & Hardware Requirements:

Operating System: Ubuntu

Programming Language: C++

### Theory:

#### 1. Introduction:

CRC is a popular method for detecting errors in digital data. It works by treating the data as a large binary number and dividing it by a predetermined binary divisor. The remainder of this division process is the CRC code, which is appended to the data before transmission. When the data is received, the same division process is applied, and if the remainder is zero, the data is assumed to be correct.

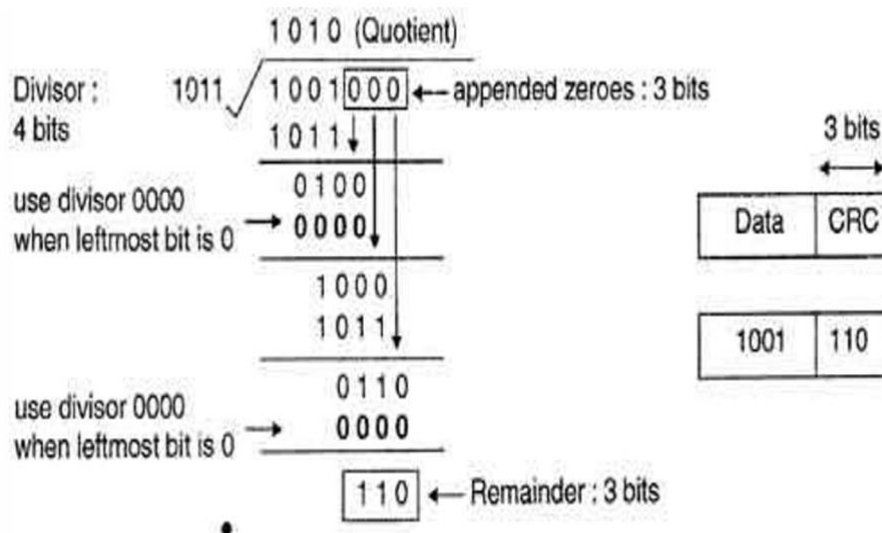
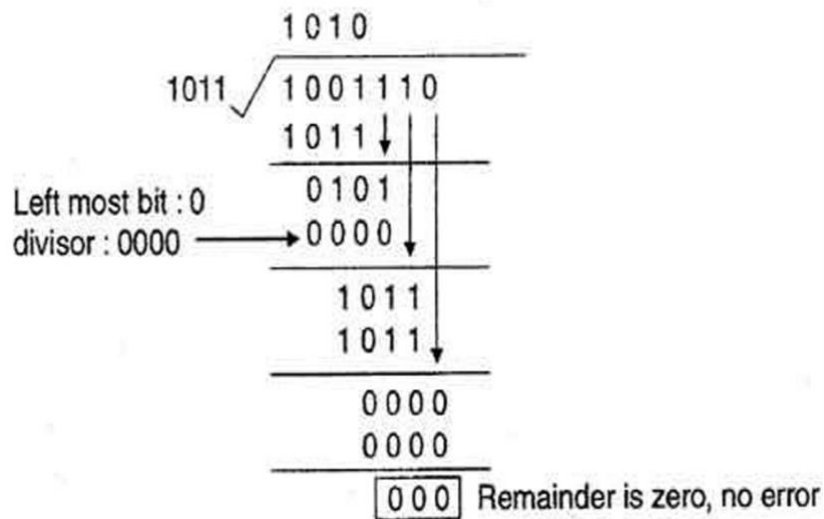


Fig 2.1: CRC Generation at sender side



**Fig 2.1: Identify Error at Receiver side**

#### **Procedure:**

- Choose a sample ASCII character (e.g., 'A') and convert it to its binary equivalent.
- Select a suitable CRC polynomial (e.g., 1101 for CRC-3).
- Append the CRC bits to the original data to form the transmitted message.
- Introduce an error in the transmitted message to test the error detection.
- Apply the CRC check at the receiving end to detect and correct errors.

**Conclusion:** Hence, we have Successfully implemented program for error detection and correction for 7/8 bits ASCII codes using CRC.

### **Assignment No:3**

**Title:** Write a program to simulate Go-Back-N and Selective Repeat modes of sliding window protocol.

**Objective:** To learn various networking protocols & layers.

**Outcome:** Apply the Error control and Flow control technique of data link layer.

#### **Software & Hardware Requirements:**

Operating System: Ubuntu

Programming Language: C++, Java

#### **Theory:**

##### **Sliding Window Protocol:**

The sliding window protocol is a method of flow control in network communication that allows a sender to send multiple frames before needing an acknowledgment for the first one. It improves the efficiency of data transmission by keeping the communication channel busy while waiting for acknowledgments.

##### **Go-Back-N (GBN):**

In the Go-Back-N protocol, the sender can send several frames specified by a window size without receiving an acknowledgment. However, if an error is detected in a frame, all subsequent frames are retransmitted, starting from the erroneous one.

##### **Selective Repeat (SR):**

Selective Repeat is more efficient than GBN. In SR, only the erroneous frames are retransmitted, rather than all subsequent frames. This allows for better utilization of the communication channel, especially in environments with higher error rates.

#### **Procedure:**

- Represent data frames as a sequence of numbers (e.g., Frame 0, Frame 1, etc.).

- Define a window size that dictates how many frames can be sent before needing an acknowledgment.
- Simulate frame errors by randomly marking certain frames as lost or corrupted.
- Implement the retransmission logic based on GBN or SR protocols.

**Conclusion:** Hence, we have successfully implemented Go-Back-N and Selective Repeat Modes of Sliding Window Protocol.

## **Assignment No:4**

**Title:** Implementation of subnetting Using packet tracer.

**Objective:** To understand and implement subnetting in a network using Cisco Packet Tracer by creating multiple subnets and configuring the IP addressing scheme for each network.

**Outcome:** Make use of different IP addressing and routing protocols for network configuration.

Tools Required:

- Cisco Packet Tracer software
- Basic knowledge of IP addressing and subnetting

### **Theory:**

Subnetting is the process of dividing a network into smaller, more manageable sub-networks or subnets. This allows efficient utilization of IP addresses and enhances security and network performance.

Steps for Subnetting:

#### **Step 1: Analyze the Network Requirements**

- Identify the IP address range (e.g., 192.168.1.0/24).
- Determine how many subnets are required.
- Calculate the subnet mask, network addresses, and host ranges.

#### **Step 2: Open Cisco Packet Tracer**

- Launch Cisco Packet Tracer software on your system.

#### **Step 3: Create the Network Topology**

##### **1. Add Devices:**

- Add routers, switches, and PCs according to your design. For example, use 1 router, 2 switches, and 4 PCs for two subnets.

##### **2. Connect Devices:**

- Use the appropriate cables (copper straight-through) to connect the devices (e.g., connect PCs to switches, switches to the router).

#### **Step 4: Configure the Router Interfaces**

1. Assign IP Addresses to Router Interfaces:
  - Click on the router, go to the CLI tab, and configure the interfaces with the subnetted IP addresses.

#### **Step 5: Assign IP Addresses to PCs**

- For each PC, assign IP addresses within the range of each subnet.
- Click on each PC, go to the Desktop tab, then click on IP Configuration to enter the IP address, subnet mask, and default gateway.

#### **Step 6: Verify Connectivity**

1. **Ping Test:**
  - Use the ping command to test connectivity between PCs in different subnets.
  - On PC1, open the Command Prompt and type:
  - Successful replies indicate that the network and subnetting have been correctly implemented.

**Conclusion:** Hence, we have successfully implemented subnetting using Cisco Packet Tracer.

## Assignment No:5

**Title:** Write a program for DNS lookup. Given an IP address as input, it should return URL and vice-versa

**Objective:** To write a program that performs Domain Name System (DNS) lookup. The program should be able to convert a given URL to its corresponding IP address and, conversely, convert a given IP address to its corresponding URL.

**Outcome:** Implement DNS lookup for forward (URL to IP) and reverse (IP to URL) resolution

### Software & Hardware Requirements:

Operating System: Ubuntu

Programming Language: C++

### Theory:

DNS (Domain Name System) is a protocol that translates human-readable domain names (e.g., [www.google.com](http://www.google.com)) into IP addresses (e.g., 172.217.10.110) and vice versa. It is essential for locating and accessing web resources across the internet.

DNS resolution works in two directions:

1. **Forward DNS Lookup (URL to IP):** This resolves a given domain name to its corresponding IP address. When a user enters a URL in a browser, DNS queries are sent to translate this domain name into the IP address where the requested website is hosted. For example, when accessing [www.google.com](http://www.google.com), the DNS resolver will return an IP address like 172.217.164.110.
2. **Reverse DNS Lookup (IP to URL):** This resolves an IP address back to its domain name. This is useful for diagnostic purposes, server logging, or security mechanisms. For example, resolving the IP 142.250.190.4 might return [google.com](http://google.com).

DNS uses a hierarchical and distributed database structure that allows efficient resolution of domain names to IP addresses. The DNS system consists of:

- **Root DNS Servers:** The highest level in the DNS hierarchy, these servers handle queries related to top-level domains (TLDs) like .com, .org, .net.

- **Top-Level Domain Servers (TLDs):** These handle the domains under them, such as .com, .net, and delegate queries to authoritative name servers for specific domains.
- **Authoritative DNS Servers:** These servers provide the actual mapping between domain names and IP addresses for a particular domain (e.g., google.com).

**Conclusion:** Hence, we have successfully implemented DNS lookups by performing both forward and reverse lookups, it showcases how DNS translates between human-readable URLs and machine-readable IP addresses.

## Assignment No:6

**Title:** Write a program using TCP socket using C++ for following:

- a. Say Hello to Each other (For all students)
- b. File transfer (For all students)
- c. Calculator (Arithmetic)

**Objective:** To develop a program in C++ that uses TCP sockets to perform the following tasks:

1. Say Hello to each other (client-server communication).
2. File transfer between client and server.
3. Implement an arithmetic calculator.

**Outcome:** Implement a transport layer protocols to facilitate communication between different nodes.

### Software & Hardware Requirements:

Operating System: Ubuntu

Programming Language: C++

### Theory:

TCP (Transmission Control Protocol) is a connection-oriented protocol that provides reliable communication between client and server over a network. TCP guarantees that data sent by one party reaches the other in the same order and without errors.

- **Sockets:** Sockets are endpoints for communication between two machines (client and server). TCP socket communication follows a client-server model where the server listens for connections and the client initiates them.
- **TCP Socket Workflow:**
  1. **Server Setup:** The server socket is created and bound to an IP address and port. It listens for incoming connection requests from clients.
  2. **Client Setup:** The client socket connects to the server's IP address and port.
  3. **Communication:** Data is exchanged between the client and the server through send/receive functions.

4. **Termination:** The connection is closed after the communication is completed.

**Conclusion:** Hence, we have Successfully Write a program using TCP socket for wired network for following a) Say Hello to Each other b)File transfer c)Calculator.