MODULE 2

CHAPTER 1 –DATA LINK LAYER

CO – Students will be able to summarize the datalink layer responsibilities and protocols

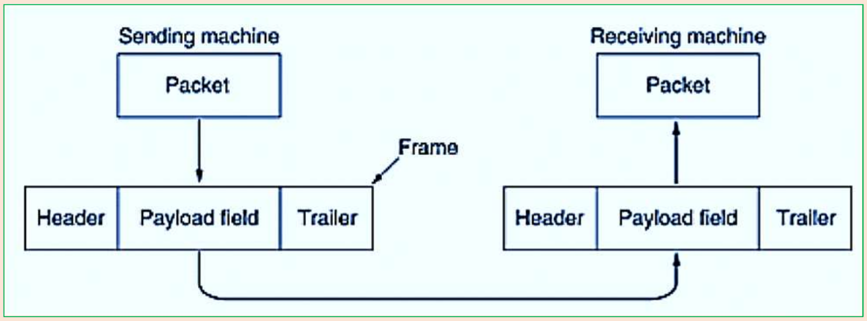
DLL DESIGN ISSUES

➢ Specific responsibilities of the data link layer include framing, addressing, flow control, error control, and media access control.

1. FRAMING

- The data link layer divides the stream of bits received from the network layer into manageable data units called frames.
- The data link layer adds a header to the frame to define the addresses of the sender and receiver of the frame.
- Each frame contains a frame header, a payload field for holding the packet, and a frame trailer

Relationship between packets and frames.



❖**Fixed-Size Framing -** In fixed-size framing, there is no need for defining the boundaries of the frames; the size itself can be used as a delimiter.

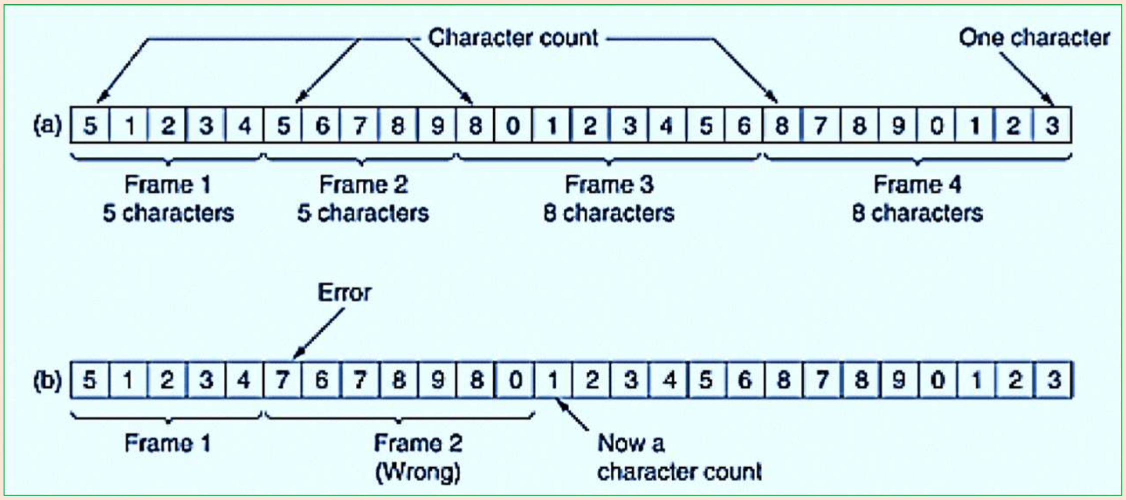
Eg: ATM wide-area network

❖Variable-Size Framing

- In variable-size framing, we need a way to define the end of the frame and the beginning of the next.
- The approaches were used for this purpose are:

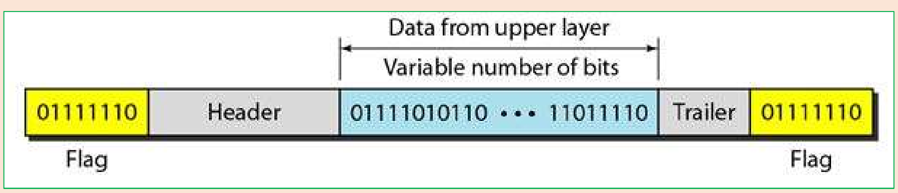
A. Character Count

- It uses a field in the header to specify the number of characters in the frame.
- When the data link layer at the destination sees the character count, it knows how many characters follow and hence where the end of the frame is.



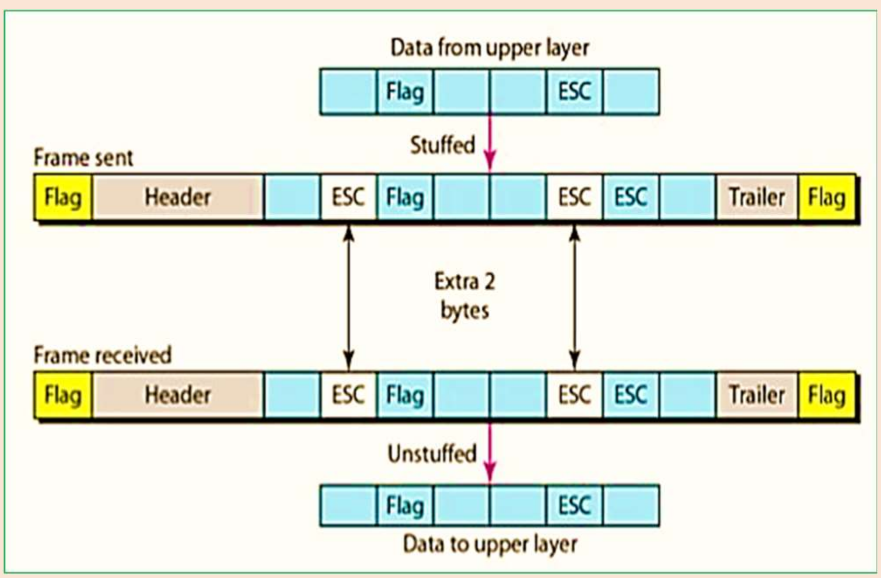
B. Flag bytes with byte stuffing

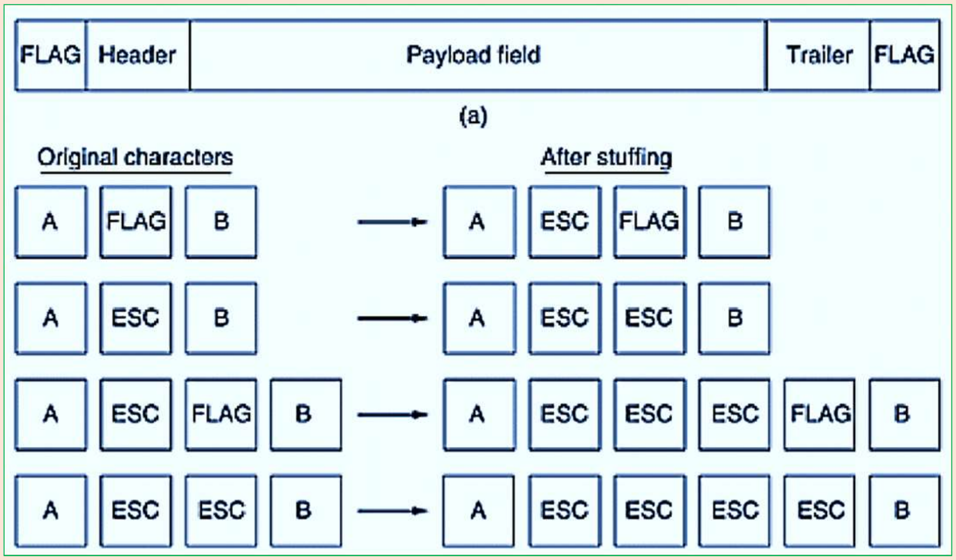
- To separate one frame from the next, an 8-bit (1-byte) flag is added at the beginning and the end of a frame.



- Any pattern used for the flag could also be part of the information. If this happens, the receiver, when it encounters this pattern in the middle of the data, thinks it has reached the end of the frame.
- To fix this problem, 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.

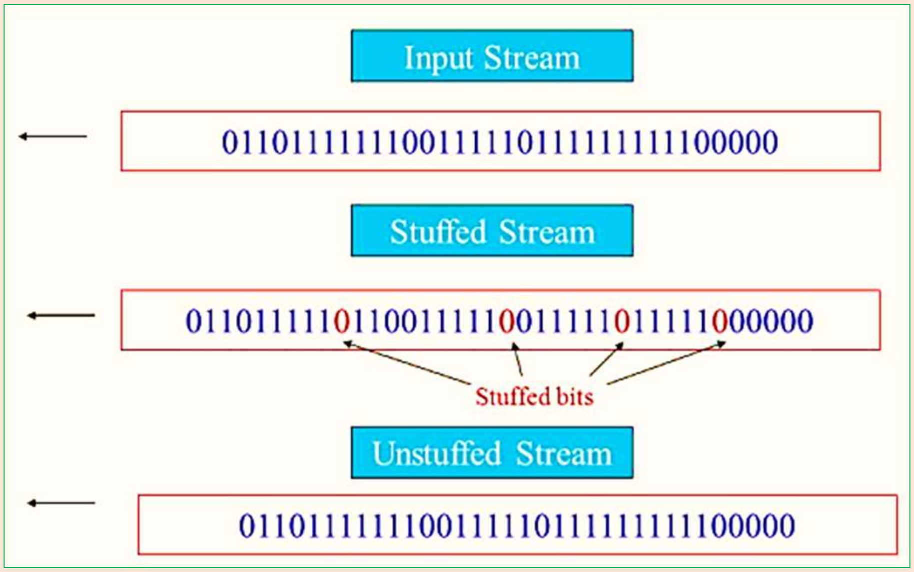
- This byte is usually called the escape character (ESC) which has a predefined bit pattern.
- Whenever the receiver encounters the ESC character, it removes it from the data section and treats the next character as data, not a delimiting flag.
- If the text contains one or more escape characters followed by a flag, the receiver removes the escape character, but keeps the flag, which is incorrectly interpreted as the end of the frame.
- To solve this problem, the escape characters that are part of the text must also be marked by another escape character.
- In other words, if the escape character is part of the text, an extra one is added to show that the second one is part of the text.





C. Bit Stuffing

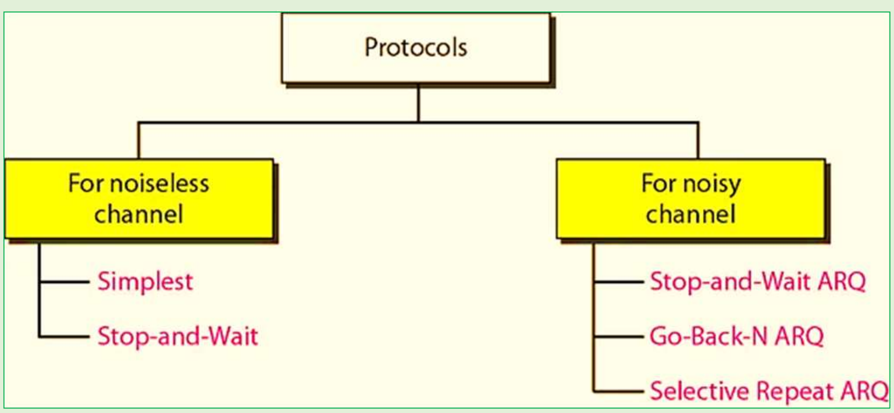
- Most protocols use a special 8-bit pattern flag 01111110 as the delimiter to define the beginning and the end of the frame
- That is, if the flag pattern appears in the data, we need to somehow inform the receiver that this is not the end of the frame.
- We do this by stuffing 1 single bit (instead of 1 byte) to prevent the pattern from looking like a flag. The strategy is called bit stuffing.
- In bit stuffing, if a 0 and five consecutive 1 bits are encountered, an extra 0 is added.
- This extra stuffed bit is eventually removed from the data by the receiver.



Prepared By Mr. EBIN PM, Chandigarh University, Punjab EDULINE 11

2. ERROR CONTROL

- Error control is both error detection and error correction.
- Any time an error is detected in an exchange, specified frames are retransmitted. This process is called automatic repeat request (ARQ).



Prepared By Mr. EBIN PM, Chandigarh University, Punjab EDULINE 12

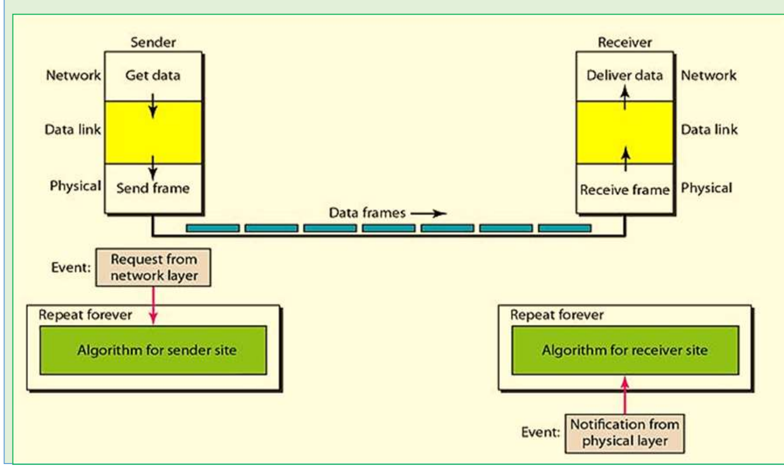
❖NOISELESS CHANNELS

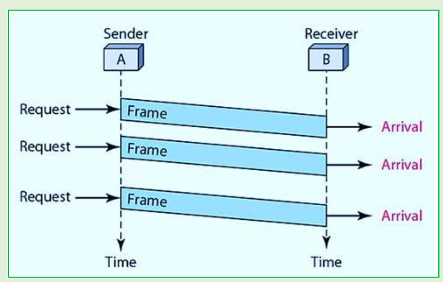
1. Simplest Protocol

- No flow control and error control
- It is a unidirectional protocol in which data frames are traveling in only one direction-from the sender to receiver.
- There is no need for flow control in this scheme. The data link layer at the sender site gets data from its network layer, makes a frame out of the data, and sends it.
- The data link layer at the receiver site receives a frame from its physical layer, extracts data from the frame, and delivers the data to its network layer.

Prepared By Mr. EBIN PM, Chandigarh University, Punjab EDULINE 13

- The data link layers of the sender and receiver provide transmission services for their network layers. The data link layers use the services provided by their physical layers (such as signaling, multiplexing, and so on) for the physical transmission of bits.





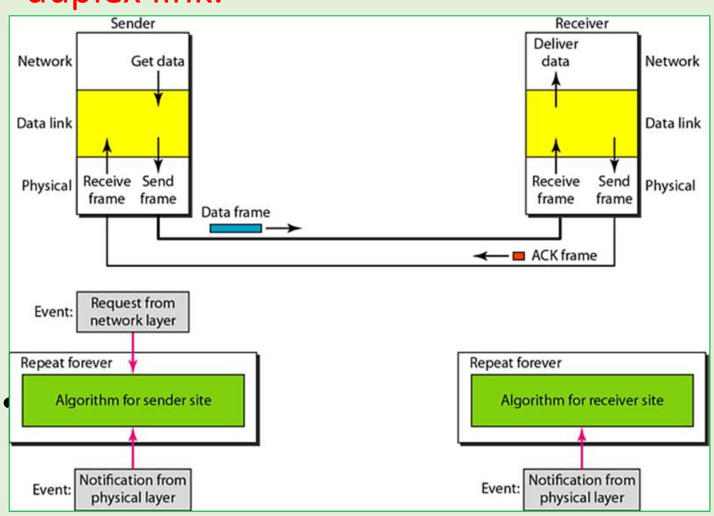
Flow Diagram

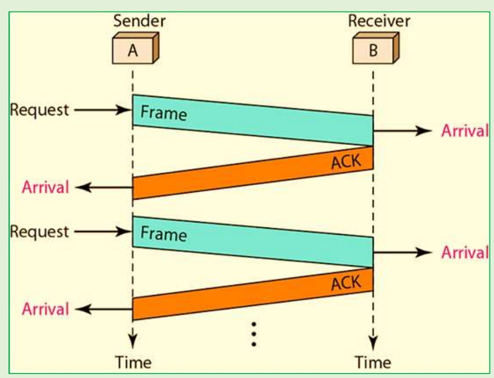
Prepared By Mr. EBIN PM, Chandigarh University, Punjab EDULINE 14

2. Stop-and-Wait Protocol

- To prevent the receiver from becoming overwhelmed with frames, we somehow need to tell the sender to slow down. There must be feedback from the receiver to the sender.
- In Stop-and-Wait Protocol , the sender sends one frame, stops until it receives confirmation from the receiver (okay to go ahead), and then sends the next frame.
- We still have unidirectional communication for data frames, but auxiliary ACK frames (simple tokens of acknowledgment) travel from the other direction.
- We add flow control to our previous protocol. After a frame is sent, the algorithm must ignore another network layer request until that frame is acknowledged.

- At any time, there is either one data frame on the forward channel or one ACK frame on the reverse channel. We therefore need a half-duplex link.





Flow Diagram

❖NOISY CHANNELS

1. Stop-and-Wait Automatic Repeat Request

- It adds a simple error control mechanism to the Stop-and-Wait Protocol
- To detect and correct corrupted frames, we need to add redundancy bits to our data frame .When the frame arrives at the receiver site, it is checked and if it is corrupted, it is silently discarded. The detection of errors in this protocol is manifested by the silence of the receiver.
- Lost frames are more difficult to handle than corrupted ones. The received frame could be the correct one, or a duplicate, or a frame out of order. The solution is to number the frames. When the receiver receives a data frame that is out of order, this means that frames were either lost or duplicated.

- The lost frames need to be resent in this protocol. the sender keeps a copy of the sent frame. At the same time, it starts a timer.
- If the timer expires and there is no ACK for the sent frame, the frame is resent, the copy is held, and the timer is restarted.
- Since the protocol uses the stop-and-wait mechanism, there is only one specific frame that needs an ACK even though several copies of the same frame can be in the network.
- Since an ACK frame can also be corrupted and lost, it too needs redundancy bits and a sequence number. The ACK frame for this protocol has a sequence number field. In this protocol, the sender simply discards a corrupted ACK frame or ignores an out-of-order one

➢Sequence Numbers

- The frames need to be numbered. This is done by using sequence numbers. A field is added to the data frame to hold the sequence number of that frame.
- One important consideration is the range of the sequence numbers. Since we want to minimize the frame size, we look for the smallest range.
- Assume we have used **x** as a sequence number; we only need to use **x + 1** after that. There is no need for **x + 2**.
- To show this, assume that the sender has sent the frame numbered x. Three things can happen.

1. The frame arrives safe and sound at the receiver site; the receiver sends an acknowledgment. The acknowledgment arrives at the sender site, causing the sender to send the next frame numbered x + 1.

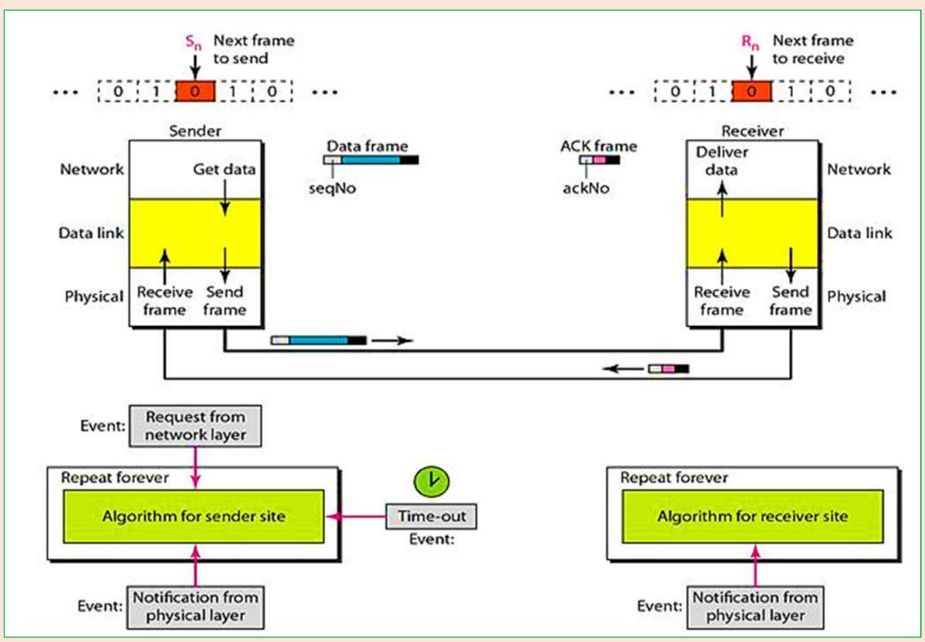
2. The frame arrives safe and sound at the receiver site; the receiver sends an acknowledgment, but the acknowledgment is corrupted or lost. The sender resends the frame (numbered x) after the time-out. Note that the frame here is a duplicate. The receiver can recognize this fact because it expects frame x + I but frame x was received.

3. The frame is corrupted or never arrives at the receiver site; the sender resends the frame (numbered x) after the time-out.

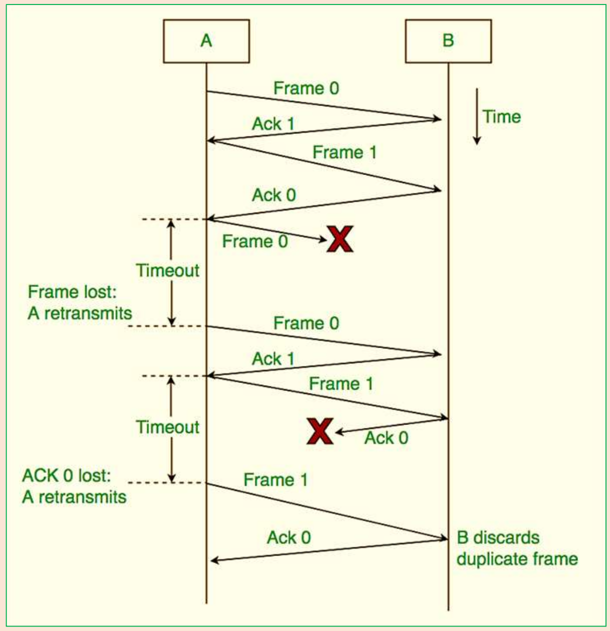
➢We can see that there is a need for sequence numbers x and x + I.

➢Acknowledgment Numbers

- The acknowledgment numbers always announce the sequence number of the next frame expected by the receiver.
- For example, if frame 0 has arrived safe and sound, the receiver sends an ACK frame with acknowledgment 1 (meaning frame 1 is expected next).



Flow diagram



2. Go-Back-N Automatic Repeat Request

- In this protocol we can send several frames before receiving acknowledgments.
- We keep a copy of these frames until the acknowledgments arrive.

➢ **Sequence Numbers**

- Frames from a sending station are numbered sequentially. Because we need to include the sequence number of each frame in the header, we need to set a limit.
- If the header of the frame allows m bits for the sequence number, the sequence numbers range from **0 to 2^m - 1**. For example, if m is 4, the only sequence numbers are 0 – 15. However, we can repeat the sequence. So the sequence numbers are

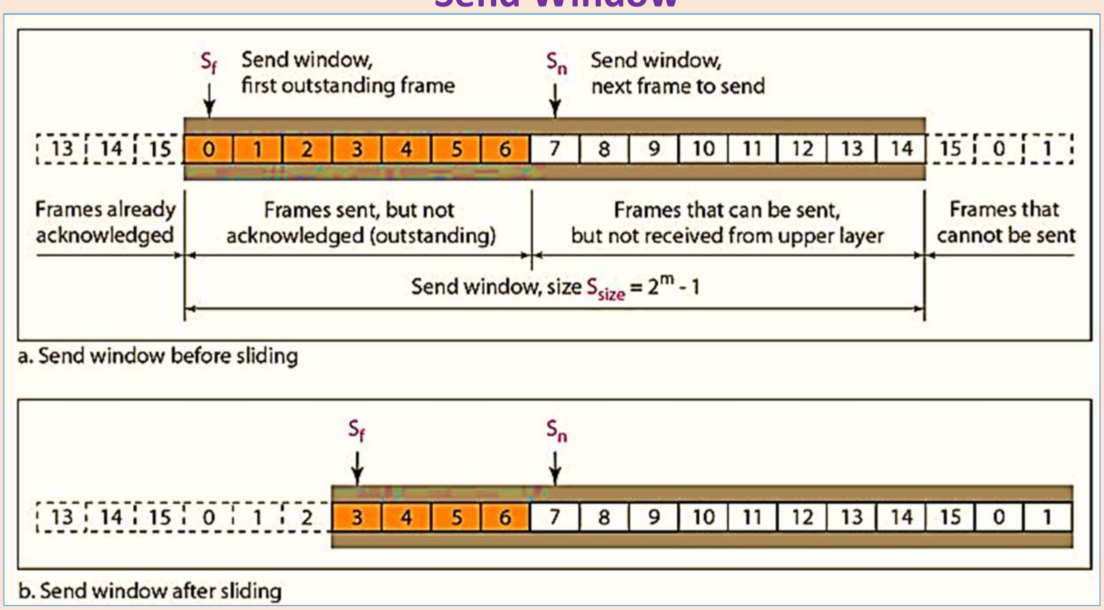
0, 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,0,1,2,3,4,5,6,7,8,9,10, 11, ...

- In the Go-Back-N Protocol, the sequence numbers are modulo2^m , where m is the size of the sequence number field in bits.

➢Sliding Window

- The sliding window is an abstract concept that defines the range of sequence numbers .
- The sender and receiver need to deal with only part of the possible sequence numbers.
- The window at any time divides the possible sequence numbers into four regions

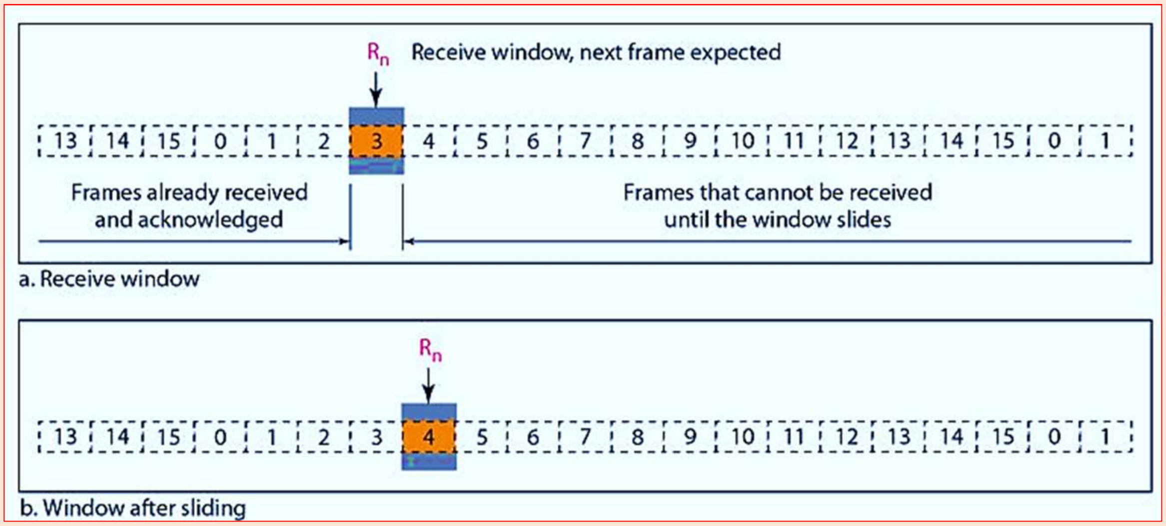
Send Window



- The window itself is an abstraction; three variables define its size and location at any time.
- **Sf**(send window, the first outstanding frame)
- **Sn** (send window, the next frame to be sent)
- **Ssize** (send window, size).
- The variable Sf defines the sequence number of the first (oldest) outstanding frame.
- The variable Sn holds the sequence number that will be assigned to the next frame to be sent.
- Finally, the variable Ssize defines the size of the window, which is fixed in our protocol.

- The acknowledgments in this protocol are cumulative, meaning that more than one frame can be acknowledged by an ACK frame.
- In Figure b, frames 0, I, and 2 are acknowledged, so the window has slide to the right three slots.
- The receive window makes sure that the correct data frames are received and that the correct acknowledgments are sent.
- The size of the receive window is always 1.
- The receiver is always looking for the arrival of a specific frame.
- Any frame arriving out of order is discarded and needs to be resent.

Receive Window



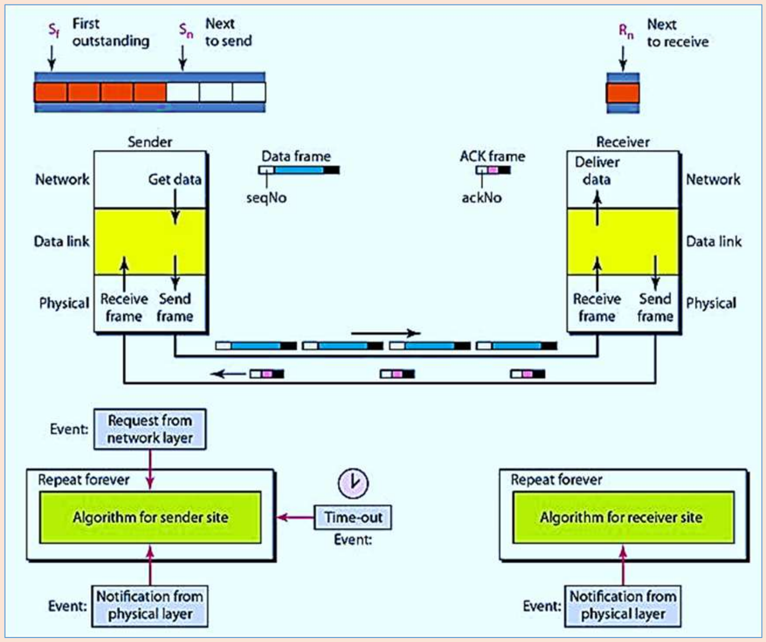
- we need only one variable Rn (receive window, next frame expected) to define this abstraction.
- The sequence numbers to the left of the window belong to the frames already received and acknowledged
- The sequence numbers to the right of this window define the frames that cannot be received. Any received frame with a sequence number in these two regions is discarded.
- Only a frame with a sequence number matching the value of Rn is accepted and acknowledged.
- The receive window also slides, but only one slot at a time. When a correct frame is received (and a frame is received only one at a time), the window slides.

➢Acknowledgment

- The receiver sends a positive acknowledgment if a frame has arrived safe and sound and in order.
- If a frame is damaged or is received out of order, the receiver is silent and will discard all subsequent frames until it receives the one it is expecting.
- The silence of the receiver causes the timer of the unacknowledged frame at the sender site to expire. This, in turn, causes the sender to go back and resend all frames, beginning with the one with the expired timer.
- The receiver does not have to acknowledge each frame received. It can send one cumulative acknowledgment for several frames

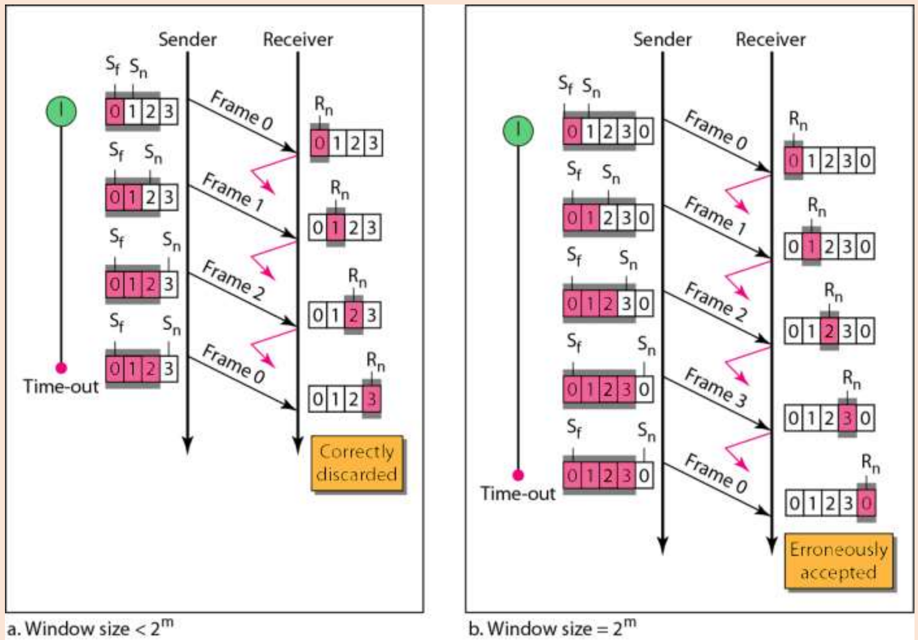
➢Resending a Frame

- When the timer expires, the sender resends all outstanding frames.
- For example, suppose the sender has already sent frame 6, but the timer for frame 3 expires. This means that frame 3 has not been acknowledged; the sender goes back and sends frames 3, 4,5, and 6 again.
- That is why the protocol is called Go-Back-N ARQ.
- In Go-Back-N ARQ, the size of the send window must be **less than 2^m** and the size of the receiver window is always **1**



➢Send Window Size

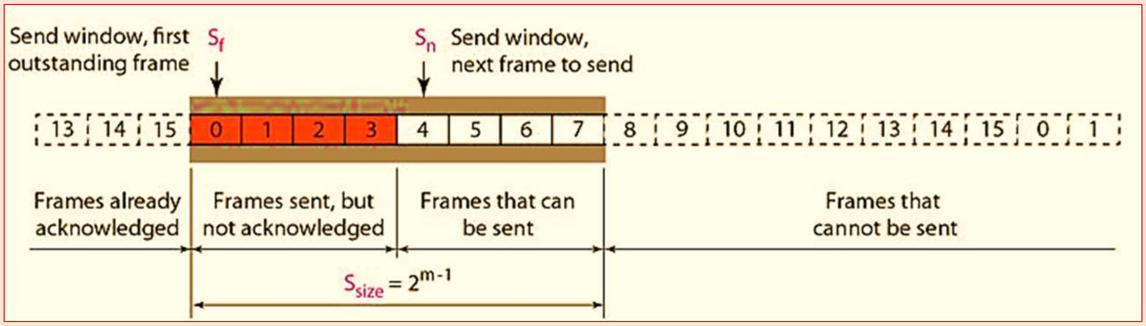
- Suppose m =2, which means the size of the window can be 2^m - 1, or 3.
- If the size of the window is 3 (less than 2^2) and all three acknowledgments are lost, the frame timer expires and all three frames are resent.
- The receiver is now expecting frame 3, not frame 0, so the duplicate frame is correctly discarded.
- On the other hand, if the size of the window is 4 (equal to 2^2) and all acknowledgments are lost, the sender will send a duplicate of frame 0.
- However, this time the window of the receiver expects to receive frame 0, so it accepts frame 0, not as a duplicate, but as the first frame in the next cycle. This is an error.



3. Selective Repeat Automatic Repeat Request

- Go-Back-N ARQ resending of multiple frames.
- For noisy links, there is another mechanism that does not resend N frames when just one frame is damaged; only the damaged frame is resent.
- This mechanism is called Selective Repeat ARQ. It is more efficient for noisy links.
- The Selective Repeat Protocol also uses two windows: a send window and a receive window
- The size of the send window is much smaller; it is 2^{m-1}
- The receive window is the same size as the send window

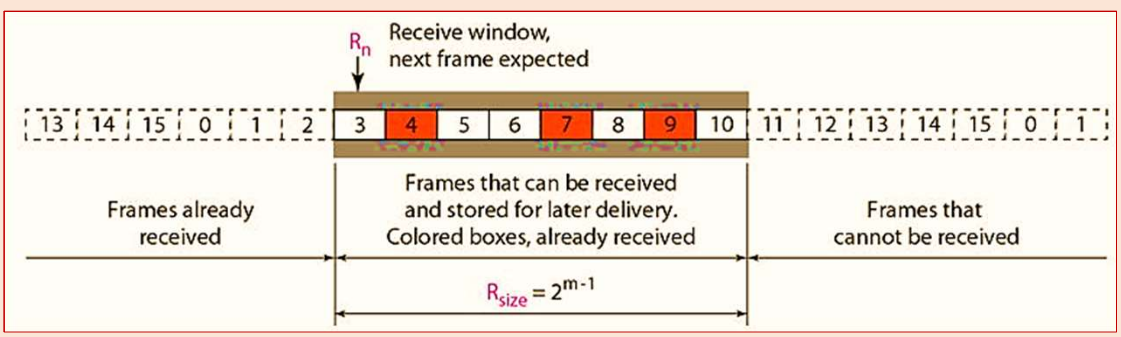
- If m = 4, the sequence numbers go from 0 to 15, but the size of the window is just 8 (it is 15 in the Go-Back-N Protocol).



- The Selective Repeat Protocol allows as many frames as the size of the receive window to arrive out of order and be kept until there is a set of in-order frames to be delivered to the network layer.

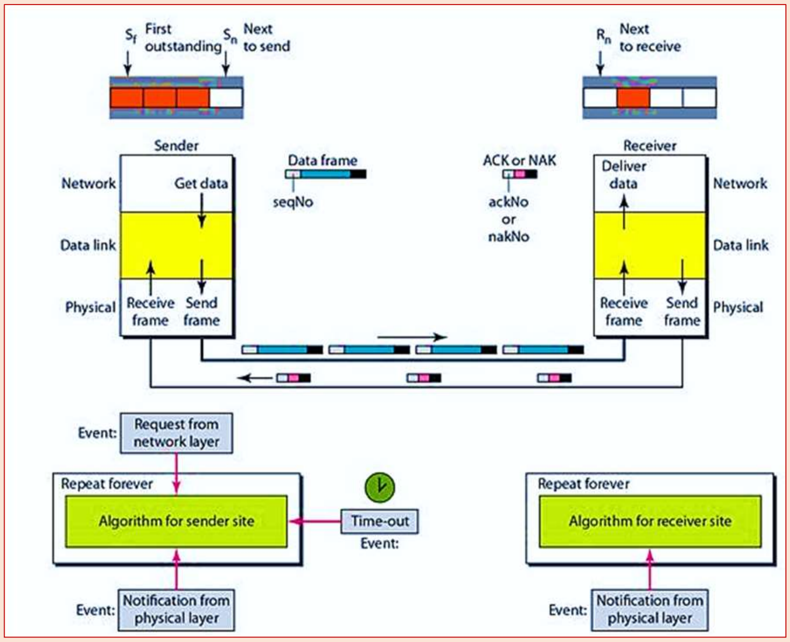
Prepared By Mr. EBIN PM, Chandigarh University, Punjab EDULINE 37

- Because the sizes of the send window and receive window are the same, all the frames in the send frame can arrive out of order and be stored until they can be delivered

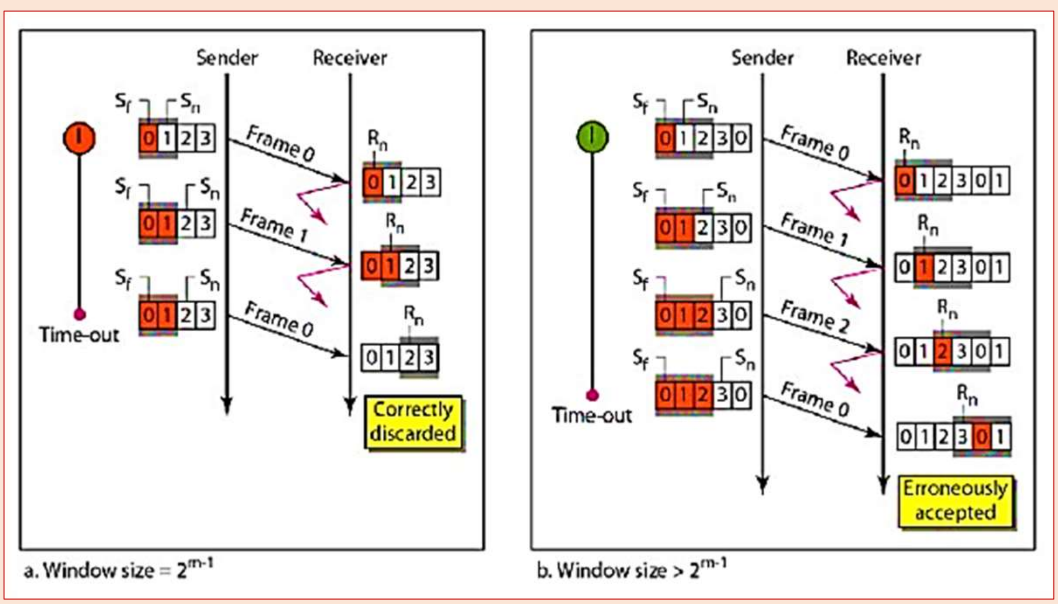


- Those slots inside the window that are colored define frames that have arrived out of order and are waiting for their neighbors to arrive before delivery to the network layer.

Prepared By Mr. EBIN PM, Chandigarh University, Punjab EDULINE 38



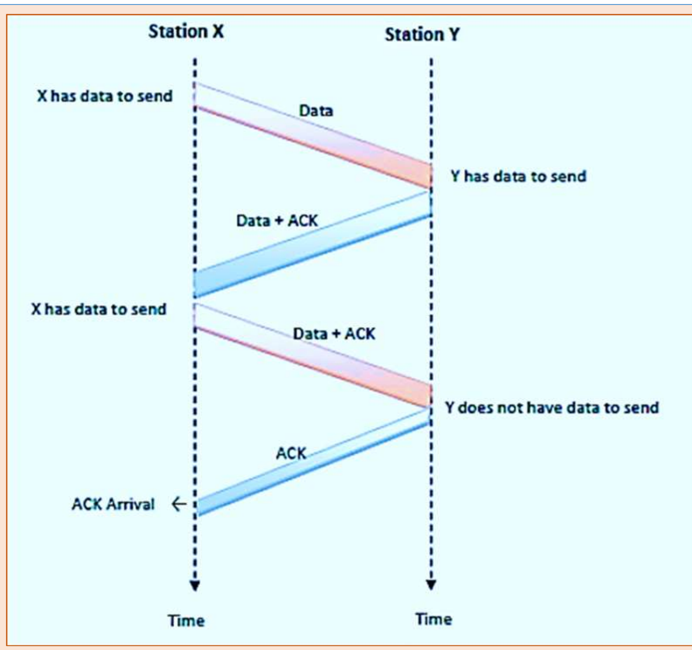
Prepared By Mr. EBIN PM, Chandigarh University, Punjab EDULINE 39



Prepared By Mr. EBIN PM, Chandigarh University, Punjab EDULINE 40

❖Piggybacking

- Communications are mostly full – duplex in nature, i.e. data transmission occurs in both directions
- In reliable full - duplex data transmission, the technique of hooking up acknowledgments onto outgoing data frames is called piggybacking.
- Piggybacking is used to improve the efficiency of the bidirectional protocols
- When a frame is carrying data from A to B, it can also carry control information about arrived (or lost) frames from B; when a frame is carrying data from B to A, it can also carry control information about the arrived (or lost) frames from A.



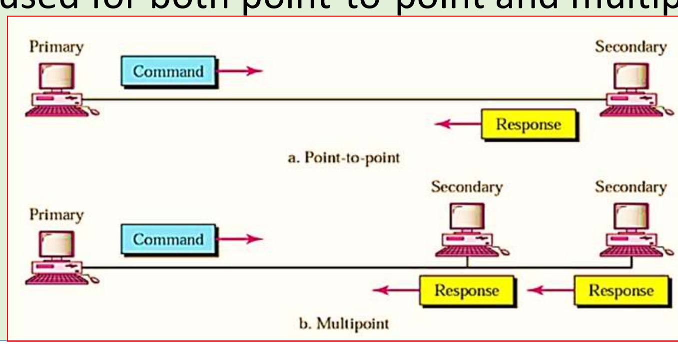
HDLC

- High-level Data Link Control (HDLC) is a bit-oriented protocol for communication over point-to-point and multipoint links.
- It implements the ARQ mechanisms
- HDLC provides two common transfer modes that can be used in different configurations:

1. Normal Response Mode (NRM)
2. Asynchronous Balanced Mode (ABM).

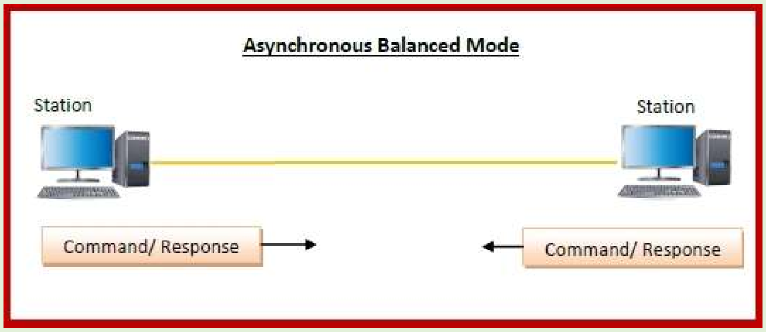
1. Normal Response Mode

- 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



2. Asynchronous Balanced Mode

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



❖Frames

- HDLC defines three types of frames:

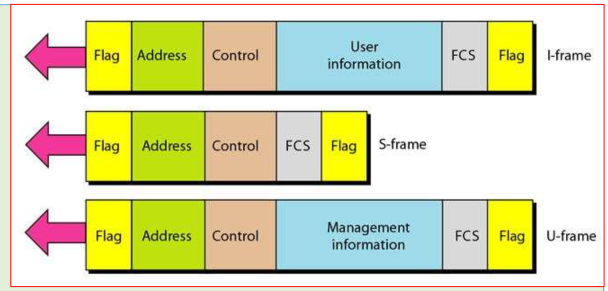
➢ **Information frames (I-frames)**

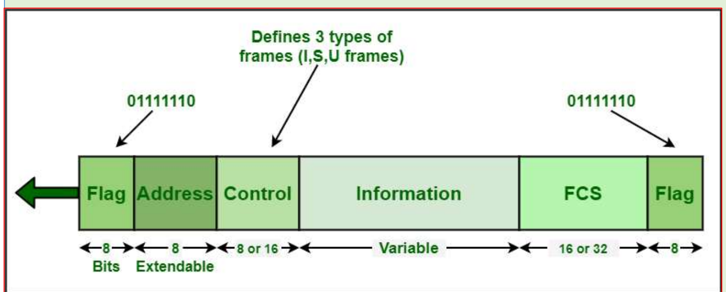
➢ **Supervisory frames (S-frames)**

➢ **Unnumbered frames (V-frames)**

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

❖Frame Format





Basic Frame Structure

Prepared By Mr. EBIN PM, Chandigarh University, Punjab EDULINE 47

- Flag field - The bit pattern 01111110, that identifies both the beginning and the end of a frame
- Address field - 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.
- Control field - The control field is a 1 or 2 byte segment of the frame used for flow and error control.
- Information field - The information field contains the user's data from the network layer or management information
- FCS field - The frame check sequence (FCS) is the HDLC error detection field. It can contain either a 2- or 4-byte ITU-T CRC

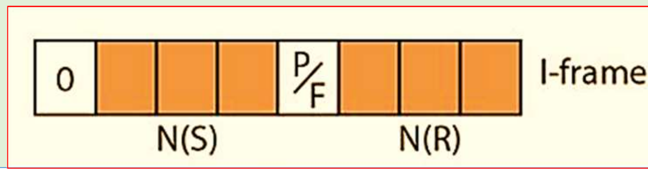
Prepared By Mr. EBIN PM, Chandigarh University, Punjab EDULINE 48

❖Control Field

- The control field determines the type of frame

➢Control Field for I-Frames

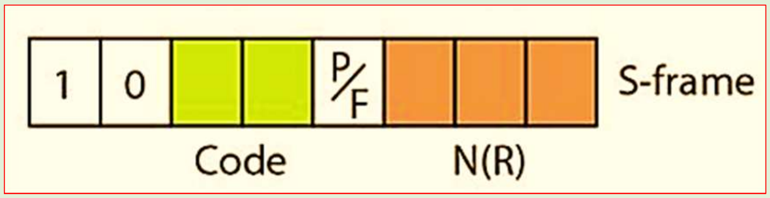
- I-frames are designed to carry user data from the network layer.
- They can include flow and error control information (piggybacking).
- The first bit defines the type. If the first bit of the control field is 0, this means the frame is an I-frame.
- The next 3 bits, called N(S),define the sequence 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 the P/F bit.
- 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

- 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



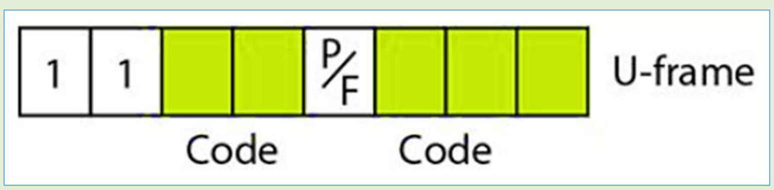
- The 2 bits called code is used to define the type of S-frame itself. With 2 bits, we can have four types of S-frames.

1. **Receive ready (RR)** - If the value of the code subfield is 00, it is an RR S-frame. This kind of frame acknowledges the receipt of a safe and sound frame or group of frames. In this case, the value N(R) field defines the acknowledgment number.
2. **Receive not ready (RNR)** - If the value of the code subfield is 10, it is an RNR S-frame. It announces that the receiver is busy and cannot receive more frames. It acts as a kind of congestion control mechanism by asking the sender to slow down. The value of N(R) is the acknowledgment number.

3. **Reject (REJ)** - If the value of the code subfield is 01, it is a REJ S-frame. This is a NAK frame. It is a NAK that can be used in Go-Back-N ARQ to improve the efficiency of the process by informing the sender, before the sender time expires, that the last frame is lost or damaged. The value of N(R) is the negative acknowledgment number.
4. **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. Note that the HDLC Protocol uses the term selective reject instead of selective repeat. 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.



U-frames Control Command and Response

Code	Command	Response	Meaning
00 001	SNRM		Set normal response mode
11 011	SNRME		Set normal response mode, extended
11 100	SABM	DM	Set asynchronous balanced mode or disconnect mode
11110	SABME		Set asynchronous balanced mode, extended
00 000	UI	UI	Unnumbered information
00 110		UA	Unnumbered acknowledgment
00 010	DISC	RD	Disconnect or request disconnect
10 000	SIM	RIM	Set initialization mode or request information mode
00 100	UP		Unnumbered poll
11 001	RSET		Reset
11 101	XID	XID	Exchange ID
10 001	FRMR	FRMR	Frame reject

MODULE 2

CHAPTER 2 – ERROR DETECTION & CORRECTION

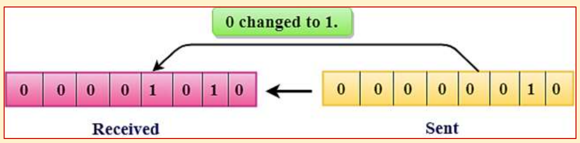
TYPES OF ERRORS

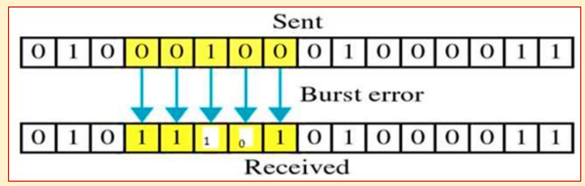
➢Single-Bit Error

- 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.

➢Burst Error

- The term burst error means that 2 or more bits in the data unit have changed from 1 to 0 or from 0 to 1.





❖REDUNDANCY

- The central concept in detecting or correcting errors is redundancy.
- To be able to detect or correct errors, we need to send some extra bits with our data.
- These redundant bits are added by the sender and removed by the receiver.
- Their presence allows the receiver to detect or correct corrupted bits.
- To detect or correct errors, we need to send extra (redundant) bits with data.

❖Detection Versus Correction

- In error detection, we are looking only to see if any error has occurred. The answer is a simple yes or no. We are not even interested in the number of errors.
- In error correction, we need to know the exact number of bits that are corrupted and more importantly, their location in the message.
- The number of the errors and the size of the message are important factors.
- If we need to correct one single error in an 8-bit data unit, we need to consider eight possible error locations; if we need to correct two errors in a data unit of the same size, we need to consider 28 possibilities.

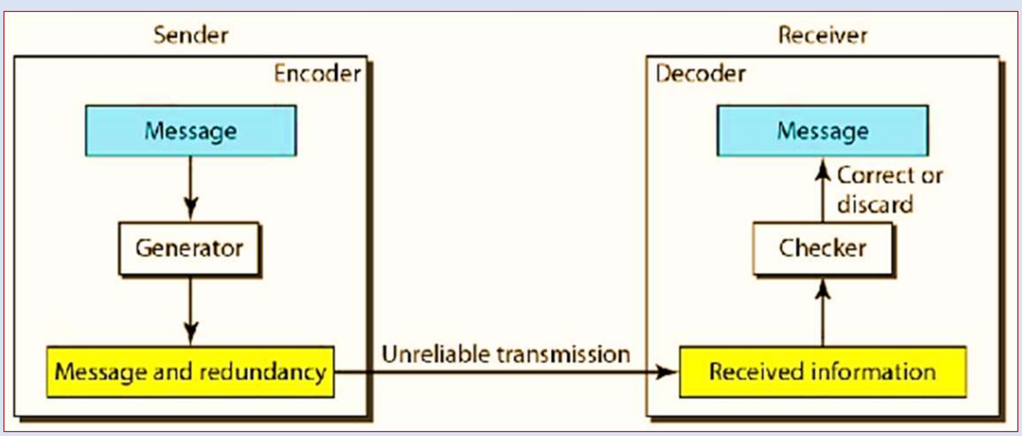
❖Forward Error Correction Versus Retransmission

- There are two main methods of error correction.
- Forward error correction is the process in which the receiver tries to guess the message by using redundant bits. This is possible if the number of errors is small.
- Correction by retransmission is a technique in which the receiver detects the occurrence of an error and asks the sender to resend the message.
- Resending is repeated until a message arrives that the receiver believes is error-free (usually, not all errors can be detected).

CODING

- Redundancy is achieved through various coding schemes.
- The sender adds redundant bits through a process that creates a relationship between the redundant bits and the actual data bits.
- The receiver checks the relationships between the two sets of bits to detect or correct the errors.
- The ratio of redundant bits to the data bits and the robustness of the process are important factors in any coding scheme.
- We can divide coding schemes into two broad categories: block coding and convolution coding.

The structure of encoder and decoder

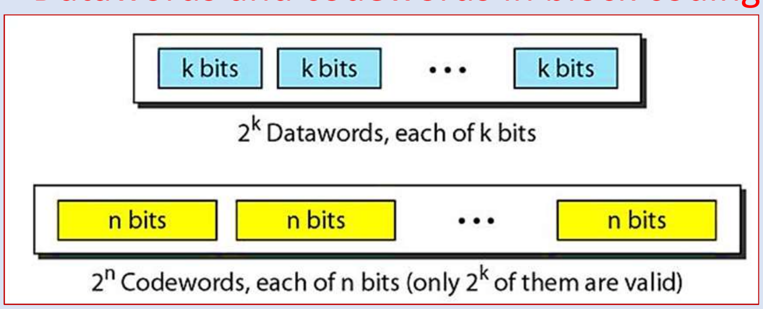


❖BLOCK CODING

- We divide our message into blocks, each of **k** bits, called **data words.**
- We add **r** redundant bits to each block to make the length **n = k + r.**
- The resulting n-bit blocks are called **code words**.
- we have a set of data words, each of size k, and a set of code words, each of size of n. With k bits, we can create a combination of 2k data words; with n bits, we can create a combination of 2n code words.
- Since n > k, the number of possible code words is larger than the number of possible data words.
- The block coding process is one-to-one; the same data word is always encoded as the same code word.

Prepared By Mr. EBIN PM, Chandigarh University, Punjab EDULINE 63

Datawords and codewords in block coding



➢Error Detection

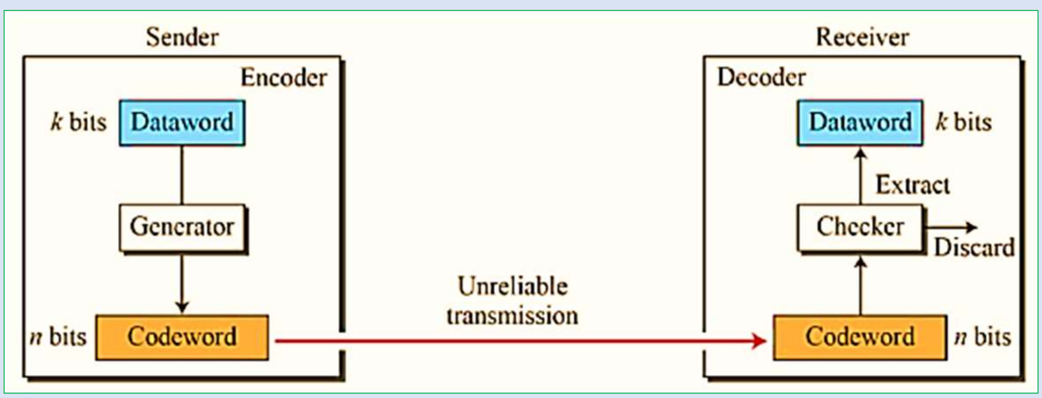
- If the following two conditions are met, the receiver can detect a change in the original codeword.

1. The receiver has (or can find) a list of valid codewords.
2. The original codeword has changed to an invalid one.

Prepared By Mr. EBIN PM, Chandigarh University, Punjab EDULINE 64

- The sender creates codewords out of datawords by using a generator that applies the rules and procedures of encoding .
- 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 codeword is not valid, it is discarded.
- If the codeword is corrupted during transmission but the received word still matches a valid codeword, the error remains undetected.
- This type of coding can detect only single errors. Two or more errors may remain undetected.

Process of error detection in block coding



Let us assume that k =2 and n =3. Table shows the list of datawords and codewords.

Datawords	Codewords
00	000
01	011
10	101
11	110

- Assume the sender encodes the dataword 01 as 011 and sends it to the receiver. Consider the following cases:

1. The receiver receives 011. It is a valid codeword. The receiver extracts the dataword 01 from it.
2. The codeword is corrupted during transmission, and 111 is received (the leftmost bit is corrupted). This is not a valid codeword and is discarded.

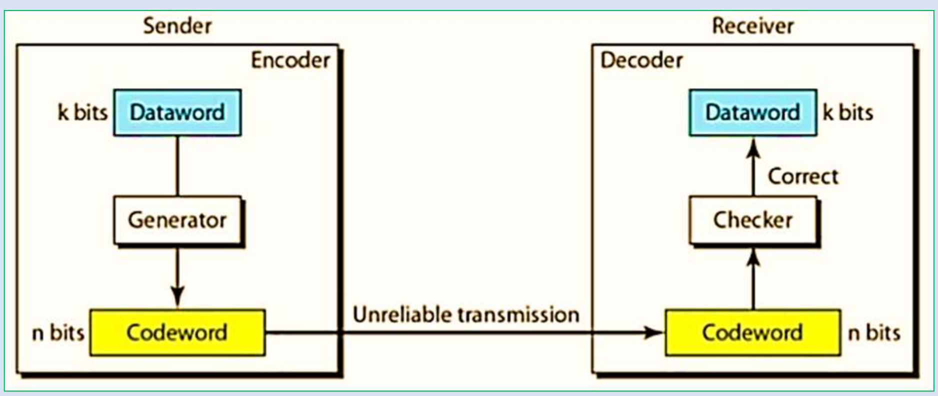
3. The codeword is corrupted during transmission, and 000 is received (the right two bits are corrupted). This is a valid codeword. The receiver incorrectly extracts the dataword 00. Two corrupted bits have made the error undetectable.

➢ An error-detecting code can detect only the types of errors for which it is designed; other types of errors may remain undetected.

➢Error Correction

- Error correction is much more difficult than error detection.
- In error detection, the receiver needs to know only that the received codeword is invalid
- In error correction the receiver needs to find (or guess) the original codeword sent.
- We can say that we need more redundant bits for error correction than for error detection.
- Following figure shows the role of block coding in error correction. The idea is the same as error detection but the checker functions are much more complex.

Structure of encoder and decoder in error correction



- Assume the dataword is 01.
- The sender consults the table (or uses an algorithm) to create the codeword 01011.
- The codeword is corrupted during transmission, and 01001 is received (error in the second bit from the right).
- First, the receiver finds that the received codeword is not in the table. This means an error has occurred. (Detection must come before correction.)
- The receiver, assuming that there is only 1 bit corrupted, uses the following strategy to guess the correct dataword.

A code for error correction

Dataword	*Codeword*
00	00000
01	01011
10	10101
11	11110

1. Comparing the received codeword with the first codeword in the table (01001 versus 00000), the receiver decides that the first codeword is not the one that was sent because there are two different bits.
2. By the same reasoning, the original codeword cannot be the third or fourth one in the table.
3. The original codeword must be the second one in the table because this is the only one that differs from the received codeword by 1 bit. The receiver replaces 01001 with 01011 and consults the table to find the dataword 01.

❖Hamming Distance

- The Hamming distance between two words (of the same size) is the number of differences between the corresponding bits.
- The Hamming distance between two words x and y is represented as **d(x, y)**
- The Hamming distance can easily be found if we apply the XOR operation (⊕) on the two words and count the number of 1s in the result.
- The Hamming distance is a value greater than zero.
- **Eg:** The Hamming distance d(000, 011) is 2 because 000 ⊕ 011 is 011 (two 1s).
- **Eg:** The Hamming distance d(10101, 11110) is 3 because 10101 ⊕11110 is 01011 (three 1s).

- The **minimum Hamming distance** is the smallest Hamming distance between all possible pairs in a set of words.
- We use d_{min} to define the minimum Hamming distance in a coding scheme.
- Eg: We first find all the Hamming distances.

  d(00000, 01011) = 3

  d(01011, 10101) = 4

  d(00000, 10101) = 3

  d(01011, 11110) = 3

  d(00000, 11110) = 4

  The d_{min} in this case is **3**

Inputs		Output
A	B	X = A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

- Any coding scheme needs to have at least three parameters:
  - The codeword size n
  - The dataword size k
  - The minimum Hamming distance d_{min}.
- A coding scheme C is written as C(n, k) with a separate expression for d_{min}

Eg: **C(5, 2) with d_{min}= 3.**

- The Hamming distance between the received codeword and the sent codeword is the number of bits that are corrupted during transmission. For example, if the codeword 00000 is sent and 01101 is received, 3 bits are in error and the Hamming distance between the two is d(00000, 01101) =3

Prepared By Mr. EBIN PM, Chandigarh University, Punjab EDULINE 75

- To guarantee the detection of up to ***s*** errors in all cases, the minimum Hamming distance in a block code must be $d_{min} = s + 1$
- To guarantee correction of up to **t** errors in all cases, the minimum Hamming distance in a block code must be $d_{min} = 2t + 1$

❖Parity bits

- A parity bit is a bit appended to a data of binary bits to ensure that the total number of 1's in the data is even or odd.
- Parity bits are used for error detection.
- There are two types of parity bits:

  Even parity bit and **Odd parity bit**

Prepared By Mr. EBIN PM, Chandigarh University, Punjab EDULINE 76

Even parity bit:

- In the case of even parity, for a given set of bits, the number of 1's are counted. If that count is odd, the parity bit value is set to 1, making the total count of occurrences of 1's an even number. If the total number of 1's in a given set of bits is already even, the parity bit's value is 0.

Odd Parity bit :

- In the case of odd parity, for a given set of bits, the number of 1's are counted. If that count is even, the parity bit value is set to 1, making the total count of occurrences of 1's an odd number. If the total number of 1's in a given set of bits is already odd, the parity bit's value is 0.

LINEAR BLOCK CODE

- In a linear block code, the exclusive OR (XOR) of any two valid codewords creates another valid codeword

1. Simple Parity-Check Code

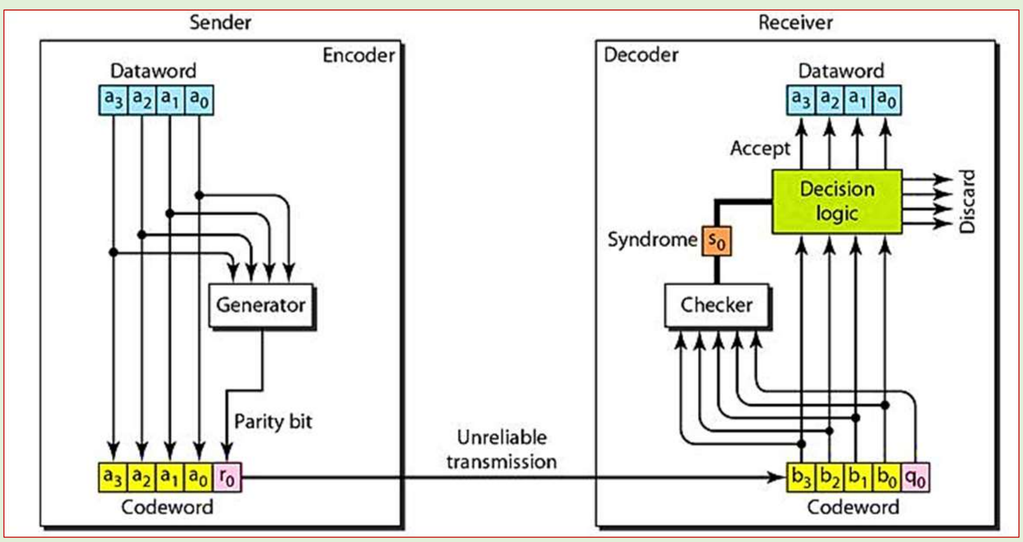
- A simple parity-check code is a single-bit error-detecting code
- In this code, a k-bit dataword is changed to an n-bit codeword where **n = k + 1**. The extra bit, called the **parity bit**, is selected to make the total number of 1s in the codeword even.
- The minimum Hamming distance for this category is d_{min} **=2** which means that the code is a single-bit error-detecting code; it cannot correct any error.

Simple parity-check code C(5, 4)

Datawords	Codewords	Datawords	Codewords
0000	00000	1000	10001
0001	00011	1001	10010
0010	00101	1010	10100
0011	00110	1011	10111
0100	01001	1100	11000
0101	01010	1101	11011
0110	01100	1110	11101
0111	01111	1111	11110

Prepared By Mr. EBIN PM, Chandigarh University, Punjab EDULINE 79

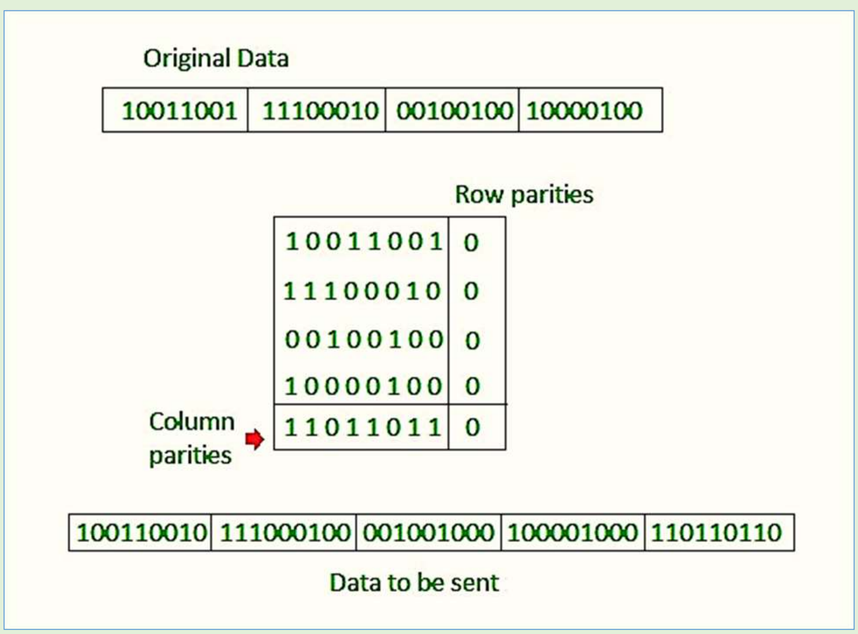
Encoder and decoder for simple parity-check code



Prepared By Mr. EBIN PM, Chandigarh University, Punjab EDULINE 80

2. Two-dimensional parity check

- In this method, the dataword is organized in a table (rows and columns)
- For each row and each column, 1 parity-check bit is calculated. The whole table is then sent to the receiver, which finds the syndrome for each row and each column.
- The two-dimensional parity check can detect up to three errors that occur anywhere in the table.
- Errors affecting 4 bits may not be detected



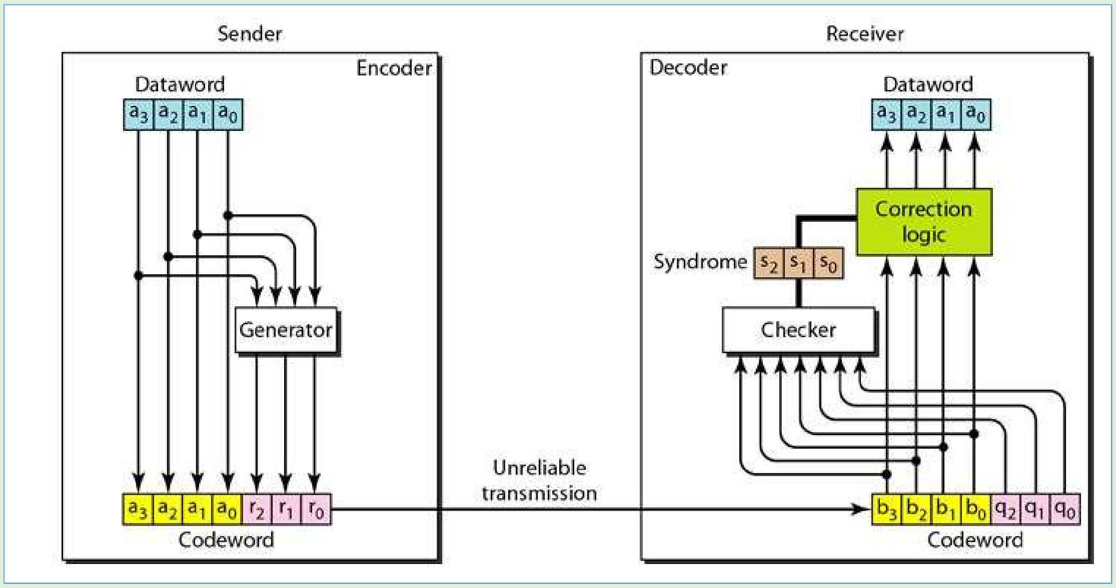
3. Hamming Codes

- Hamming codes are error-correcting codes
- These codes were originally designed with d_{min}= 3, which means that they can detect up to two errors or correct one single error.
- relationship between n and k in a Hamming code. We need to choose an integer m >= 3. The values of n and k are then calculated from mas n = 2^m -1 and k = n - m. The number of check bits r =m.
- For example, if m =3, then n =7 and k = 4. This is a Hamming code C(7, 4) with d_{min} =3.

Hamming code C(7, 4)

Datawords	*Codewords*	*Datawords*	*Codewords*
0000	0000000	1000	1000110
0001	0001101	1001	1001011
0010	0010111	1010	1010001
0011	0011010	1011	10111 00
0100	0100011	1100	1100101
0101	01011 10	1101	1101000
0110	0110100	1110	1110010
0111	0111001	1111	1111111

The structure of the encoder and decoder for a Hamming code



❖Hamming Code Structure

➢All the bit positions that are power of 2 are marked as parity bits (1,2,4,8,…..) and other bits are for data.

Eg: 7 bit hamming code

D7	D6	D5	P4	D3	P2	P1

Parity bit values :

Eg: 1101

D7	D6	D5	P4	D3	P2	P1
1	1	0	?	1	?	?

P1 : check 1 bit & skip 1bit (1,3,5,7,9,……)

P2 : check 2 bit & skip 2 bit (2,3,6,7,10,11,…)

P4 : check 4 bit & skip 4 bit (4,5,6,7,12,13,14,15,……)

D7	D6	D5	P4	D3	P2	P1
1	1	0		1		

P1 D3 D5 D7 ⟶ P1 101 ⟶ P1 = 0

P2 D3 D6 D7 ⟶ P2 111 ⟶ P2 =1

P4 D5 D6 D7 ⟶ P4 011 ⟶ P4 = 0

1	1	0	0	1	1	0

❖DETECTING & CORRECTING ERROR

Suppose the receiver receives the data **1011011**

D7	D6	D5	P4	D3	P2	P1
1	0	1	1	0	1	1

Step 1: analyze bit 1,3,5,7 (for P1) = P1 D3 D5 D7 = 1011

Odd number of 1s. So error exist and we put **P1=1**

Step 2: analyze bit 2,3,6,7 (for P2) = P2 D3 D6 D7 = 1001

even number of 1s. So no error and we put **P2=0**

Step 3: analyze bit 4,5,6,7 (for P4) = P4 D5 D6 D7 = 1101

Odd number of 1s. So error exist and we put **P4=1**

Correcting Error

- Error word E =

P4	P2	P1
1	0	1

- Decimal value of the error E = 5 which shows that the 5th bit is in error.
- So we write the correct word by simply inverting only the 5th bit

D7	D6	D5	P4	D3	P2	P1
1	0	0	1	0	1	1

CYCLIC CODES

- Cyclic codes are special linear block codes with one extra property.
- In a cyclic code, if a codeword is cyclically shifted (rotated), the result is another codeword.
- For example, if 1011000 is a codeword and we cyclically left-shift, then 0110001 is also a codeword
- cyclic codes have a very good performance in detecting single-bit errors, double errors, an odd number of errors, and burst errors.
- They can easily be implemented in hardware and software

❖Cyclic Redundancy Check

- Cyclic Redundancy Check (CRC) is used in networks such as LANs and WANs.
- It is an error detection method based on binary division.
- Here a sequence of redundant bits called CRC bits are appended to the end of data, so that the resulting data unit become exactly divisible by second predetermined binary number.
- At the destination side, the incoming data is divisible by the same number.
- If the remainder is 0, then the data is accepted , otherwise rejected.

➢CRC generation at sender side

Step 1 : Find the length of the divisor "L"

Step 2 : Append "L-1" bits to the original message

Step 3 : Perform binary division (XOR) Operation

Step 4 : Remainder of the division = CRC

Eg : Find the CRC for the data block 100100 with the divisor 1101?

- Here L=4 . So 3 zeros will be append to the original message.
- So the data block become **100100000**

➢ **How receiver detect error using CRC**

CHECK SUM

- It is an error detection method .
- The checksum is used in the Internet by several protocols
- The checksum is based on the concept of redundancy
- Suppose our data is a list of five 4-bit numbers that we want to send to a destination. In addition to sending these numbers, we send the sum of the numbers.
- For example, if the set of numbers is (7, 11, 12, 0, 6), we send (7, 11, 12,0,6,36), where 36 is the sum of the original numbers. The receiver adds the five numbers and compares the result with the sum.

- If the two are the same, the receiver assumes no error, accepts the five numbers, and discards the sum. Otherwise, there is an error somewhere and the data are not accepted.
- The Internet has been using a 16-bit checksum.