Department of Information Technology Engineering
National Institute of Technology Srinagar

Hazratbal, Srinagar, Jammu and Kashmir - 190006, India.

Computer Networks Project: Develop a network simulator implementing entire protocol stack

Group Members:

Akhilesh Kumar (2022BITE029)

Pavan Garg (2022BITE081)

Dev Karan (2022BITE036)

B.Tech. ITE (6th Semester)

Submitted to:

Dr. Iqra Altaf Gillani

Department of Information Technology Engineering National Institute of Technology Srinagar Hazratbal, Srinagar, Jammu and Kashmir - 190006, India.

Implement Physical layer functionalities.

• **Minimum deliverables:** Your simulator should be at least capable of

- Creating end devices, hubs
- Creating connections between them to form a topology
- Sending and receiving data

• **Test cases:** You can check your simulator's working on the following cases:

- Create two end devices with a dedicated connection and enable data transmission between them. Note that this is only a simulation as no real data can be shared.
- Create a star topology with five end devices connected to a hub and enable communication within end devices. This should follow exact working principles of a hub.

Objective:

The goal of this project is to visually simulate data communication between multiple devices connected to a central hub. The simulator demonstrates data transmission, acknowledgment (ACK), and device communication in a star topology.

🎨 **Features Implemented:**

1. Device Class

- **Attributes:** id, name, x, y (position), and methods for:
  - Drawing the device.
  - Sending and receiving data.
  - Sending and receiving ACKs with appropriate visual cues.
- **Visual Effects:**
  - Arrows to simulate data flow.
  - Acknowledgment (ACK) represented by a green tick.
  - Ignored data shown with a red cross.

2. Hub Class

- **Attributes:** connectedDevices (vector storing connected devices).
- **Methods:**
  - Connecting devices to the hub.
  - Receiving and broadcasting data.
  - Handling ACK communication between devices.

- **Hub Visualization:**
    - Rectangle hub with holes representing connection ports for devices.
    - Each connected device visually links to the hub.

3. Data Transmission Process

- **Sender sends data to the hub.**
    - The hub broadcasts the data to all connected devices.
    - The intended receiver receives the data and sends ACK back to the hub.
    - Hub forwards ACK to the sender.
- **Error Handling:**
    - Unintended devices ignore the data.

🖥️ **Simulation Details:**

- **Topology:** Star topology with 5 devices (A, B, C, D, E) connected to a central hub.
- **Visualization:**
    - Clear graphical interface with device labels and data arrows.
    - Device radius and hub holes are made larger for better visibility.
    - Messages and acknowledgments displayed using text-based labels.

🔁 **Test Case:**

1. **Simulation Initialization:**
    - Devices A, B, C, D, and E are drawn.
    - Hub is connected to all devices.
2. **Data Flow Example:**
    - Device A sends a message to Device D.
    - The hub receives and forwards the data.
    - Device D acknowledges the receipt by sending an ACK.
    - Device A receives the ACK.

🚩 **Challenges Addressed:**

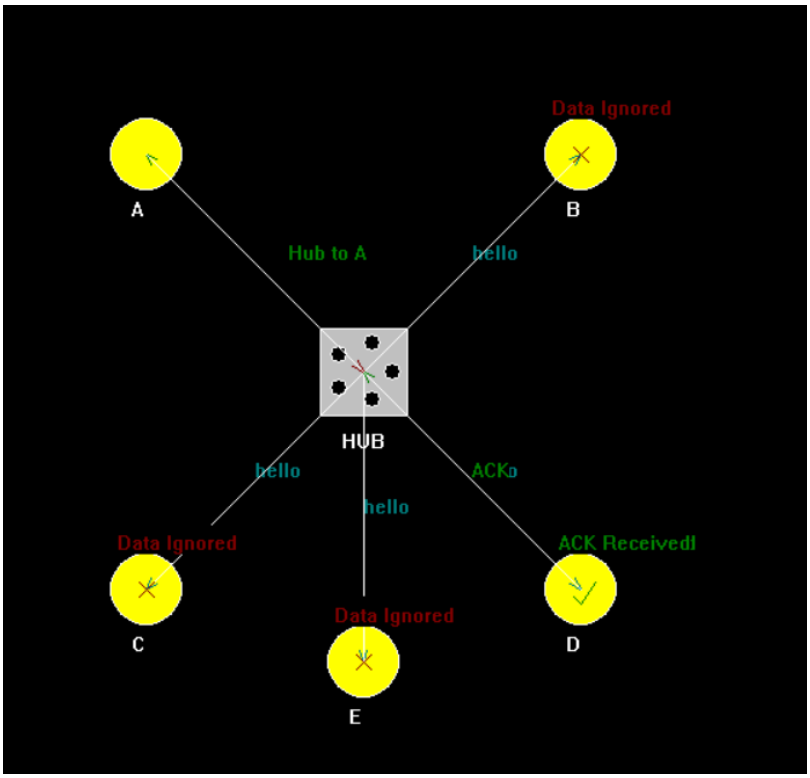
- Managing visual representation of data flow.
- Simulating proper ACK handling.

- Implementing ignore mechanism for unintended receivers.

📝 **Potential Enhancements:**

- Add error detection and retransmission.
- Enable multi-hop communication.
- Enhance visualization with real-time packet delays.

✅ **Status:** Successfully implemented star topology simulation with data and ACK flow between devices!




```
Enter sender device name (A/B/C/D/E): A
Enter receiver device name (A/B/C/D/E): D
Enter message: hello
A sent data to Hub: "hello"
B ignored the data.
C ignored the data.
D received data: "hello"
ACK sent from D to Hub.
ACK received by D from Hub
E ignored the data.
ACK sent from D to Hub.
ACK delivered to A from Hub.
```

Report on Enhanced Network Switch Simulation

🎯 **Objective:**

The project simulates the functionality of a Layer 2 (Data Link Layer) network switch, demonstrating the following key protocols:

- **Error Control Protocol:** To detect and discard corrupted frames using a checksum verification.
- **CSMA/CD (Carrier Sense Multiple Access with Collision Detection):** To handle data collisions and apply a backoff mechanism.
- **Sliding Window Protocol:** To efficiently manage flow control by transmitting multiple frames and ensuring successful delivery through acknowledgments.
- **Dynamic Frame Forwarding and MAC Learning:** To dynamically learn MAC addresses and forward frames based on MAC table entries.

The simulation also visualizes frame transmission between multiple devices connected to switches and hubs.

📚 **1. Overview of the Simulation:**

The simulation models the behavior of a switch in a network environment where multiple devices are connected. Each device communicates with others through the switch, which dynamically learns the MAC addresses of connected devices and forwards frames accordingly.

🎥 **2. Key Functionalities and Protocols:**

🔍 **2.1 Dynamic MAC Learning and Frame Forwarding:**

- **MAC Learning:** The switch learns the MAC address of each device that sends a frame and updates its MAC table.
- **Frame Forwarding:** When a frame arrives at the switch, it checks its MAC table to determine whether to unicast, broadcast, or discard the frame.
- **Unicast:** If the destination MAC is known, the frame is forwarded to the appropriate port.
- **Broadcast:** If the destination MAC is unknown, the frame is broadcasted to all ports except the source.

✅ **Outcome:**

- Efficient forwarding of frames between devices.
- Updated MAC table with learned addresses.

⚡ 2.2 Error Control: Checksum Verification

- **Checksum Calculation:** Each frame has a checksum generated based on the frame data.
- **Error Detection:** Upon receiving a frame, the switch recalculates the checksum and compares it to the transmitted checksum. If there's a mismatch, the frame is discarded.

✅ **Outcome:**

- Corrupted frames are detected and discarded.
- Reliable data transmission is ensured.

📡 2.3 CSMA/CD (Carrier Sense Multiple Access with Collision Detection)

- **Carrier Sensing:** The switch checks whether the channel is busy before sending a frame.
- **Collision Detection:** If a collision occurs, the switch implements a backoff mechanism and retries sending the frame after a random time interval.
- **Backoff Time:** A random delay is introduced to avoid continuous collisions.

✅ **Outcome:**

- Collisions are detected and managed effectively.
- Reduced data loss due to collisions.

🪟 2.4 Sliding Window Protocol: Flow Control

- **Window Mechanism:** Frames are sent in windows of a specified size to optimize data transfer.
- **Acknowledgment:** After sending a set of frames, the switch waits for acknowledgments (ACKs) before proceeding with the next set.
- **Frame Retransmission:** If an acknowledgment is not received for a frame, it is retransmitted.

✅ **Outcome:**

- Improved data flow efficiency.
- Reduced frame retransmissions and waiting times.

🔎 3. Test Cases and Execution:

📝 3.1 Test Case 1: Single Switch with 5 Devices

- **Scenario:** 5 devices are connected to a single switch, and frames are exchanged between them.

- **Protocol Demonstrated:**
    - MAC learning
    - Unicast and broadcast forwarding
    - Error control through checksum

✅ **Expected Outcome:**

- Frames are forwarded successfully.
- Correct frames are acknowledged, and corrupted frames are discarded.

📝 **3.2 Test Case 2: Error Control (Checksum Mismatch)**

- **Scenario:** A frame with incorrect checksum is transmitted.
- **Protocol Demonstrated:**
    - Error detection and frame discard.

❌ **Expected Outcome:**

- Frame with checksum mismatch is discarded.
- No acknowledgment is sent.

📝 **3.3 Test Case 3: CSMA/CD (Collision Detection)**

- **Scenario:** Two devices try to send data at the same time, causing a collision.
- **Protocol Demonstrated:**
    - Collision detection
    - Backoff and retransmission

✅ **Expected Outcome:**

- Collision is detected.
- Frame is retransmitted after a backoff delay.

📝 **3.4 Test Case 4: Sliding Window Protocol**

- **Scenario:** Multiple frames are sent with a window size of 3.
- **Protocol Demonstrated:**
    - Flow control
    - Frame retransmission on failure

✅ **Expected Outcome:**

- Frames within the window are transmitted and acknowledged.
- Frames with missing acknowledgments are retransmitted.

📊 **4. Topology Simulation:**

⭐ **4.1 Single Switch Topology**

- **Topology:** 5 devices connected to 1 switch.
- **Broadcast Domains:** 1
- **Collision Domains:** 5 (each port forms a collision domain)

✅ **Expected Outcome:**

- Communication occurs within a single broadcast domain.
- Collisions may occur if devices transmit simultaneously.

⭐ **4.2 Multi-Switch Topology with Hubs**

- **Topology:**
  - 2 hubs, each with 5 connected devices.
  - The hubs are connected to each other using a switch.
- **Broadcast Domains:** 2
- **Collision Domains:** 10 (each device connected to a hub shares a collision domain)

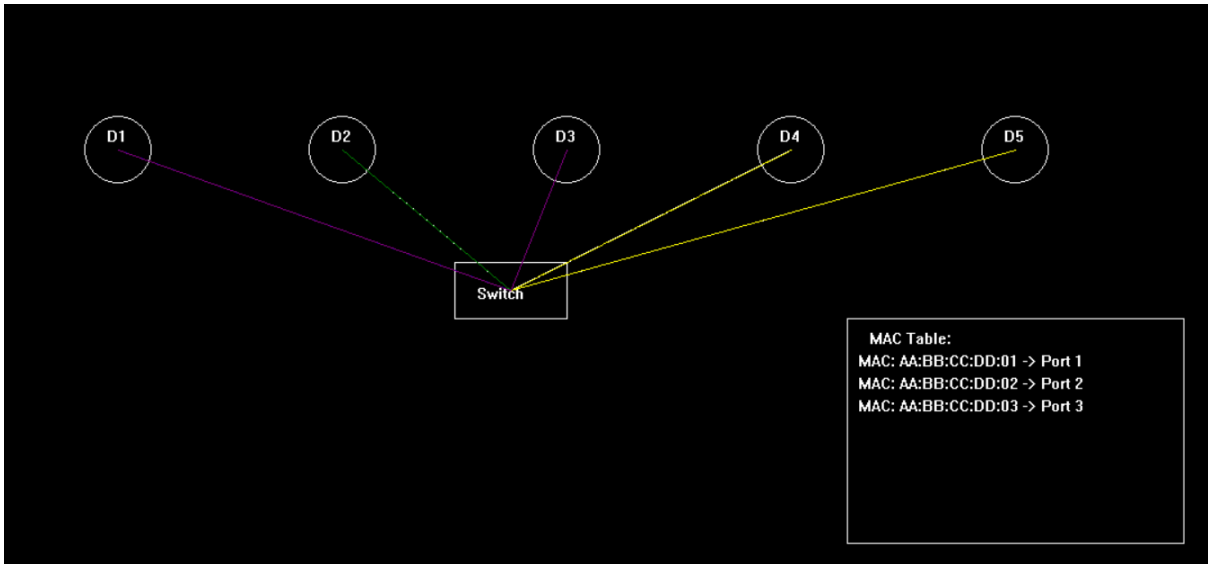
✅ **Expected Outcome:**

- Communication between devices connected to different hubs goes through the switch.
- Multiple collision and broadcast domains are created.

🎁 **5. Conclusion:**

The simulation successfully demonstrates the functionality of a Layer 2 switch with error control, collision management, and flow control. The tested scenarios confirm that:

✅ MAC learning and dynamic frame forwarding work as expected.
✅ Error detection ensures data integrity by discarding corrupted frames.
✅ CSMA/CD effectively handles collisions and minimizes data loss.
✅ Sliding window protocol improves flow control and ensures reliable frame delivery.




```
Enter the number of switches to connect: 1

Enter the number of devices for Switch 1: 5

Choose Mode for Switch 1:
1. Manual Simulation
2. Run Test Cases
2

Running Test Cases...

Test Case 1: Single Switch with 5 Devices

Sending Frame: Hello | Source: AA:BB:CC:DD:01 | Destination: AA:BB:CC:DD:04
Destination MAC unknown. Broadcasting frame.
Broadcasting to Port 2
Broadcasting to Port 3
Broadcasting to Port 4
Broadcasting to Port 5

Test Case 2: Error Control (Checksum Mismatch)

Sending Frame: Error | Source: AA:BB:CC:DD:02 | Destination: AA:BB:CC:DD:05
Error detected in frame. Frame discarded.

Test Case 3: CSMA/CD (Collision Detection)

Sending Frame: Retry | Source: AA:BB:CC:DD:03 | Destination: AA:BB:CC:DD:01
Unicasting frame to Port 1
Sending ACK from Port 1 to Port 3

Test Case 4: Sliding Window Protocol

Sliding Window [Window Size: 3]
Sending frame 1 | Data: Data0
```

```
Sending Frame: Data0 | Source: AA:BB:CC:DD:01 | Destination: AA:BB:CC:DD:02
Unicasting frame to Port 2
Sending ACK from Port 2 to Port 1
Sending frame 2 | Data: Data1

Sending Frame: Data1 | Source: AA:BB:CC:DD:01 | Destination: AA:BB:CC:DD:02
Unicasting frame to Port 2
Sending ACK from Port 2 to Port 1
Sending frame 3 | Data: Data2

Sending Frame: Data2 | Source: AA:BB:CC:DD:01 | Destination: AA:BB:CC:DD:02
Unicasting frame to Port 2
Sending ACK from Port 2 to Port 1
ACK received for frame 1.
Frame 2 lost. Retransmitting...

Sending Frame: Data1 | Source: AA:BB:CC:DD:01 | Destination: AA:BB:CC:DD:02
Unicasting frame to Port 2
Sending ACK from Port 2 to Port 1
ACK received for frame 3.

Sliding Window [Window Size: 3]
Sending frame 4 | Data: Data3

Sending Frame: Data3 | Source: AA:BB:CC:DD:01 | Destination: AA:BB:CC:DD:02
Unicasting frame to Port 2
Sending ACK from Port 2 to Port 1
Sending frame 5 | Data: Data4

Sending Frame: Data4 | Source: AA:BB:CC:DD:01 | Destination: AA:BB:CC:DD:02
Unicasting frame to Port 2
Sending ACK from Port 2 to Port 1
ACK received for frame 4.
ACK received for frame 5.

All test cases executed successfully!
```