

MODULE 2

DATA LINK LAYER – ERROR DETECTION AND CORRECTION

1.1 INTRODUCTION TO DATA LINK LAYER

Data Link Layer is the second layer of the OSI (Open Systems Interconnection) model. It is responsible for the reliable transmission of data across a physical network link. This layer ensures that the data transferred between two directly connected nodes is error-free, well-framed, and properly synchronized. The Data Link Layer takes the raw bits from the Physical Layer and organizes them into frames (structured units of data). It also provides error detection and correction, flow control, and link management to maintain smooth communication between devices.

- Is responsible for moving frames from node to node or computer to computer.
- Can move frames from one adjacent computer to another, cannot move frames across routers.
- Requires MAC address or physical address.
- Protocols defined include Ethernet Protocol and Point-to-Point Protocol (PPP).
- Device example: Switch
- Two sublayers: **Logical Link Control (LLC)** and the **Media Access Control (MAC)**

1. Logical Link Control (LLC) Sublayer

- **Full form:** Logical Link Control
- **Position:** Upper sublayer of the Data Link Layer
- **Main Function:** Handles communication between the Network Layer and the Data Link Layer.
- **Purpose:** Provides logical link establishment, maintenance, and termination.

- **Functions:**

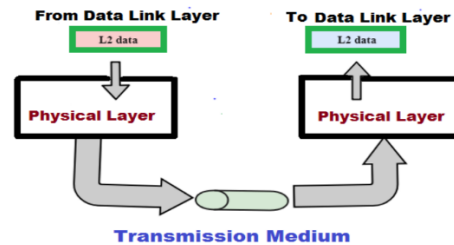
1. Identifies network layer protocols (like IP, IPX).
2. Provides **error detection** (not correction) and also provides an interface to network layer by protocols like **IP (Internet Protocol)** and **ARP (Automatic Repeat Protocol)**.
3. Controls **flow of data** between sender and receiver.
4. Manages **frame synchronization**.

- **Addressing:** Uses **Service Access Points (SAPs)** to identify the protocol type.

- **Example Protocol:** IEEE 802.2 (LLC protocol).

2. Media Access Control (MAC) Sublayer

- **Full form:** Media Access Control
- **Position:** Lower sublayer of the Data Link Layer
- **Main Function:** Controls how devices on the same network medium (like LAN) access the shared physical medium.
- **Purpose:** Regulates **when and how** data can be transmitted over the network.
- **Functions:**
 1. Provides hardware (MAC) addressing — every device has a unique MAC address.
 2. Determines who can use the medium at a given time (media access control).
 3. Defines frame format and frame transmission rules.
 4. Detects and sometimes helps in error handling during transmission.
- **Protocols:** Ethernet (IEEE 802.3), Wi-Fi (IEEE 802.11).



Data Link Layer

1.2 ERROR DETECTION AND ERROR CORRECTION

1.2.1 INTRODUCTION TO ERROR

In data communication, an error refers to any change or corruption that occurs in the transmitted data during its journey from the sender to the receiver.

Errors mainly happen because of noise, signal distortion, interference, or hardware faults in the transmission medium.

As a result, the received data may not match the data that was originally sent, leading to incorrect or incomplete information. Detecting and correcting these errors is an essential part of reliable communication.

There are two main types of errors: **single-bit errors** and **burst errors**. A single-bit error affects only one bit of data, while a burst error affects a group of bits. To ensure accuracy, the Data Link Layer of the OSI model uses various error detection and correction techniques such as parity check, checksum, and cyclic redundancy check (CRC). These methods help identify and sometimes correct the errors, maintaining the integrity and reliability of data transmission in a network.

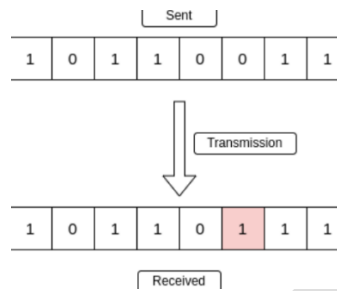
1.2.2 TYPES OF ERRORS

There are two main types of errors:

1. **Single-bit Errors**
2. **Burst Errors**

1. Single-bit Errors

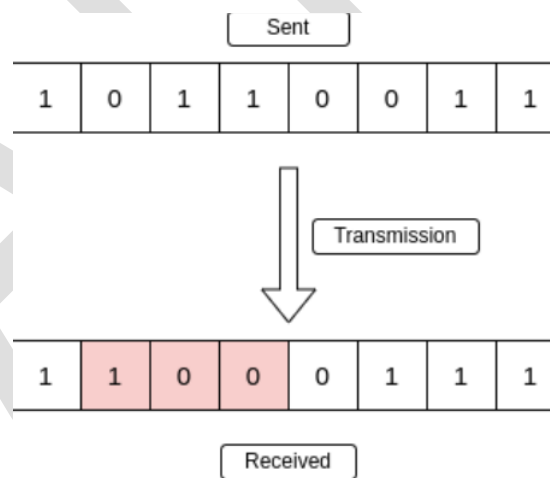
Whenever bits flow from one point to another, they are subject to unpredictable errors. The term single-bit error means that only 1 bit of a given data unit (such as a byte character, or packet) is changed from 1 to 0 or from 0 to 1.



Single Bit Errors

2. Burst Bit Errors

The term burst error that 2 or more bits in the data unit have changed from 1 to 0 or from 0 to 1. A burst error is more likely to occur than a single-bit error because the duration of the noise signal is normally longer than the duration of 1 bit, which means that when noise affects data, it affects a set of bits. The number of bits affected depends on the data rate and duration of noise. For example, if we are sending data at 1 kbps, a noise of 1/100 second can affect 10 bits; if we are sending data at 1 Mbps, the same noise can affect 10,000 bits.



Burst Bit Error

1.3 ERROR DETECTION

How can errors be detected by using block coding? If the following two conditions are met, the receiver can detect a change in the original codeword.

1. The receiver has (or can find) a list of valid codewords.

2. The original codeword has changed to an invalid one.

The role of block coding in error detection. The sender creates code words out of datawords by using a generator that applies the rules and procedures of encoding (discussed later). Each codeword sent to the receiver may change during transmission. If the received codeword is the same as one of the valid codewords, the word is accepted; the corresponding dataword is extracted for use. If the received code-word is not valid, it is discarded. However, if the codeword is corrupted during transmission but the received word still matches a valid codeword, the error remains undetected. One of the most commonly used techniques for error detection is **Block Coding**.

1.3.1 BLOCK CODING

One of the most commonly used techniques for error detection is **Block Coding**. In block coding, we divide our message into blocks, each of m bits, called **datawords**. We add r redundant bits to each block to make the length $n = m + r$.

The resulting n -bit blocks are called **codewords**. How the extra r bits are chosen or calculated is something we will discuss later. For the moment, it is important to know that we have a set of datawords, each of size m , and a set of codewords, each of size n .

With m bits, we can create a combination of 2^m datawords with n bits, we can create a combination of 2^n codewords. Since $n > m$, the number of possible codewords is larger than the number of possible datawords.

The block coding process is one-to-one; the same data-word is always encoded as the same codeword. This means that we have $2^n - 2^m$ codewords that are not used. We call these codewords invalid or illegal.

The trick in error detection is the existence of these invalid codes, as we discuss next. If the receiver receives an invalid codeword, this indicates that the data was corrupted during transmission.

The main three steps for block coding they are

1. Division
2. Substitution
3. Combination

1. Division

The original data stream is **divided into fixed-size blocks** of m bits.

Each block is treated separately for coding.

This makes transmission easier to manage and helps identify where errors occur.

Example: If data = 110101010111, and block size = 4 bits → Blocks are divided as: 1101, 0101, 0111.

2. Substitution

Each k -bit block is then **replaced (substituted)** with a unique n -bit codeword.

The extra bits added are called **redundant bits**, used for **error detection** ($n = m + r$).

A **codebook or lookup table** defines which codeword represents which data block.

Example: If data = 110101010111, and block size = 4 bits → 11010, 01011, 01110
(Adding Redundant bits).

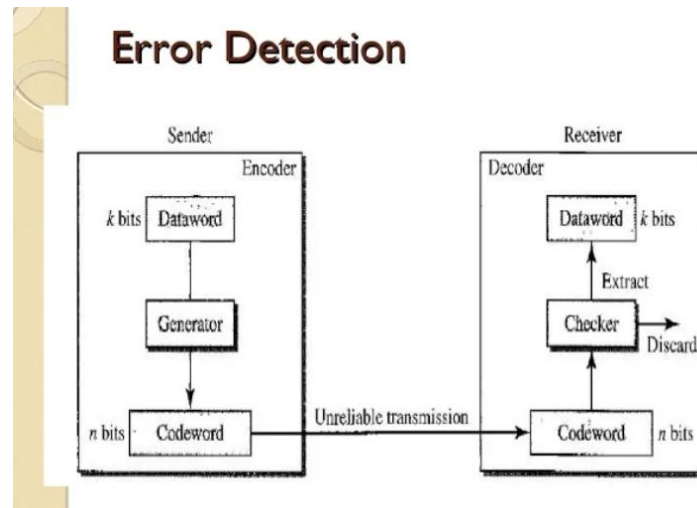
3. Combination

After substitution, all the **n -bit codewords** are **combined** together to form the **final encoded bit stream**.

This complete bit stream is then transmitted through the communication channel.

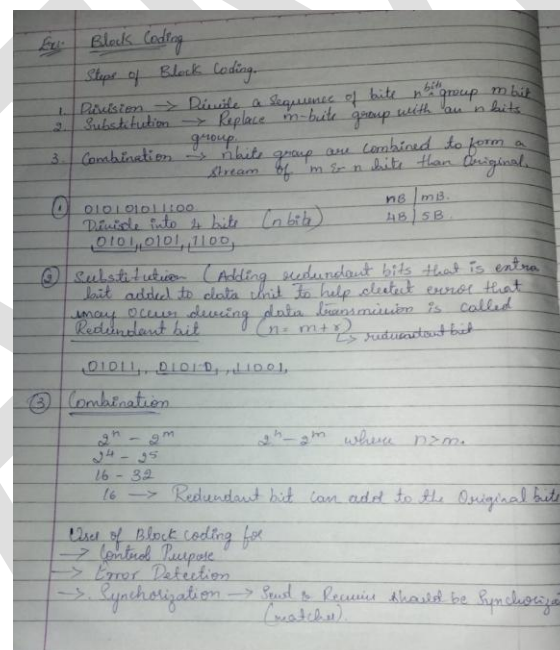
$2^n - 2^m$ where $n > m$ bits.

Error Detection



Error Detection and Block Coding (same Diagram for both)

Example:



Example for block coding

2.DATA LINK CONTROL(DLC)

2.1 DLC SERVICES

The data link control (DLC) deals with procedures for communication between two adjacent nodes-node-to-node communication-no matter whether the link is dedicated or broadcast. Data link control functions include framing and flow and error control. In this section, we first discuss framing, or how to organize the bits that are carried by the physical layer. We then discuss flow and error control.

The main services of DLC are:

- 1. Framing**
- 2. Flow Control**
- 3. Error Control**
- 4. Connectionless and Connection Oriented Services**
- 5. Data Link Layer Protocols**
- 6. High-Level Data Link Control (HDLC)**

1. Framing

☐ **Physical Layer:**

- Responsible for moving bits in the form of signals from the source to the destination.
- Provides bit synchronization to ensure proper timing during transmission.

☐ **Data-Link Layer:**

- Packs bits into frames so that each frame can be distinguished from another.
- Similar to the postal system where inserting a letter into an envelope separates one message from another.
- The envelope acts as a delimiter and carries the sender and receiver addresses.
- This is important because the postal system, like the network, supports many-to-many communication.

□ Framing in the Data-Link Layer:

- Separates messages from one source to the intended destination.
- Adds a sender address and a destination address:
 - The destination address indicates where the packet should go.
 - The sender address helps the receiver acknowledge the receipt.

□ Why Not Use One Large Frame?

- Although the entire message could be placed in one large frame, it is not efficient.
- Large frames make flow and error control difficult.
- If a single-bit error occurs in a large frame, the entire frame must be retransmitted.
- Dividing a message into smaller frames ensures that:
 - Errors affect only a small portion of data.
 - Retransmission is limited to the damaged frame only.

Frame Size

Frames can be of fixed or variable size. In fixed-size framing, there is no need for defining the boundaries of the frames, the size itself can be used as a delimiter. An example of this type of framing is the ATM WAN, which uses frames of fixed size called **cells**.

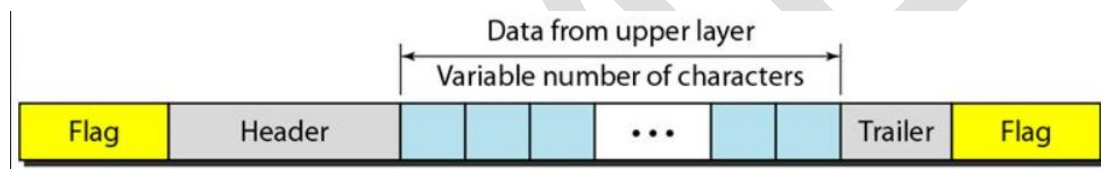
Our main discussion in this chapter concerns variable-size framing, prevalent in local-area networks. In variable-size framing, we need a way to define the end of one frame and the beginning of the next. Historically, two approaches were used for this purpose:

- 1. character-oriented methods or approach**
- 2. Byte Stuffing and Unstuffing**
- 3. bit-oriented method or approach**

1. character-oriented methods or approach

- In character-oriented (or byte-oriented) framing, the data to be carried are 8-bit characters from a coding system such as ASCII (see Appendix A).

- ☐ The header normally carries:
 - The source address,
 - The destination address, and
 - Other control information.
- ☐ The trailer carries error detection redundant bits.
- ☐ Both the header and trailer are multiples of 8 bits.
- ☐ To separate one frame from the next, an 8-bit (1-byte) flag is added at the:
 - Beginning of a frame, and
 - End of a frame.
- ☐ The flag is composed of protocol-dependent special characters.
- ☐ The flag signals the start or end of a frame.
- ☐ Figure shows the format of a frame in a character-oriented protocol.

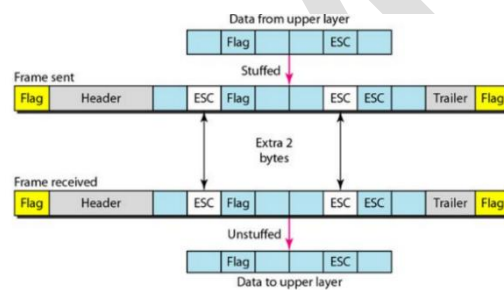


A Frame in a Character Oriented Framing

- ☐ Character-oriented framing was popular when only text was exchanged by the data-link layers.
- ☐ The flag could be chosen as any character not used for text communication.
- ☐ Now, we transmit different types of data such as:
 - Graphs
 - Audio
 - Video
- ☐ In such cases, any character used as a flag could also appear in the actual data.
- ☐ If this happens, the receiver may mistakenly think it has reached the end of the frame when it encounters the flag pattern inside the data.
- ☐ To solve this problem, a byte-stuffing strategy was introduced in character-oriented framing.
- ☐ In byte stuffing (or character stuffing):

- A special byte is added to the data section of the frame when there is a character with the same pattern as the flag.
 - The data section is stuffed with an extra byte to avoid confusion.
- This special byte is called the Escape character (ESC) and has a predefined bit pattern.
- When the receiver encounters the ESC character, it:
- Removes it from the data section, and
 - Treats the next character as data, not as a flag.

2. Byte Stuffing and Unstuffing



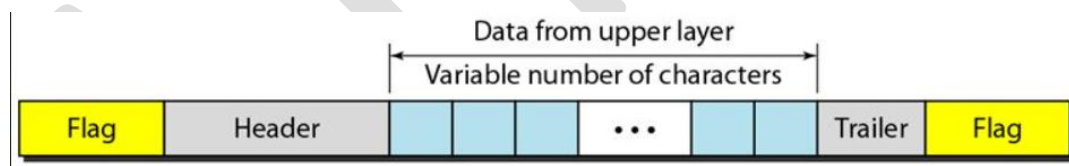
Byte Stuffing and Unstuffing

- Byte stuffing is the process of adding one extra byte whenever there is a flag or escape character in the text.
- Byte stuffing using the escape character (ESC) allows the flag to appear inside the data section of the frame without confusion.
- However, this method creates another problem:
- What happens if the text contains one or more escape characters followed by a byte with the same pattern as the flag?
- In this case:
- The receiver removes the escape character,
 - But keeps the next byte, which may be incorrectly treated as part of the text.
- To solve this issue:
- If the escape character itself is part of the text,
 - An extra escape character is added to mark that the second one is part of the text.

- The universal coding systems used today, such as Unicode, have 16-bit and 32-bit characters.
- Character-oriented protocols create another problem in data communications because they conflict with 8-bit character systems.
- Therefore, the tendency in modern communication is moving toward bit-oriented protocols, which will be discussed next.

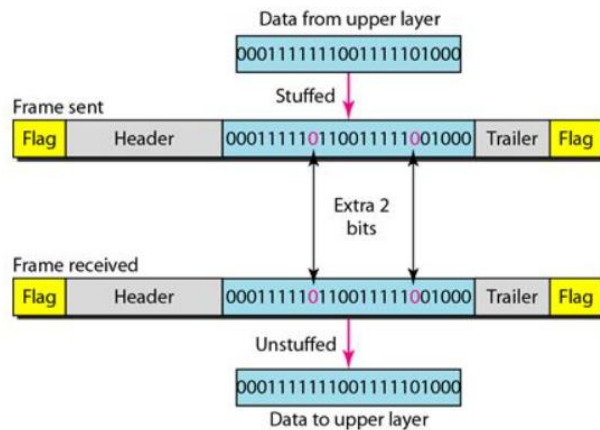
3. Bit-Oriented Framing

- In bit-oriented framing, the data section of a frame is a sequence of bits.
- These bits can be interpreted by the upper layer as:
 - Text
 - Graphics
 - Audio
 - Video, and so on.
- Besides the header (and possibly a trailer), a delimiter is required to separate one frame from another.
- Most protocols use a special 8-bit pattern flag — 01111110 — as the delimiter.
- This flag marks the beginning and the end of each frame.



- The flag used in bit-oriented framing can create the same type of problem seen in character-oriented protocols.
- If the flag pattern appears inside the data, the receiver might mistake it for the end of the frame.
- To avoid this confusion, a technique called bit stuffing is used.
- In bit stuffing:
 - Whenever a 0 followed by five consecutive 1s (11111) is encountered in the data,
 - An extra 0 bit is inserted (stuffed) into the data stream.

- This extra stuffed bit is later removed by the receiver when the data is received.
- The extra 0 is added after one 0 and five 1s, no matter what the next bit is.
- This ensures that the flag pattern (01111110) never appears accidentally inside the data.

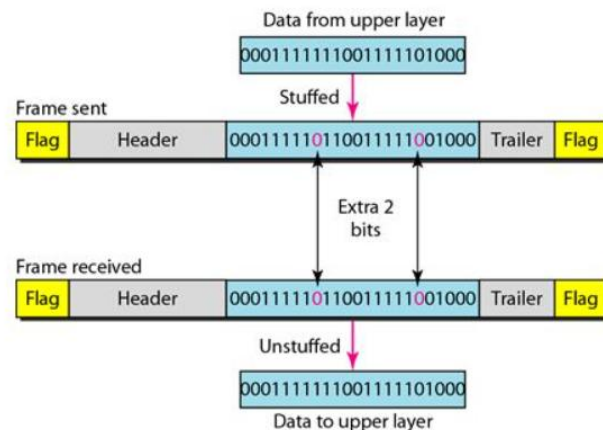


- ☐ Bit stuffing is the process of adding one extra 0 whenever five consecutive 1s follow a 0 in the data.
- ☐ This ensures that the receiver does not mistake the pattern 01111110 (the flag) as part of the data.
- ☐ Bit stuffing is done at the sender, and bit removal is done at the receiver.
- Even if there is a 0 after five 1s, the sender still stuffs an extra 0.
- ☐ The receiver removes this extra stuffed 0 when reading the data.
- ☐ **Example:**
- If the flag-like pattern 01111110 appears in the data, it becomes 011111010 after bit stuffing.
- This prevents it from being mistaken as a real flag.
 - ☐ The actual flag (01111110) is not stuffed by the sender and is correctly recognized by the receiver.

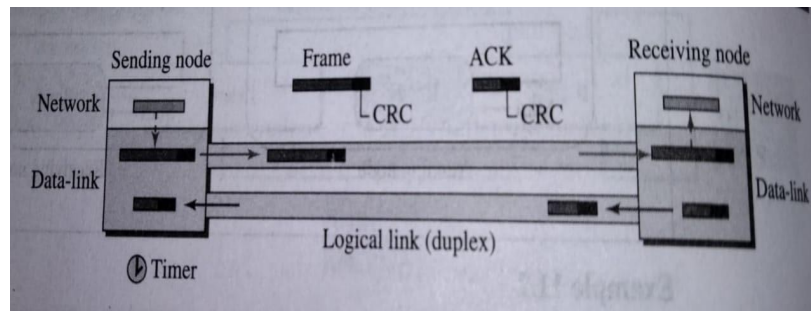
2. FLOW CONTROL

- ☐ Whenever one entity produces items and another entity consumes them, there must be a balance between production and consumption rates.
- ☐ If items are produced faster than they can be consumed:
 - The consumer may become overwhelmed.
 - Some items may need to be discarded, leading to data loss.
- ☐ If items are produced more slowly than they can be consumed:
 - The consumer must wait, causing the system to become less efficient.

- Flow control deals with maintaining this balance between sender (producer) and receiver (consumer).
- The main goal of flow control is to prevent data loss and ensure smooth data transmission at the receiver site.

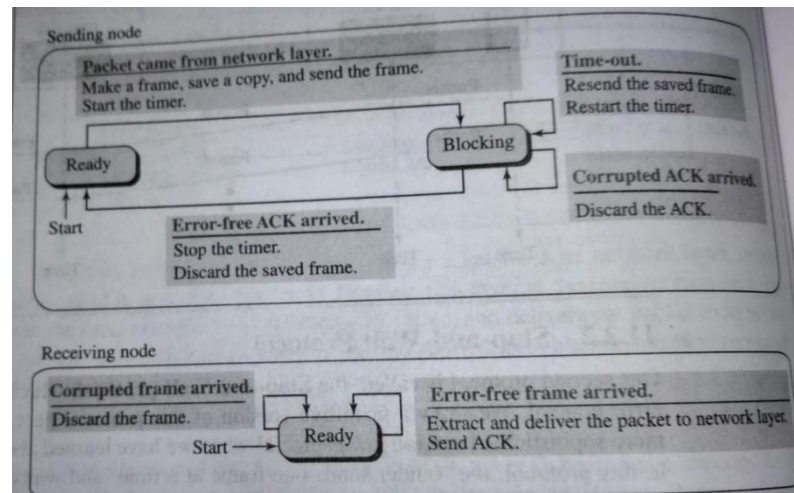


- In communication at the data-link layer, there are four entities involved:
 1. Network layer at the sending node
 2. Data-link layer at the sending node
 3. Network layer at the receiving node
 4. Data-link layer at the receiving node
 - In real systems, there can be complex relationships involving multiple producers and consumers (as discussed in Chapter 23).
 - However, for simplicity, we ignore the interactions between the network and data-link layers.
 - Instead, we focus only on the relationship between the two data-link layers — one at the sender and the other at the receiver.
- **Techniques in flow control**
 1. **Stop-and-Wait** (send one frame, wait for acknowledgment before sending next)
 2. **Sliding Window** (multiple frames sent before acknowledgment, more efficient)
1. **Stop – and – Wait Protocol**



Stop and Wait Protocol

- Stop-and-Wait protocol provides both flow control and error control.
- The sender sends one frame at a time and waits for an acknowledgment before sending the next frame.
- Each data frame includes a CRC for error detection.
- If the CRC is incorrect, the frame is considered corrupted and is silently discarded by the receiver.
- The receiver's silence indicates that a frame was either lost or corrupted.
- The sender starts a timer after sending a frame.
- If an acknowledgment is received before the timer expires, the timer is stopped, and the next frame is sent.
- If the timer expires before receiving an acknowledgment, the sender resends the previous frame.
- The sender keeps a copy of the frame until its acknowledgment arrives.
- Once the acknowledgment is received, the sender discards the frame copy and sends the next one.
- Only one frame and one acknowledgment can be in the channel at any time.



FSM Primitive Stop and Wait Protocol

Sender States

The sender is initially in the ready state, but it can move between the ready and blocking state.

Ready States

When the sender is in this state, it is only waiting for a packet from the network layer. If a packet comes from the network layer, the sender creates a frame, saves a copy of the frame, starts the only timer and sends the frame. The sender then moves to the blocking state.

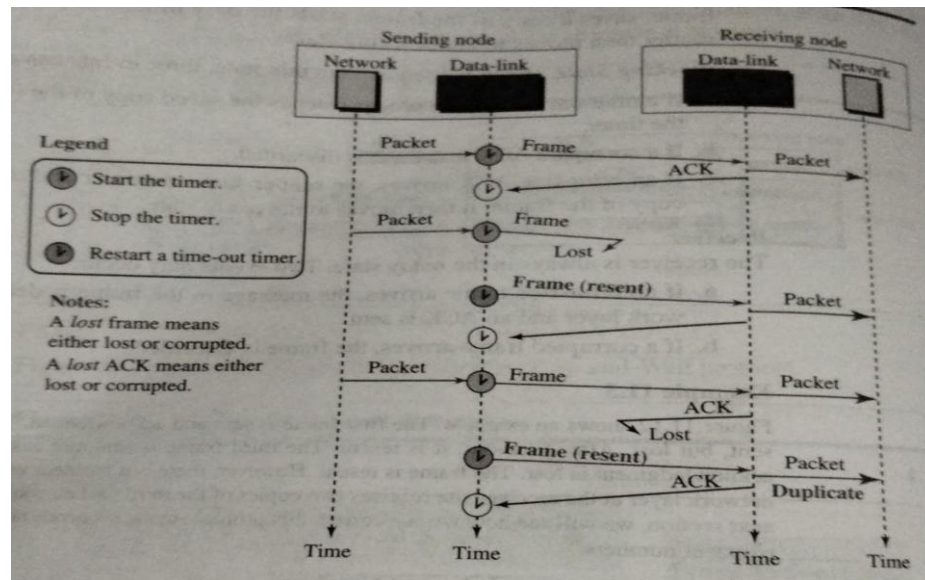
Blocking State

When the sender is in this state, three events can occur.

- If a time-out occurs, the sender resends the saved copy of the frame and restarts the timer.
- If a corrupted ACK arrives, it is discarded.
- If an error-free ACK arrives, the sender stops the timer and discards the saved copy of the frame. It then moves to the ready state.

• Receiver

- The receiver is always in the ready state. Two events may occur:
 - If an error-free frame arrives, the message in the frame is delivered to the network layer and an ACK is sent
 - If a corrupted frame arrives, the frame is discarded.



Flow Diagram for Stop and Wait

- The first frame is sent successfully and acknowledged by the receiver.
- The second frame is sent but gets lost during transmission.
- After a time-out, the sender resends the second frame.
- The third frame is sent and acknowledged successfully, but the acknowledgment is lost.
- Due to the lost acknowledgment, the sender resends the third frame.
- The receiver receives the same third frame twice, resulting in duplicate data being delivered to the network layer.
- This duplication is an error in the Stop-and-Wait scheme.
- To fix this issue, sequence numbers and acknowledgment numbers are introduced in the next section.

Sequence and Acknowledgment Numbers

- Duplicate packets can cause serious problems, such as repeating actions (e.g., ordering the same item twice).
- To prevent duplicates, sequence numbers are added to **data frames** and acknowledgment numbers to **ACK frames**.

- Sequence numbers help the receiver identify new frames and detect duplicates.
- In the Stop-and-Wait protocol, numbering is simple and alternates between **0 and 1** (i.e., 0, 1, 0, 1, ...).
- Acknowledgment numbers also alternate as **1, 0, 1, 0, ...** corresponding to the next expected frame.
- The acknowledgment number indicates the **sequence number of the next frame** the receiver expects.
- **Automatic Repeat Request** mechanism ensures reliable and duplicate-free data transmission between sender and receiver.

2. Sliding Window

Two Techniques in Sliding Window

1. Go-Back-N (GBN) Protocol

2. Selective Repeat

1. Go-Back-N (GBN) Protocol

Introduction

- Go-Back-N (GBN) is a sliding window protocol used in the data-link layer for reliable data transmission.
- It allows the sender to transmit multiple frames before needing an acknowledgment from the receiver.
- The receiver, however, is simpler: it only needs to acknowledge the last correctly received frame in order.

Key Features

1. Sliding Window at Sender

- The sender can send N frames without waiting for an acknowledgment.
- N = window size, a fixed number of frames.

2. Receiver Behaviour

- The receiver accepts frames only in order.

- If a frame is missing or erroneous, all subsequent frames are discarded (even if received correctly).

3. Acknowledgment (ACK)

- Receiver sends an ACK for the last correctly received frame in sequence.
- The sender goes back and retransmits all frames starting from the missing/erroneous frame.

4. Error Handling

- Errors can be lost frames, corrupted frames, or lost acknowledgments.
- GBN handles this by retransmitting all frames from the lost frame onward.

Working Principle

1. Sender can send frames 0, 1, 2, ..., N-1 within the window.
2. Receiver checks the sequence number of incoming frames:
 - If the frame is in order, it is accepted, and an ACK is sent.
 - If the frame is out of order, it is discarded, and no ACK is sent for that frame.
3. If the sender times out (doesn't receive ACK), it retransmits all frames from the last acknowledged frame.

Advantages

- Efficient for low error rates because multiple frames are sent before waiting for ACK.
- Simple receiver design.

Disadvantages

- If there is an error, all subsequent frames are retransmitted, even if they were received correctly.
- Can be inefficient for high-error networks.

- **Example**

- **Sender:** [0] [1] [2] [3] [4]

↓ ↓ ↓

- **Receiver:** [0] [1] X [3] [4]

- **Frame 2 lost → Receiver discards 3, 4**

- → ACK (1) sent last
- → Sender retransmits from frame 2 onward

2. Selective Repeat Protocol

Introduction

- Selective Repeat (SR) is a sliding window protocol used in the data-link layer for reliable data transmission.
- Unlike Go-Back-N (GBN), SR retransmits only the specific frames that are lost or erroneous, not all subsequent frames.
- This makes SR more efficient, especially in high-error networks.

□ Sliding Window at Sender and Receiver

- Both sender and receiver maintain a window of size N (number of frames that can be sent or received without waiting for acknowledgment).

□ Receiver Behaviour

- The receiver accepts frames that arrive out of order.
- It buffers out-of-order frames until missing frames are received.
- Only when the missing frames arrive, it delivers frames to the upper layer in order.

□ Acknowledgment (ACK)

- Receiver sends an individual ACK for each correctly received frame.
- ACK contains the sequence number of the frame being acknowledged.

□ Error Handling

- If a frame is lost or corrupted, only that frame is retransmitted.
- No need to retransmit subsequent frames, unlike GBN.

Working Principle

1. Sender can transmit frames 0, 1, 2, ..., N-1 within its window.

2. Receiver accepts frames in or out of order:
 - If a frame is in order, deliver it to the upper layer and send ACK.
 - If a frame is out of order, buffer it and send ACK.
3. Retransmission occurs only for frames that are not acknowledged within a timeout period.

Advantages

- More efficient than Go-Back-N for networks with high error rates.
- Minimizes retransmissions — only erroneous or lost frames are resent.
- Receiver can accept out-of-order frames, improving throughput.

Disadvantages

- Complex receiver design because it needs buffering for out-of-order frames.
- More complex sequence number management compared to GBN.

Example :

Sender: [0] [1] [2] [3] [4]

↓ ↓ ↓ ↓

Receiver: [0] [1] X [3] [4]

(stores 3,4)

→ Sender retransmits only frame 2

→ Receiver now has 2,3,4 and delivers in order.

3. ERROR CONTROL

- The physical layer of a network may not always be reliable, so data packets can get corrupted.
- To prevent corrupted packets from reaching the network layer, error control is implemented at the data-link layer.
- This is usually done using **two main methods**, both involving a CRC (Cyclic Redundancy Check) added to the frame header.

Method 1: If a frame is corrupted, it is simply discarded. If it is correct, it is delivered to the network layer. This method is commonly used in wired LANs like Ethernet.

Method 2: If a frame is corrupted, it is discarded, but acknowledgments are also used to manage both **flow control** and **error control** between sender and receiver.

Combination of Flow and Error Control

- Flow control and error control can work together in a network.
- The acknowledgment (ACK) sent for flow control can also indicate that a packet was received without errors.
- If the sender does not receive an acknowledgment, it means there was a problem with the sent frame.
- Frames carrying such acknowledgments are called **ACK frames** to distinguish them from regular data frames.
- This concept is applied in many simple network protocols.

4. CONNECTIONLESS AND CONNECTION-ORIENTED SERVICES

- DLC protocols can be **Connectionless** or **Connection-Oriented**.
- More details are covered in **Network and Transport Layers**.

1. Connectionless Protocol

- Frames are sent **independently**, with **no relationship** between them.

- “Connectionless” \neq no physical connection the transmission medium still exists.
- Common in **LAN data-link protocols**.
- **Example: Ethernet (IEEE 802.3)**
- Frames are sent independently, no numbering, no guaranteed order.
- Common in wired LANs.

2.Connection-Oriented Protocol

- **Logical connection** is established before data transfer (Setup Phase).
- Related frames are transmitted in order (Transfer Phase).
- Connection is terminated after transfer (Teardown Phase).
- Frames are **numbered**; receiver ensures correct **order** before delivering to network layer.
- Rare in wired LANs; found in some **point-to-point protocols, wireless LANs, and WANs**.
- Establishes a logical connection before sending frames.
- Used in some Wireless WANs.

5. DATA LINK LAYER PROTOCOLS

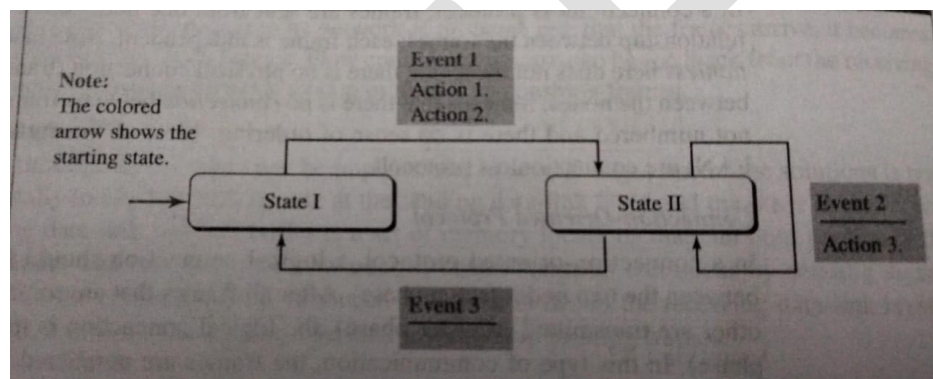
- The four protocols have been defined for the data-Link Layer to deal with Flow and Error Control.
- **Four traditional protocols:**
 1. **Simple**
 2. **Stop-and-Wait**
 3. **Go-Back-N**
 4. **Selective-Repeat**
- **Simple and Stop-and-Wait** are still used at the data-link layer.
- **Go-Back-N and Selective-Repeat** are mostly used at the transport layer.
- **Finite State Machine (FSM) Concept:**

- FSM models the behavior of a data-link-layer protocol.
- **States:** The machine has a **finite number of states (limited or countable;)** always in one state at a time.

(The system can move between these states based on events, but it **cannot have unlimited states.**)

- A **state** represents the **current condition or situation** of the system.
- It defines **what the system is doing or waiting for** at a specific time.
- The system **stays in a state** until an **event** occurs that causes it to change.
- **Example:**
- “Waiting for frame to send”
- “Waiting for acknowledgment”
- **Initial State:** The **initial state** is the **starting point** of the FSM when the system is first turned on or begins operation.
- It is the **default state** before any event occurs.
- From this state, the machine reacts to the first event and may transition to another state.
- **Example:**
- When the system starts, it is in the “Idle” or “Waiting to send” state.
- **Events:** Defining the list of action to performed and determine the next state.
- **Actions:** Tasks performed when an event occurs.
- **Event:**
- An **event** is something that **happens or occurs** that causes a **change** in the system’s state.
- It acts as a **trigger** for the machine to perform an action or move to another state.
- **Example:** A frame is received, timer expires, or acknowledgment arrives. (through the IP address)

- **Action:**
- An action is the response or task performed by the machine when an event occurs.
- It defines what the system does as a result of the event.
- **Example:** Send a frame, start a timer, or discard a frame.
- **Event = What happens**
- **Action =** What the system does in response
- **Trigger =** A trigger is something that causes or starts an action or event. It's the reason why a system changes from one state to another.



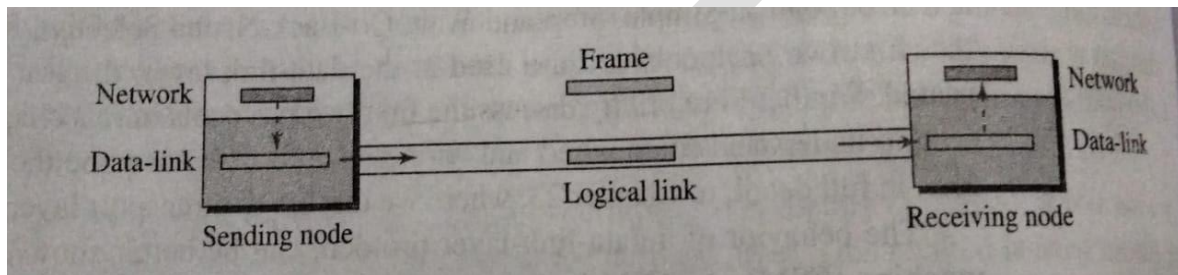
Data Link Protocol

Example (FSM with 3 States):

- Starts at **State I**.
- **Event 1:** Perform actions 1 & 2 → move to **State II**.
- **State II:**
 - Event 1 → perform action 3 → stay in State II.
 - Event 3 → no action → move back to **State I**.
 - The figure shows a machine with three states.
 - There are only three possible events and three possible actions.

- The machine starts in state I. If event 1 occurs, the machine performs actions 1 and 2 and moves to state II.
- When the machine is in state II, two events may occur. If event 1 occurs, the machine performs action 3 and remains in the same state, state II.
- If event 3 occurs, the machine performs no action, but move to state I.

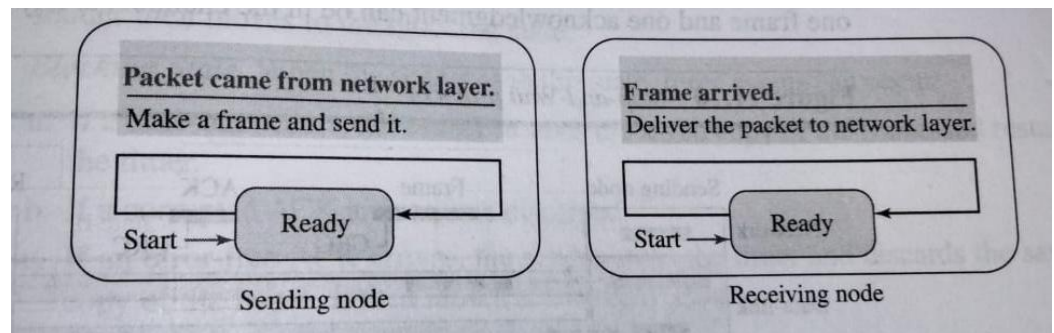
1. SIMPLE PROTOCOL



Simple Protocol

- The first data-link layer protocol has no flow control and no error control.
- Assumes the receiver can handle all incoming frames immediately.
- The receiver is never Overloaded by data.
- Frames are sent continuously without waiting for acknowledgment.
- The data-link layer at the sender gets a packet from its network layer, makes a frame out of it, and sends the frame.
- The data-link layer at the receiver receives a frame from the link, extracts the packet from the frame, and delivers the packet to its network layer.
- The data-link layers of the sender and receiver provide transmission services for their network layers.

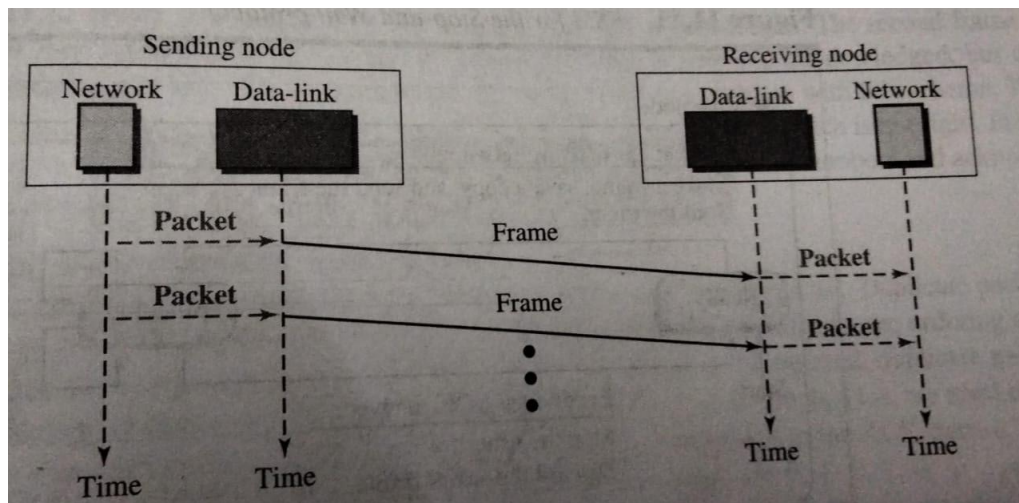
FSMs for the Simple Protocol



Finite State Machines (FSMs) for Simple Protocol

- The sender site should not send a frame until its network layer has a message to send.
- The receiver site cannot deliver a message to its network layer until a frame arrives.
- Each FSM has only one state, the ready state.
- The sending machine remains in the ready state until a request comes from the process in the network layer.
- When this event occurs, the sending machine **encapsulates** the message in a frame and sends it to the receiving machine.
- The receiving machine remains in the ready state until a frame arrives from the sending machine.
- When this event occurs, the receiving machine **decapsulates** the message out of the frame and delivers it to the process at the network layer.

Flow Diagram



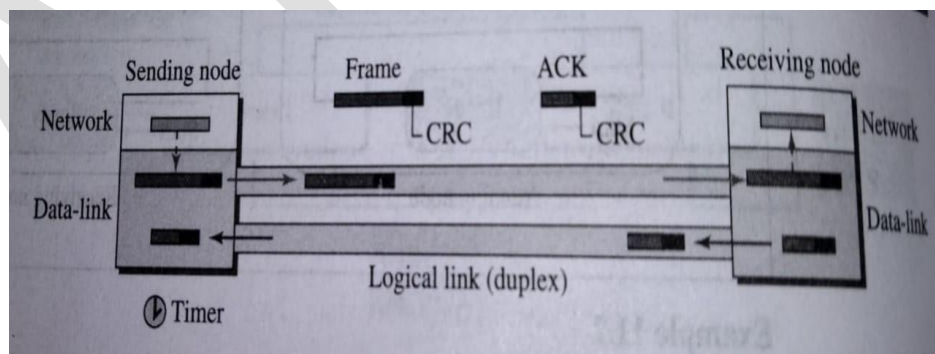
Flow of Data (Left to Right)

- Network → Data-Link (Sender side):
- The Network layer at the sender gives a packet to the Data-Link layer.
- The Data-Link layer encapsulates the packet into a frame.
- Data-Link (Sender) → Data-Link (Receiver):
- The frame is transmitted across the physical link to the receiver's Data-Link layer
- Data-Link → Network (Receiver side):
- The Data-Link layer at the receiver receives the frame.
- It extracts (decapsulates) the packet from the frame.
- Then it delivers the packet to its Network layer.
- **Key Points**
- There is **no flow control** or **error control** in this simple protocol.
- The **receiver can always handle incoming frames immediately**.
- Each **frame** carries one **packet** from sender to receiver.

- This diagram shows the **basic service** provided by the **Data-Link layer** — transferring packets between two directly connected nodes.

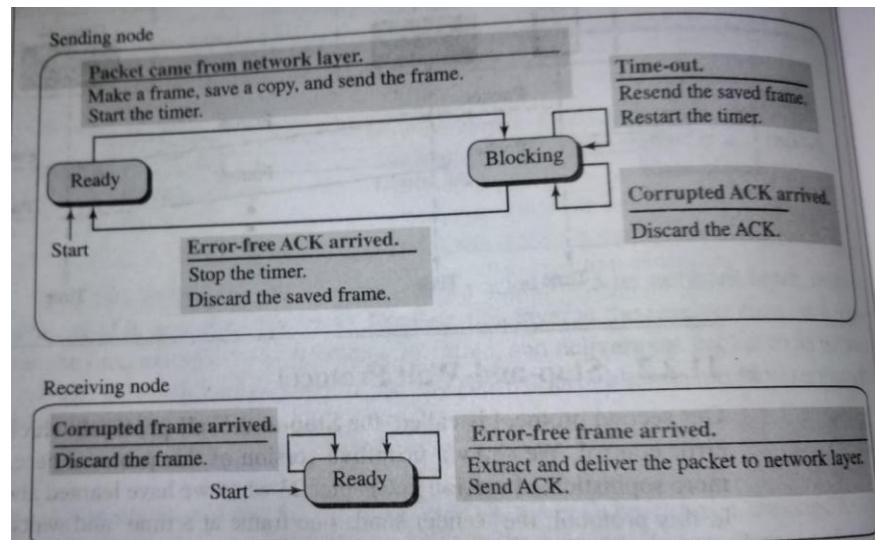
2 STOP – AND – WAIT PROTOCOL

- Stop-and-Wait protocol provides both flow control and error control.
- The sender sends one frame at a time and waits for an acknowledgment before sending the next frame.
- Each data frame includes a CRC for error detection.
- If the CRC is incorrect, the frame is considered corrupted and is silently discarded by the receiver.
- The receiver's silence indicates that a frame was either lost or corrupted.
- The sender starts a timer after sending a frame.
- If an acknowledgment is received before the timer expires, the timer is stopped, and the next frame is sent.
- If the timer expires before receiving an acknowledgment, the sender resends the previous frame.
- The sender keeps a copy of the frame until its acknowledgment arrives.
- Once the acknowledgment is received, the sender discards the frame copy and sends the next one.
- Only one frame and one acknowledgment can be in the channel at any time.



Stop and Wait Protocol

FSM Primitive Stop and Wait Protocol



- **Sender States**

1. The sender is initially in the ready state, but it can move between the ready and blocking state.
2. Ready States
3. When the sender is in this state, it is only waiting for a packet from the network layer.
4. If a packet comes from the network layer, the sender creates a frame, saves a copy of the frame, starts the only timer and sends the frame. The sender then moves to the blocking state.

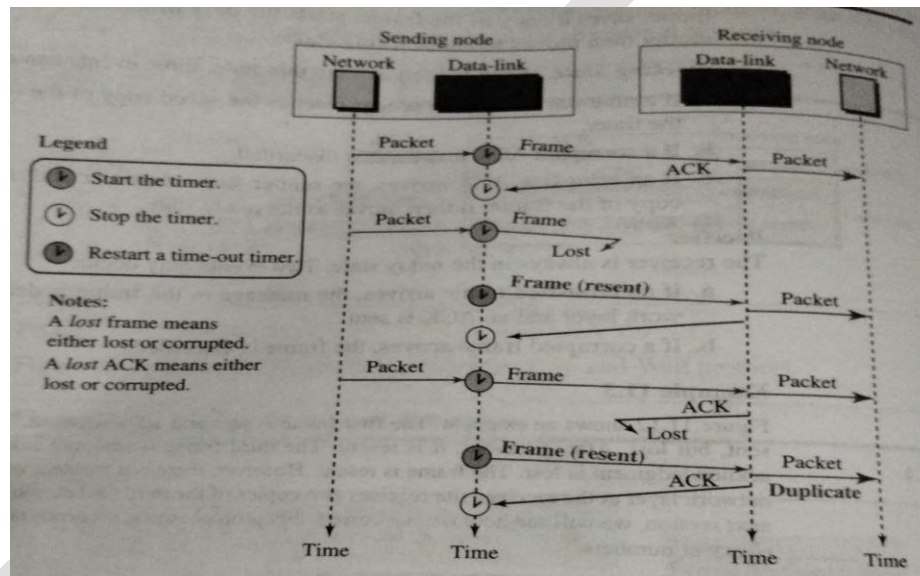
- **Blocking State**

1. When the sender is in this state, three events can occur.
 - a. If a time-out occurs, the sender resends the saved copy of the frame and restarts the timer.
 - b. If a corrupted ACK arrives, it is discarded.
 - c. If an error-free ACK arrives, the sender stops the timer and discards the saved copy of the frame. It then moves to the ready state.

- **Receiver**

1. The receiver is always in the ready state. Two events may occur:
2. **a.** If an error-free frame arrives, the message in the frame is delivered to the network layer and an ACK is sent
3. **.b.** If a corrupted frame arrives, the frame is discarded.

Flow Diagram for Stop and Wait Protocol



- The first frame is sent successfully and acknowledged by the receiver.
- The second frame is sent but gets lost during transmission.
- After a time-out, the sender resends the second frame.
- The third frame is sent and acknowledged successfully, but the acknowledgment is lost.
- Due to the lost acknowledgment, the sender resends the third frame.
- The receiver receives the same third frame twice, resulting in duplicate data being delivered to the network layer.
- This duplication is an error in the Stop-and-Wait scheme.

- To fix this issue, sequence numbers and acknowledgment numbers are introduced in the next section.

Sequence and Acknowledgment Numbers

- Duplicate packets can cause serious problems, such as repeating actions (e.g., ordering the same item twice).
- To prevent duplicates, sequence numbers are added to data frames and acknowledgment numbers to ACK frames.
- Sequence numbers help the receiver identify new frames and detect duplicates.
- In the Stop-and-Wait protocol, numbering is simple and alternates between 0 and 1 (i.e., 0, 1, 0, 1, ...).
- Acknowledgment numbers also alternate as 1, 0, 1, 0, ... corresponding to the next expected frame.
- The acknowledgment number indicates the sequence number of the next frame the receiver expects.
- Automatic Repeat Request mechanism ensures reliable and duplicate-free data transmission between sender and receiver.

3 Go and Back N

4 Selective Repeated Protocol (Refer in Flow Control)

5 HIGH-LEVEL DATA LINK CONTROL (HDLC)

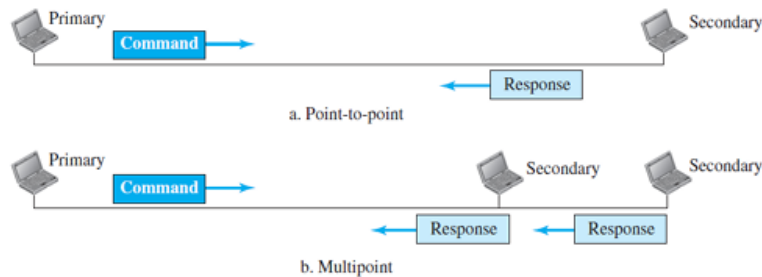
High-level Data Link Control (HDLC) is a bit-oriented protocol for communication over point to-point and multipoint links. It implements the Stop-and-Wait protocol.

Configurations and Transfer Modes

HDLC provides two common transfer modes that can be used in different configurations:

1. Normal Response Mode

In normal response mode (NRM), the station configuration is unbalanced. We have one primary station and multiple secondary stations. A primary station can send commands; a secondary station can only respond. The NRM is used for both point-to-point and multiple-point links, as shown in Figure.



2. Asynchronous Balanced Mode

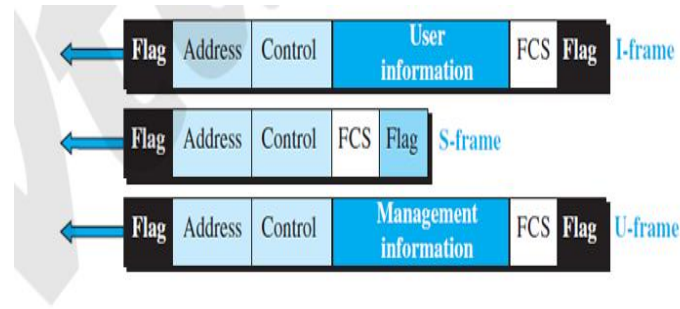
In asynchronous balanced mode (ABM), the configuration is balanced. The link is point-to-point, and each station can function as a primary and a secondary (acting as peers), as shown in Figure. This is the common mode today.



Framing

To provide the flexibility necessary to support all the options possible in the modes and configurations just described, HDLC defines three types of frames:

- 1. I-frames (information frames)** are used to transport user data and control information relating to user data (piggybacking).
- 2. S-frames (supervisory frames)** are used only to transport control information. V-frames are reserved for system management.
- 3. U frames (unnumbered frames)** Information carried by U-frames is intended for managing the link itself.



Flag field-The flag field of an HDLC frame is an 8-bit sequence with the bit pattern 01111110 that identifies both the beginning and the end of a frame and serves as a synchronization pattern for the receiver.

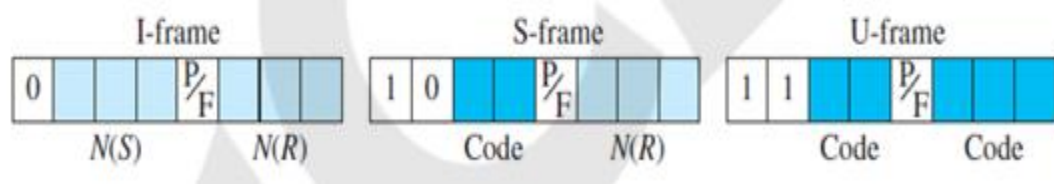
Address field- The field of an HDLC frame contains the address of the secondary station. If a primary station created the frame, it contains a **to address**. If a secondary creates the frame, it contains a **from address**. An address field can be **1 byte or several bytes long**, depending on the needs of the network.

Control field -The control field is a 1- or 2-byte segment of the frame used for flow and error control. The interpretation of bits in this field depends on the frame type.

Information field-The information field contains the user's data from the network layer or management information. Its length can vary from one network to another.

FCS field-The frame check sequence (FCS) is the HDLC error detection field. It can contain either a 2- or 4-byte CRC.

Control Field- The control field determines the type of frame and defines its functionality.



The control field determines the type of frame and defines its functionality.

Control Field for I-Frames

I-frames are designed to carry user data from the network layer. In addition, they can include flow and error control information (piggybacking). The subfields in the control field are used to define these functions.

- The first bit defines the type. If the first bit of the control field is 0, this means the frame is an I-frame.
- The next 3 bits, called N(S), define the sequence number of the frame.
- The last 3 bits, called N(R), correspond to the acknowledgment number when piggybacking is used.
- The single bit between N(S) and N(R) is called as P/F.
- The P/F field is a single bit with a dual purpose. It has meaning only when it is set (bit = 1) and can mean poll or final. It means poll when the frame is sent by a primary station to a secondary (when the address field contains the address of the receiver). It means final when the frame is sent by a secondary to a primary (when the address field contains the address of the sender).

Control Field for S-Frames

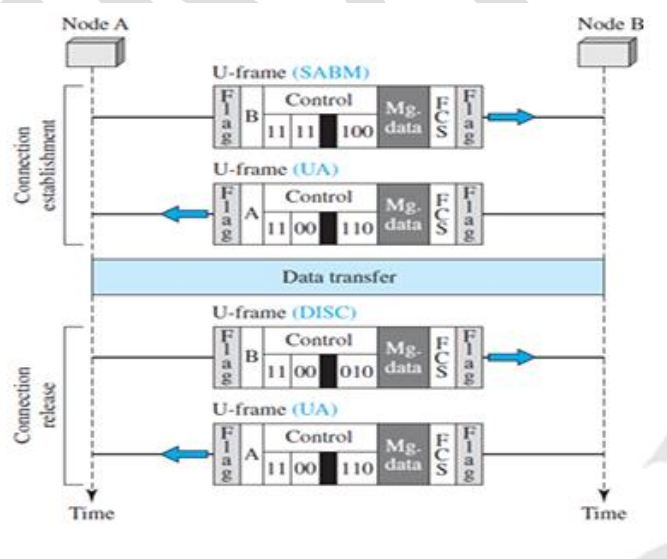
Supervisory frames are used for flow and error control whenever piggybacking is either impossible or inappropriate. S-frames do not have information fields. If the first 2 bits of the control field is 10, this means the frame is an S-frame. The last 3 bits, called N(R), corresponds to the acknowledgment number (ACK) or negative acknowledgment number (NAK) depending on the type of S-frame. The 2 bits called code is used to define the type of S-frame itself. With 2 bits, we can have four types of S-frames, as described below:

- 1. Receive ready (RR)** If the value of the code subfield is 00, it is an RR S-frame. Receive not ready (RNR). If the value of the code subfield is 10, it is an RNR S-frame. The value of NCR is the acknowledgment number.
- 2. Reject (REJ)** If the value of the code subfield is 01, it is a REJ S-frame. The value of NCR is the negative acknowledgment number.

3. Selective reject (SREJ) If the value of the code subfield is 11, it is an SREJ S-frame. This is a NAK frame used in Selective Repeat ARQ. The value of N(R) is the negative acknowledgment number.

Control Field for U-Frames

Unnumbered frames are used to exchange session management and control information between connected devices. U-frame codes are divided into two sections: a 2-bit prefix before the P/F bit and a 3-bit suffix after the P/F bit. Together, these two segments (5 bits) can be used to create up to 32 different types of U-frames. Figure shows how U-frames can be used for connection establishment and connection release. Node A asks for a connection with a set asynchronous balanced mode (SABM) frame; node B gives a positive response with an unnumbered acknowledgment (UA) frame. After these two exchanges, data can be transferred between the two nodes (not shown in the figure). After data transfer, node A sends a DISC (disconnect) frame to release the connection; it is confirmed by node B responding with a UA (unnumbered acknowledgment).



POINT-TO-POINT PROTOCOL

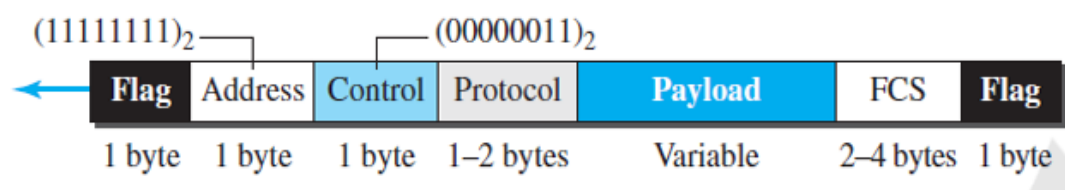
Although HDLC is a general protocol that can be used for both point-to-point and multipoint configurations, the most common protocols for point-to-point access is the Point-to-Point Protocol (PPP). PPP is by far the most common. PPP provides several services:

1. PPP defines the format of the frame to be exchanged between devices. PPP defines how two devices can negotiate the establishment of the link and the exchange of data.
2. PPP defines how network layer data are encapsulated in the data link frame.
3. PPP defines how two devices can authenticate each other.
4. PPP provides multiple network layer services supporting a variety of network layer protocols.
5. PPP provides connections over multiple links.
6. PPP provides network address configuration. This is particularly useful when a home user needs a temporary network address to connect to the Internet. On the other hand, to keep PPP simple, several services are missing:
 1. PPP does not provide flow control. A sender can send several frames one after another with no concern about overwhelming the receiver.
 2. PPP has a very simple mechanism for error control.
 3. PPP does not provide a sophisticated addressing mechanism to handle frames in a multipoint configuration.

Framing

PPP is a byte-oriented protocol. Framing is done according byte-oriented protocols.

Frame Format



Flag.- A PPP frame starts and ends with a 1-byte flag with the bit pattern 01111110.

Address- The address field in this protocol is a constant value and set to 11111111 (broadcast address). During negotiation (discussed later), the two parties may agree to omit this byte.

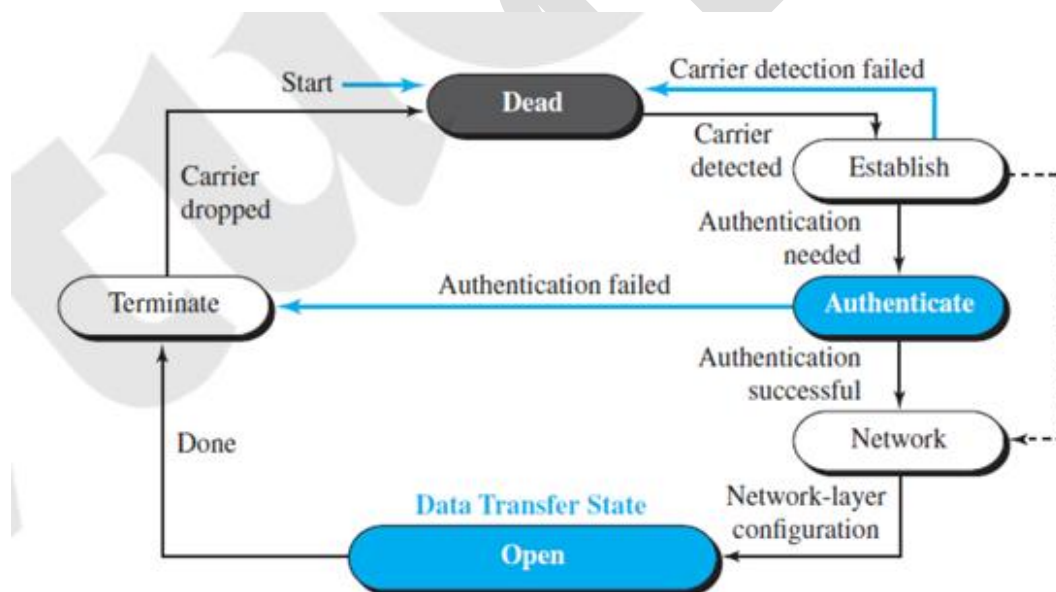
Control.- This field is set to the constant value 11000000.

Protocol-The protocol field defines what is being carried in the data field: either user data or other information. This field is by default 2 bytes long, but the two parties can agree to use only 11 byte.

Payload field- This field carries either the user data or other information. The data field is a sequence of bytes with the default of a maximum of 1500 bytes; but this can be changed during negotiation.

FCS-The frame check sequence (FCS) is simply a 2-byte or 4-byte standard CRC. Transition Phases

A PPP connection goes through phases which can be shown in a transition phase diagram.



Dead -In the dead phase the link is not being used. There is no active carrier and the line is quiet.

Establish- When one of the nodes starts the communication, the connection goes into this phase. In this phase, options are negotiated between the two parties.

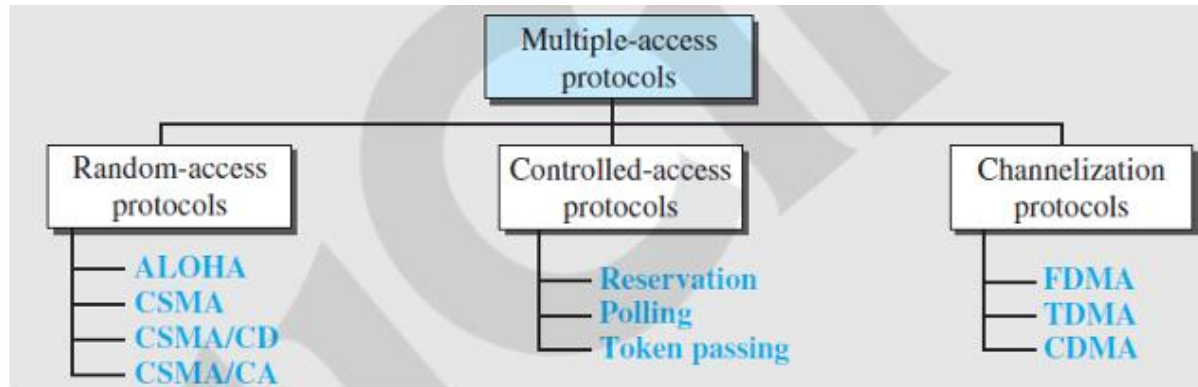
Authenticate- The authentication phase is optional; the two nodes may decide, during the establishment phase, not to skip this phase.

Network- In the network phase, negotiation for the network layer protocols takes place. Open In the open phase, data transfer takes place.

Terminate In the termination phase the connection is terminated.

Media Access control

When nodes or stations are connected and use a common link, called a multipoint or broadcast link, we need a multiple-access protocol to coordinate access to the link. Many protocols have been devised to handle access to a shared link. All of these protocols belong to a sublayer in the data-link layer called media access control (MAC). We categorize them into three groups.



RANDOM ACCESS

In random access or contention methods, no station is superior to another station. A station that has data to send uses a procedure defined by the protocol to make a decision on whether or not to send. Two features give this method its name.

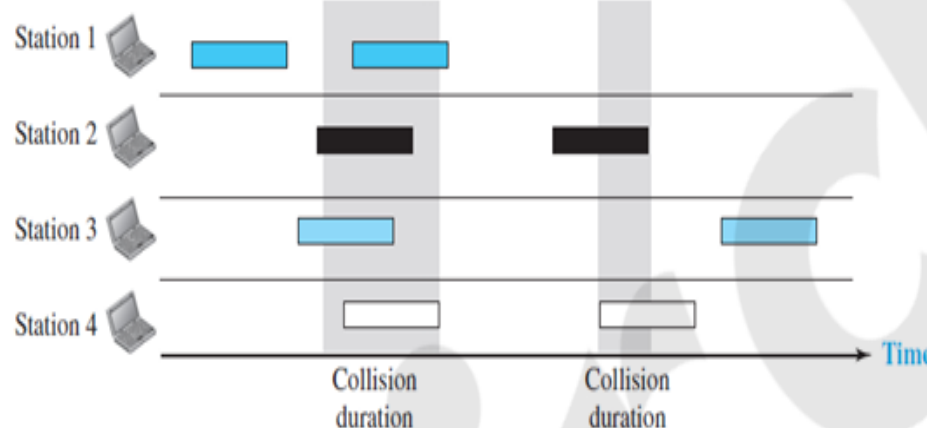
1. There is no scheduled time for a station to transmit. Transmission is random and hence called random access.
2. No rules specify which station should send next. Station compete with one another to access the medium and hence called contention methods. If more than one station tries to send, there is an access conflict-collision-and the frames will be either destroyed or modified.

1. ALOHA

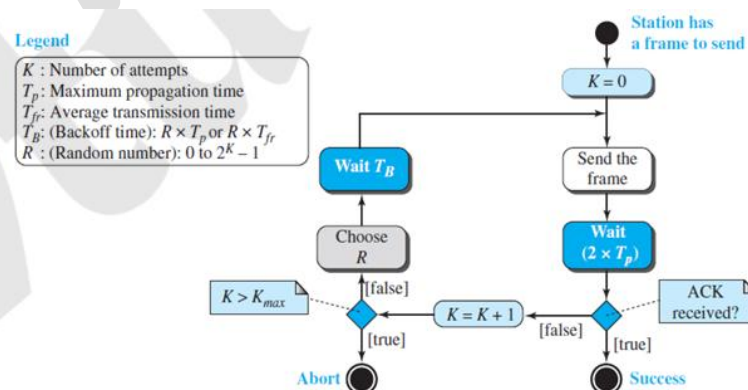
ALOHA, the earliest random access method was developed at the University of Hawaii in early 1970. The medium is shared between the stations and hence has potential arrangements.

Pure ALOHA

- The original ALOHA protocol is called pure ALOHA. The idea is that each station sends a frame whenever it has a frame to send.
- Since there is only one channel to share, there is the possibility of collision between frames from different stations.
- There are four stations (unrealistic assumption) that contend with one another for access to the shared channel. Only two frames survive: frame 1.1 from station 1 and frame 3.2 from station 3.



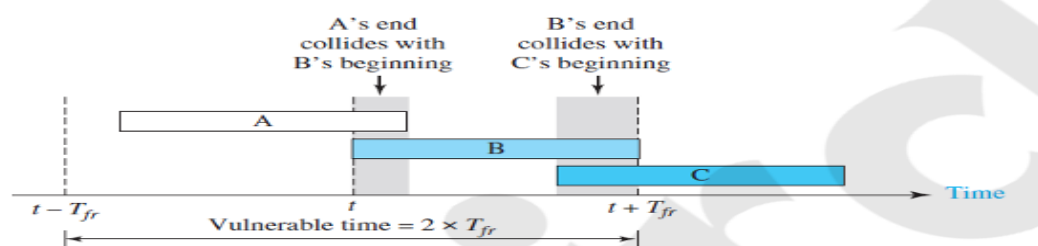
- The pure ALOHA protocol relies on acknowledgments from the receiver. If the acknowledgment does not arrive after a time-out period, the station resends the frame.
- If all these stations try to resend their frames after the time-out, the frames will collide again. Each station waits for a random time before resending, called the back-off time T_B .
- In the second method after a maximum number of retransmission attempts K_{max} a station must give up and try later.



Vulnerable time- Station

A sends a frame at time t . Now imagine station B has already sent a frame between $t - T_{fr}$ and t . This leads to a collision between the frames from station A and station B. The end of B's frame collides with the beginning of A's frame. On the other hand, suppose that station C sends a frame between t and $t + T_{fr}$. Here, there is a collision between frames from station A and station C. The beginning of C's frame collides with the end of A's frame.

Pure ALOHA vulnerable time = $2 \times T_{fr}$

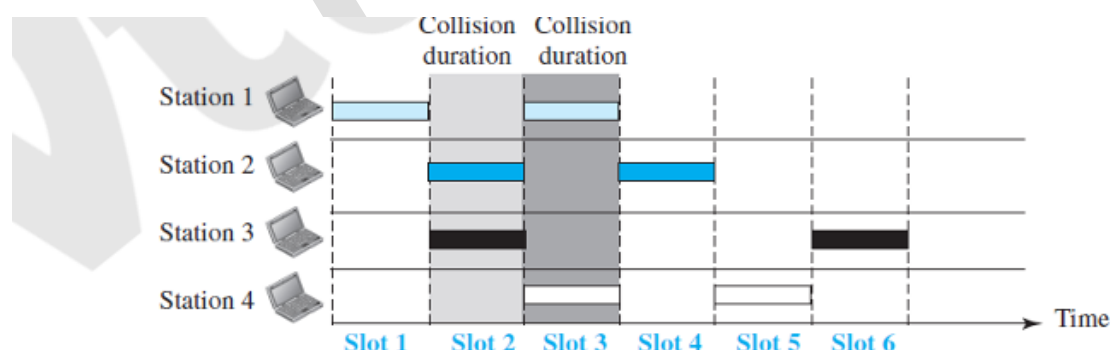


Throughput

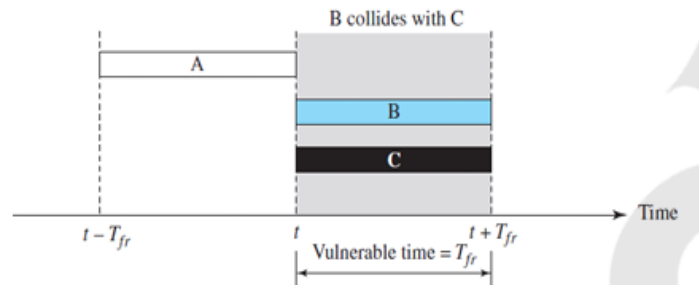
Let us call G the average number of frames generated by the system during one frame transmission time. Then it can be proved that the average number of successful transmissions for pure ALOHA is $S = G \times e^{-2G}$. The maximum throughput S_{max} is 0.184, for $G = 1$.

2. Slotted ALOHA

In slotted ALOHA we divide the time into slots of T_{fr} and force the station to send only at the beginning of the time slot.

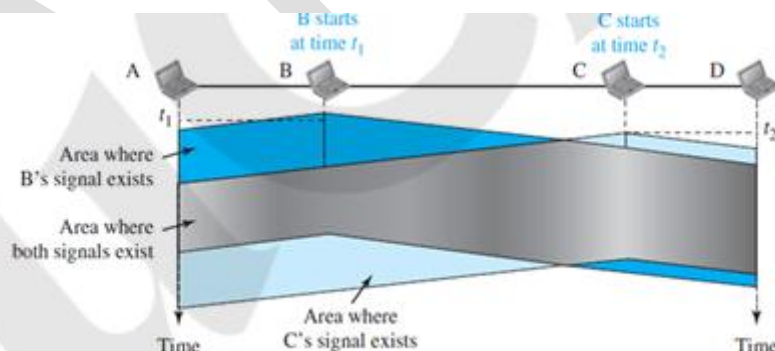


The vulnerable time for slotted ALOHA is one-half that of pure ALOHA. Slotted ALOHA vulnerable time = T_{fr} Throughput It can be proved that the average number of successful transmissions for ALOHA is $S = G \times e^{-G}$. The maximum throughput S_{max} is 0.368, when $G = 1$.



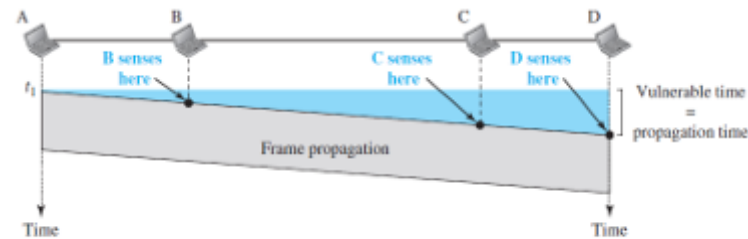
2. CARRIER SENSE MULTIPLE ACCESS (CSMA)

- Carrier sense multiple access (CSMA) requires that each station first listen to the medium before sending. The principle "sense before transmit" or "listen before talk."
- CSMA can reduce the possibility of collision, but it cannot eliminate it. Stations are connected to a shared channel.
- The possibility of collision still exists because of propagation delay; when a station sends a frame, it still takes time for the first bit to reach every station and for every station to sense it.



Vulnerable Time

The vulnerable time for CSMA is the propagation time T_p . This is the time needed for a signal to propagate from one end of the medium to the other.



Persistence Methods

This method tells, What should a station do if the channel is busy? What should a station do if the channel is idle? Three methods have been devised

1. I-persistent - In this method, after the station finds the line idle, it sends its frame immediately.

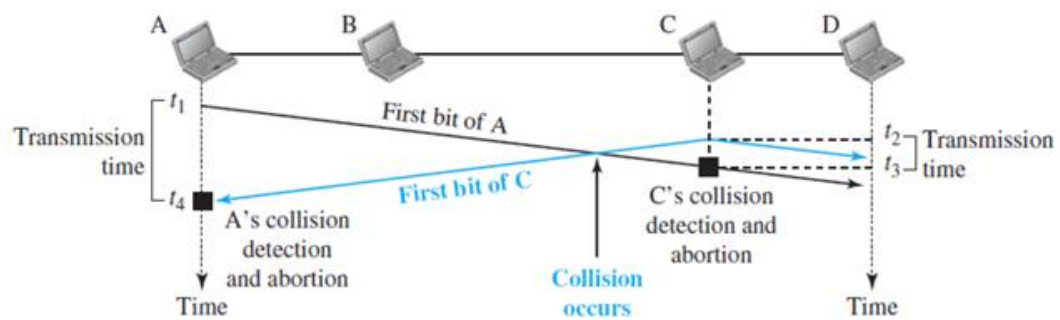
2. Nonpersistent- In this method, a station that has a frame to send senses the line. If the line is idle, it sends immediately. If the line is not idle, it waits a random amount of time and then senses the line again.

3. p-Persistent -This method is used if the channel has time slots with a slot duration equal to or greater than the maximum propagation time. In this method, after the station finds the line idle it follows these steps:

1. With probability p , the station sends its frame.
2. With probability $q = 1 - p$, the station waits for the beginning of the next time slot and checks the line again.
 - a. If the line is idle, it goes to step 1.
 - b. If the line is busy, it acts as though a collision has occurred and uses the backoff procedure .



1. At time t_1 , station A has executed its persistence procedure and starts sending the bits of its frame.
2. At time t_2 , station C has not yet sensed the first bit sent by A. Station C executes its persistence procedure and starts sending the bits in its frame, which propagate both to the left and to the right.
3. The collision occurs sometime after time t_2' . Station C detects a collision at time t_3 when it receives the first bit of A's frame.
4. Station C immediately aborts transmission.
5. Station A detects collision at time t_4 when it receives the first bit of C's frame; it also immediately aborts transmission.



Minimum Frame Size

→ For CSMA/CD to work, a restriction on the frame size is required.

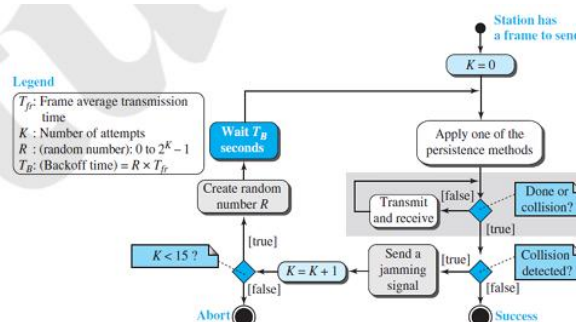
→ Before sending the last bit of the frame, the sending station must detect a collision, if any, because it does not keep a copy of the frame.

→ Therefore, the frame transmission time T_{fr} must be at least two times the maximum propagation time T_p .

Procedure

It is similar to the one for the ALOHA protocol, but there are differences.

1. Addition of the persistence process.
2. In ALOHA, we first transmit the entire frame and then wait for an acknowledgment. In CSMA/CD, transmission and collision detection is a continuous process.
3. A short jamming signal that enforces the collision in case other stations have not yet sensed the collision.



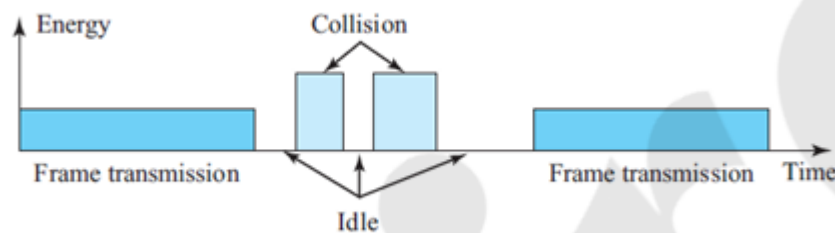
Energy Level

The level of energy in a channel can have three values: zero, normal, and abnormal.

→ At the zero level, the channel is idle.

→ At the normal level, a station has successfully captured the channel and is sending its frame.

→ At the abnormal level, there is a collision and the level of the energy is twice the normal level. A station that has a frame to send or is sending a frame needs to monitor the energy level to determine if the channel is idle, busy, or in collision mode.



Throughput

The throughput of CSMA/CD is greater than that of pure or slotted ALOHA. The maximum throughput occurs at a different value of G and is based on the persistence method and the value of p in the p -persistent approach.

1. For I-persistent method the maximum throughput is around 50 percent when $G = 1$.
2. For nonpersistent method, the maximum throughput can go up to 90 percent when G is between 3 and 8.

4.CARRIER SENSE MULTIPLE ACCESS WITH COLLISION AVOIDANCE (CSMA/CA)

The basic idea behind CSMA/CD is that a station needs to be able to receive while transmitting to detect a collision.

→ When there is no collision, the station receives one signal: its own signal. When there is a collision, the station receives two signals: its own signal and the signal transmitted by a second station.

→ To distinguish between these two cases, the received signals in these two cases must be significantly different.

→ In a wired network, the received signal has almost the same energy. In case of collision, the detected energy almost doubles.

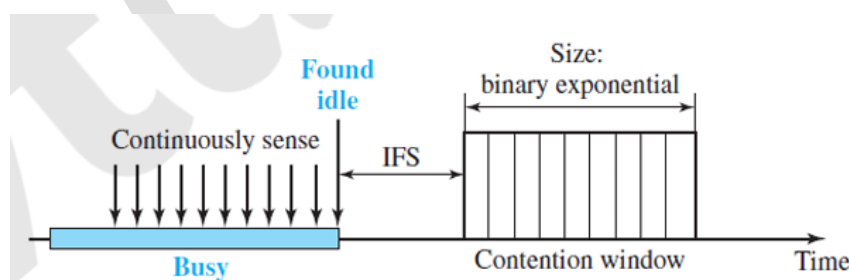
→ In a wireless network, much of the sent energy is lost in transmission. The received signal has very little energy. Collision may add very little additional energy. This is not useful for effective collision detection.

→ Collision must be avoided on wireless network, using the following strategies

1. Interframe Space (IFS) Collisions are avoided by deferring transmission even if the channel is found idle. When an idle channel is found, the station does not send immediately. It waits for a period of time called the interframe space or IFS.

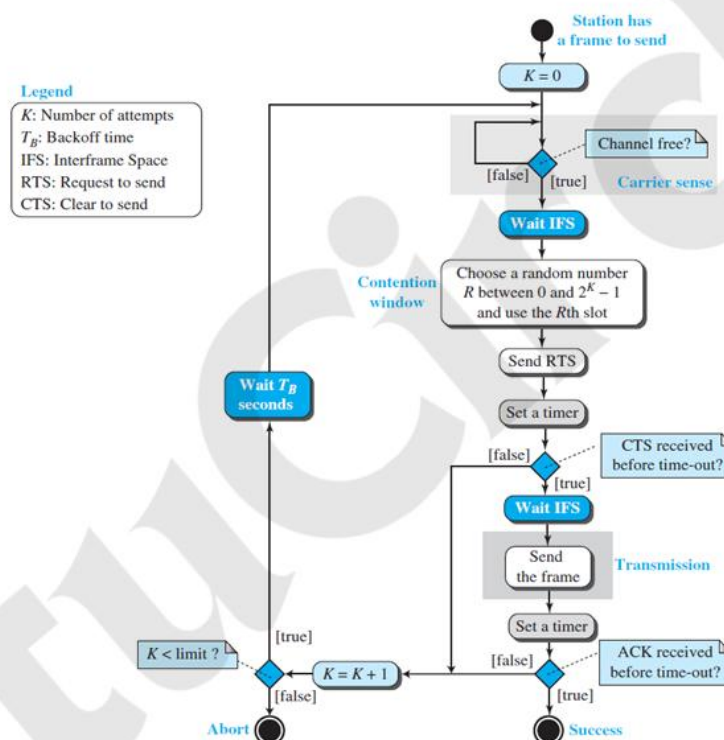
2. Contention Window The contention window is an amount of time divided into slots. A station that is ready to send chooses a random number of slots as its wait time. The number of slots in the window changes according to the binary exponential back-off strategy (set to one slot the first time and then doubles each time the station cannot detect an idle channel after the IFS time).

3. Acknowledgment With all these precautions, there still may be a collision resulting in destroyed data. In addition, the data may be corrupted during the transmission. The positive acknowledgment and the time-out timer can help guarantee that the receiver has received the frame.



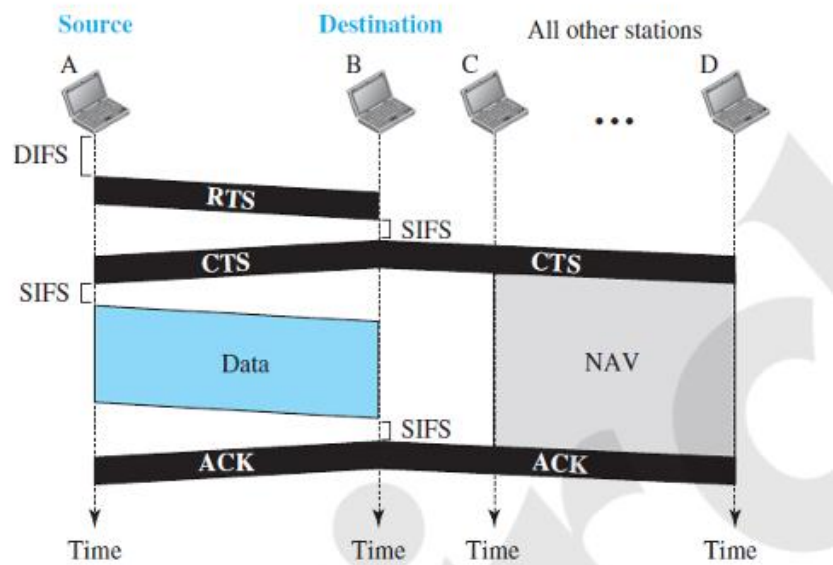
Procedure

- The channel needs to be sensed before and after the IFS. The channel also needs to be sensed during the contention time.
- For each time slot of the contention window, the channel is sensed.
- If it is found idle, the timer continues; if the channel is found busy, the timer is stopped and continues after the timer becomes idle again.



Frame Exchange Time

Line Figure shows the exchange of data and control frames in time.



1. Before sending a frame, the source station senses the medium by checking the energy level at the carrier frequency.

- a. The channel uses a persistence strategy with backoff until the channel is idle.
 - b. After the station is found to be idle, the station waits for a period of time called the DCF interframe space (DIFS); then the station sends a control frame called the request to send (RTS).
2. After receiving the RTS and waiting a period of time called the short interframe space (SIFS), the destination station sends a control frame, called the clear to send (CTS), to the source station. This control frame indicates that the destination station is ready to receive data.
3. The source station sends data after waiting an amount of time equal to SIFS. 4. . The destination station, after waiting an amount of time equal to SIFS, sends an acknowledgment to show that the frame has been received.

Network Allocation Vector

Collision avoidance aspect of this protocol is accomplished by a feature called NAV.

- When a station sends an RTS frame, it includes the duration of time that it needs to occupy the channel.
- The stations that are affected by this transmission create a timer called a network allocation vector (NAV) that shows how much time must pass before these stations are allowed to check the channel for idleness.
- Each station, before sensing the physical medium to see if it is idle, first checks its NAV to see if it has expired.

Collision During Handshaking

Two or more stations may try to send RTS frames at the same time. These control frames may collide. However, because there is no mechanism for collision detection, the sender assumes there has been a collision if it has not received a CTS frame from the receiver. The back off strategy is employed, and the sender tries again.

CONTROLLED ACCESS

In controlled access, the stations consult one another to find which station has the right to send. A station cannot send unless it has been authorized by other stations.

1. Reservation

- In the reservation method, a station needs to make a reservation before sending data.
- Time is divided into intervals.
- In each interval, a reservation frame precedes the data frames sent in that interval.

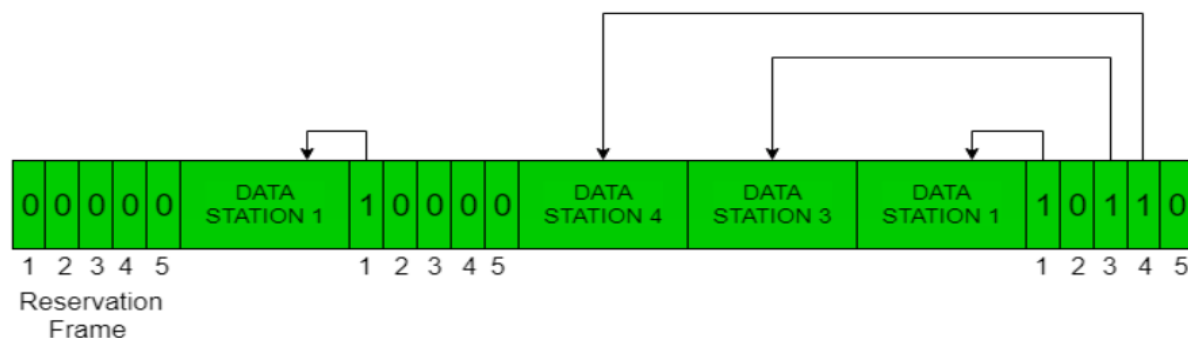


Figure shows a situation with five stations and a five-minis lot reservation frame. In the first interval, only stations 1, 3, and 4 have made reservations. In the second interval, only station 1 has made a reservation.

2. Polling

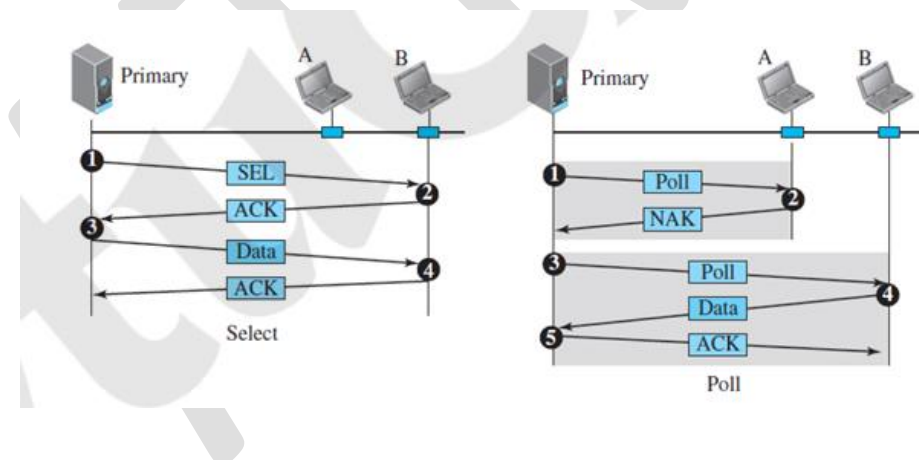
→ Polling works with topologies in which one device is designated as a primary station and the other devices are secondary stations.

→ The primary device controls the link; the secondary devices follow its instructions.

→ The primary device, therefore, is always the initiator of a session.

→ If the primary wants to receive data, it asks the secondary's if they have anything to send; this is called poll function. Secondary responds with DATA if any, else negative ack (NAK) if it has nothing to send.

→ If the primary wants to send data, it tells the secondary to get ready to receive; this is called select function. It transmits SEL and waits for acknowledgement from secondary before sending data.



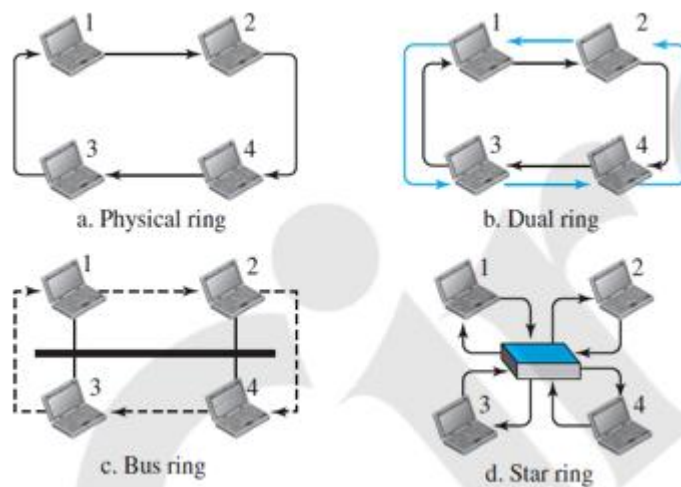
3. Token Passing

→ In the token-passing method, the stations in a network are organized in a logical ring.

→ A special packet called a token circulates through the ring.

→ The possession of the token gives the station the right to access the channel and send its data.

- When a station has some data to send, it holds the token and sends its data and releases the token once data is sent.
- When a station receives the token and has no data to send, it just passes the data to the next station.
- Stations must be limited in the time they can have possession of the token.
- The token must be monitored to ensure it has not been lost or destroyed.
- Stations do not have to be physically connected in a ring; the ring can be a logical one.



- The problem with this topology is that if one of the links-the medium between two adjacent stations fails, the whole system fails.
- The dual ring topology uses a second (auxiliary) ring which operates in the reverse direction compared with the main ring.