Data Link Layer(Flow and Error Control)

Design issues:

The Data Link Layer is the second layer in the OSI (Open Systems Interconnection) model and is responsible for providing reliable transfer of data between two adjacent nodes on a network. Some of the design issues related to the Data Link Layer are:

Framing: The Data Link Layer needs to segment the data received from the Network Layer into frames before transmission. The design must ensure that the frames are of a fixed or variable size and can be easily identified by the receiver.

Flow Control: The Data Link Layer should ensure that the sender does not overwhelm the receiver by transmitting data at a faster rate than it can handle. The design must incorporate mechanisms for flow control to avoid buffer overflows and data loss.

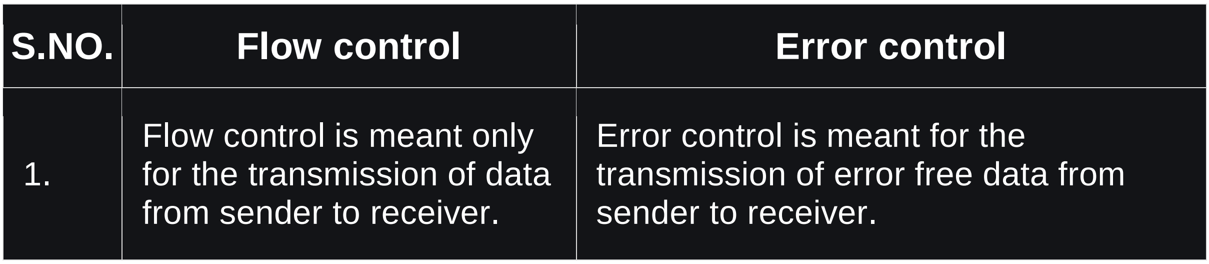
Error Control: The Data Link Layer should ensure that data transmitted is received correctly without errors. The design must incorporate error detection and correction mechanisms to detect and recover from transmission errors.

Access Control: The Data Link Layer should provide a mechanism for multiple devices to share the same transmission medium. The design must ensure that only one device transmits at a time to avoid data collisions.

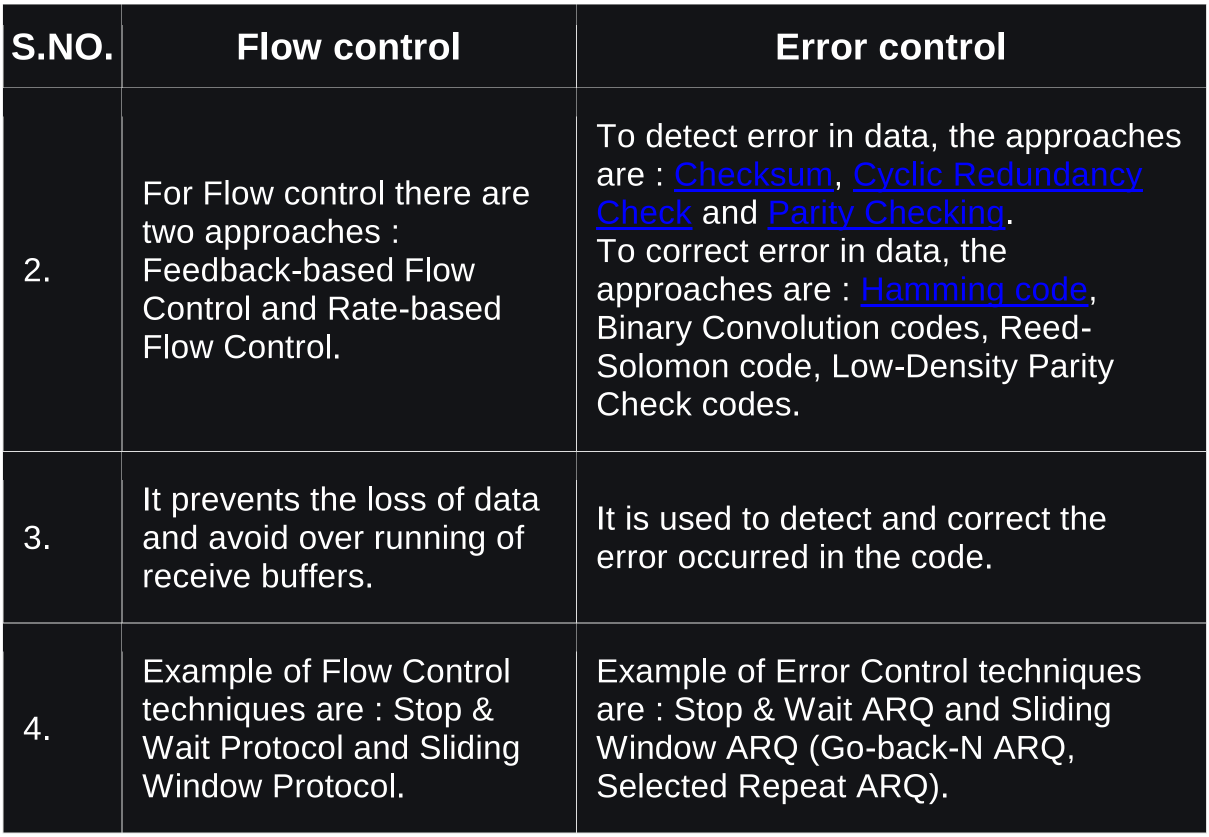
Addressing: The Data Link Layer needs to assign unique addresses to devices connected to the network. The design must ensure that the addressing scheme is unique and easy to manage.

Media selection: The Data Link Layer must be designed to work with different types of media such as wired or wireless. The design must ensure that the protocols used are compatible with the selected media.

Error Detection & Correction



S.NO.	Flow control	Error control
1.	Flow control is meant only for the transmission of data from sender to receiver.	Error control is meant for the transmission of error free data from sender to receiver.



S.NO.	Flow control	Error control
2.	For Flow control there are two approaches : Feedback-based Flow Control and Rate-based Flow Control.	To detect error in data, the approaches are : Checksum, Cyclic Redundancy Check and Parity Checking. To correct error in data, the approaches are : Hamming code, Binary Convolution codes, Reed-Solomon code, Low-Density Parity Check codes.
3.	It prevents the loss of data and avoid over running of receive buffers.	It is used to detect and correct the error occurred in the code.
4.	Example of Flow Control techniques are : Stop & Wait Protocol and Sliding Window Protocol.	Example of Error Control techniques are : Stop & Wait ARQ and Sliding Window ARQ (Go-back-N ARQ, Selected Repeat ARQ).

Error detection and correction are techniques used to ensure that data transmitted over a communication channel is received without errors. In digital communication systems, errors can occur due to various factors such as noise, interference, attenuation, and distortion. Error detection and correction techniques help to minimize these errors and ensure that the received data is accurate.

Error Detection

Error detection involves adding additional bits to the data being transmitted to detect any errors that may occur during transmission. The most common techniques for error detection include:

Parity Check: Parity bits are added to the transmitted data to detect errors. Even parity involves adding an extra bit such that the total number of 1's in the transmitted data and parity bit is even, and odd parity involves adding an extra bit such that the total number of 1's in the transmitted data and parity bit is odd.

Checksum: A checksum is a value computed from the transmitted data that is used to detect errors. The sender computes the checksum and includes it in the transmitted data. The receiver then computes the checksum of the received data and compares it to the checksum received from the sender to detect any errors.

Cyclic Redundancy Check (CRC): CRC is a more robust error detection technique that involves generating a polynomial code from the transmitted data. The receiver performs the same calculation and compares the generated code to the one received from the sender to detect errors.

Error Correction

Error correction involves identifying and correcting errors that occur during transmission. The most common techniques for error correction include:

Automatic Repeat Request (ARQ): ARQ involves retransmitting the data that is lost or corrupted during transmission. The receiver sends a request to the sender to resend the data, and the sender retransmits it.

Forward Error Correction (FEC): FEC involves adding redundancy to the transmitted data such that the receiver can use the redundant data to correct errors. The receiver uses the redundant data to reconstruct the original data in case of errors.

In summary, error detection and correction techniques are critical in digital communication systems to ensure that data is transmitted and received accurately. Different techniques can be used depending on the level of accuracy required, the available resources, and the characteristics of the communication channel.

Flow control & Error Control

The data link layer is the second layer of the OSI model and is responsible for the reliable transmission of data between two devices over a physical link. Within the data link layer, two important functions are flow control and error control.

Flow control is the process of managing the rate of data transmission between two devices to prevent one device from overwhelming the other with too much data. This is typically achieved using a technique called sliding window protocol. The sliding window protocol enables the sender to transmit a specific number of data frames before waiting for an acknowledgment from the receiver. The receiver sends an acknowledgement frame to the sender to acknowledge that the data has been received. This helps to prevent overloading the receiver's buffer and allows both devices to operate at the same speed.

Error control is the process of ensuring that the data being transmitted between two devices is error-free. This is achieved using a technique called error detection and correction. Error detection involves adding additional bits to the data being transmitted, which can be used to detect errors that occur during transmission. These additional bits are known as parity bits or checksums. Error correction involves retransmitting any frames that have been detected as being corrupted or lost during transmission. This is done by using a technique called Automatic Repeat Request (ARQ), which involves retransmitting a frame if it is not acknowledged by the receiver within a specified time period.

Both flow control and error control are critical in ensuring reliable data transmission between devices over a physical link. They help to prevent errors, lost data, and overloading of the receiver's buffer, which can result in poor performance or even data loss.

Sliding Window Protocols

Sliding Window Protocol is a flow control protocol used in computer networks to manage the flow of data between two devices or nodes that communicate with each other over a communication channel. It is used in the Data Link Layer of the OSI model to ensure reliable transmission of data.

The Sliding Window Protocol allows the sender to transmit multiple frames before receiving an acknowledgment (ACK) from the receiver. The sender maintains a sliding window of fixed size, which consists of a sequence of frames that it can transmit without receiving an ACK. The receiver, in turn, maintains a corresponding sliding window of the same size that indicates the frames it expects to receive.

As the sender transmits frames, it slides the window to the right, and the receiver slides its window to the right as it receives frames. If a frame is lost or damaged, the receiver sends a negative acknowledgment (NAK) to the sender, requesting retransmission of the lost frame. The sender then retransmits the lost frame and resumes transmission from where it left off.

There are two types of sliding window protocols: Go-Back-N and Selective Repeat. In the Go-Back-N protocol, if a frame is lost, the sender retransmits all the frames that were sent after the lost frame. In the Selective Repeat protocol, only the lost frame is retransmitted, and the receiver discards any duplicate frames that it receives.

Overall, Sliding Window Protocol is an efficient method for managing the flow of data over a communication channel, and it helps to ensure reliable transmission of data between two devices or nodes in a network.

…………………………………….

Here are the steps involved in the Sliding Window Protocol, presented as a list of points:

- The sender divides the data into fixed-size frames and assigns a sequence number to each frame.
- The sender sends the first frame and starts a timer.
- If the sender does not receive an ACK for the first frame before the timer expires, it resends the frame and restarts the timer.
- The sender maintains a sliding window of fixed size that consists of a sequence of frames it can transmit without receiving an ACK.
- As the sender transmits frames, it slides the window to the right.
- The receiver maintains a corresponding sliding window of the same size that indicates the frames it expects to receive.
- If the receiver receives a frame successfully, it sends an ACK to the sender, indicating the next expected frame.
- If the receiver detects an error in a received frame, it sends a NAK to the sender, requesting retransmission of the lost frame.
- If the sender receives a NAK from the receiver, it retransmits the lost frame and resumes transmission from where it left off.
- If the sender receives an ACK for a frame, it removes the frame from the sliding window and slides the window to the right.
- If the sender receives an ACK for all frames in the sliding window, it sends the next set of frames.
- The process continues until all frames have been transmitted and received successfully.

ARQ: Stop & Wait , Go Back n, Selective Repeat

These terms are related to the way data is transmitted over a network, specifically in the context of error control and flow control. Here's a simple explanation of each:

Stop & Wait: In this method, the sender sends one packet of data and waits for an acknowledgement (ACK) from the receiver before sending the next packet. This ensures that the packets are received in order, but it can be slow and inefficient if there is a lot of network latency or the packets are small.

Go Back n: In this method, the sender sends multiple packets of data without waiting for an ACK. The receiver buffers the packets and sends an ACK for each packet it receives successfully. If the sender does not receive an ACK for a particular packet, it assumes that all subsequent packets have been lost and retransmits all the packets from that point onwards.

Selective Repeat: This is similar to Go Back n, but instead of retransmitting all subsequent packets when a packet is lost, the sender only retransmits the specific lost packet. This can be more efficient than Go Back n if there are only a few packets that need to be retransmitted.

Here are some examples of when these methods might be used:

Stop & Wait might be used when transmitting small amounts of data over a low-latency network, such as a local area network (LAN).

Go Back n might be used when transmitting large amounts of data over a high-latency network, such as a satellite link.

Selective Repeat might be used when transmitting data over a network with a moderate amount of latency and occasional packet loss, such as a cellular data network.

Overall, the choice of which method to use depends on factors such as the size and type of data being transmitted, the characteristics of the network, and the desired level of reliability and efficiency.

Examples of DLL protocols – HDLC, PPP

HDLC (High-Level Data Link Control) and PPP (Point-to-Point Protocol) are both examples of Data Link Layer (DLL) protocols used for reliable data transmission over communication links. Here's a brief explanation of each:

HDLC: HDLC is a bit-oriented protocol that provides reliable, error-free data transmission over point-to-point or multipoint communication links. It uses a sliding window protocol for flow control, allowing the sender to transmit multiple frames before receiving an acknowledgement from the receiver. HDLC supports both connection-oriented and connectionless modes of operation and is commonly used in high-speed data communications such as in wide area networks (WANs).

PPP: PPP is a protocol used to establish a direct connection between two nodes over a serial link. It is a lightweight protocol that provides authentication, encryption, and compression capabilities. PPP supports multiple network layer protocols, including IP, IPX, and AppleTalk. It uses a three-step handshake to establish a connection and supports various authentication methods such as PAP and CHAP. PPP is commonly used for dial-up connections, leased lines, and virtual private networks (VPNs).

In summary, HDLC and PPP are two examples of DLL protocols that are used for reliable data transmission over communication links. While HDLC is commonly used in WANs, PPP is used in a variety of network environments, including dial-up connections and VPNs.