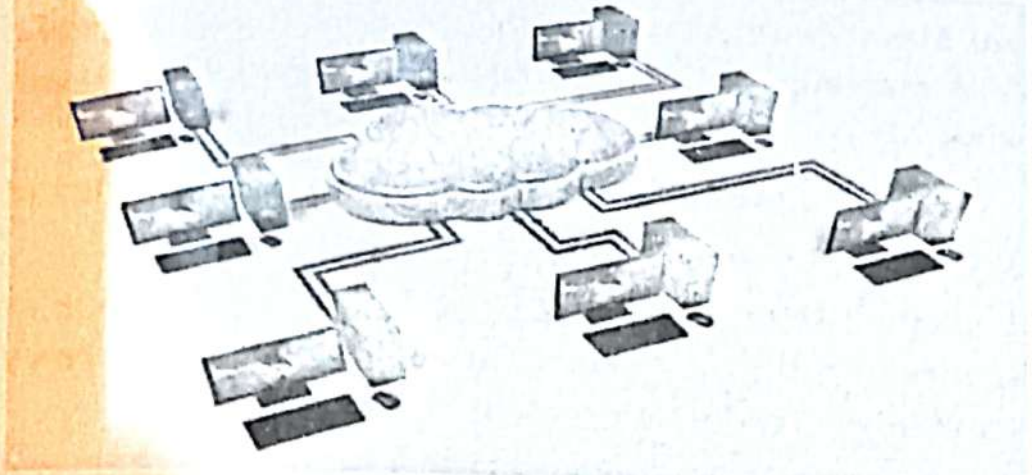


5

DATA LINK LAYER

INTRODUCTION

- Whenever an exchange of information between two devices is required an interconnecting transmission medium between two devices is required.
- This medium or line carries the electrical signals.
- At the same time a standard interface and a mechanism to convert binary digits into electrical signals and vice versa is required. Once the information is transmitted & received at other end it should be checked for correctness because errors may be introduced in the data bits due to electrical signal impairments like noise, linear distortions etc. So there should be some mechanism to control transmission errors.
- Some times it is possible that receiving device is not ready for the incoming bits and in that case may be some of the bits are lost.
- So a data flow control mechanism is also required. These error and flow control functions are known as **data link control**.
- The interconnecting transmission medium which carries electrical signals from one device to another is known as line or transmission line.
- It can be simply a copper wire, a cable, a group of paired wires or some thing else.
- The data link control is concerned with the transfer of data over a serial data link or line which can be either a point to point physical

circuit like twisted pair wire, coaxial cable, optical fiber or a radi based channel like a satellite link or a physical or logical link throug a switched network.

- In case of cables one or more amplifiers or repeaters are inserted a intervals to restore signals at original levels.
- One of the two transmission modes can be used, synchronous o asynchronous modes and they are based on either character oriente or bit oriented transmission control protocol.
- The data link control includes flow control, error control an associated link/line management functions.
- The specific set of rules and procedures for carrying data link contro functions is called **data link protocol**.
- The very first task of data link control is to format the user data a series of frames also known as configuring the frame.
- Then it looks for error and flow control and finally into link management. Let us discuss this layer in detail.

5.1 DATA LINK LAYER DESIGN ISSUE

- The data link layer transforms the physical layer, a raw transmission facility, to a link responsible for node-to-node (hop-to-hop) communication. Specific responsibilities of the data link layer include *framing, addressing, flow control, error control,* and *media access control.*
- 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.
- If the rate at which the data are absorbed by the receiver is less than the rate at which data are produced in the sender, the data link layer imposes a flow control mechanism to avoid overwhelming the receiver.
- The data link layer also adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged, duplicate, or lost frames.
- When two or more devices are connected to the same link, data link layer protocols are necessary to determine which device has control over the link at any given time.

- Assume a single link from A to B. Assume service provided by physical layer is that it will deliver bits to B in the order in which they are sent. So what does data link layer have to do ? The Data Link layer performs a number of separate duties, ***including packetizing, addressing, error control, flow control, and access control*** *as shown in Fig. 5.1*

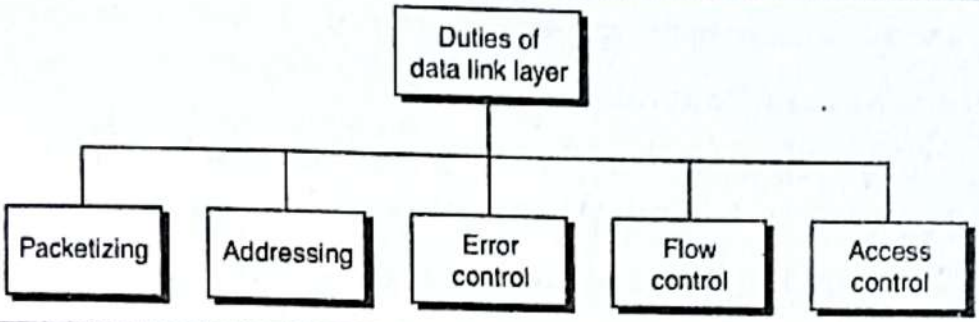


Fig. 5.1 Functions of Data link layer

- **Packetizing :**
  - The packet coming form upper layer must be encapsulated in the appropriate packet defined by the data link layer of the under laying LAN or WAN.
  - Different protocols have different names for the packet.
  - Most LANs/WANs refer to packet as ***frame.*** ATM, WAN refers as ***Cell.***
- **Addressing :**
  - The data link layer is the lowest layer in the OSI model that is concerned with addressing, labeling information with a particular destination location.
  - Each device on a network has a unique number, usually called a *hardware address* or *MAC address,* that is used by the data link layer protocol to ensure that data intended for a specific machine gets to it properly.
  - **Next-hop address** is used to carry a frame across the **LAN.**
  - **Virtual circuit address** is used to carry a frame across **WAN.**
- **Error Control :**
  - Network must be able to transfer data from one device to another with complete accuracy.
  - The data link layer handles errors (caused by signal attenuation, noise etc.) that occur at the lower levels of the network stack.

- **Flow Control :**
  - Flow control is the set of producers that tells the sender how much data it can transmit before it must wait for an acknowledgment from the receiver.
  - The flow of data must not be allowed to overwhelm the receiver.
  - Receiving device must be able to tell the transmitting device to send few frames or stops temporarily.
- **Medium Access Control :**
  - When computers use a shared medium (cable or air), there must be a method to control the access to the medium.
  - This refers to the procedures used by devices to control access to the network medium.
  - Since many networks use a shared medium (such as a single network cable, or a series of cables that are electrically connected into a single virtual medium) it is necessary to have rules for managing the medium to avoid conflicts. For example, Ethernet uses the CSMA/CD method of media access control, while Token Ring uses token passing.
- Because the Data-Link Layer is very complex, it is sometimes divided into sublayers (as defined by the IEEE 802 model). The lower sublayer, called the **Media Access Control (MAC) sublayer,** provides ***network access.***
- The upper sublayer, called the **Logical Link Control (LLC) Sublayer,** is concerned with ***sending and receiving packets and error checking.***
- **The Logical Link Control (LLC) Sublayer** defines connection establishment and maintains links between communicating devices.
  - This is usually considered a DLL sublayer; it provides services to the network layer above it and hides the rest of the details of the data link layer to allow different technologies to work seamlessly with the higher layers. Most local area network technologies use the IEEE 802.2 LLC protocol.
- **The Media Access Control (MAC) Sublayer** provides controls for how multiple devices use the same media.
  - Devices that operate at this level include bridges, intelligent hubs & network interface cards (NICs).

5.1.1 Functions :

Data link layer has several functions to perform. These functions include

1. Providing a well defined service to the network layer,
2. Determining how the bits of the physical layer are grouped into frames,
3. Dealing with transmission errors,
4. Regulating the flow of frames so that slow receivers are not swamped by fast senders, and
5. General link management

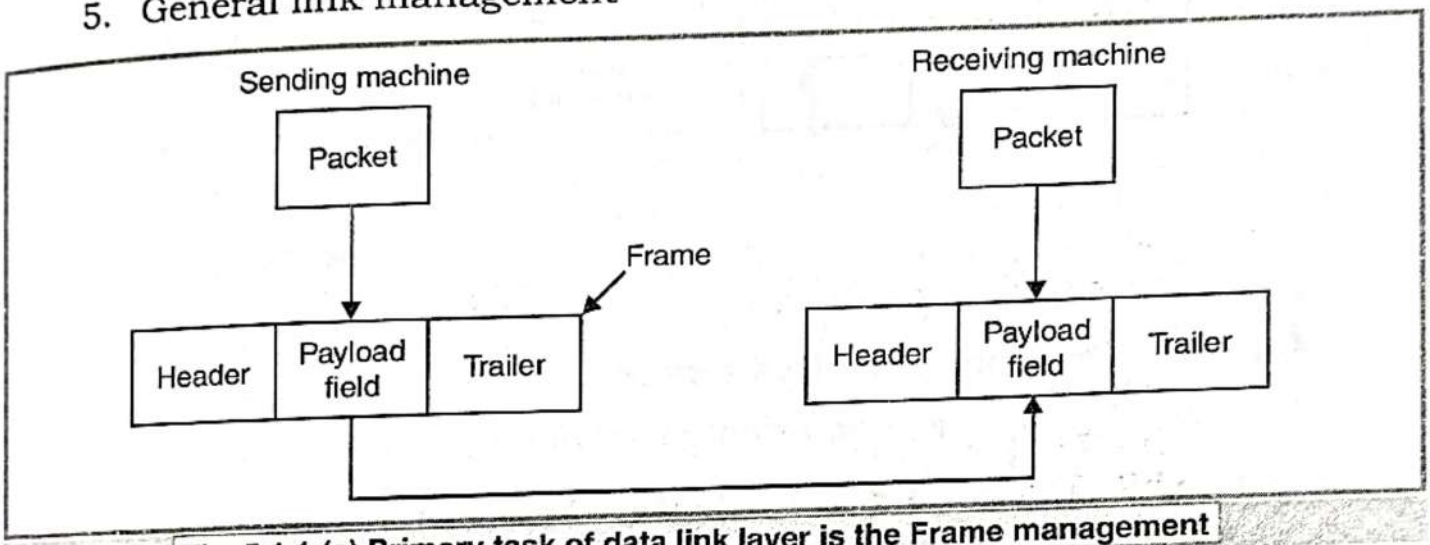


Fig. 5.1.1 (a) Primary task of data link layer is the Frame management

To accomplish these tasks, the data link layer takes the packets it gets from the network layer and encapsulates them into frames for transmission. Each frame contains a frame header, a payload field for holding the packet, and a frame trailer, as illustrated in Fig. 5.1.1(a). Primary task of data link layer is the Frame management.

5.2 SERVICE PROVIDED BY DATA LINK LAYER TO NETWORK LAYER

- The main service to be provided is to transfer data from the network layer on the sending machine to the network layer of the receiving machine.
- The virtual path followed for such a communication is shown in Fig. (a). It is not the actual path.
- The actual path followed by the data from sending machine to destination is shown in Fig. (b) which follows all the layers below the network layer, then the physical medium, then layer 1, 2, 3 of receiving machine.

5.2.1 Types of Service Provided :

- Data link layer can be designed to offer different types of services. Some of them are as follows :

1. Unacknowledged connectionless service.

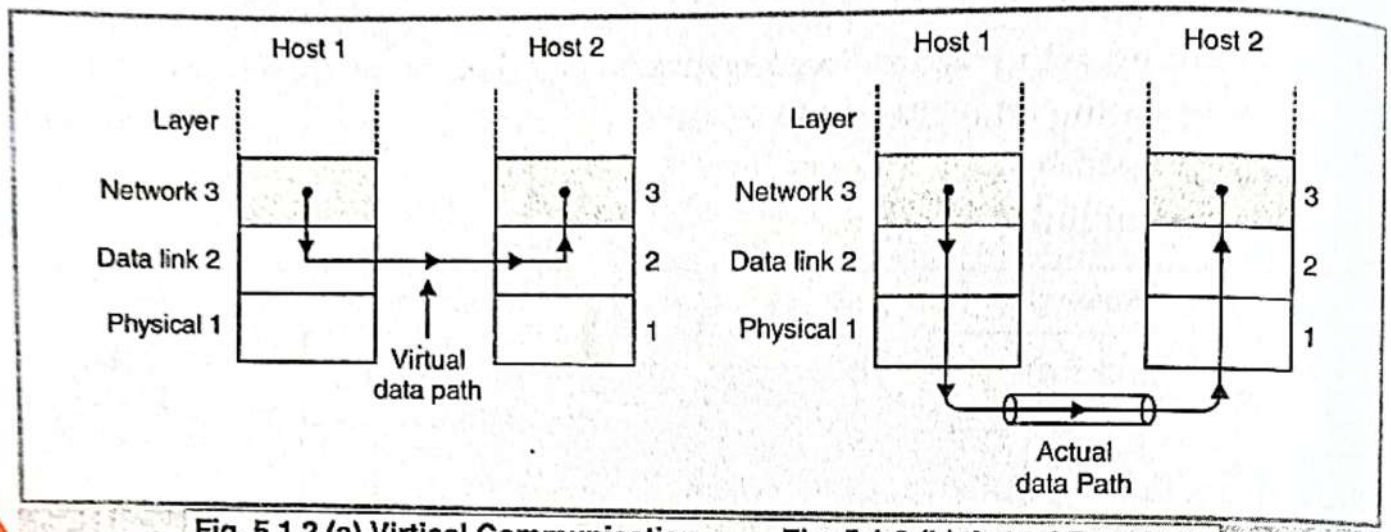


Fig. 5.1.2 (a) Virtical Communication Fig. 5.1.2 (b) Actual Data Path

2. Acknowledged connectionless service.
3. Acknowledged connection oriented service.

5.2.1.1 Unacknowledged Connectionless Service :

- In this type of service, the destination machine does not send back any acknowledgement of receiving frames.
- It is a connectionless service. So no connection is established before communication or released after it is over.
- If a frame is lost due to noise, then no attempt to made is recover it.
- This service is suitable only if the error rate is low. It is suitable for real time traffic such as speech.

5.2.1.2 Acknowledged Connectionless Service :

This is the next step to improve reliability.

- In this service, there are no connections used but for each frame received, the receiver sends an acknowledgement to the sender.
- If a frame is not received within specified time, then the sender will retransmit it.
- This service is suitable for communication over unreliable channels.

5.2.1.3 Acknowledged Connection Oriented Service :

- This is the most sophisticated one.

- The source and destination machines establish a connection before transferring the data.
- A specific number is given to each frame being sent and the data link layer guarantees that each transmitted frame is received.
- All the frames are guaranteed to be received in the same order as the order of transmission. The data transfer takes place by following three distinct phases given below.
  1. Connection is established.
  2. One or more frames are actually transmitted.
  3. The connection is released.

5.3 FRAME DESIGNING/FRAMING

- Breaking the bit stream into frames is called as framing.
- A frame has a predefined structure. It contains some control fields along with user data.
- In case of detection of an error the whole frame is retransmitted.
- Thus a frame is processed as one entity for flow control and error control.
- Each frame consists of three components in general.
- The three components are header, data bits and trailer as shown in fig. 5.2.
- The header and trailer has various formats depending on type of protocol used.

Fig. 5.2 Data Frame

- The header may contain one or more fields.
- Similarly trailer may also contain one or more fields.
- Both header & trailer fields contain data link control protocol information.
- For example high level data link control (HDLC) frame contains 6 fields - 3 in Header, 1 data bits and 2 in trailer.
- The first field in flag which identifies start of a frame.

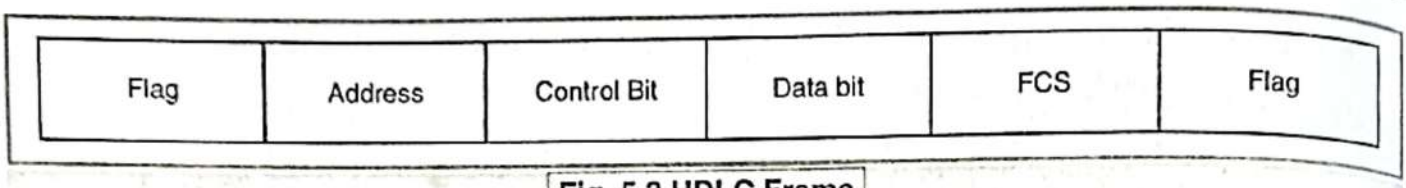


Fig. 5.3 HDLC Frame

- The second field of header is 'Address' which contains address of the destination.
- The third field contains sequence number of the frame, acknowledgements of receipt of frames or other control informations.
- Hence this field is known as control Bit field.
- The fourth field contains the data bit.
- The fifth field contains error detection bit and hence is known as 'Frame Check Sequence' i.e. FCS the last field contains another flag indicating end of the frame.
- A HDLC frame format is shown in fig. 5.3
- Whether it is frame format of HDLC or other data link control Protocol - each frame is such designed that the beginning and end of frame along with its various fields are easily located by the receiver. The receiver can locate the data bits too very easily.
- A frame can have one of the three types
  1. Fixed format - fixed length
  2. Fixed format - Variable length
  3. Variable format - Variable length
- In first case the fields of frame have fixed size and the format is decided once.
- In second case the field length are variable. In third case all the fields are optional and if a field is present its size is variable.
- So a start and end delimiter are required for this type of frame formats.
- The bits to be transmitted is first broken into discrete frames at the data link layer.
- In order to guarantee that the bit stream is error free, the checksum of each frame is computed.
- When a frame is received, the data link there, recomputes the checksum.
- If it is different from the checksum present in the frame, then the data link layer knows that an error has occurred.

- It then discards the bad frame and sends back a request for retransmission.
- Breaking the bit stream into frames is called as framing.
- One way of doing it is by inserting time gaps between frames as shown in Fig. 5.4.

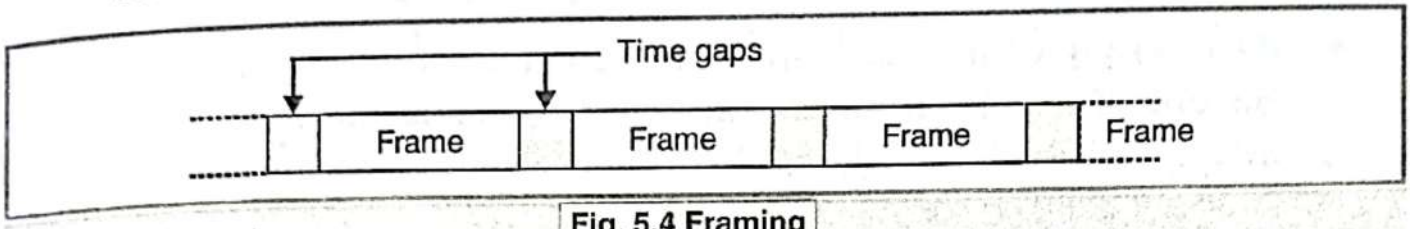


Fig. 5.4 Framing

- But practically this framing technique does not work satisfactorily, because networks generally do not make any guarantees about the timing.

So some other methods are devised.

5.3.1 Framing Methods :

Following methods are used for carrying out framing.

1. Character count
2. Starting and ending characters, with character stuffing.
3. Starting and ending flags with bit stuffing.
4. Physical layer coding violations.

5.3.1.1 Character Count :

- In this method, a field in the header is used to specify the number of characters in the frame.
- This number helps the receiver to know the number of characters in the frame following this count.

The character count method is illustrated in Fig. 5.5

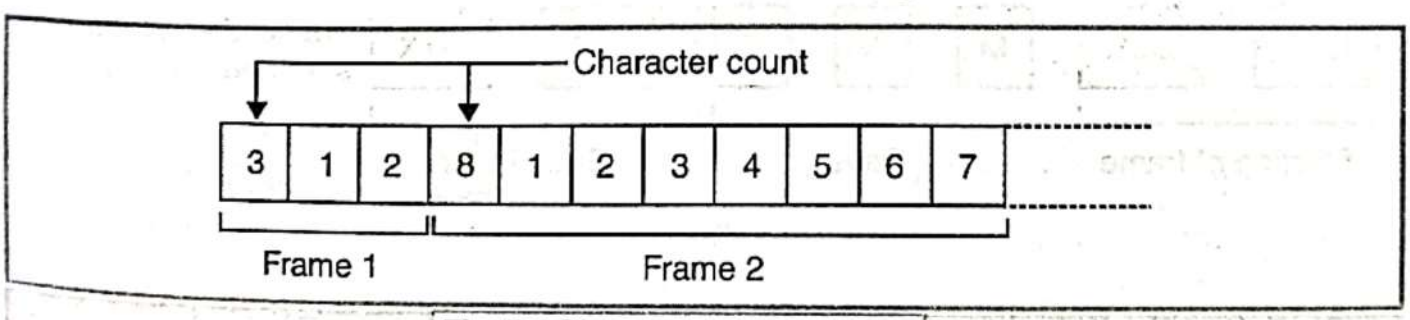


Fig. 5.5 Character count method

- There are two frames in the Fig. *i.e.* 3 and 8.

- First frame has three (baracter (3, 1, 2) and second frame has 8 characters (8, 1, 2, 3, 4, 5, 6, 7).
- It is clear that first bit tells how quary bit in a frame.
- The disadvantage of this method is that, an error can change the character count.
- If the wrong character count number is received then the receiver will get out of synchronization and will be unable to locate the start of next frame.

5.3.1.2 Starting and Ending Character with Character Stuffing :

- In this scheme, a starting character before the starting of each frame and an ending character at the end of each frame, is used.
- Each frame is preceded by the transmission of ASCII character sequence DLE STX. (DLE stands for data link escape and STX is start of TeXt).
- After each frame the ASCII character sequence DLE ETX is transmitted.
- Here DLE stands for Data Link Escape and ETX stands for End of TeXt.
- So if the receiver loses the synchronization, it just has to search for the DLE STX or DLE ETX characters to return back on track. This is shown in Fig.

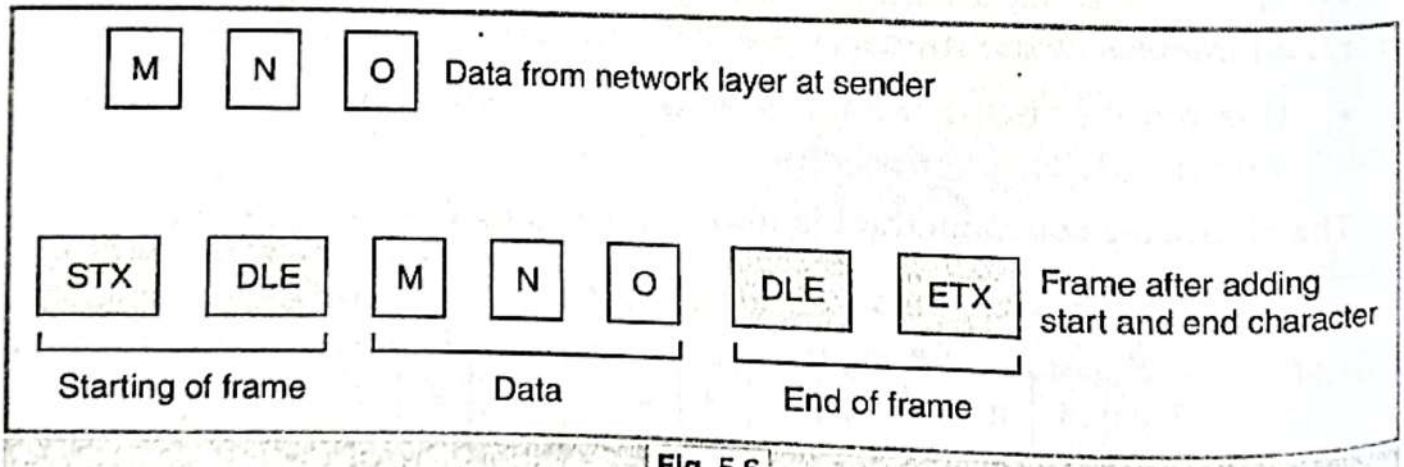


Fig. 5.6

Character Stuffing :

- The problem with the above system is that the characters DLE STX or DLE ETX can be a part of data as well.

- If so, they will be misinterpreted by the receiver as start or end of frame.
- This problem is solved by using a technique called character stuffing. Which is as follows.
- The data link layer at the sending end inserts an ASCII DLE character just before each accidental DLE character in the data.
- The data link layer at the receiving end will remove these DLE characters before handling over the data to the network layer.

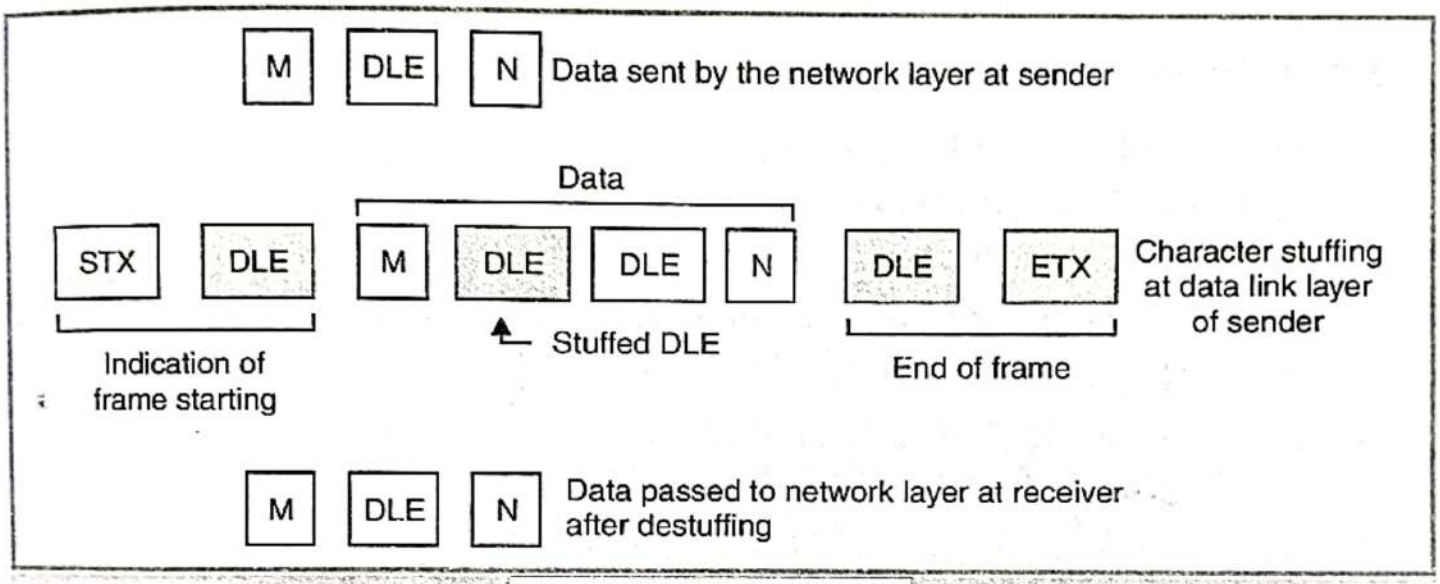


Fig. 5.7 Character stuffing

- The main disadvantage of this frame method is that we have to use the 8 bit character and ASCII code.
- The problem can be solved by next framing technique.

5.3.1.3 Starting and Ending Flags, with Bit Stuffing :

- This technique allows the framed to contain an arbitrary number of bits and codes different from ASCII code.
- At the beginning and end of each frame, a specific bit pattern 0111 1110 called **flag byte** is transmitted.
- Since there are six consecutive 1s in this byte a technique called **bit stuffing** which is similar to character stuffing is used.

Bit stuffing :

- Whenever the sender data link layer detects the presence of five consecutive ones in the data, it automatically stuffs a 0 hit into the outgoing bit stream.

- This is called bit stuffing and it is illustrated in Fig. below

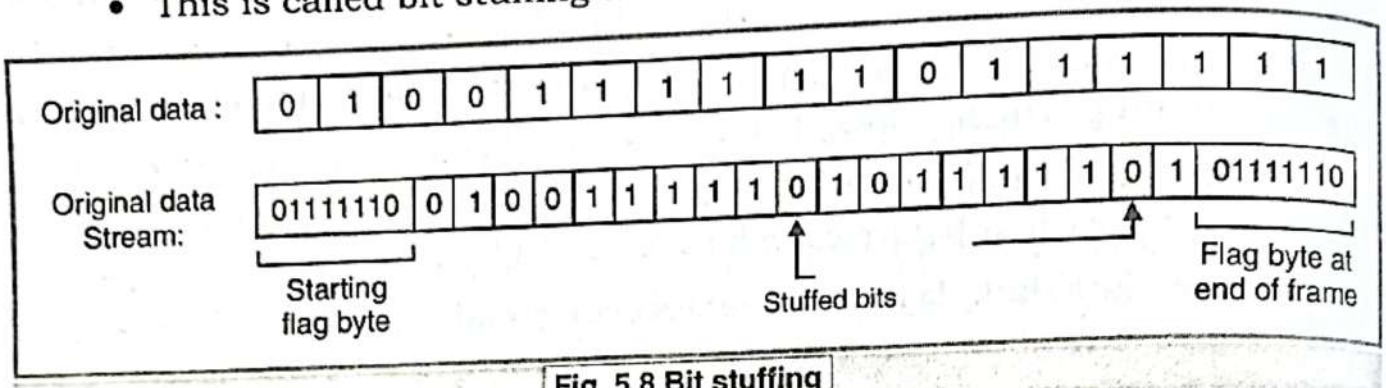


Fig. 5.8 Bit stuffing

5.3.1.4 Physical Layer Coding Violations :

- This is called destuffling.
- When a receiver detects presence of five consective ones in the received bit stram. It automatically the "0" bit following the five 1's.
- This method of framing is applicable only to the networks in which the encoding on the physical medium contains some redundancy.
- Some LANs encode each bit of data using two physical bits. The Manchester coding is generally used.
- Normally a 1 bit is encoded into a 10 pair and a 0 bit is encoded into a 01 pair as shown in Fig.
- This use of invalid physical code is a part of 802 LAN standards.

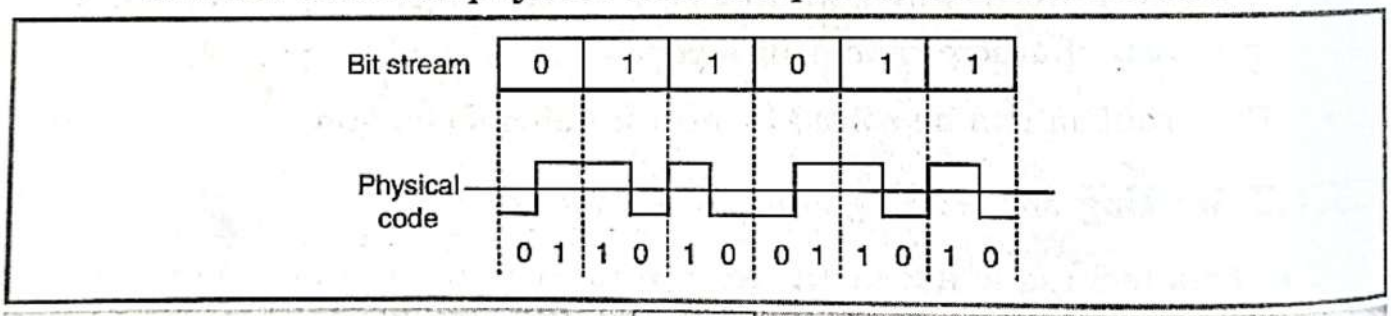


Fig. 5.9

5.4. FLOW CONTROL

- It is an important component of data link control.
- It is concerned with controlling the rate of transmission of frame or characters on a link in such a manner that the receiver always has sufficient space i.e. buffer storage resources to accept them prior to processing.
- The problem to be handled is what to do with the sender which systematically wants to transmit the frames at faster rate than the capacity of the receiver.

- For example in case of character by character transmission (Character oriented link) if the remote computer is servicing many terminals at a time it may become temporarily overloaded.
- At that time it will be unable to process all characters sent to it at the available transmission rate.
- Similarly if the transmission is frame oriented then there may be shortage of buffer storage if an indeterminate number of frames are sent to it.
- So a control to regulate the flow of incoming frames/data is required. This control is mostly in form of acknowledgement (ACK) which is sent by receiver.
- ACK is king of feed back mechanism from the receiver to sender.
- We are going to discuss some flow control techniques based on these principles.
- It is a set of procedures that tells the senders how much data it can transmit before it must wait for anacknowledgement from the receiver otherwise there will be overflow of the data.
- Each receiver has a block of memory called buffer, reserved for storing incoming data until they are processed.
- If the buffer begins to fillup, the receiver must be able to tell the sender to stop transmission until it is once again able to receive.
- Similarly the transmitter also has a buffer for storing the bit oi the transmission is stopped.
- The two main purposes served by acknowledgement are.
  * Acknowledging receipt of all previous frame.
  * Make it clear the sending end to transmit the next data frame.

A few commonly used flow control mechanisms are as follows:

5.4.1 X-On/X-Off mechanism :

- It is a flow control facility which ensures that a terminal does not send any further characters until an overload condition has been cleared.
- For doing this the computer returns a special control characters X-OFF to the controlling device which is inside the terminal like keyboard.

- This character instructs the controlling device to ease transmission.
- When the controlling device receives this X-off character it responds in one of the two manners.
  (a) Either it ignores any further character entered at a key board.
  (b) It buffers the entire character into a local buffer attached to terminal until it gets a clearing message from computer.
- This way computer gets a relief from entertaining an overload. As soon as the over load is cleared the computer returns another special characters X-On informing terminal control device to restart sending characters.
- The same mechanism is also used by printers and other output devices to control the flow of output coming from computer. In such cases the control device within printer or output device sends X-ON and X-OFF characters to control flow of characters.
- The mechanism is shown in fig. 5.10.

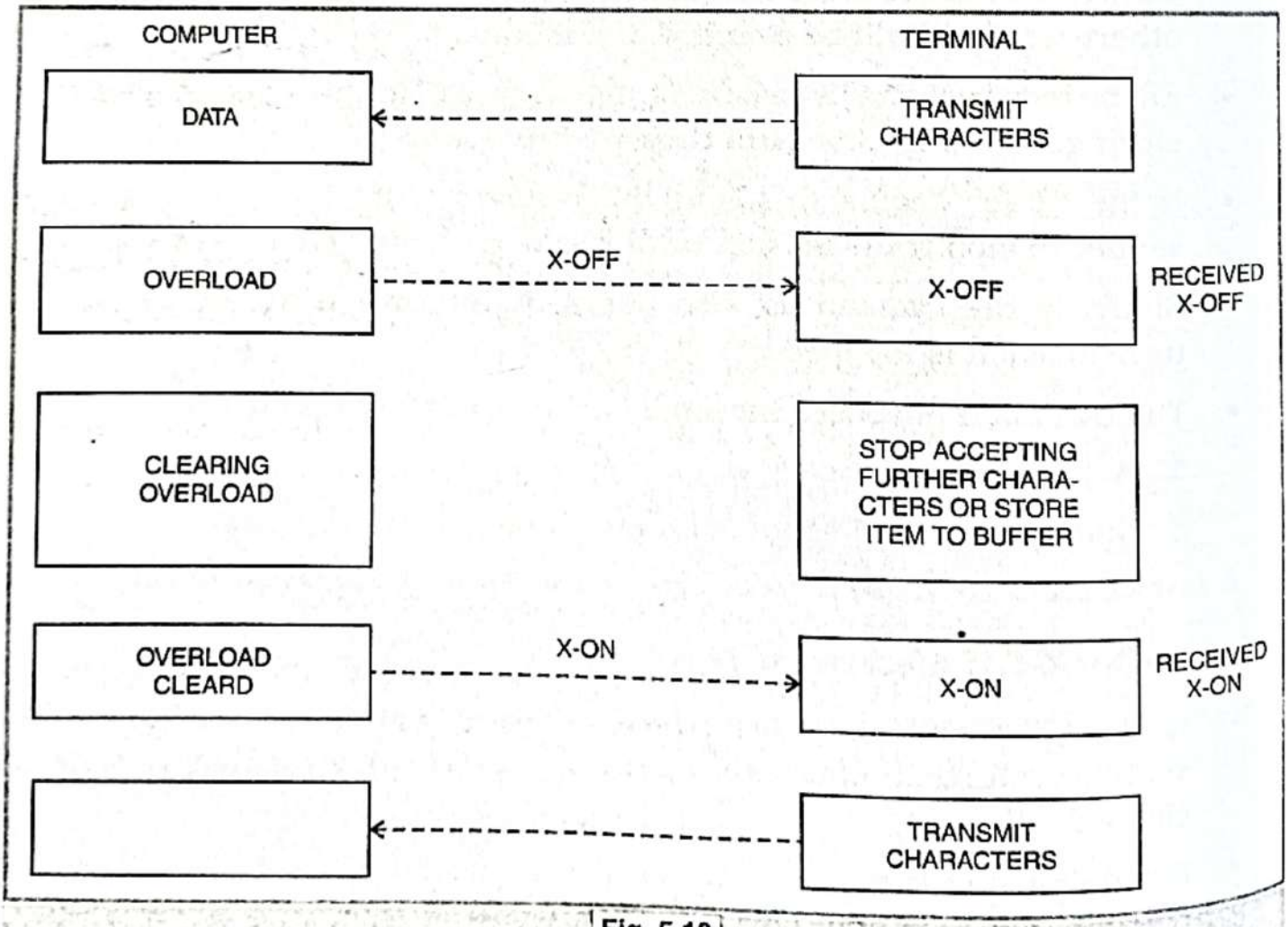


Fig. 5.10

5.4.2. Stop and Wait Flow Control Mechanism

This mechanism can be used between two computers or DTE's as well as a computer and terminal.

- In this mechanism the sending end sends one data frame at a time and then stops further transmission. It waits for the receiver's response i.e. an acknowledgement from receiver.
- The receiver can temporarily stop flow of data frames by withholding acknowledgement or by sending message like wait for acknowledgement or an ACK and WAIT better known as WACK. On receipt of WACK the sender stops and waits for acknowledgement ACK to send next data frame.
- The sending end has a timer to keep the time limits and it challenges the receiver after time out if no acknowledgement is received.
- When the receiver is challenged either it sends ACK for previous frame or if required it sends WACK.
- There are chances of loss of frames in such cases because if a frame is delivered to wrong destination because of technical faults the receiver has no knowledge about it.
- So when after time out the sender/transmitter challenges it the receiver sends ACK for previous frame and sender understands it an acknowledge for the frame which is actually lost without knowledge of sender as well as receiver.
- Though there are no chances of loss of ACK as sender always enquire it after time out. To over come this problem a concept of odd and even frame acknowledgements is used.
- ACK 1 is used for odd frames and ACK for even frames.
- Mostly it keeps track of all frames and there is no chance of missing of a single frame. But if two consecutive frames are delivered to wrong destination then it can't be managed with this technique too, as the last frame send as well as received both are even or odd.
- The mechanism's are shown in fig. 5.10 and fig. 5.11.

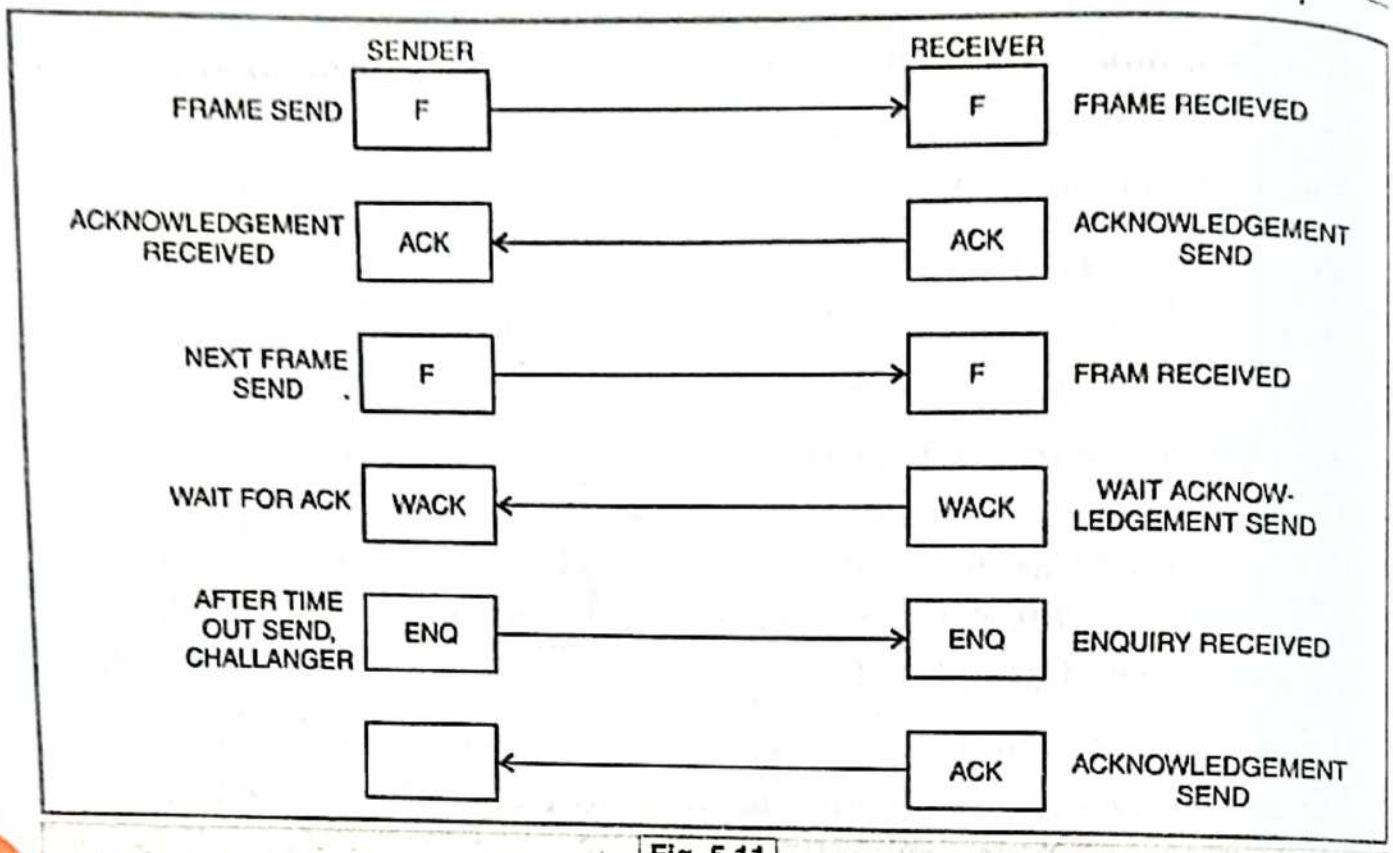


Fig. 5.11

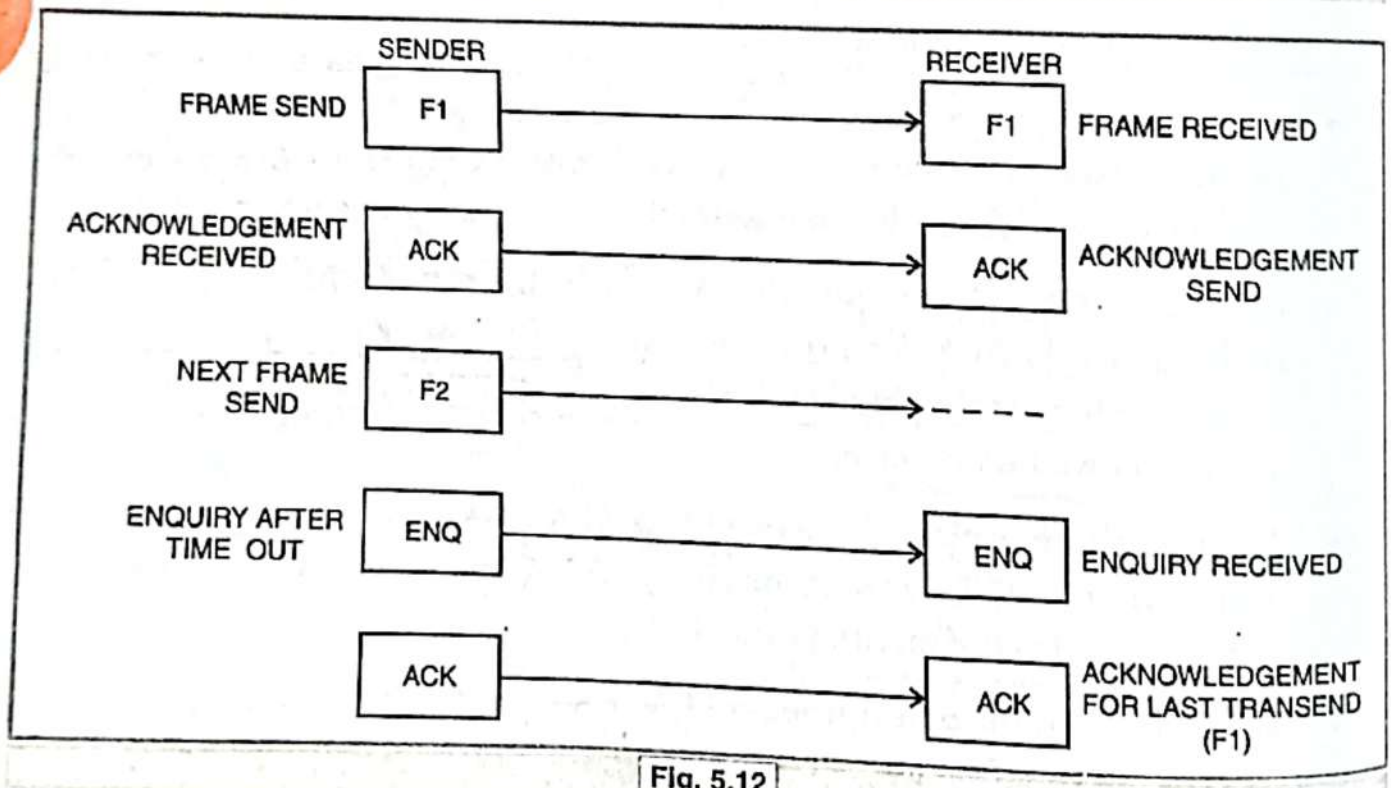


Fig. 5.12

- In this mechanism, as far as the link utilization is concerned, much time is consumed in transmission of data and if propagation time is large this mechanism becomes very inefficient from the point of view of link utilization.

(The frame F2 is lost without knowledge of seder as well as recivers)

If the frame size is of N bits and data rate is R then frame transmission time t_f is given by

- If t_p is the propagation time then the frame will be received at receiving end after time t_f+t_p. If sender sends frame and receiver sends acknowledgement without delay and sends next frame just after receiving acknowledgement then link utilization efficiency U can be given by

$$U = \frac{t_f}{t_f+2t_p} = \frac{1}{1+2t_p/t_f}$$

or

$$U = \frac{1}{1+2t}$$

where

$$U = \frac{t_p}{t_f}$$

- The stop and wait mechanism is only suitable for small propagation time i.e. When it is comparable to or less than frame transmission time.
- The link utilization is very poor for satellite link as well as for large propagation time.
- This can be reduced using large data frames but some other technical problems may arise in that case like other transmission errors etc.
- A better approach is the sliding window concept which overcomes the limitations of this mechanism.

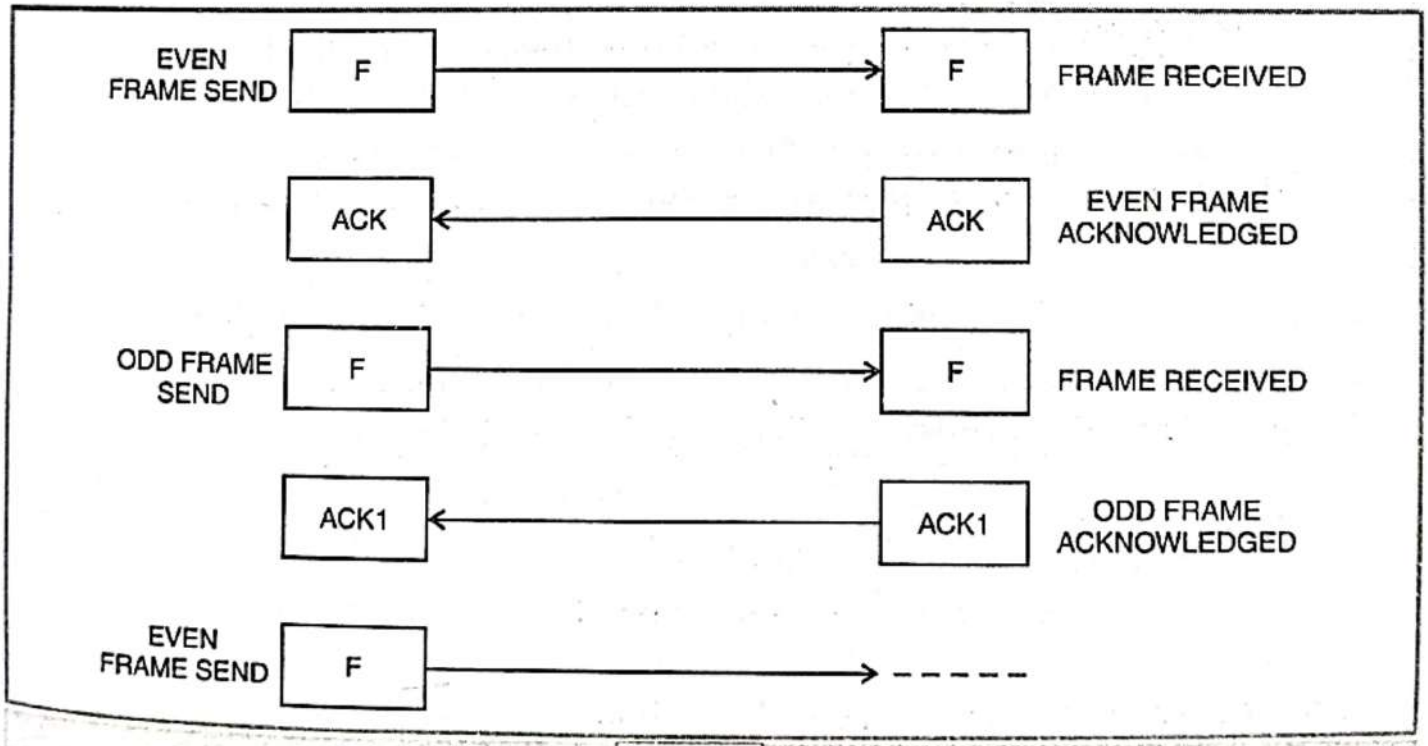


Fig. 5.13

5.4.3 Sliding Window Flow Control Mechanism

- This mechanism overcomes the limitations of stop and wait mechanism. It is a universally adopted method.
- As far as link utilization is concerned it is very efficient because in this mechanism the frames are transmitted one after the another without waiting for acknowledgement also contains number telling up to which number of frames are received properly.
- Each data frame is identified by a sequence number.
- The sending end has a window of fixed number of data frames.
- These frames are ready for transmission and can be sent without waiting for any acknowledgement.
- But a copy of these transmitted frames remain in the window till an acknowledgement is received.
- The acknowledgement (NAK) depending on detection of errors by receiver.
- If its is a positive acknowledgment like RR-5 or ACK-5 then up to 4th frame slide out from window and next four frames slide in and are ready for transmission.
- If a negative acknowledgement is received like NAK-5 then depending on request or error correction techniques used, it retransmits the frame starting from F5 in case of go-back-N mechanism and retransmits only F5 for selective repeat mechanism.
- Any how up to frame F4 the frames slide out from window and next four slide in for transmission in both the cases. A third case may also arise that is if the acknowledgement is RNR-N which is equivalent to WACK in stop and wait mechanism.
- In that case the frames up to the number (N-1) slide out and next frames slide in the window.
- How many frames will slide in, it depends on size of window.
- The number of frames which can stay at a time in a window is known as a **size of window.**
- Its typical value is seven i.e. most of the windows have their size seven and seven frames may stay in it.
- But its size can be less or more too depending on frame size of sender.
- RR stands for 'receive ready' and RNR stands for 'receive not ready. N stands for sequence number of the frame.

- If an acknowledgement like RR-N is received.
- Then it ensures the correct receipt the frames 1,2,3.....up to (n–1).
- When an acknowledgement is received the frame up to previous number are slided out from window and their copy is deleted. If x frames are deleted then next x frames slide in the window for transmission.
- RNR-N acknowledges up to frame (N-1) and request to stop further transmission temporarily.
- The sender waits for an RR-N and only after receiving it transmits next frame.
- A timer is provided with the sender and if it does not receive an acknowledgement from receiver in time then after time out the sender requests or enquires about the acknowledgement.
- An example of sliding window mechanism is shown in fig. 5.14 for a window of size 9.
- In fig. 5.14 the acknowledgement RR-3 indicates the correct receiving of F1 & F2 hence F1 & F2 slide out from window and F10 & F11 slide in.
- When it gets RNR-8 indicating the correct receiving up to F7 the frames F3 to F7 slide out of the window and now the window is occupied by the frames F8 to F15. The sender now waits for resuming transmission indication i.e. RR-8. Once it receives RR-8 it will restart transmission from F8 onwards.
- The sequence numbering of frames is done using fixed no. of bits. If no of bits available for sequence numbering is n then total 2^n frames can be numbered starting from 0 to 2^n-1.
- For example if bits used for sequencing are four then the frames will be numbered as 0 to 15.
- The seventeenth frame will have the number 0, eighteenth will have 1 and so on, i.e. the count restarts from zero once the 2^nth frame is slided in the window. Here one thing is to be noted that within a window no sequence number should be duplicated.
- Hence the no. of bits used for sequencing is taken in a manner such that its combinations should be larger than the size of window. Hence for an n bit sequence number the maximum window size is restricted to 2^n 1.

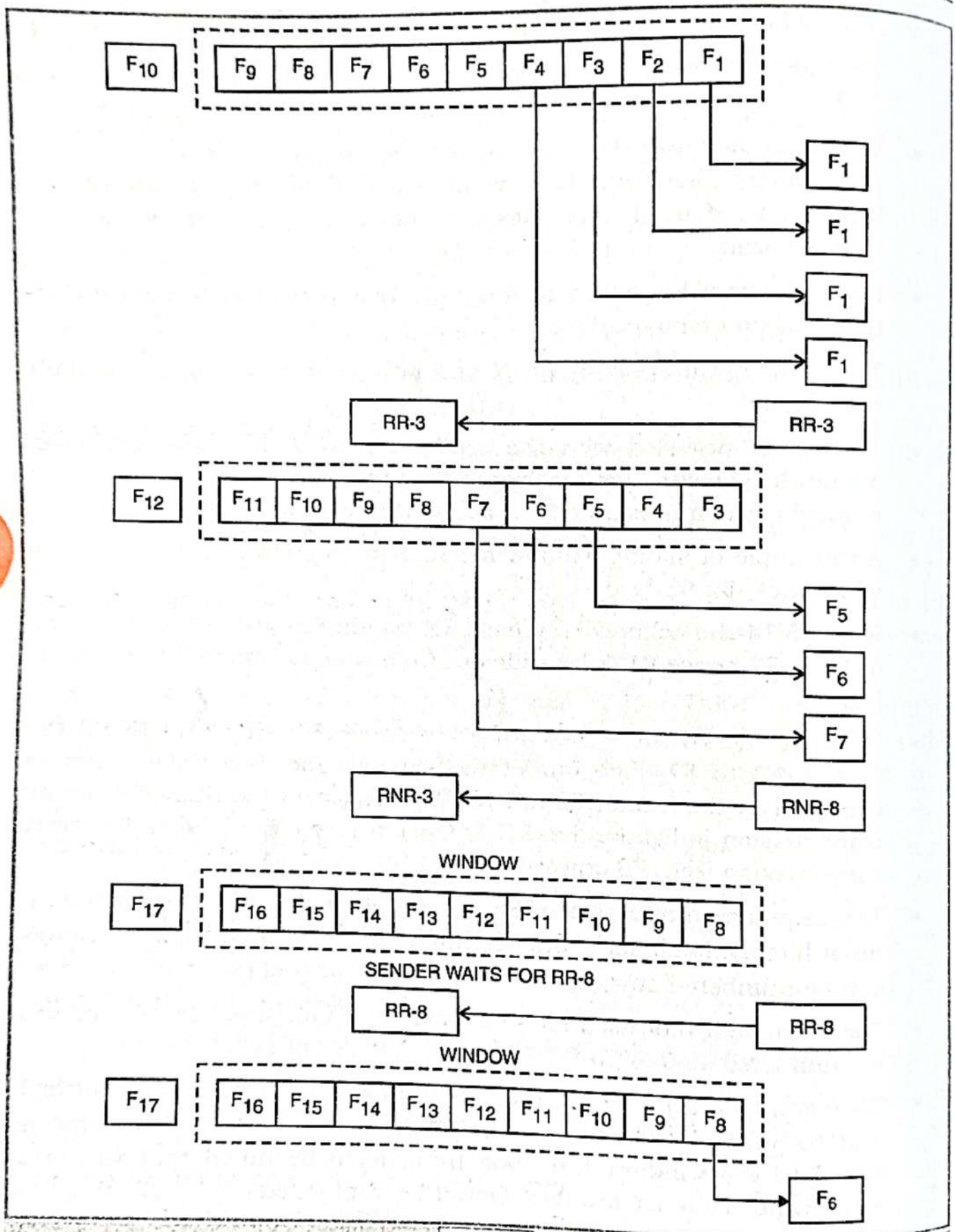


Fig. 5.14

- The link utilization is better in sliding window mechanism as compared to stop and wait mechanism, as the transmission rate is faster in this case.

- For calculating link utilization efficiency one of the two cases may arise.
- In first case the sender receives an acknowledgement before exhausting the window. i.e.

$$W.t_f \geq t_f + 2t_p$$

or $$W \geq 1 + \frac{2t_p}{t_f}$$

or $$W \geq 1 + 2t \text{ where } t = \frac{t_p}{t_f}$$

- In second case the sender exhausts the window before it revieves an acknowlegement

i.e. $$W.t_1 \leq t_f + 2t_p$$

or $$W \leq 1 + 2t$$

- If the reciever sends the acknoledgement the earlies the or

$$U = \frac{w\, t_f}{t_f + 2t_p} \quad U = \frac{w}{1+2t}$$

- If the size of window is 1 then the sliding window mechanism is nothing but stop and wait mechanism.
- But for larger window the link utilization efficiency increases.
- In case of satellite links the window size is very large even as large as 127 frames.
- So they are very efficiency in point of view of link utilization.
- The sliding window flow control mechanism also allows a high value to t i.e. a large data frame without impairments and then link utilization efficiency increases both the way.

5.5 DATA LINK ERROR CONTROL

- When we enter into a computer by means of a key board we use to press character keys one by one and the characters are displayed on screen directly.
- The character displayed is different from required one we generally use a delete or backspace key to remove it and we again enter the character using corresponding key.

- This error control is known as manual error control and is used directly between computer and peripheral.
- Now if the terminal is connected to a remote computer a similar type of operation takes place.
- This time the characters entered are not displayed on screen directly, instead they are transmitted to remote computer character by character.
- The remote computer reads it and after storing into memory retransmits it to the terminal to be displayed.
- If it is different from the required one then user transmits a delete character in the same manner and once it is deleted then sends next required character in the same manner.
- This mode of error control is known as **Echo checking**.
- Now if a computer is transferring blocks of characters i.e. frames to another computer via a serial data link then it needs an efficient automated error control procedure or program which could control the error automatically without any intervention from the user.
- In such cases the receiver computer checks the received frame for all possible transmission errors and then sends an acknowledgement to sender computer.
- If it founds an error it sends request to retransmit the frame.
- This mode of error control is known as **automatic repeat request (ARQ).** There are two types of ARQ.

  A. Idle RQ

  B. Continuous RQ

(A) Idle RQ :

- This mechanism is used with character oriented or byte oriented data transmission schemes.
- This operates in a half duplex mode as the sender after transmitting a frame must wait until it receives an acknowledgement from the receiver.
- More common name of this mechanism is **stop and wait mechanism.**
- If receiver finds no error in the data frame it sends a positive acknowledgement i.e. ACK and if an error is found in received data frame it responds with a negative acknowledgement i.e. NAK.

- This is how the content errors are checked. Now to control flow integrity errors the sender is equipped with a timer.

The scheme is shown in fig. 5.15.

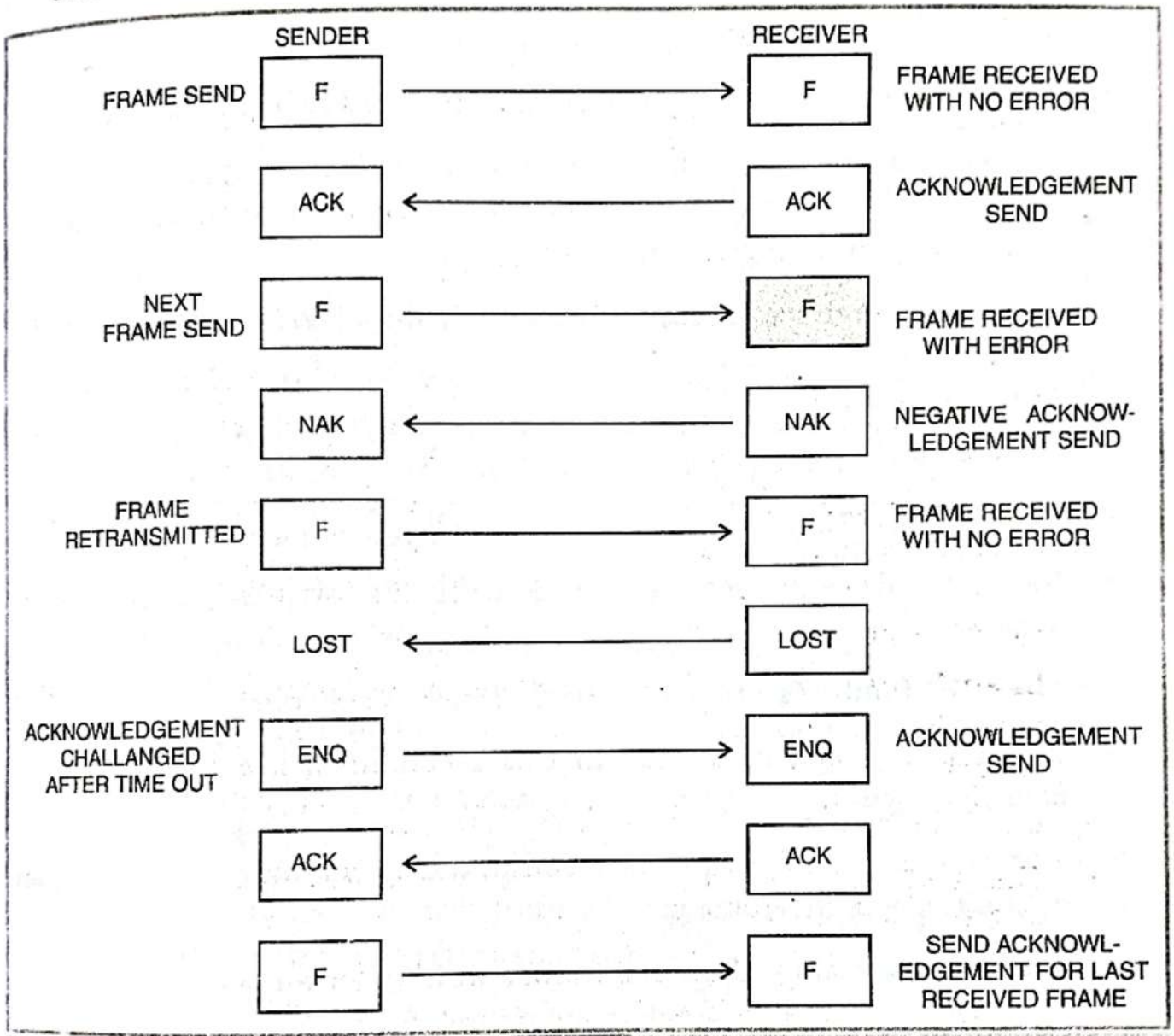


Fig. 5.15

- After transmitting a frame the sender waits for acknowledgement for a specified time period.
- If an acknowledgement ACK or NAK is not received within that time period then it challanges the receiver by sending an enquiry (ENQ).
- The sender sends the acknowledgement whenever enquired for the last data frame recived.
- This mechanism is very simple but there are chances for loss of frames as discussed earlier.
- To overcome this problem an alternative scheme with odd and even data frames is used.

- The frames are marked as odd or even and sender and receiver both keep track of odd and even designation of each frame.
- There is no chance of loss of acknowledgement because it is always enquired by the sender.
- From the link utilization point of view this scheme is inefficient.
- In this scheme each frame has two or more copies known as duplicates because after transmitting one frame the sender stops and waits for acknowledgement.
- In case of negative acknowledgement it may have to retransmit the frame.
- So until a positive acknowledgement is received a copy of frame remains with sender.

(B) Continuous RQ :

- Initially this scheme was used with bit oriented transmission schemes.
- The size of buffer storage is increased in this scheme.
- Two types of schemes used in this mechanism are go-back-N and selective repeat.
- The sender transmits frames continuously without waiting for an acknowledgement frame.
- Now there are more than one frames waiting for an acknowledgement and hence a copy of each transmitted frame is retained with the sender, Until they are positively acknowledged.
- The list of transmitted frames along with their contents retained with sender is known as retransmission list.
- The list operates on FIFO queue discipline Now each frame contains a unique identifier or sequence number for its identification.
- The receiver sends acknowledgement with frame sequence number and the acknowledgement is for a group instead of individual frame. For example ACK-6 acknowledges the correct receiving of frames up to sequence number 5.
- Once an ACK is received the acknowledged frames are removed from retransmission list.

- A better known name for this mechanism is sliding window mechanism.
- There are no chances of loss of frames in this scheme.
- As soon as an out of sequence frame is received a request for retransmission of missing frame is sent.
- This is made possible by receiver by keeping track of sequence numbers of incoming data frames.
- For correcting errors in received data frames one of the two mechanisms can be used.
- In go-back-N scheme the receiver requests the sender to retransmit data frames starting from frame with sequence number N and this request is send by sending a REJ-N message/request.
- For example a request REJ-5 request the sender to retransmit data frames starting from sequence number 5.
- In selective repeat mechanism the receiver requests the sender to retransmit the missing data frame only or a frame which is received with data errors.
- This is done by sending a request as SREJ-N.
- The sender then retransmits the data frame with sequence number N and then starts further transmission from frame next to last transmitted frame (not retransmitted).
- No doubt the selective repeat is best mechanism among all.
- With the link utilization point of view this mechanism is very efficient.

5.5.1 ERROR DETECTION

- When data is being transmitted from one machine to another, it may be possible that data become corrupted on its way.
- Some of the bits may be altered, damaged or lost during transmission, such a condition is known as error.
- The error may occur because of noise on line, attenuation and delay distortion.

- For reliable communication, it is important that errors are detected and corrected.

5.5.1.1 Type of errors

There are two main types of errors in transmissions:

1. Single bit error
2. Burst error

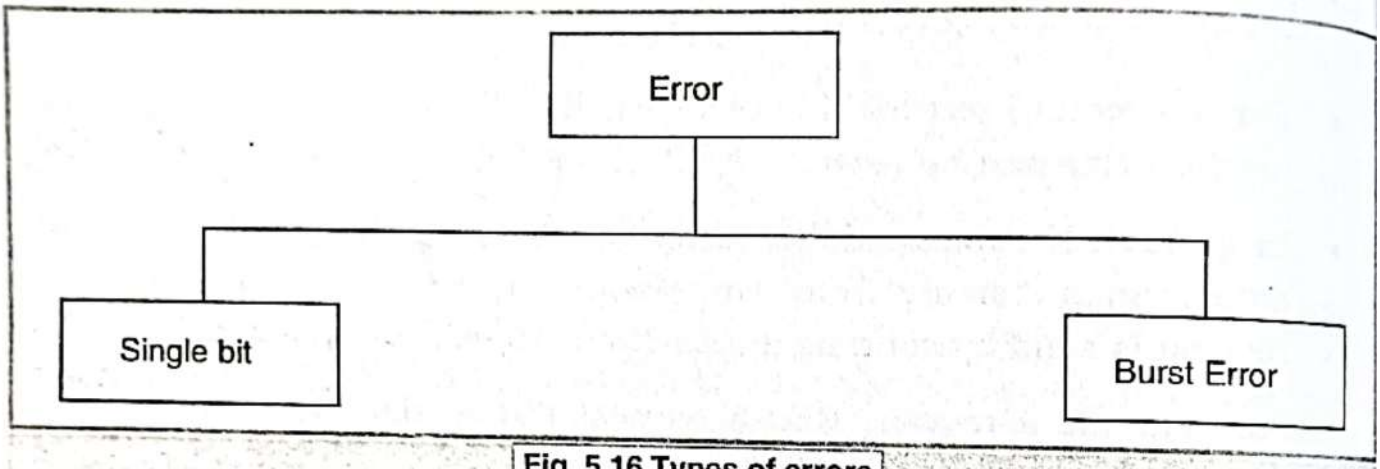


Fig. 5.16 Types of errors

1. Single bit error: It means only one bit of data unit is changed from 1 to 0 or from 0 to 1 as shown in fig. 5.16.

- Single bit error can happen in parallel transmission where all the data bits are transmitted using separate wires.
- Single bit errors are the least likely type of error in serial transmission.

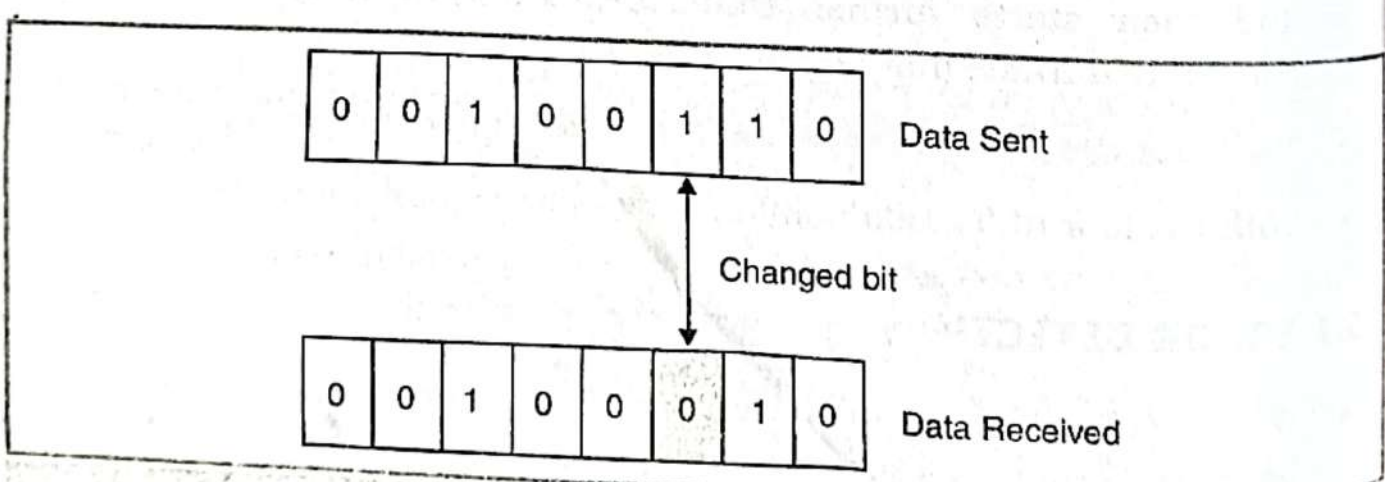


Fig. 5.17 Single bit error

2. Burst Error:

- It means two or more bits in data unit are changed from 1 to 0 from 0 to 1 as shown in fig. 5.17.

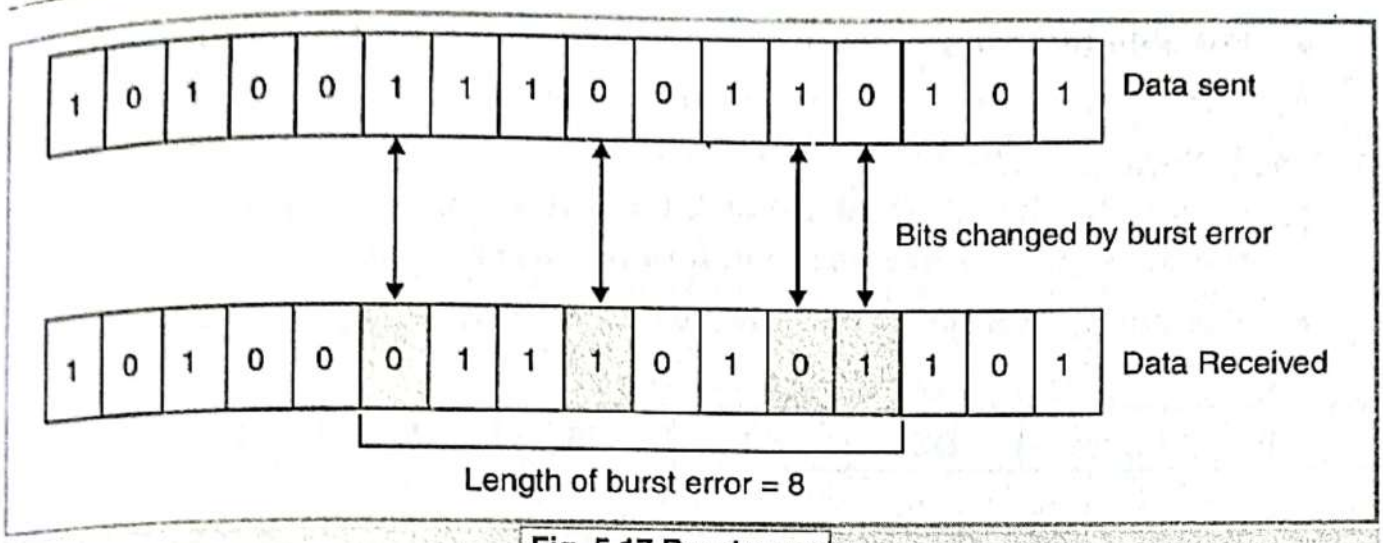
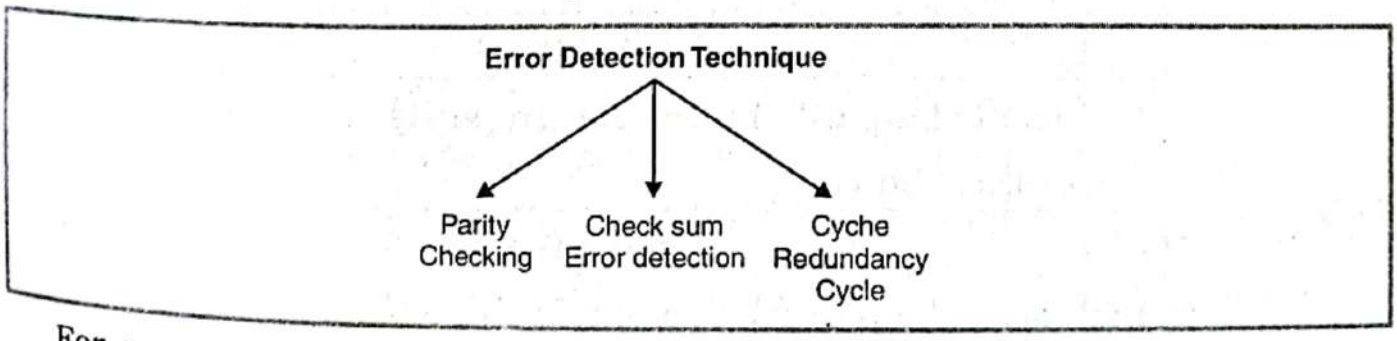


Fig. 5.17 Burst error

- In burst error, it is not necessary that only consecutive bits are changed.
- The length of burst error is measured from first changed bit to last changed bit. As shown in fig. 5.18, length of burst error is 8, although some bits are unchanged in between.
- Burst error is most likely to occur in a serial transmission. The noise occurring for a longer duration affects multiple bits.
- The number of bits affected depends on the data rate & duration of noise. For *e.g.* if data rate is 1 kbps, a noise of 1/100 second can affect 10 bits.

5.5.1.2 Error detection techniques

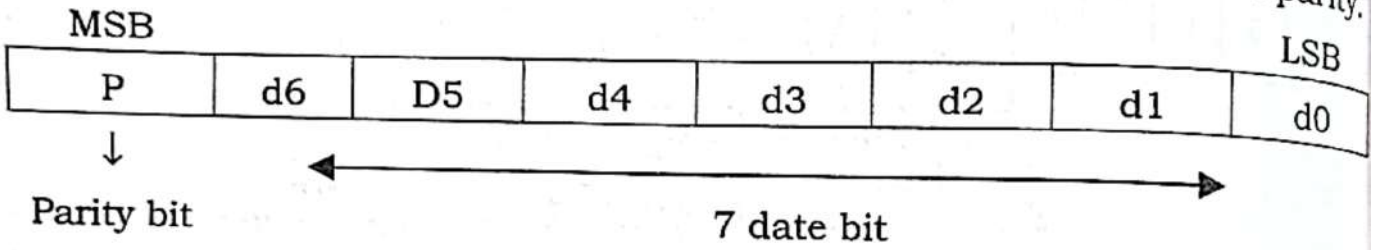
- Error detection means inspecting the data so as to check whether or not data is changed during the transmission.
- An error detection system is needed at the receiver end that should check the incoming data & report the number & type of errors that have occurred.
- The most common technique used for error detection are :



For error detection and correction it is necessary to add some check bit to a block of the data bits. These check bits are also known as redundant bit because they do not carry any useful information.

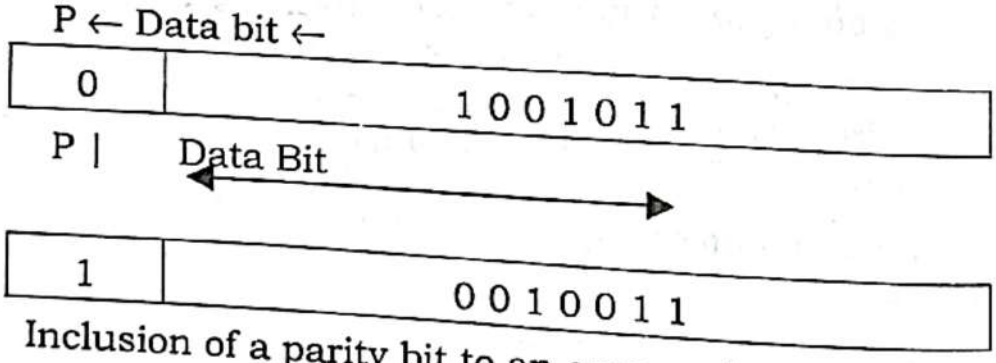
1. Parity Checking

- The simplest technique for detecting error is to add an extra bit known as parity bit to each word being transmitted.
- Generally the MSB of an 8-bit word is used as parity bit and in remaining 7 bits are used as data or message bit
- The parity of 8 bit transmitted word may be even parity or add parity.

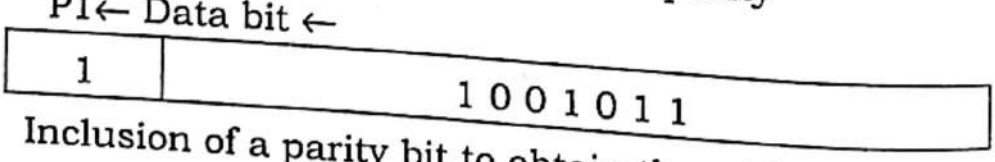


- Even parity means the number of 1's in the given word including the parity bit should b even (2, 4, 6, ...)
- Odd parity means the number of 1's in the given word including the parity bit should be odd (1, 3, 5, ...)
- **Use of parity bit**
- The parity bit can be set to 0 or 1 depending on the type of parity required.
- For odd parity this bit is set to 1 or 0 at the transmitter such that number of "1bits" in the entire word is odd.
- For even parity this bit is set to 1 or 0 such has the number of "1 bits" in the entire word is even.

Example



Inclusion of a parity bit to an even parity

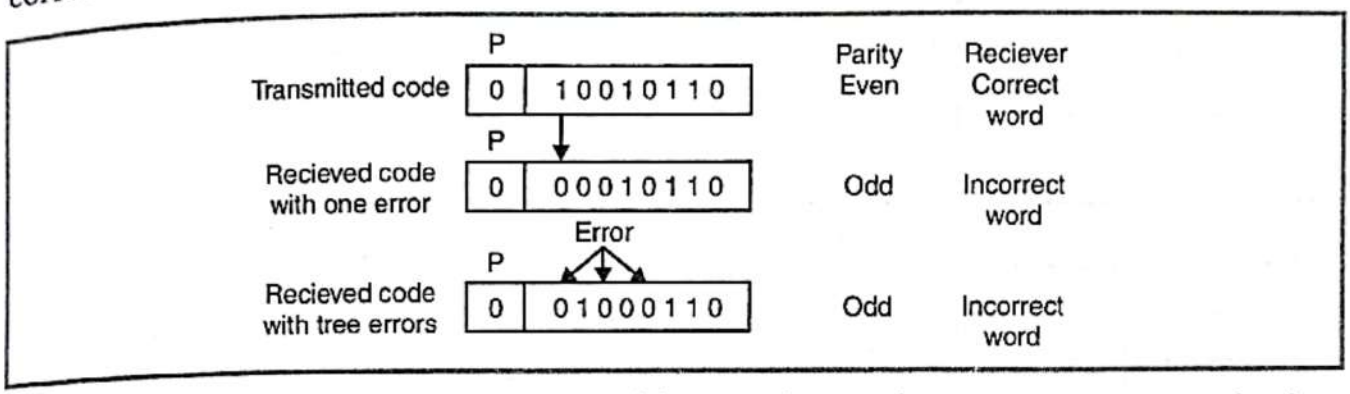


Inclusion of a parity bit to obtain the odd parity

- How does error detection take place ?

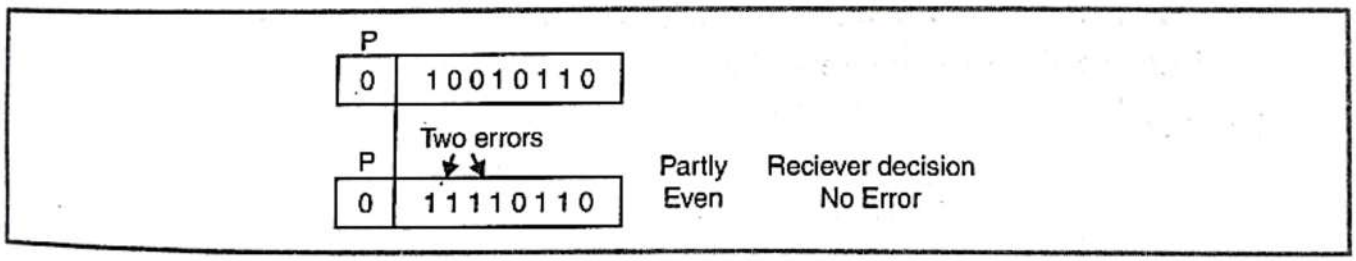
1. The parity checking at the receiving can detect the presence of an

error if the parity of the reviewing signal is different from the expected parity *i.e.* If it is known that the parity of the transmitted signal has an even parity if received signal has odd parity must conclude that received signal is not correct.



- When a single error or odd number of error occures during transmission the parity of the code word changes. It will detect an error.
- If the presence of an error occures or detected the receiver will ignore the received byte and request for retransmission of the same byte to the transmitter.
- **When does parity checking fails to detect errors ?**

If the number of errors introduced in transmitted code is even number, then the parity received code word will not change. It will remain even and the receiver will fail to detect the presence of error.



- The receiver cannot detect the presence of error if the number of errors is even. *i.e.* 2, 4, 6, ...

Limitations

1. It is not suitable for detecting the multiple errors.
2. It cannot reveal the location of erroneous bit. It cannot correct the error.
3. It is very difficult to detect the brust error.

5.5.1.2.1 Redundancy

- Redundancy is the method in which some extra bits are added to the data so as to check whether the data contain error or not.

- These redundant bits are added by the sender and removed by the receiver as soon as the accuracy. of the transmission has been determined (see fig. 5.19)

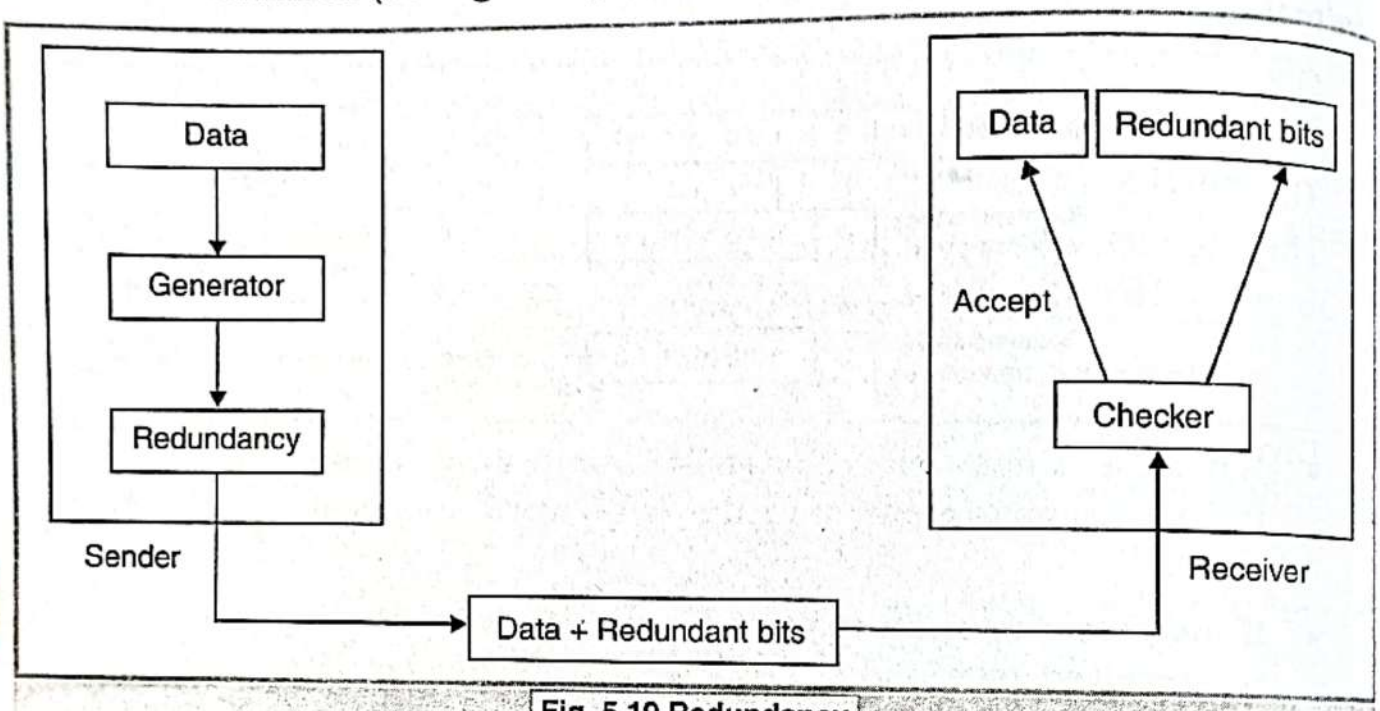


Fig. 5.19 Redundancy

- As shown in fig. 5.19 the data generated by source is given to device called generator that analyses the data and create redundant bits.
- These redundant bits are then added to the data and (data + redundant) bits are transmitted to receiver.
- Receiver then passes this data through a checking function. This checking function inspects the data for errors.
- If there is no error, it separates the redundant bits and accepts the data. If data is erroneous then it discards it.

2. Check sum

Example 1. The code word is received as 1100 1001 01011 check whether there are errors in the received codeword . If the divisor is 10101

```
        1111
10101 ) 1100100101011  <---  Recieved
        10101                codework
         11000
         10101
          11010
          10101
           11111
           10101
           10101
           10101
           0000
```

Zero remainder shows no error in the received code word.

Example 2. Generate the CRC code for the data word of 1100 10101. The divisor is 10101.

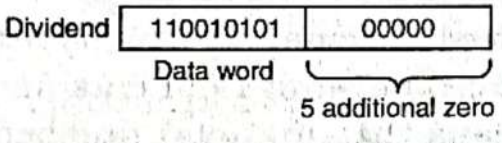
Solution

Given Data word 1100 10101

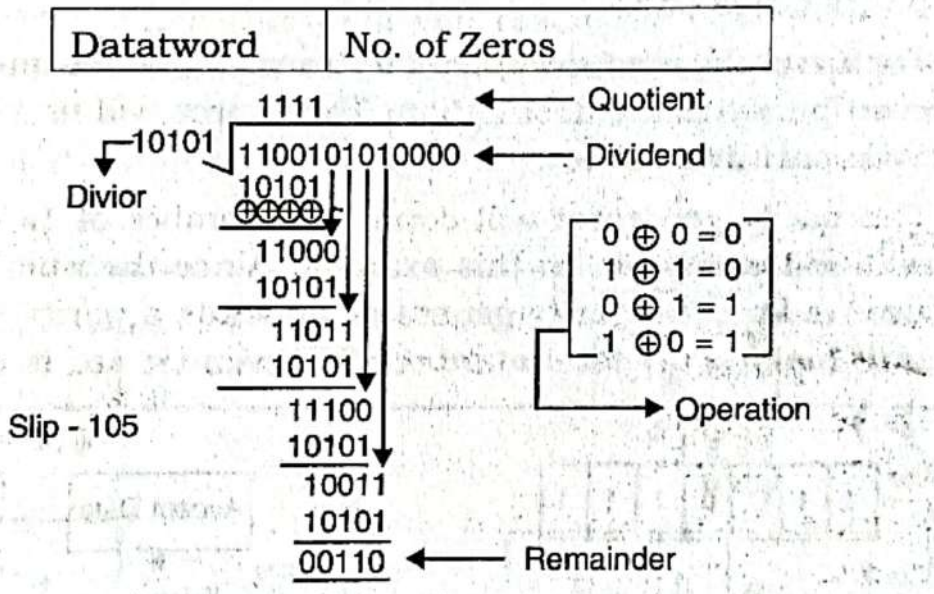
Divisor 10101

The number of data bits = m = 9

The number of bits in code word = N.



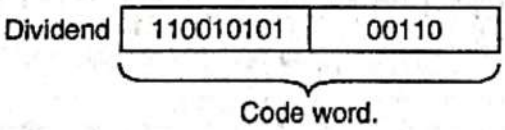
Dividend



In CRC remainder will be appended to the data word to get the code word.

$$\therefore \quad \begin{array}{r} 1100101010000 0 \\ +110 \\ \hline 1100101010001 10 \end{array}$$

Now the complete code word will be



Hence the complete code word is : 11001010100110

5.5.1.2.2 Types of redundancy check/Error detection Methods or Codes

The various error detection codes or methods used are:

1. Vertical Redundancy Check (VRC)/Parity Check.

2. Longitudinal Redundancy Check (LRC).
3. Cyclic Redundancy Check (CRC).
4. Checksum.

1. Vertical Redundancy Check (VRC)/Parity Check

(a) This method is also known as parity check.

(b) In this method, a redundant bit called parity bit is added to each data unit.

(c) This method may include even parity or odd parity. Even parity means that total number of 1s in data (including parity bit) is even. Odd parity means that the total number of 1s in data (including parity bit) is odd.

(d) For example, if source wants to transmit a data unit 1100111 using even parity to the destination. The source will first pass this unit to even parity generator.

(e) The parity generator will count the number of 1s in data unit and will add parity bit. In this example, since the number of 1s in data unit is five, the parity generator appends a parity bit 1 to this data unit making the total number of 1s even *i.e.* six. (see fig. 5.13)

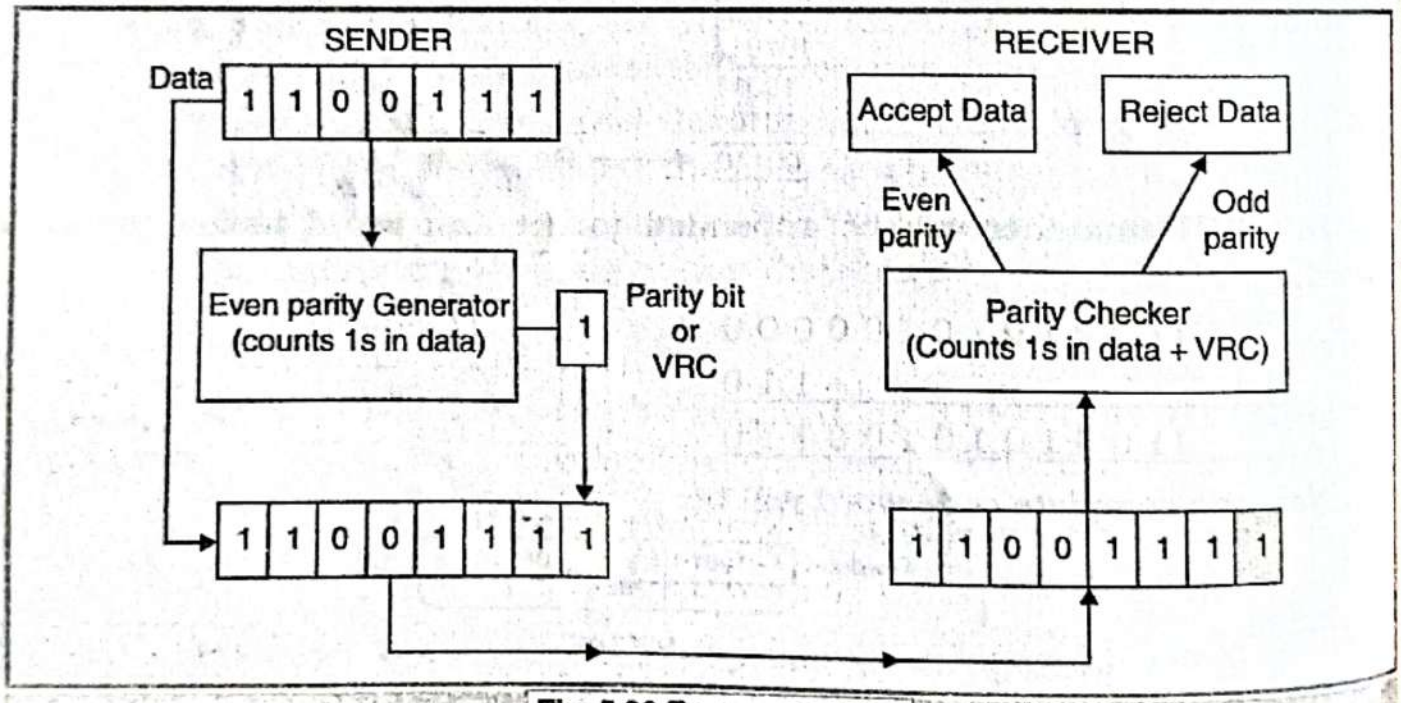


Fig. 5.20 Even parity VRC

(f) This data along with parity bit is then transmitted across the network. In this case 11001111 will be transmitted.

(g) At the destination, this data is passed to parity checker. The parity checker counts the number of 1s in data. If the number of ones

comes out to be odd, for *e.g.*, five or seven then destination will come to know that there is some error in the data. The receiver then rejects such erroneous data unit.

(h) VRC can detect all single bit error.

(i) It can also detect burst errors but only in those cases where number of bits changed is odd *i.e.* if number of bits changed are 1, 3, 5, 7 .... etc.

(j) The major disadvantage of this method is that it is not able to detect burst error if number of bits changed is even i.e., 2, 4, 6, 8.... etc.

For example : If original data is 1100111. After adding VRC, data unit that will be transmitted is 11001111. Suppose on the way 2 bits are changed 01011111. When this data will reach the destination, the parity checker will count the number of 1s in data and that comes out to be even *i.e.* 8. Now in this case, parity is not changed, it is still even. The destination will assume that there is no error in data even though data is erroneous. (See fig. 5.21)

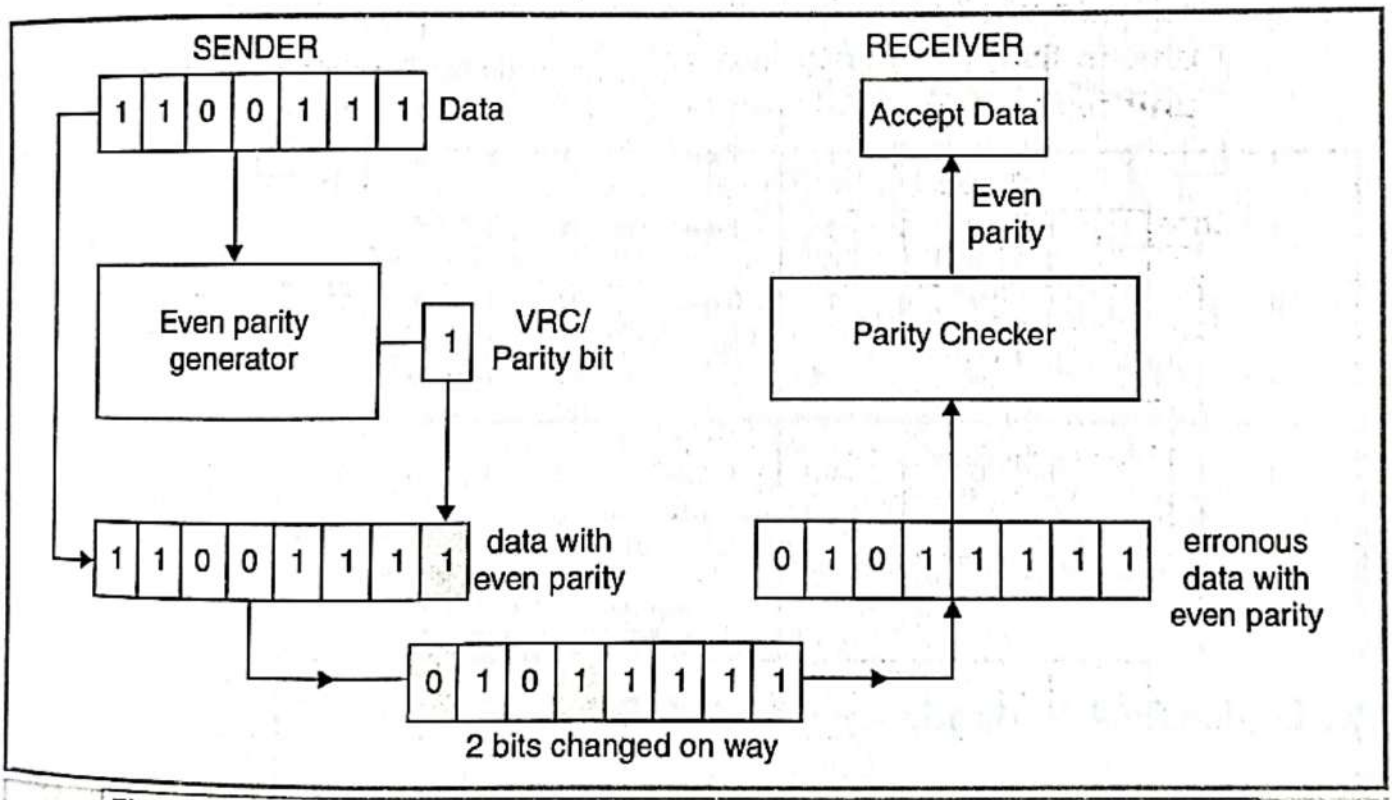


Fig. 5.21 Erroneous data accepted by receiver with number of changed bits = 2

(a) Two-dimensional parity check

- In this method, the data to be sent is organised into table of rows and columns (see fig. 5.22).

1	1	0	0	1	1	1	1
1	0	1	1	1	0	1	0
0	1	1	1	0	0	1	0
0	1	0	1	0	0	1	1
0	1	0	1	0	1	0	1

Row parities

Column parities

1	1	0	0	1	1	1	1
1	0	[1]	1	1	0	1	1 ←
0	1	1	1	0	0	1	0
0	1	0	1	0	0	1	1
0	1	0	1	0	1	0	1

↑

(a) One error affects two parities arrows points to parity bit affected

1	1	0	0	1	1	1	1
1	0	[1]	1	[1]	0	1	1
0	1	1	1	0	0	1	0
0	1	0	1	0	0	1	1
0	1	0	1	0	1	0	1

↑ ↑

(b) Two errors affect two parities

1	[1]	0	0	1	1	1	1 ←
1	0	[1]	1	1	0	1	1 ←
0	1	[1]	1	0	0	1	0 ←
0	1	0	1	0	0	1	1
0	1	0	1	0	1	0	1

↑

(c) Three errors affect four parities

1	1	0	[0]	[1]	1	1	1
1	0	1	1	1	0	1	1
0	1	1	[1]	[0]	0	1	0
0	1	0	1	0	0	1	1
0	1	0	1	0	1	0	1

(d) Four errors cannot be detected

Fig. 5.22 Generating rows and column partities

2. Logitudinal Redundancy Check (LRC)

- In this method, a block of bits is divided into table or matrix of rows and columns.
- In order to detect the error, a redundant row of bits is added to the whole block and this block is transmitted to the receiver.
- The receiver uses this redundant row to detect error. After checking the data for errors, receiver accepts the data and discards the redundant row of bits.

- For example if a block of 32 bits is to be transmitted, it is divided into matrix of four rows and eight columns is shown in fig. 5.23.
- In this matrix of bits, a parity bit (odd or even) is calculated for each column.
- Thus we get a redundant row that contains 8 bits, one for each column (see fig. 5.23)
- This means that 32 bits data plus 8 redundant bits are transmitted to the receiver.

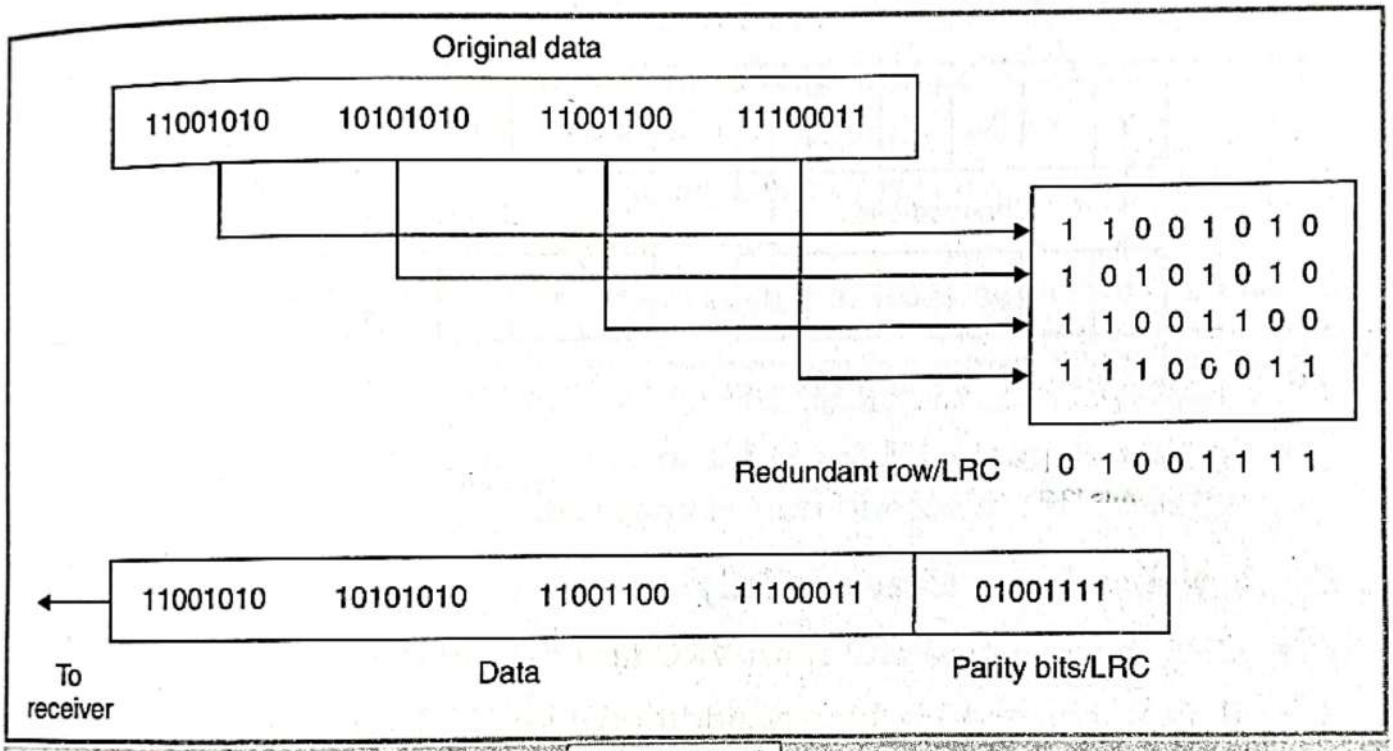


Fig. 5.23 LRC

- When this data reaches the destination, receiver uses LRC to detect error in data.

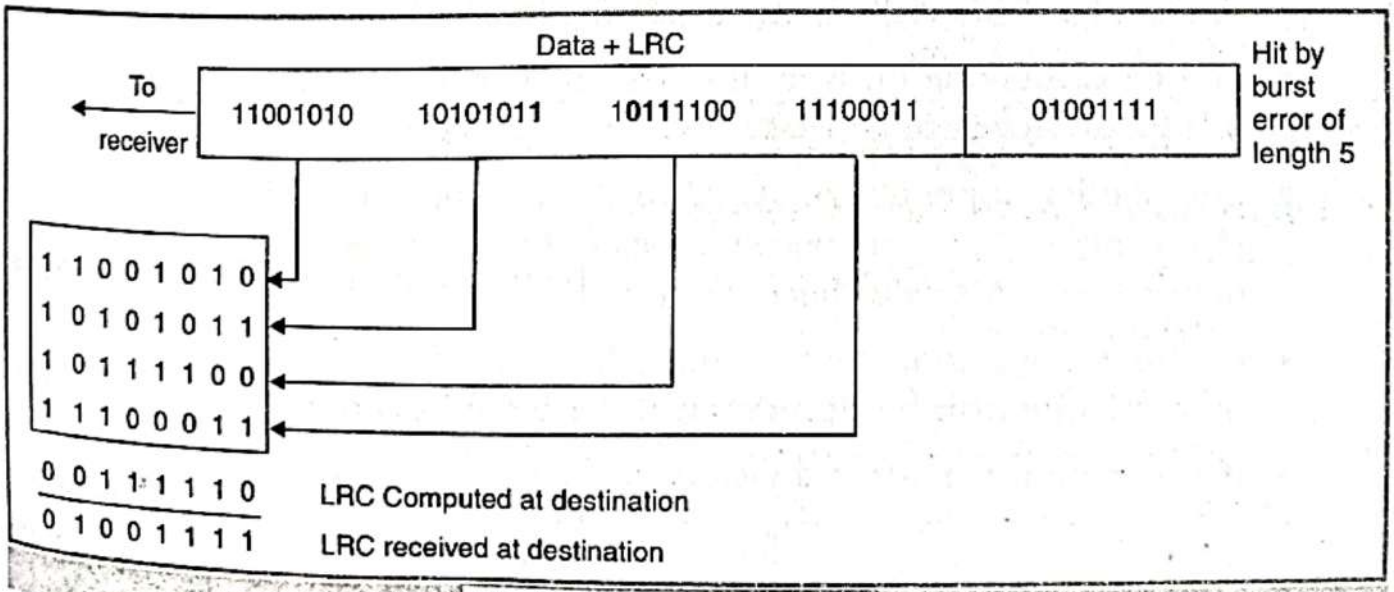


Fig. 5.24 Burst error & LRC

- Now, the LRC received by the destination does not match with newly computed LRC.
- The destination comes to know that data is erroneous, so it discards it.
- The main problem with this method is that, it is not able to detect the error if two bits in a data unit are damaged and two bits in exactly the same position in other data unit are also damaged. For example if data 110011 010101 is changed to 010010 110100 (see fig. 5.25).

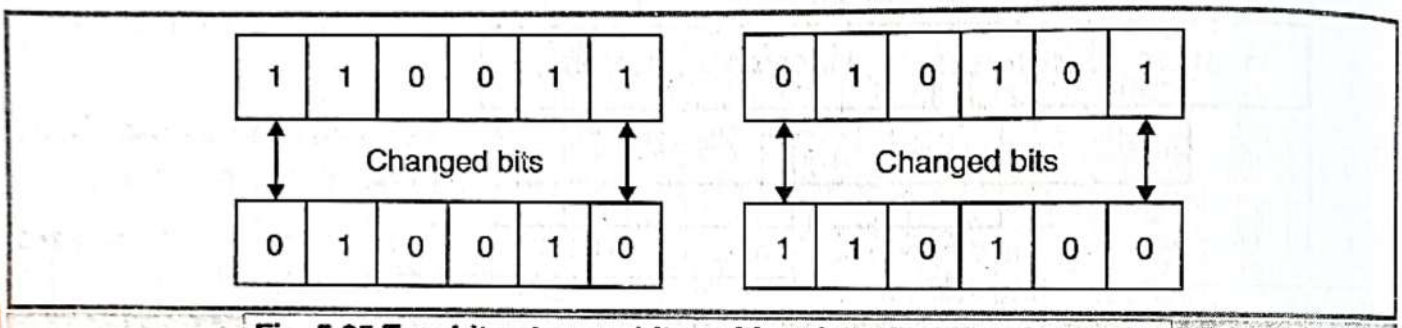


Fig. 5.25 Two bits at same bit position damaged in 2 data units

- In this example 1st & 6th bit in one data unit is changed. Also the 1st & 6th bit in second unit is changed.

2. Cyclic Redundancy Check (CRC)

- CRC is more powerful than VRC and LRC in detecting errors.
- It is not based on binary addition like VRC and LRC. Rather it is based on binary division.
- At the sender side, the data unit to be transmitted is divided by a predetermined divisor (binary number) in order to obtain the remainder. This remainder is called CRC.
- The CRC has one bit less than the divisor. It means that if CRC is of n bits, divisor is of n+1 bit.
- The sender appends this CRC to the end of data unit such that the resulting data unit becomes exactly divisible by predetermined divisor *i.e.* remainder becomes zero.
- At the destination, the incoming data unit *i.e.* data + CRC is divided by the same number (predetermined binary divisor).
- If the remainder after division is zero then there is no error in the data unit & receiver accepts it.
- If remainder after division is not zero, it indicates that the data unit has been damaged in transit and therefore it is rejected.

- The various steps followed in the CRC method are (see fig. 5.26)
  1. A string of n 0s is appended to the data unit. The length of predetermined divisor is n+1.
  2. The newly formed data unit *i.e.* original data + string of n 0s is divided by the divisor using binary division and remainder is obtained. This remainder is called CRC.
  3. Now, string of n 0s appended to data unit is replaced by the CRC remainder (which is also of n bit).
  4. The data unit + CRC is then transmitted to receiver.
  5. The receiver on receiving it, divides data unit + CRC by the same divisor & checks the remainder (see fig. 5.27).
  6. If the remainder of division is zero, receiver assumes that there is no error in data and it accepts it.
  7. If remainder is non-zero then there is an error in data and receiver rejects it.

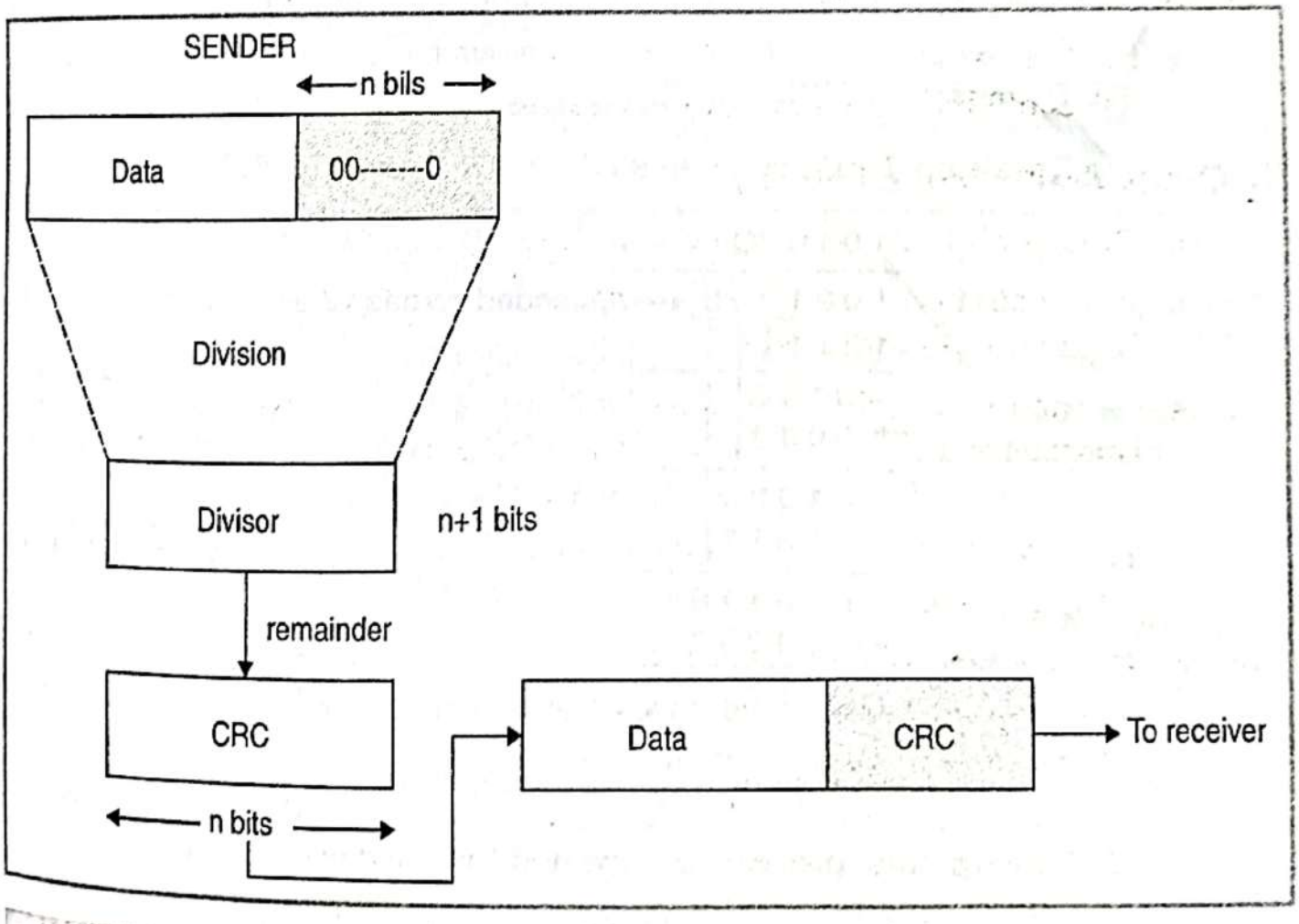


Fig. 5.26 CRC generator at sender side

- For example, if data to be transmitted is 1001 and predetermined divisor is 1011. The procedure given below is used :

1. String of 3 zeroes is appended to 1011 as divisor is of 4 bits. Now newly formed data is 1011000.

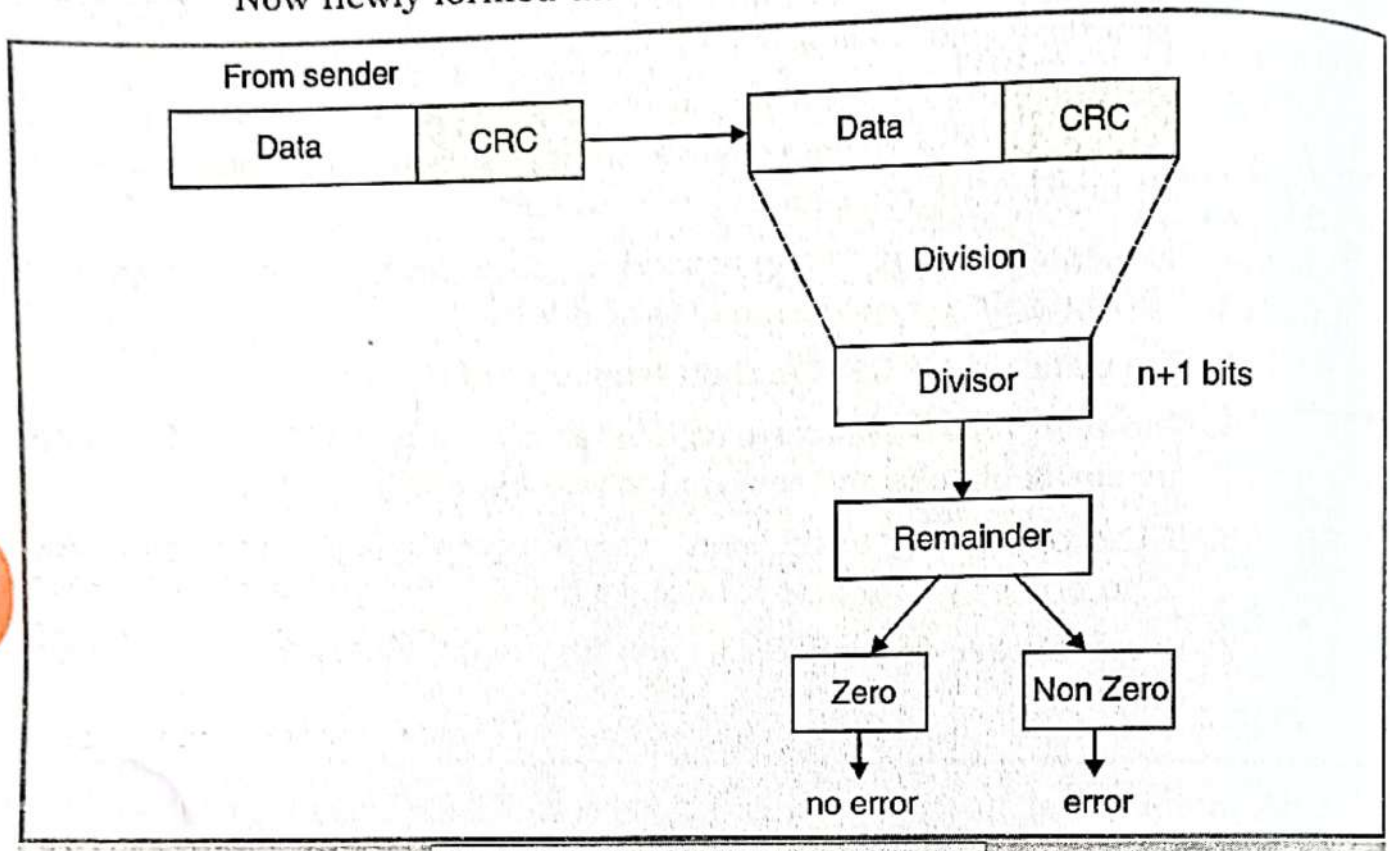


Fig. 5.27 CRC checker at receiver side

1. Data unit 1001000 is divided by 1011. (see fig. 5.28)

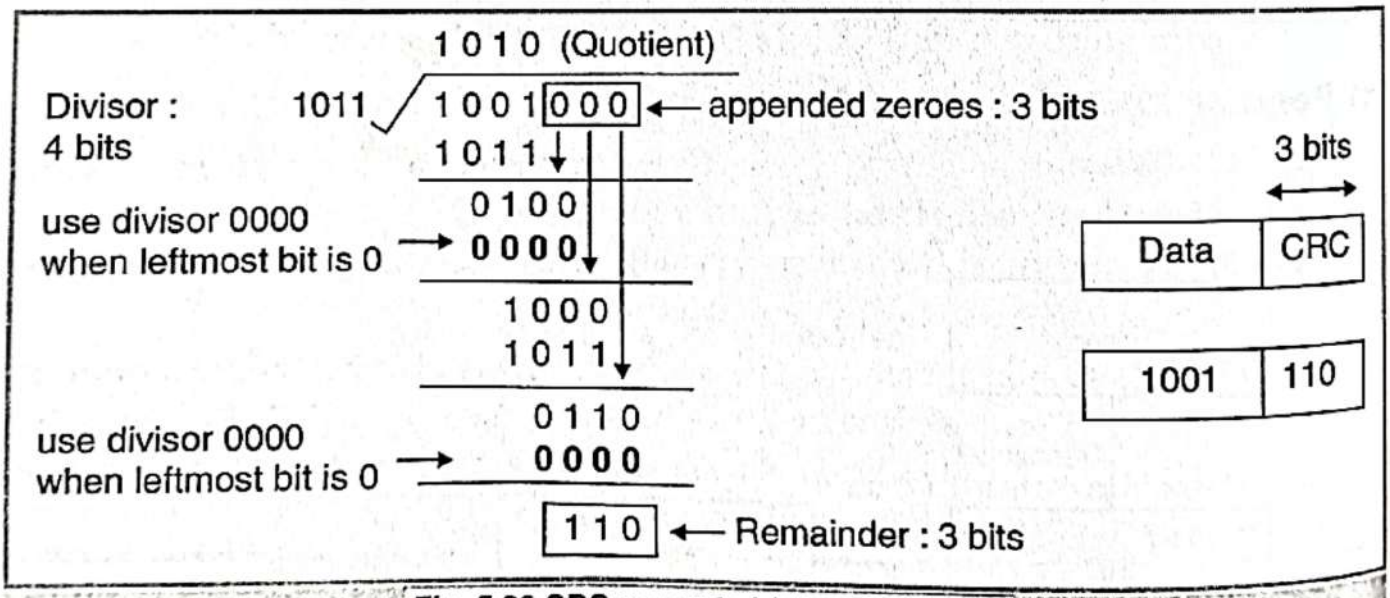


Fig. 5.28 CRC generated (Binary division)

2. During this process of division, whenever the leftmost bit of dividend or remainder is 0, we use a string of 0s of same length as divisor. Thus in this case divisor 1011 is replaced by 0000.
3. At the receiver side, data received is 1001110.
4. This data is again divided by a divisor 1011 (see fig. 5.29)

5. The remainder obtained is 000, it means there is no error.

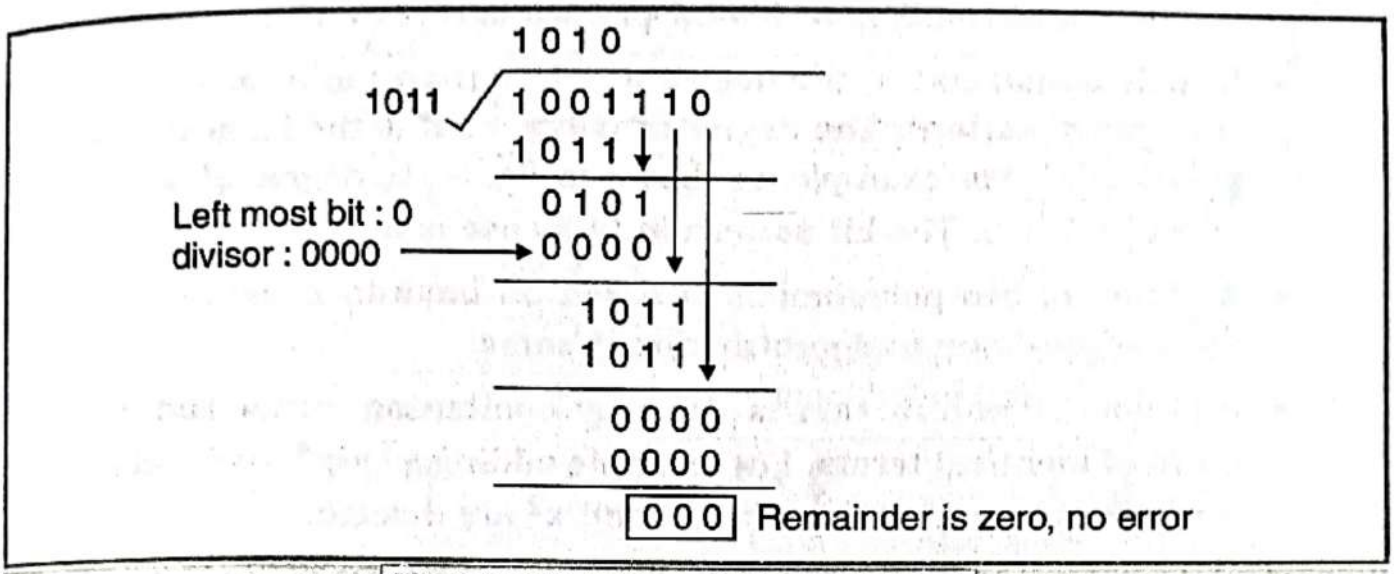


Fig. 5.29 CRC decoded (binary division)

- CRC can detect all the burst errors that affect an odd number of bits.
- The probability of error detection and the types of detectable errors depends on the choice of divisor.
- Thus two major requirement of CRC are :
  (a) CRC should have exactly one bit less than divisor.
  (b) Appending the CRC to the end of the data unit should result in the bit sequence which is exactly divisible by the divisor.

(a) Polynomial codes

- A pattern of 0s and 1s can be represented as a polynomial with coefficient of 0 and 1.
- Here, the power of each term shows the position of the bit and the coefficient shows the values of the bit.
- For example, if binary pattern is 100101, its corresponding polynomial representation is $x^5 + x^2 + 1$ (see fig. 5.30). Figure 5.30 (b) shows the polynomial where all the terms with zero coefficient are removed and x^1 is replaced by x and x^0 by 1.

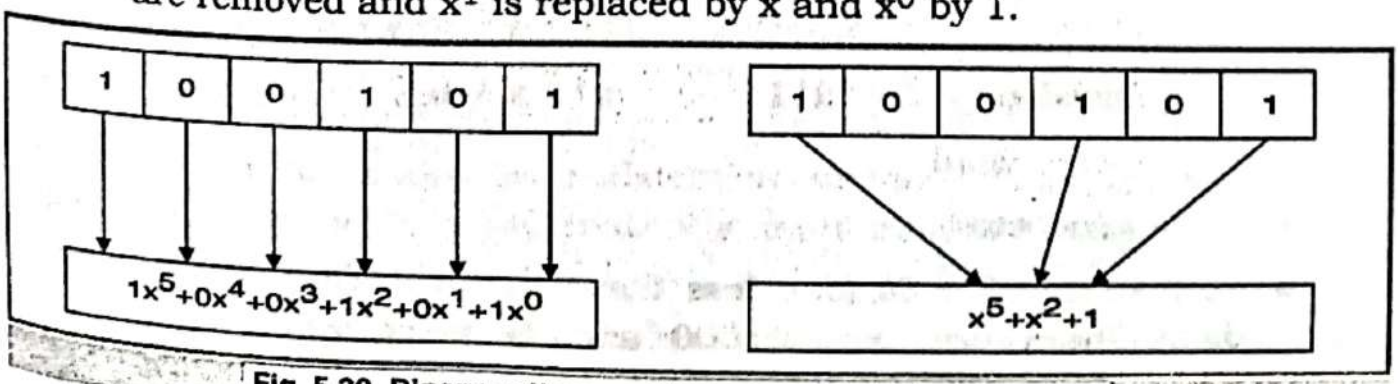


Fig. 5.30. Binary pattern and its polynomial representation

- The benefits of using polynomial codes is that it produces short codes. For example here a 6-bit pattern is replaced by 3 terms.
- In polynomial codes, the degree is 1 less than the number of bits in the binary pattern. The degree of polynomial is the highest power in polynomial. For example as shown in fig. 5.30, degree of polynomial $x^5 + x^2 + 1$ is 5. The bit pattern in this case is 6.
- Addition of two polynomials is based on modulo-2 method. In such as case, addition and subtraction is same.
- Addition or subtraction is done by combining terms and deleting pairs of identical terms. For example adding $x^5 + x^4 + x^2$ and $x^6 + x^4 + x^2$ gives $x^6 + x^5$. The terms x^4 and x^2 are deleted.
- If three polynomials are to be added and if we get a same term three times, a pair of them is detected and the third term is kept. For example, if there is x^2 three times then we keep only one x^2.
- In case of multiplication of two polynomials, their powers are added. For example, multiplying $x^5 + x^3 + x^2 + x$ with $x^2 + x + 1$ yields :

  $(x^5 + x^3 + x^2 + x)(x^2 + x + 1)$

  $= x^7 + x^6 + x^5 + x^5 + x^4 + x^3 + x^4 + x^3 + x^2 + x^3 + x^2 + x$

  $= x^7 + x^6 + x^3 + x$

  In this, first polynomial is multiplied by all terms of second. The result is then simplified and pairs of equal terms are deleted.
- In case of division, the two polynomials are divided as per the rules of binary division, until the degree of dividend is less than that of divisor.

(b) CRC generator using polynomials

- If we consider the data unit 1001 and divisor or polynomial generator 1011 their polynomial representation is :

Data	1001	$x^3 + 1$
Division (polynomial generator)	1011	$x^3 + x + 1$

- Now string of n 0s (one less than that of divisor) is appended to data. Now data is 1001000 and its corresponding polynomial representation is $x^6 + x^3$.

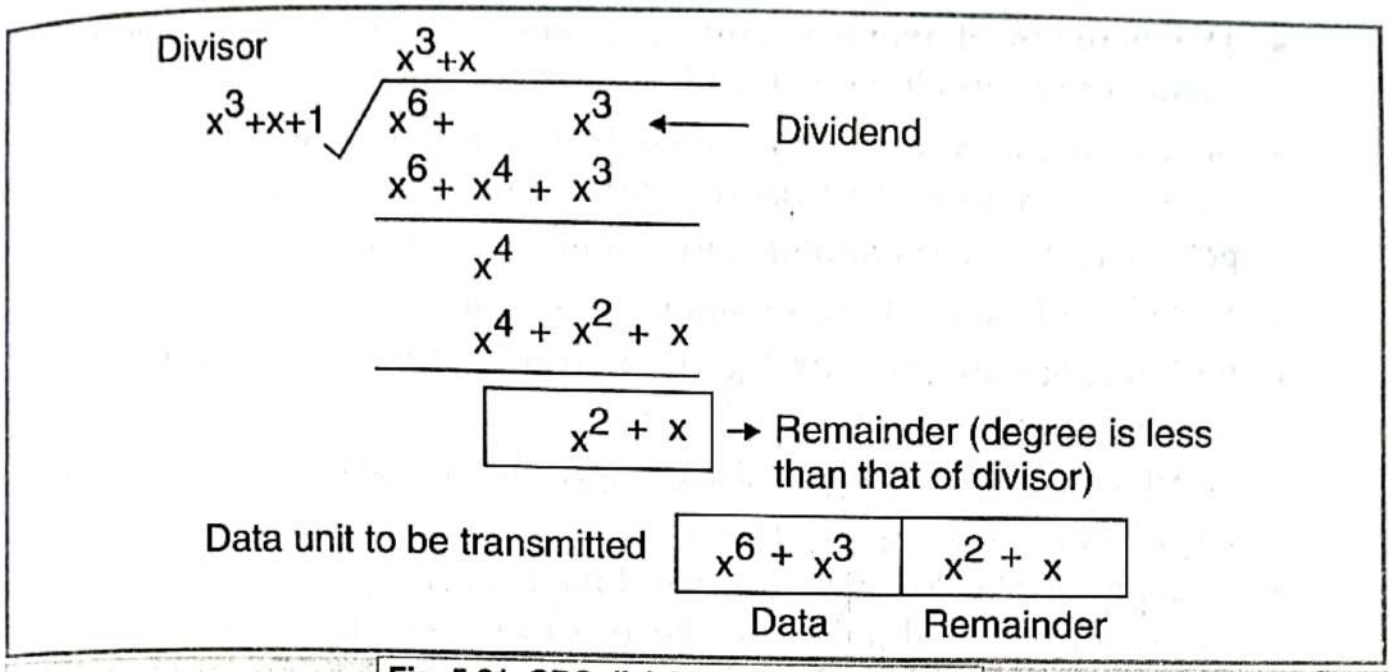


Fig. 5.31. CRC division using polynomial

- The division of x^6+x^3 by x^3+x+1 is shown in fig. 5.31.
- The polynomial generator should have following properties:
  1. It should have atleast two terms.
  2. The coefficient of the term x^0 should be 1.
  3. It should not be divisible by x.
  4. It should be divisible by x+1.
- There are several different standard polynomials used by popular protocols for CRC generation. These are :

Name	Polynomial	Application
CRC-8	$x^8 + x^2 + x + 1$	ATM header
CRC-10	$x^{10} + x^9 + x^5 + x^4 + x^2 + 1$	ATM AAL
CRC-16	$x^{16} + x^{12} + x^5 + 1$	HDLC
CRC-32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	LANs

2. Checksum

- Checksum is the error detection method used by upper layer protocols and is considered to be more reliable than LRC, VRC and CRC.
- This method make use of checksum generator on sender side and checksum checker on receiver side.

- At the sender side, checksum generator divides the data into equal subunits of n bit length. This length is generally of 16 bits.
- These subunits are then added together using one's complement method. This sum is also of n bits.
- The resultant sum is then complemented.
- This complemented sum (called checksum) is appended to the end of original data unit and is then transmitted to receiver (see fig. 5.32).
- Receiver upon receiving data + checksum pass it to checksum checker.
- Checksum checker divides this data unit into various subunits of equal length and add all these subunits. These subunits also contain checksum as one of the subunit (see fig. 5.32).
- The resultant sum is then complemented.
- If the complemented result is zero, it means the data is intact and error free.
- If the result is non-zero, the data contains an error and receiver rejects it.

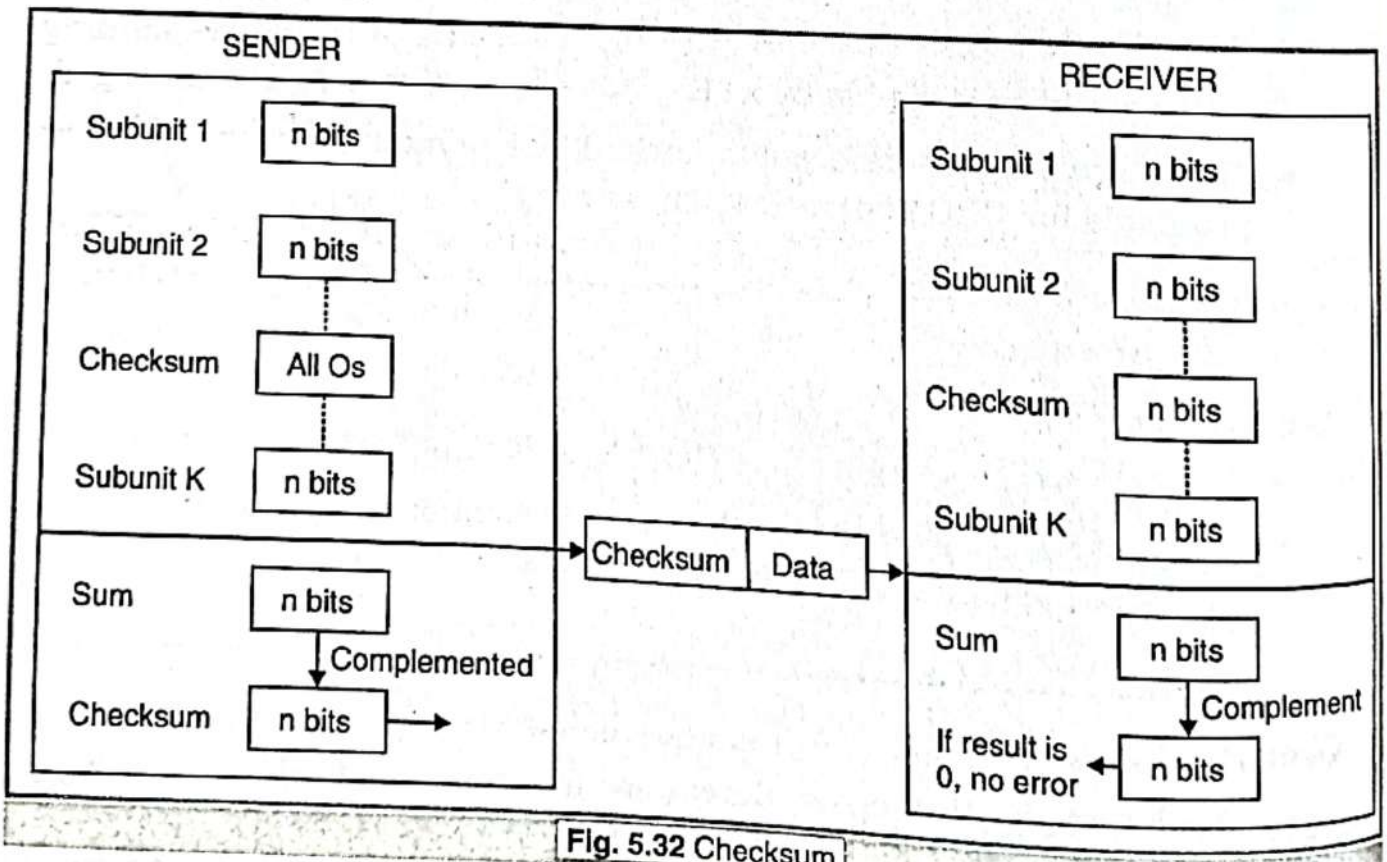


Fig. 5.32 Checksum

- For example, if the data unit to be transmitted is 10101001 00111001, the following procedure is used at the sender site and receiver site.

Sender site :

1 0 1 0 1 0 0 1	subunit 1
0 0 1 1 1 0 0 1	subunit 2
1 1 1 0 0 0 1 0	sum(using 1s complement)
0 0 0 1 1 1 0 1	checksum (complement of sum)

Data transmitted to receiver is :

10101001 00111001 00011101

Data Checksum

Receiver site :

1 0 1 0 1 0 0 1	subunit 1
0 0 1 1 1 0 0 1	subunit 2
0 0 0 1 1 1 0 1	checksum
1 1 1 1 1 1 1 1	sum
0 0 0 0 0 0 0 0	sum's complement

Result is zero, it means no error.

- The checksum detects all errors involving an odd number of bits as well as the error involving even number of bits.
- The main problem in this method is that the error goes undetected if one or more bits of a subunit are damaged and the corresponding bit or bits of opposite value in second subunit are also damaged. This is due to the fact that the sum of those columns remain unchanged.

For example, if the data transmitted along with checksum is 10101001 00111001 00011101. But the data received at destination is 00101001 10111001 00011101

Receiver site

0 0 1 0 1 0 0 1	1st bit of subunit 1 is damaged
1 0 1 1 1 0 0 1	1st bit of subunit 2 is damaged
0 0 0 1 1 1 0 1	checksum
1 1 1 1 1 1 1 1	sum
0 0 0 0 0 0 0 0	Ok 1s complement

Although data is corrupted, error is undetected.

5.5.3 Error Correction

- Error correction means rectifying all the bits that have been altered or changed during the transmission of data. Error correction can be done in two ways :

1. By retransmission of data from sender to receiver.
2. By the use of error correction codes at the receiver side.

- In this section, error correction using correction codes will be undertaken.
- Error correcting codes are more sophisticated than error detecting codes and require more redundancy bits.
- The error correction codes are generally used to correct errors up to three bits only.
- For burst errors or multiple bit errors the number of redundancy bits required is very high. So in such cases retransmission in the preferred method.

5.5.3.1 Single bit error correction

- Errors can be detected by the addition of a redundant (parity) bit to the data unit.
- A single additional bit can detect single-bit errors in any sequence of bits because it must distinguish between only two conditions : error or no error.
- A bit has two states (0 and 1).
- However two states are not enough to correct it.
- An error occurs when the receiver reads a bit 1 as a 0 or a 0 bit as a 1.
- To correct the error, the receiver simply reverses the value of the altered bit.
- In order to correct errors, it must be known which bit is in error.
- For example, to correct a single bit error in an ASCII character, the error correction code must determine which of the seven bits has changed. In this case, we need to have eight different states.
- These states are : no error, error in position 1, error in position 2, error in position 3 and so on up to error in position 7.
- Therefore to show all these eight states, we required more than one redundant bit.
- In order to indicate these eights states (000 to 111) we need three redundant bits. Therefore the total number of bits that will be transmitted in ASCII Code becomes ten (seven plus three).
- However an error can also occur in the redundant bit themselves.

- In that case three redundant bits are not sufficient to indicate which bit has undergone change as three redundant bits can indicate eight states only.

5.5.3.2 Redundancy bits

- To calculate the number of redundancy bits (r) required to correct a given number of data bits (m) we must find a relationship between m and r. If there are m data bits and r redundancy bits in data, the length of resulting code is m + r (see fig. 5.33)

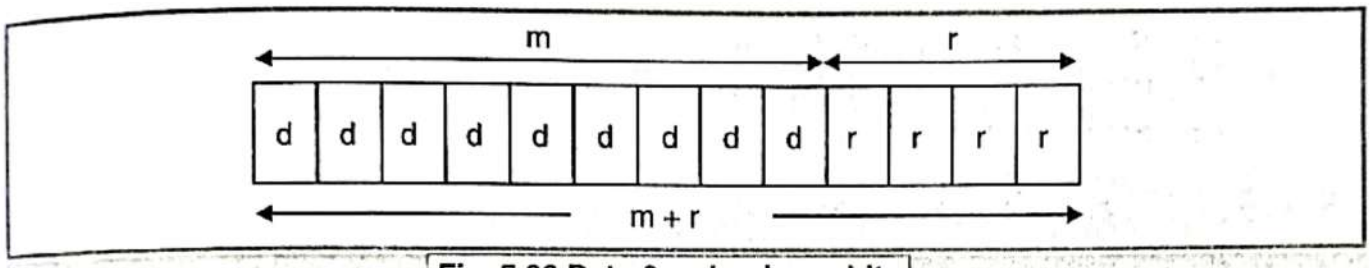


Fig. 5.33 Data & redundancy bits

- Now, if the total number of bits in transmittable unit is m+r, then r must be able to indicate at least m + r + 1 different states. Of these, one state means no error and m+r states indicate the location of an error in each of the m+r positions.
- Therefore r bits indicate 2^r difference states. Thus 2^r must be equal to or greater than m+r+1.

$$2^r \quad m + r + 1$$

- The value of r is determined by plugging in the value of m *i.e.* the original length of data unit to be transmitted. For example if value of m is 7, the smallest value of r that can satisfy this equation is 4.

$$2^4 \quad 7 + 4 + 1$$

- The relationship between the some possible m values (data bits) and the r value (redundancy bits) is shown in table:

No. of data bits (m)	No. of redundant bits (r)	Total (m+r)
1	2	3
2	3	5
3	3	6
4	3	7
5	4	9
6	4	10
7	4	11

5.5.3.3 Hamming Code

- Hamming code is technique developed by R.W. Hamming for error correction.
- This method corrects the error by finding the state at which the error has occurred.

5.5.3.3.1 Determining the positions of redundancy bits

- Till now, we know the exact number of redundancy bits required to be embedded with the particular data unit.
- We know that to detect errors in a 7 bit code, 4 redundant bits are required.

Now, the next task is to determine the positions at which these redundancy bits will be placed within the data unit.

- These redundancy bits are placed at the positions which correspond to the power of 2.
- For example in case of 7 bit data, 4 redundancy bits are required, so making total number of bits as 11. The redundancy bits are placed in position 1, 2, 4 and 8 as shown in fig. 5.347

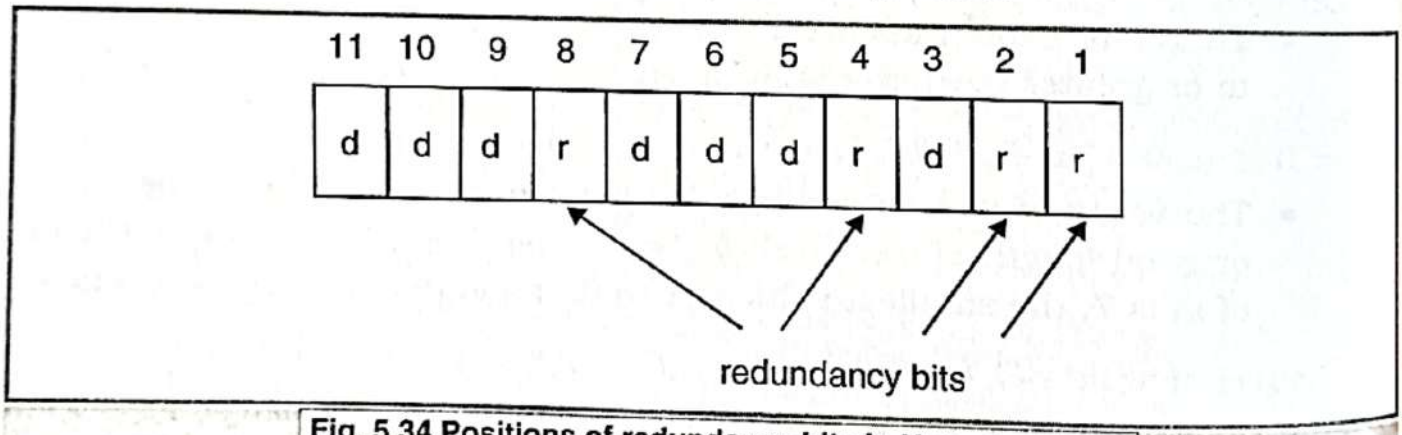


Fig. 5.34 Positions of redundancy bits in Hamming Code

5.5.3.3.2 Heaming Distance

- Consider two code words having the elements

Code word No. 1	11	0	1	0	1	0	0
	↕			↕		↕	
Code word No. 2	10	1	0	1	1	1	0

Note that the bits 2, 4, 8 are different from each other so heaming distance is 3 **Minimum distance** d_{min} :

- The minimum distance is defined as the smallest heaming distance between any pair of code word.

- The error detection is always possible when the number of transmission errors in the code word is less than d_{min} (minimum distance), because then the erroreous word is not a valid word.
- But when the number of error equals or exceeds d_{min}, the erroneous code word may correspond to another valid code word and errors cannot be detected.
- The error detection and correction capability of code depends on the d_{min} (minimum distance) as show in the table.

Role of d_{min} for detection and correction of error

Detection upto "s" error per word	$d_{min} \geq (s+1)$
Correct upto "t" error per word	$d_{min} \geq (2\ t+1)$
Correct upto "t" error and detect s >t errors per used	$d_{min} \geq (t+s+1)$

Example. For a hamming distance of 5 how many errors can be detected ? How may errors can be corrected ?

Solution

Let us assume minimum heaming distance

$$d_{min} = 5$$

(i) Number of errors that can be detected can be obtained from

$$(S+1) \leq d_{min}$$

$$\therefore \quad S \leq 5-1$$

$$S \leq 4$$

Thus at most 4 errors can be detected

(ii) Number of errors that can be corrected (t) can be obtained from

$$(2t+1) \leq d_{min}.$$

$$\therefore \quad t \leq 2$$

Thus at most 2 errors can be corrected.

Example. For a hamming distance of 6 how many error can be detected ? How many can be corrected ?

Solution

1. No. of errors that can be detected as

$$S \leq d_{min} - 1$$

$$\therefore \quad S \leq 6-1 \qquad s = 5$$

2. No. of errors can be corrected

$$t \le \frac{d_{min} - 1}{2} = \frac{6-1}{2} = \frac{5}{2} = 2.5$$

So at most 2 errors can be corrected.

5.5.3.3.3 Forward Error Correction :

- In this method the code set is so designed that it is possible for the receiver to detect as well as to correct the errors.
- The errors are detected with their location.
- Once the error is located the bit is reversed to correct the error. There are many such methods available.

Some commonly used forward error correction methods are as follows.

(*a*) Block Parity method

(*b*) Hamming code method

(*c*) Convolution code method.

(a) Block Parity Method :

- This method is based on the block sum check detection method.
- Two parity check bits are generated for each data byte.
- These two parity bits are known as transverse redundancy check bit and longitudinal redundancy check bit.
- The TRC is generated over the row and LRC is appended to each data block.
- If there is any error in transmitted data block the longitudinal redundancy check fails in one of the rows and transverse redundancy check fails in one of the column.
- The bit which is common to this row & column is the bit with error.
- If there is a single error it can be corrected by locating and then inverting the bit but multiple errors can not be detected with exact location neither they can be corrected.
- Let us check the received bit stream as follows & correct the error if any - 111010101, 110101011, 111101011, 110101110, 001011011, with even party schemes.

	1	1	1	0	1	0	1	0	1
	1	1	0	1	0	1	0	1	1
WRONG PARITY →	1	1	1	1	0	1	0	1	1
	1	1	0	1	0	1	1	1	1
	0	0	1	0	1	1	0	1	0

↑ WRONG PARITY

Fig. 5.35

- The bit at cross section as '1' is the wrong bit it can be inverted to correct data word.

(B) *Hamming code method :*

- This method detects the error as well as its position too.
- The identified bit is inverted to get the correct code word.
- The code is known as block code because the data to be transmitted is treated as single frame or block for encoding and decoding process.
- During encoding a few additional bits are added to the data word to obtain a large code word.
- If data word is of size n bits and code word is a size p bits then the code is known as (p, n) hamming code.
- The ratio of n/p is known as code efficiency while is known as redundancy.
- For example let for a 4 bit data word, 3 parity bits are used to make 7 bit code word then the code is known as (7,4) block code or (7,4) hamming code.
- The code efficiency or code rate will be 4/7 and redundancy.
- Thus in a code word of length p there are n data bits m parity bits such that.

  $p = m+n$

  and $2^m >> p+1$
- The generated code word is transmitted and at receiving end each data bit is checked by a number of parity bits.

- For checking the data bits the rule used is such that for each data bit position the sum of powers of 2 finds the parity bits position.
- For example - for data bit D_3 the parity bits. P_1 and P_2 will be used because $3 = 2^0 + 2^1 = 1 + 2$ and hence 1 & 2. Data bit D_5 will use $2^0 + 2^2$ i.e. 2 and 4 i.e. P_2 and P_4 and Data bit D_7 will use $2^0 + 2^1 + 2^2$ i.e. P_1, P_2 and P_4.
- Thus for a code words of length 7 as

  P_1 P_2 P_3 P_4 D_5 D_6 D_7
- There will be three partiy bit as P_1, P_2 and P_4 and four data bit D_3, D_5, D_6 and D_7.
- For generaling P_1, P_2 and P_4 as well as for checking data bits the following combination will be used.

(*A*)
1. P_1, P_2, D_3
2. P_1, P_4, D_5
3. P_2, P_4, D_6
4. P_1, P_2, P_4, D_7

or the same can be written in other way as

(*B*)
1. P_1, D_3, D_5, D_7
2. P_2, D_3, D_6, D_7
3. P_4, D_5, D_6, D_7

- For 7 bit 8 data 1, 2, 4, 7 are parity bits rest are data bits.
- The scheme (*B*) is generally used for producing party bits and also for checking data bits.

(c) Convolution Code

- The hamming single bit code can correct single bit errors and can detect two bit errors but multiple bit errors can not be detected using it.
- The same is the problem with block parity method. One more thing with block codes is that the output codeword depends only on the current data block being encoded so they are memory less codes.
- On the other hand convolution codes use continuous stream of source bit and produce a continuous stream of encoded bits.
- Each bit in encoded sequence depends on previous sequence of source bits.

- Hence convolution code method uses some form of memory.
- Shift register of a fixed finite length known as constraint length is used for this purpose.
- The convolution operation is performed using one or more exclusion or gates.
- For example a convolution code of constraint length 5 is generated bit by bit using last 5 data bits.
- The shift register. A convolution encoder of constraint length 3 is shown in fig. 5.36.
- It has 3 bit shift register of provide memory, two ex-OR gates to provide modulo-2 addition i.e. convolution operation.
- There are two outputs hence the encoder is known as 1/2 (n/p) convolution encoder with constraint length 3.
- Here n is size of data word & p is size of code word.
- As the encoder uses memory so some specific method/s are required to determine the output encoded bit sequence for a known input sequence.

One of the three techniques can be used for this purpose. They are –

- Tree diagram.
- State diagram.
- Trellis diagram

Let us discuss these methods in brief.

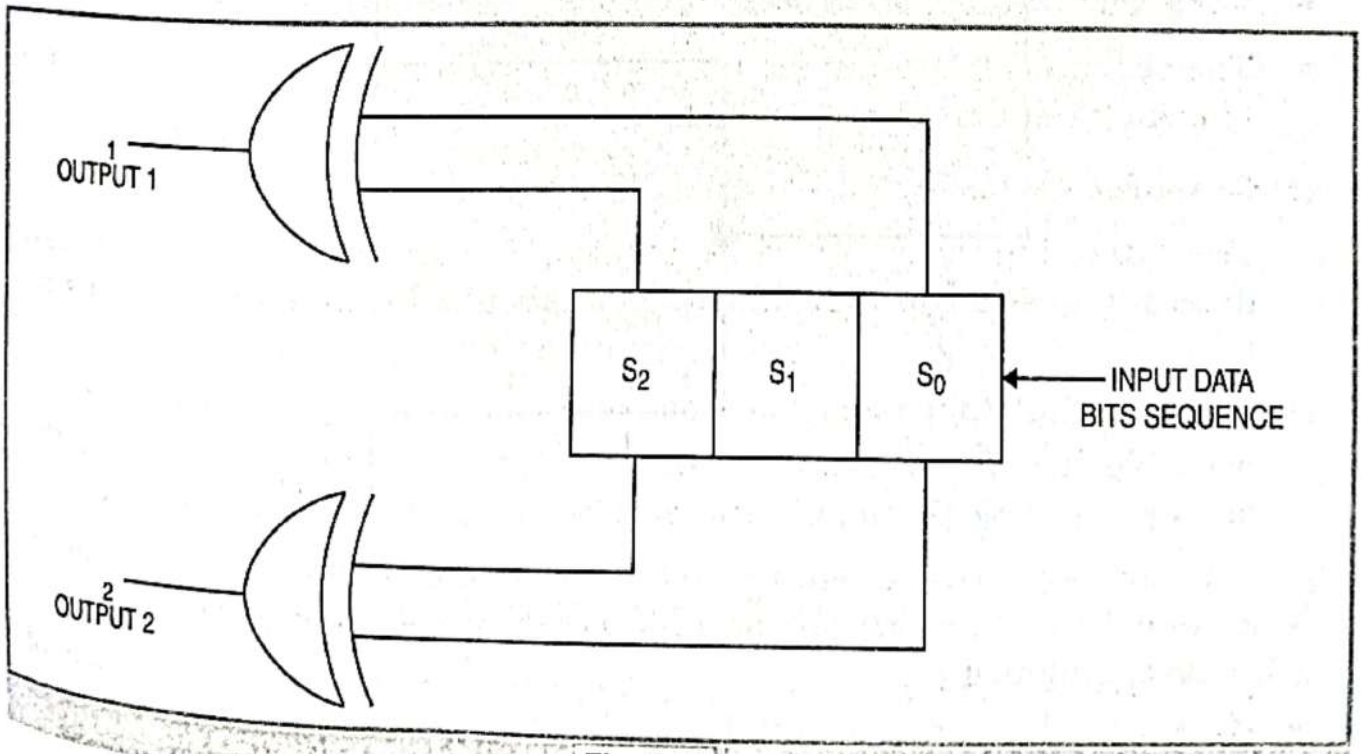


Fig. 5.36

(I) Tree Diagram :

- A tree starts from a single point-known as root.
- The data bits equal to constraint length are input from this point.
- The branching point of tree is known as node.
- The upper branch corresponds to zero input bit and lower branch to input bit '1'. The output is shown at outer side of each branch line.
- See fig. 5.37 for initial data bits = 000 in a constraint length 3 encoder.

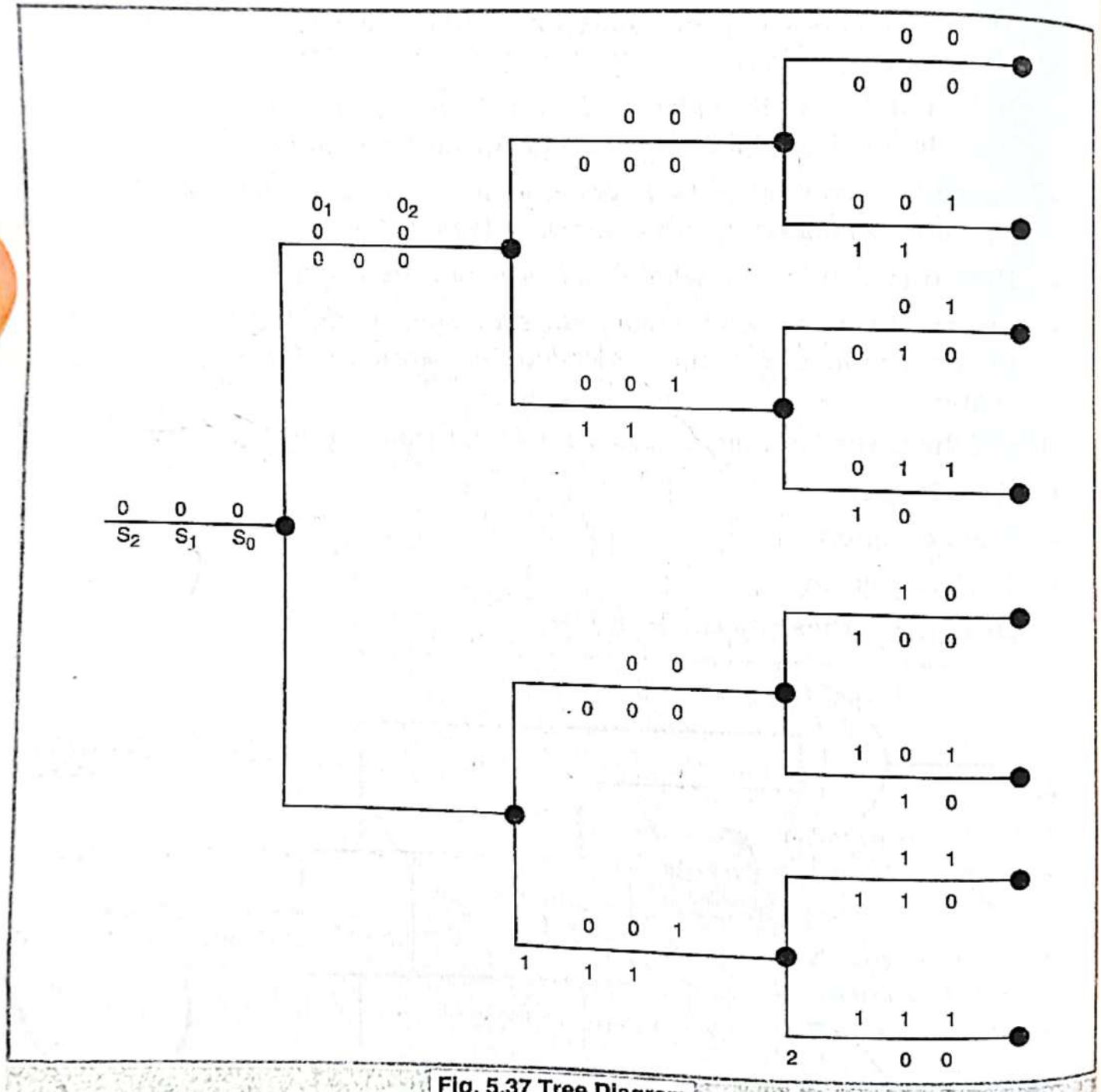


Fig. 5.37 Tree Diagram

(II) State Diagram :

- In a tree diagram the number of branches doubles for each new bit

but after second branch level the tree is repetitive because there are only four unique outputs possible.

- They are known as states.
- If we represent these output states as A = 00, B = 01, C = 10 and D = 11 and an zero representing state transmitted for input bit (0 or 1), label on each zero represents input bit/output bits.
- The state diagram for these conditions is shown in fig. 5.38.
- If initial state of encoder is 00 and data bit sequence is 11101011 then the encoded output will be 1110011111010010.

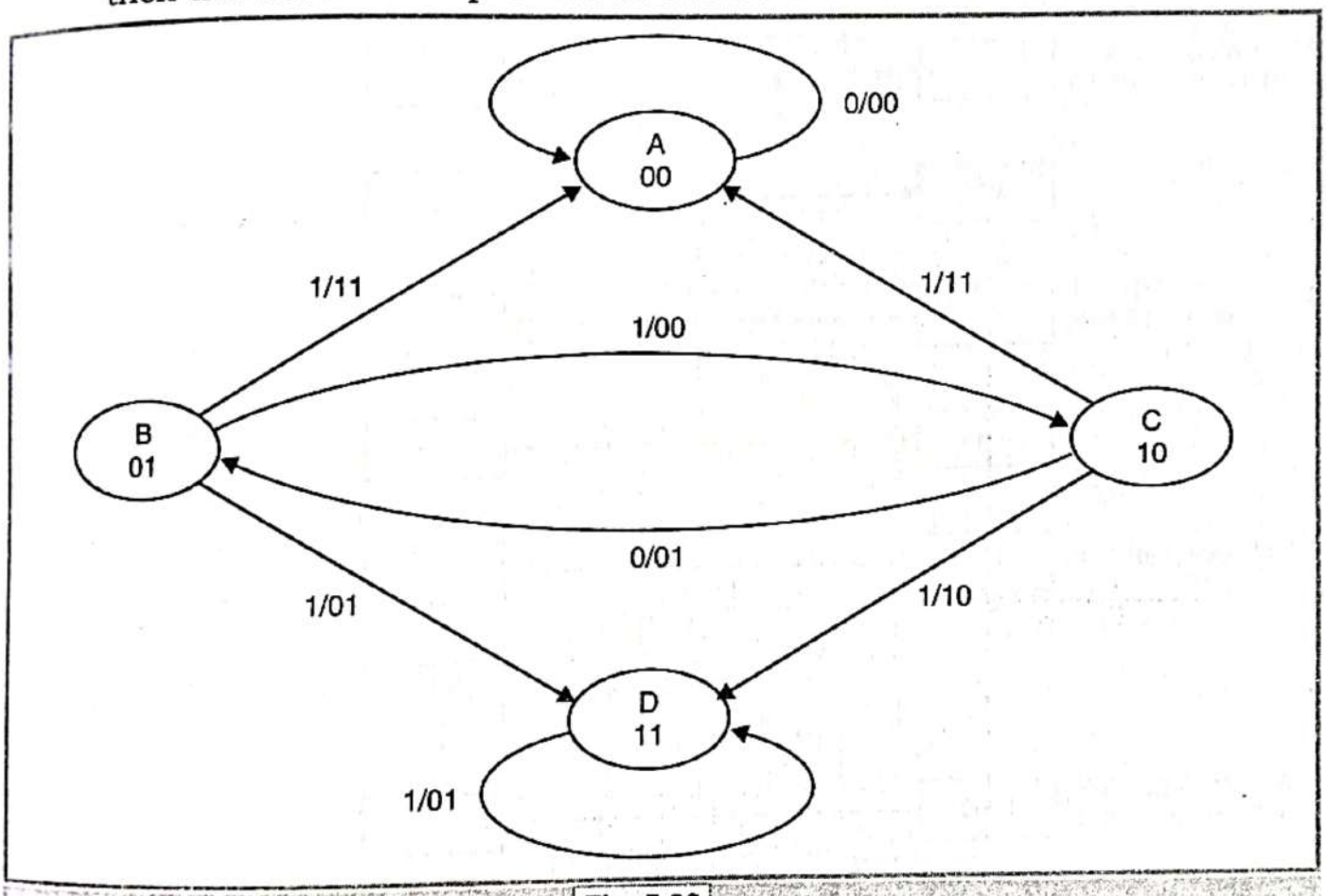


Fig. 5.38

(III) Trellis Diagram :

- Once the states of encoder are identified with the help of tree diagram the trellis diagram can be drawn.
- A trellis diagram shows the outputs of encoder for all the possible input bit sequences.
- It is an alternative way of representing states of state diagram.
- While making trellis is diagram a convention used here is that all arrows for down ward transition represent a '1' input bit and upward transition represent '0' input bit.
- With this convention the trellis diagram for states of fig 5.37 is shown

in fig 5.38 and the convolution code generation is shown in following table.

Present State	Input Bit	Next State	Output Bit
A	0	A	0 0
A	1	C	1 1
C	0	B	0 1
B	1	C	0 0

From table it is clear the encoded output bit sequence is 00110100

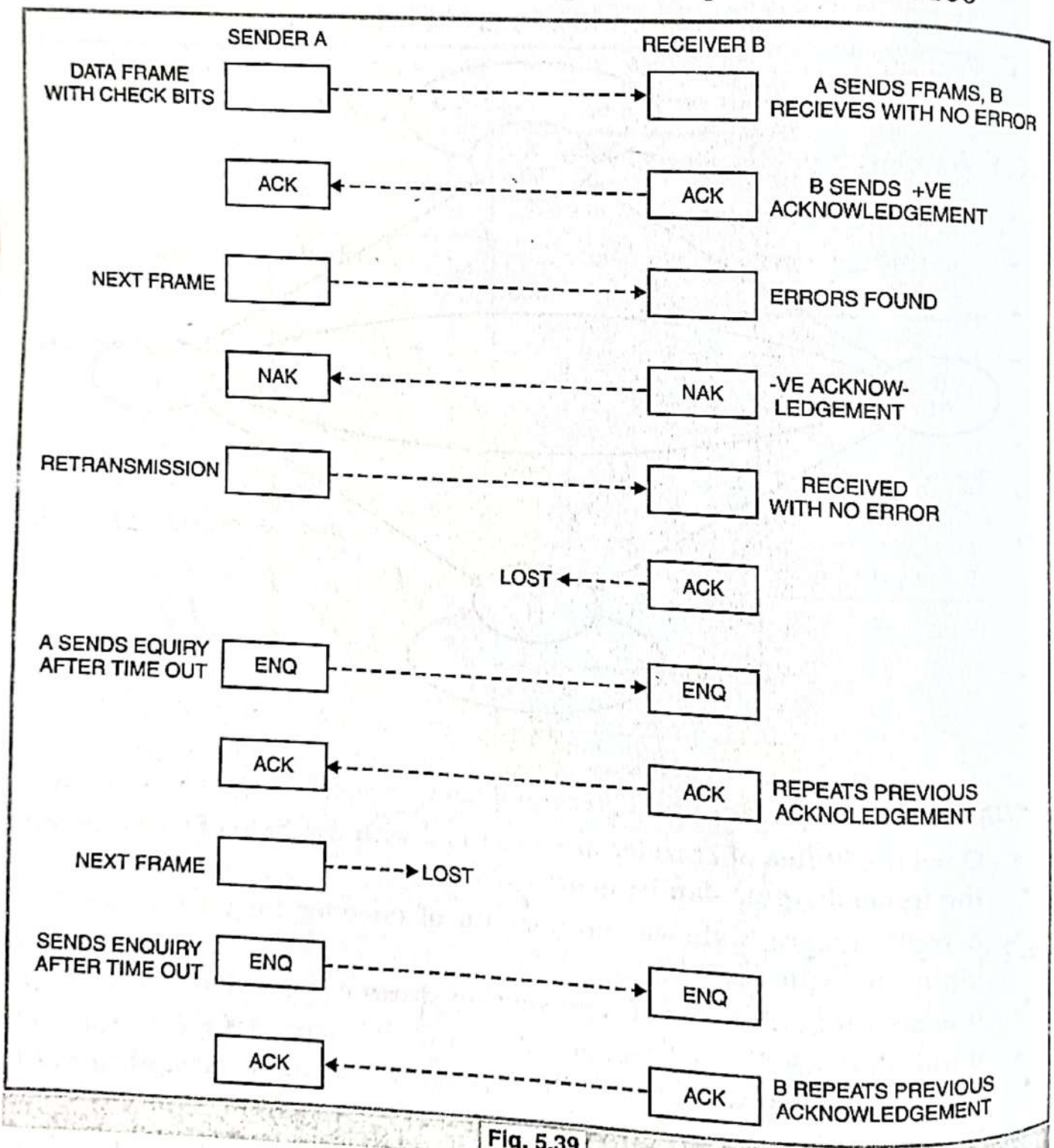


Fig. 5.39

- At receiving end the trellis diagram is searched for received code sequence selecting output sequence nearest to the received sequence.
- The minimum distance path thus obtained corrects the data bit received wrongly.

5.6 ELEMENTARY DATA LINK PROTOCOL

- In general the word protocol referes to the set of rules or conventions for executing particular task.
- A protocol in data communication is the set or rules of specification used to implement one or more layers of the OSI model.

5.6.1 An Unristicted Simplex Protocol

- It is the simplest possible protocol.
- The transmission of the data takes place in only one direction
- It is unidirectional protocol.
- It is always assumed that network layer of sender and receiver are always ready.
- It is also assumed that processing time is ignored.
- It has enough buffer space.
- It is assumed that the channel is having no noise *i.e.* noise free. So it does not lose any frame.
- All this is unrealistic.
- This protocol is also called 'Utopia."
- This protocol has two different procedures, namely a sender and receiver.
- Sender runs in data link layer machine and receiver runs in data link layer machine.
- No sequence or acknowledgement is used.

5.6.2 Simplex Stop and Wait Protocol

- In the unrestricted simplex protocol the assumption that the receiving network layer can process the data with zero processing time.
- In this protocol it is assumed that a finite processing time is essential.
- It is also assumed that the communication channel is assumed to be noise free and communication is simplex *i.e.* only in one direction.

- This protocol deals with the problem of flooding *i.e.* how to prevent the sender from flooding the receiver with data transfer faster than its processing speed.
- In this a small dummy frame is sent back from the receiver to the transmitter to indicate that it can send the next frame.
- The transmitter sends one frame and then wait for the acknowledgement. Hence the name is stop and wait.
- The best thing about this protocol is that the incoming frame is always an acknowledgement.

5.6.3 Simplex Protocol for Noisy Channel

- In this protocol it is assumed that the sender's window and receiver's window need not be of the same size.
- They may be of same size or they can grow or shrink as the frames are sent or received.
- The receiving window corresponds to frames the receiver may accept. Any frame that falls out of the receiving window is discarded.

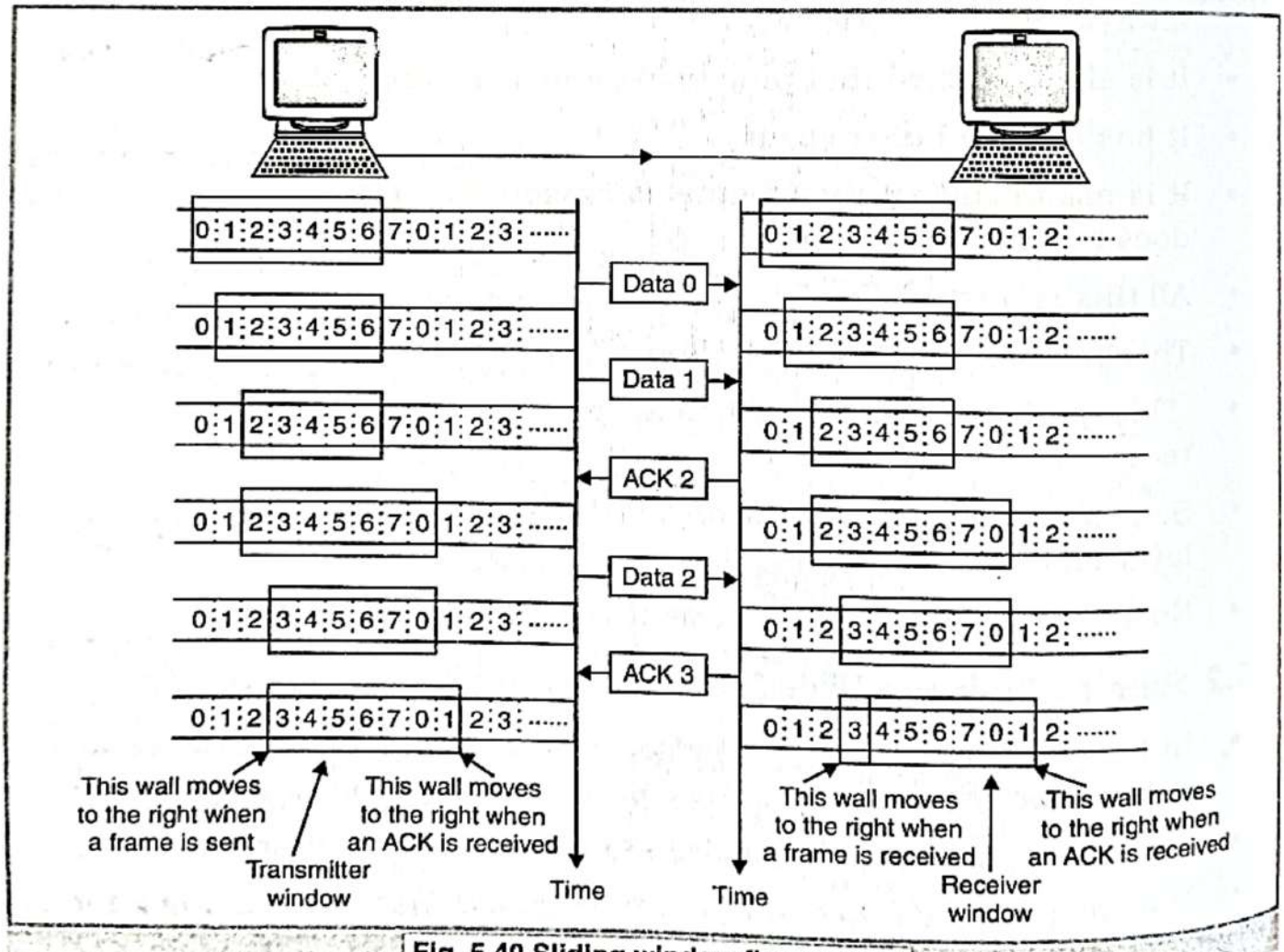


Fig. 5.40 Sliding window flow control

- At the beginning of the transmission of the frames the transmitter, the receiver window contains $n-1$ spaces for frames.
- The sliding window of the receiver shrinks from left when frame of the data are received.
- The sliding window of the receiver expends to right when acknowledgements are sent.
- The receiver window therefore represents not the number of frames received but the number of frames that may still be received before an ACK must be sent. e.g. of frames 0 through 3 (i.e. 4) have arrived but have not been acknowledged the window in the receiver contain 7 – 4 = 3 frame space.
- As each ACK is sent out, the receiving window expands to include as many as new spaces as newly acknowledged frames.
- The window expands to include a number of new frame spaces (5–2 = 3) equal to the number of previously acknowledge frames.

5.6.4 A one bit sliding window protocol

- This protocol is called one bit because the maximum window size is equal to 1.
- It uses the stop wait technique. The sender sends one frame and waits to get its acknowledgement.
- Only after receiving its acknowledgement, it transmits the next.
- Hence one bit sliding window protocol is called stop and wait protocol. Communication channel is noisy and the error may occur during the data travelling over it.
- Framers may either be damaged or 1ost completely.
- In this protocol, the sender waits for a positive acknowledgement before advancing to the next data item.
- So it is called as PAR (P→ Positive, A→ Acknowledgement with R→ Retransmission) or ARQ (A → Automatic RQ → Request)
- One thing should be noted that due to Retransmission, there is always possibility of duplication of frame at the receiving end. To avoid this, the sender puts the sequence.
- There are some added features for retransmission which are added to the basic flow control mechanism.

1. The transmitter have a copy of last frames transmitted until it receives an acknowledgement for those frames.
2. For identification purpose both data and ACK frames are numbered alternatively 0 and 1. A data "0" is acknowledged by an ACK 1 frame indicating that the receiver bas received data 0 and is expecting that next data frame #1.
3. If an error occurs while transmission, the receiver sends a NAK frame back to the transmitter for retransmission of the corrupted frame. NAK frame which are not numbered tell the transmitter to retransmit

the header of each frame it sends.

- The receiver will check the sequence number of each arriving frame to check the possibility of duplication. If a frame is duplicated then this receiver will discard it.
- The operations can be divided into two works :

1. Normal Operation 2. Time Out

1. Normal Operation.

- After transmission one packet, the sender waits for acknowledgement (ACK) from the receiver before transmitting in next packet.
- In this way, the sender can recognise that the previous packet is transmitted successfully and we could say " stop and wait"
- It guarantees reliable transfer between nodes.
- To support this, the sender keeps the reorder of each packet it sends.
- Also to avoid the confusion caused by delayed or duplicate ACKs "Stop-and-wait" sends each packet with unique sequence number and receives that numbers in each ACKs.

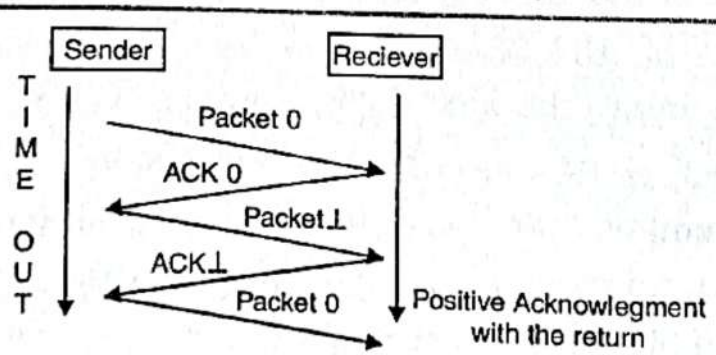


Fig. 5.41 Positive Acknowledge with retransmission

2. Time Out

- If the sender does not receive ACK for the previous sent packet after a

certain period of time, the sender time out and retransmits that packet again.

- There are two cases when the sender does not receiver "ACK"
  1. When ACK is lost
  2. When the frame itself is not transmitted.
- To support this feature, the sender keeps the timer for each packet.

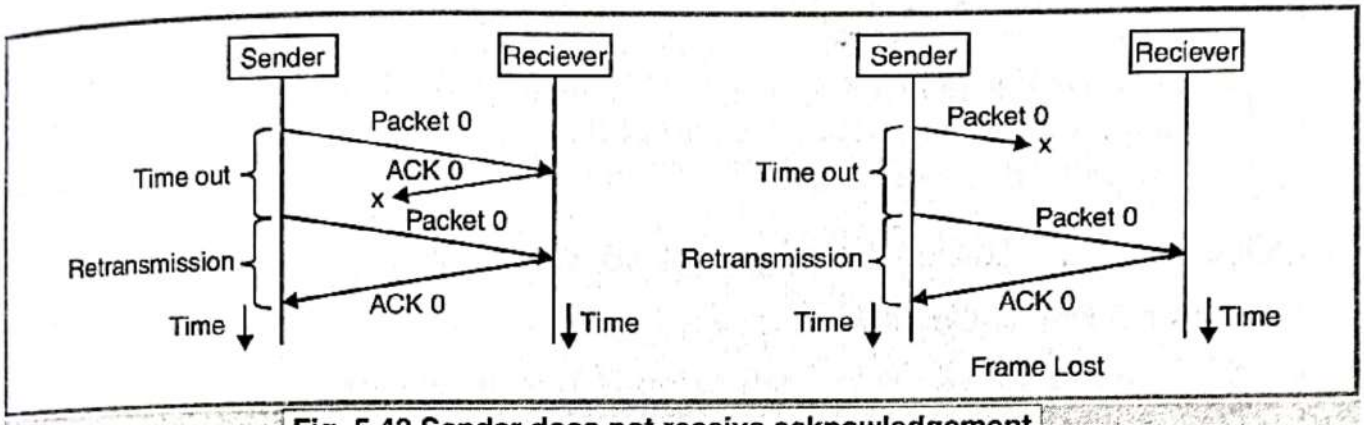


Fig. 5.42 Sender does not receive acknowledgement

- The problem with all the above protocols is that there is uni direclimal flow of the data but practically there is a bi-directional flow of data. This is called as Full duplex.

A. Bidirectional or full duplex flow of the data in Data Link Layer

- The protocol used in unidirectional flow of data is called sliding window protocol.

A.1 Sliding Window Protocol

- In the sliding window protocol, each out bound frame contains a sequence number, ranging for 0 to 2^{n-1}
- The value of n can be arbitrary.

A.2 Sliding window

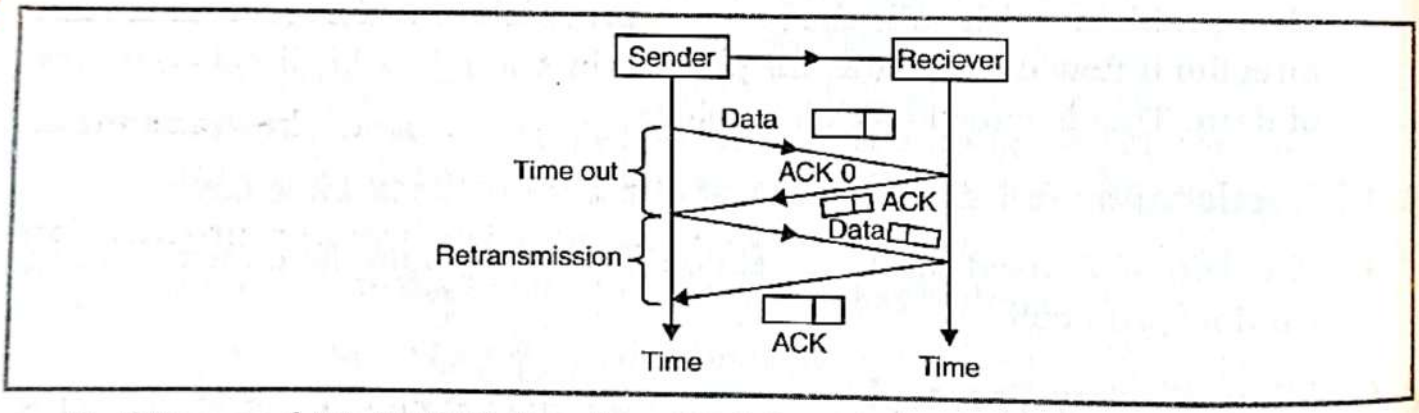
- Sliding window refers to imaginary boxes at the transmitter and receiver.
- This window holds the frames at either ends and provides the upper limit on the number of frames that can be transmitted before requiring an acknowledgement.
- Hence we can say that, at may point of time, the sender maintains a set of sequence number corresponding to frame it is permitted to send.

- These frames which are being permitted to be sent are said to be falling in the sending window.
- The receiver also maintains a receiving window. It corresponds to the set of the frames it is permitted to accept.
- The operation of this protocol is based on ARQ (automatic repeat request) principle.
- So the sliding window is also called ARQ protocol.
- If it receives a positive acknowledgement (ACK) it transmits the next frame. If it receives the negative acknowledgement (NAK) it retransmits the same frame.

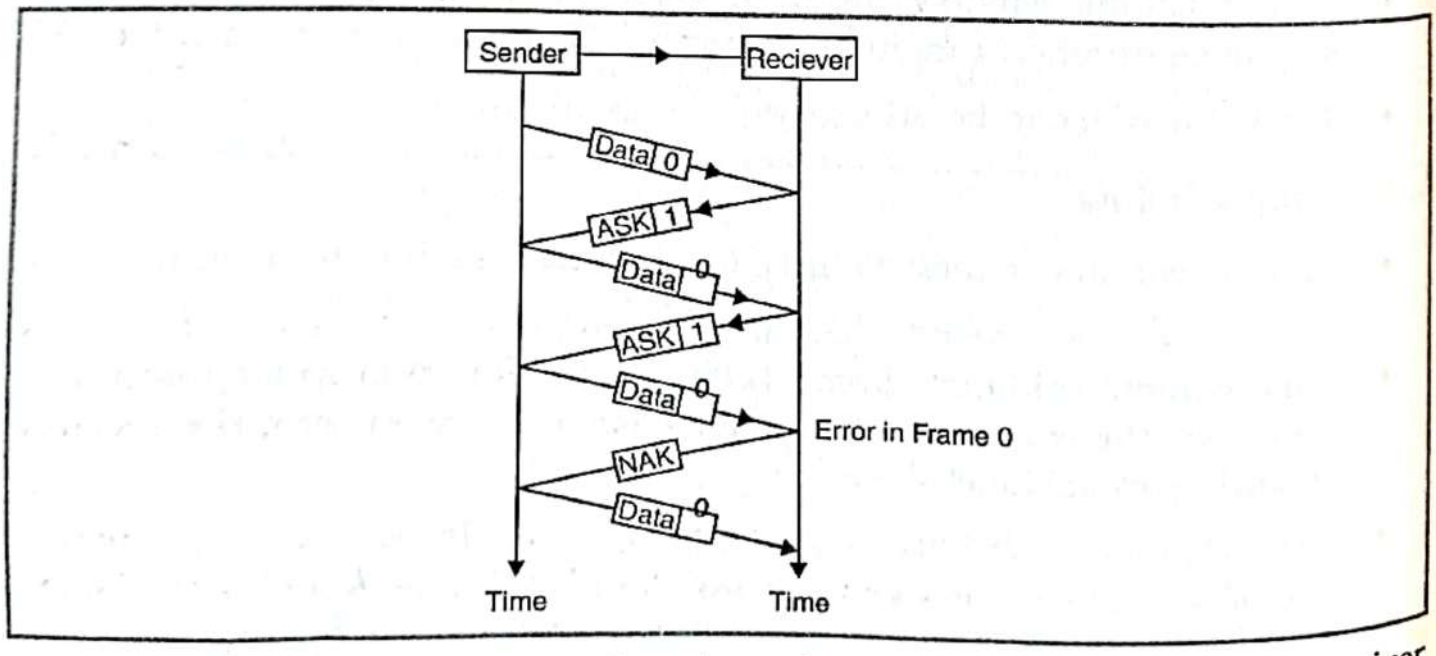
A.3 Operation of Sliding Window Protocol

1. Under Normal Condition

- No frame is lost so retransmission is not necessary.
- Everything is normal.



2. Stop and Wait ARQ for damaged frame



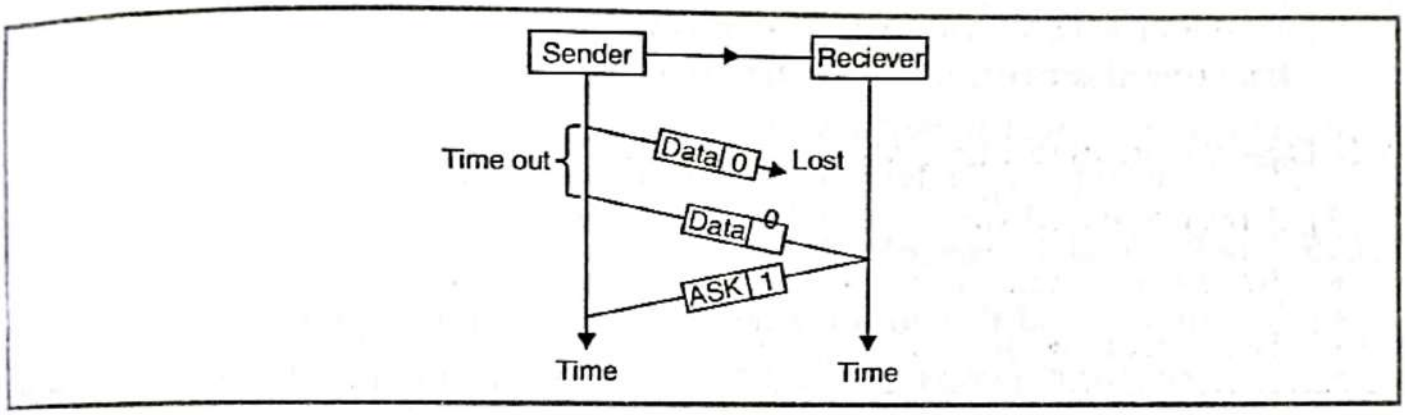
- The transmitter transmits data frame numbered 0. The receiver

returns an ACK1 indicates that the frame numbered 0 is received without an error.

- The process goes on this way, but if error occurs the receiver sends NAK requesting retransmission of the corrupted data frame.

3. Stop and Wait ARQ for lost data frame

- This situation occures when data frame is lost.
- If the transmitter does not receive any type of acknowledgement from the receiver within specified time it retransmit the same frame again.



4. Stop and Wait for lost acknowledgement

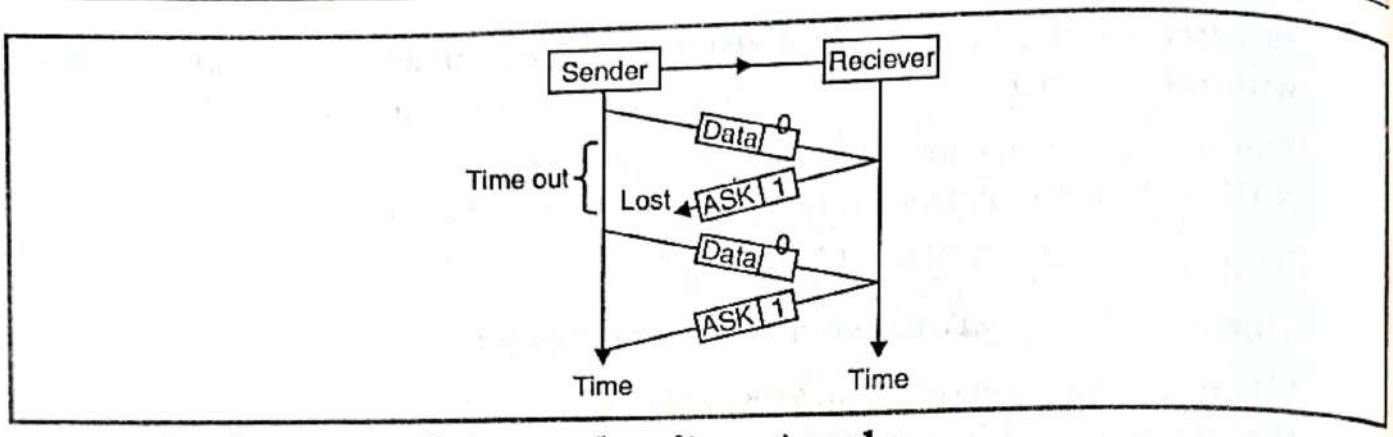
- The acknowledgement sent by the receiver is lost, the transmitter retransmits the same data frame if its timer goes off..
- Stop and wait ARQ protocol becomes inefficient when in propagation delay is much grater than the time to transmit a frame.

Example :

- Let us assume that we are transmitting the frame 800 bit long over a channel that has a speed of 1Mbps and also assume that the time taken for transmission of the frame and its acknowledgement is 30 ms.
- The number of bits can be transmitted over the channel is 30 ms is equal to

$$30 \times 10^{-3} \times 1\times 10^{-6} = 30{,}000 \text{ bits}$$

- But in stop and wait ARQ only 800 bits can be transmitted in this time period. This inefficiency is due to the fact that in stop and wait ARQ the transmitter waits, for an acknowledgment from the receiver before sending the next frame.
- The product of the bit rate and delay that elapses before an action can take place is called Delay bandwidth product. The Delay-bandwidth product helps in measuring the lost opportunity in terms of transmitted bits.



Disadvantage of stop and wait protocol :

- Problem with the stop and wait protocol is that it is very inefficient. At any one moment, only one frame is in transition.
- The sender will have to wait one round trip time before sending next. The waiting can be long for the slow network such as satellite link.

5.6.5 Go-Back-N Protocol

- In this method the data frames are assigned a sequence number.
- The receiver keeps track of sequence numbers of incoming data frames.
- The transmitter transmits the numbered data blocks continuously one after the another, without waiting for acknowledgement.

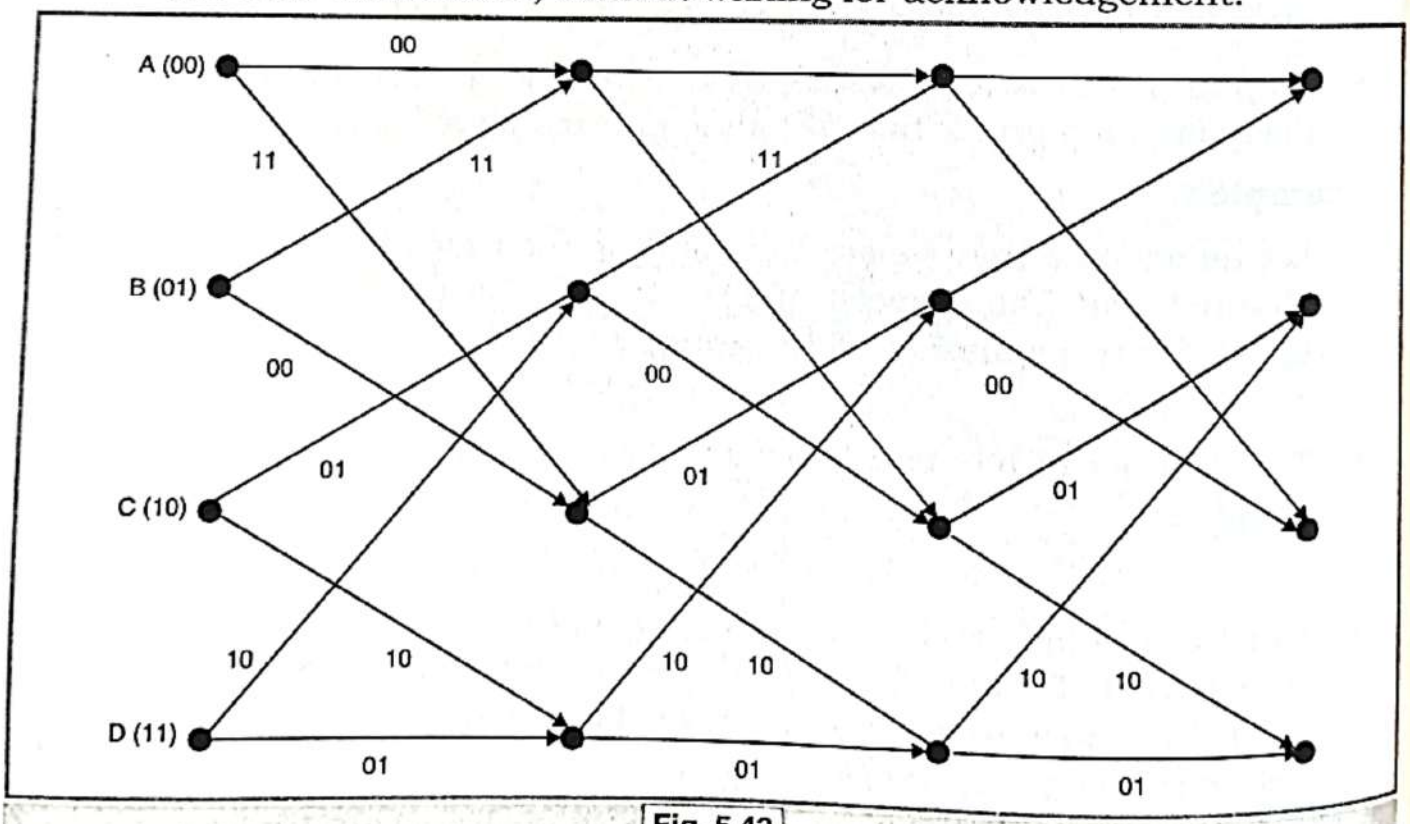


Fig. 5.43

- The receiver sends numbered acknowledgements as ACK-N,
- for example ACK-5 means up to frame no.5 there are no errors. If in

the mean time any error is detected by receiver it sends REJ-N message i.e. retransmit the frames starting from frame no.-N.

- In that case after receiving REJ-N message the transmitter retransmits all blocks starting from N. For example an REJ-5 message will request transmitter to send from frame No 5 onwards i.e. frame 5 is rejected so transmitter will go back to frame 5 and will start retransmitting frame 5 onwards. The scheme is shown in fig.

5.6.6 Selective Retransmission Protocol

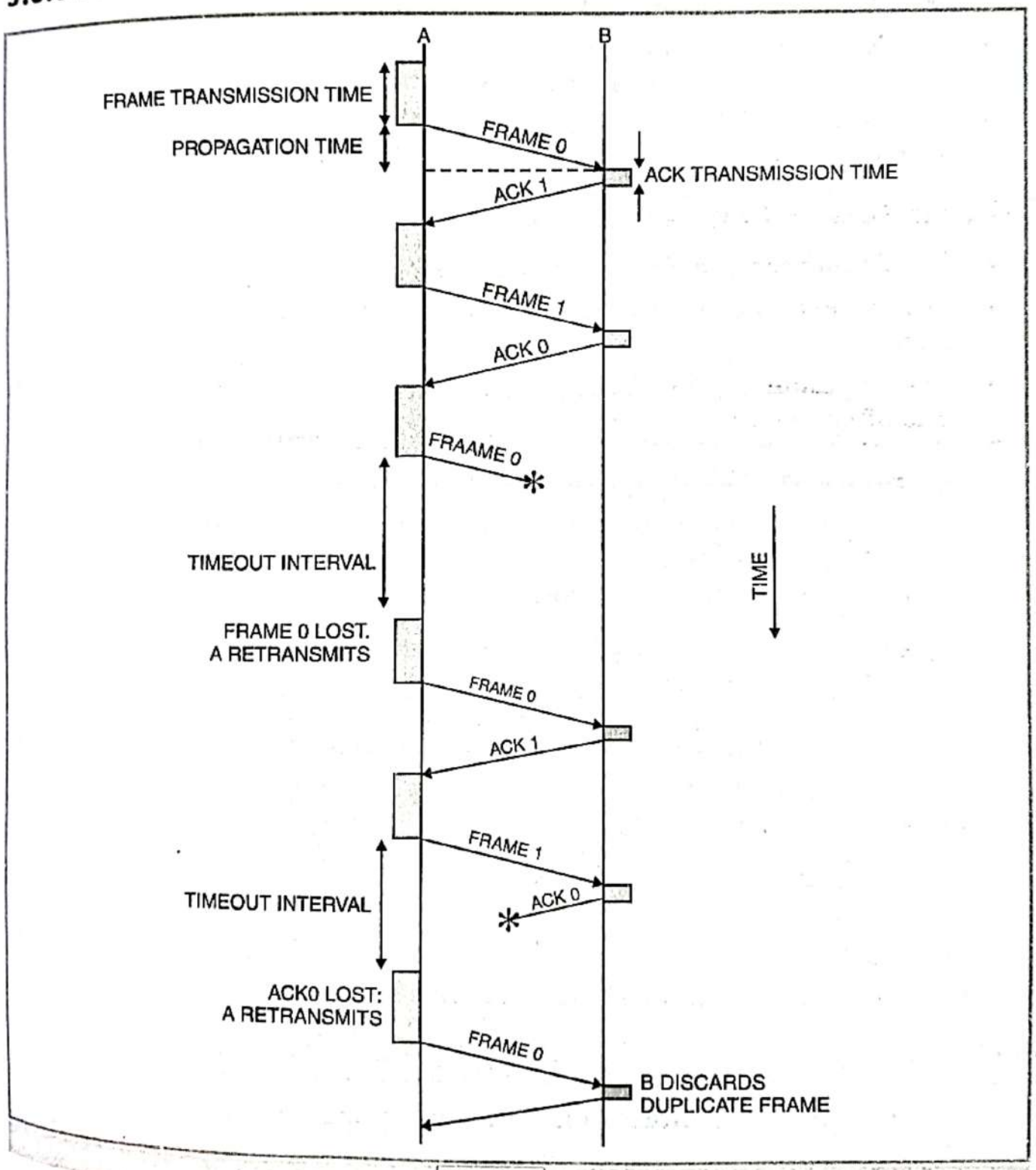


Fig. 5.44

- In this method the transmitter sends the data blocks or frames in a numbered sequence as in go-back-n method.
- But this time if error is detected in any block the receiver sends SREJ-N acknowledgement with a request to retransmit N^{th} data block only.
- For example if error is detected in 4th frame.
- The receiver will send message SREJ-4 and transmitter will send 4th block again. The mechanism is shown in fig.
- REJ and SREJ both correspond to negative acknowledgement;.
- One is for simple request to send from n^{th} frame onwards is REJ-N while SREJ-N is selective reject for retransmitting the nth frame only.
- Such error controls in which receiver sends acknowledgement to sender for correct receipt and requests for retransmission for wrong receipts of data are known as automatic repeat requests (ARQ).
- Two basic types of ARQ are idle RQ and continuous RQ.
- Idle RQ are used with byte oriented data transmission schemes while continuous RQ are used with bit oriented schemes.
- Stop and wait method is an idle RQ and go back n and selective repeat are continuous RQ methods.
- There are so many control methods but no error control method is hundred percent effective, there are always some undetected content errors and flow integrity errors.

5.7 FRAMES

- HDLC defines three types of frames; **Information frames (I-frames,) Supervisory frames(S-frames),** and **Unnumbered frames (U-frames);**
- Each type of frame works as an envelope for the transmission of a different type of message.
- I-frames are used to transport user data and control information relating to user data. S-frames are used only to transport control information, primarily data link layer flow and error controls.
- U-frames are reserved for system managements.
- Infomration carried by U-frames is intended for managing the link itself.

- Each frame in HDLC may contain up to six fields; a beginning flag field, an address field, a control field, an information field, a frame check sequence (FCS) field, and an ending flag field.
- In multiple frame transmission, the ending flag of one frame can double as the beginning **flag** of the next frame.

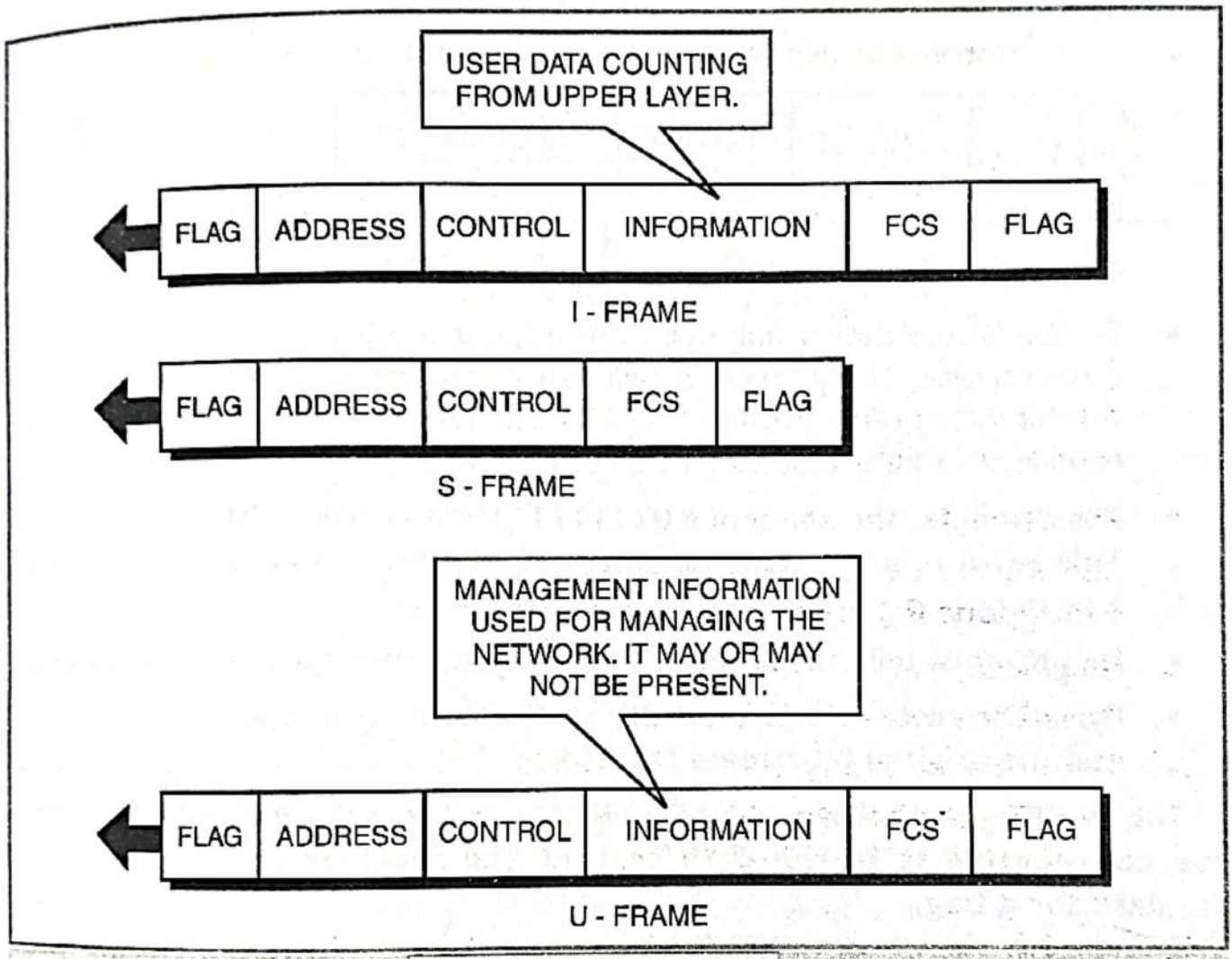


Fig. 5.44 HDLC frame types

(1) Flag Field

- The flag field of an HDLC frame is an eight-bit sequence with a bit pattern 01111110 that identifies both the beginning and end of a frame and serves as a synchronozation pattern for the receiver.
- The flag field is the closest that HDLC comes to a control character that might be misread by a receiver.
- The flag field is also, therefore, HDLC's only potential cause of transparency problems.
- Once a station finds a flag on the line, determines that the frame is addressed to it, and begins reading the transmission, it is watching

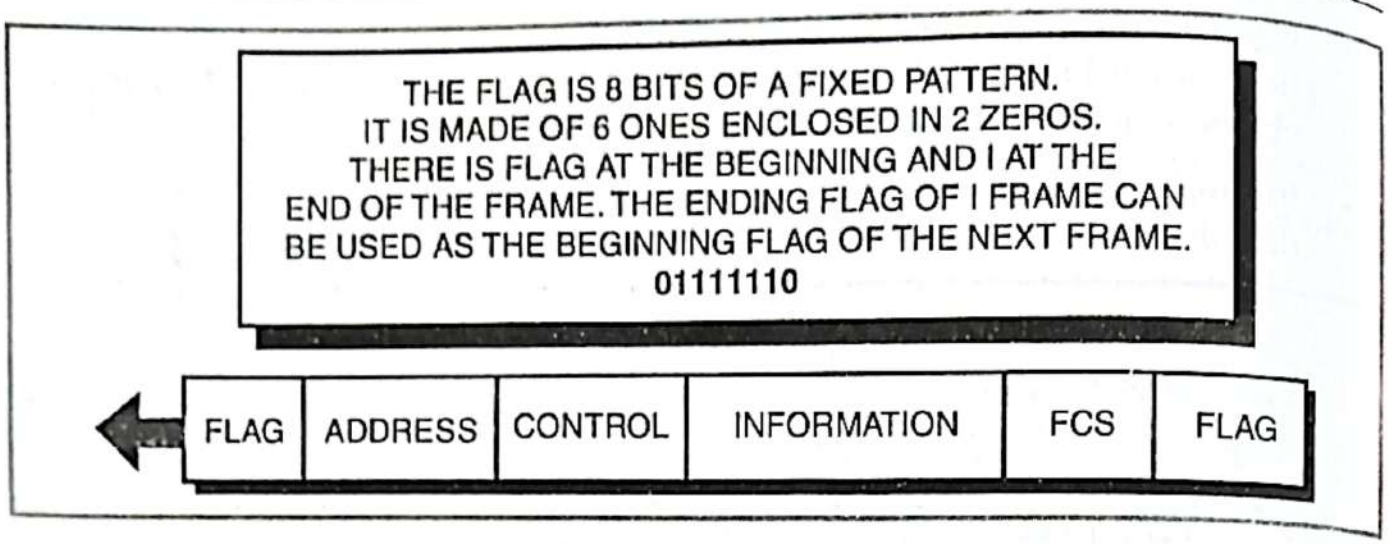


Fig. 5.45 HDLC flag field

- To guarantee that a flag does not appear inadvertently anywhere else in the frame, HDLC uses a process called **bit stuffing**. Every time a sender wants to transmit a bit sequence having more than five consecutive Is, it inserts (stuffs) one redundant 0 after the fifth 1.
- For example, the sequence 011111111000 becomes 0111110111000. This extra 0 is inserted regardless of whether the sixth bit is another 1 or not.
- Its presence tells the receiver that the current sequence is not a flag.
- Once the receiver has seen the stuffed 0, it is dropped from the data and the original bit stream is restored.

Bit stuffing is the process of adding one extra 0 whenever there are five consecutive Is in the data so that the receiver does not mistake the data for a flag.

- The following Figure Shows bit stuffing at the sender and bit removal at the receiver.
- Note that even if we have a 0 after five 1s, we stuff a 0.
- The 0 will be removed by the receiver.
- With three exceptions, bit stuffing is required whenever five 1s occur consecutively.
- The exceptions are when the bit sequence really is a flag, when the transmission is being aborted, and when the channel is being put into idle.
- The flowchart in Figure shows the process the receiver follows to identify and discard a stuffed bit.
- As the receiver reads the incoming bits, it counts 1s.

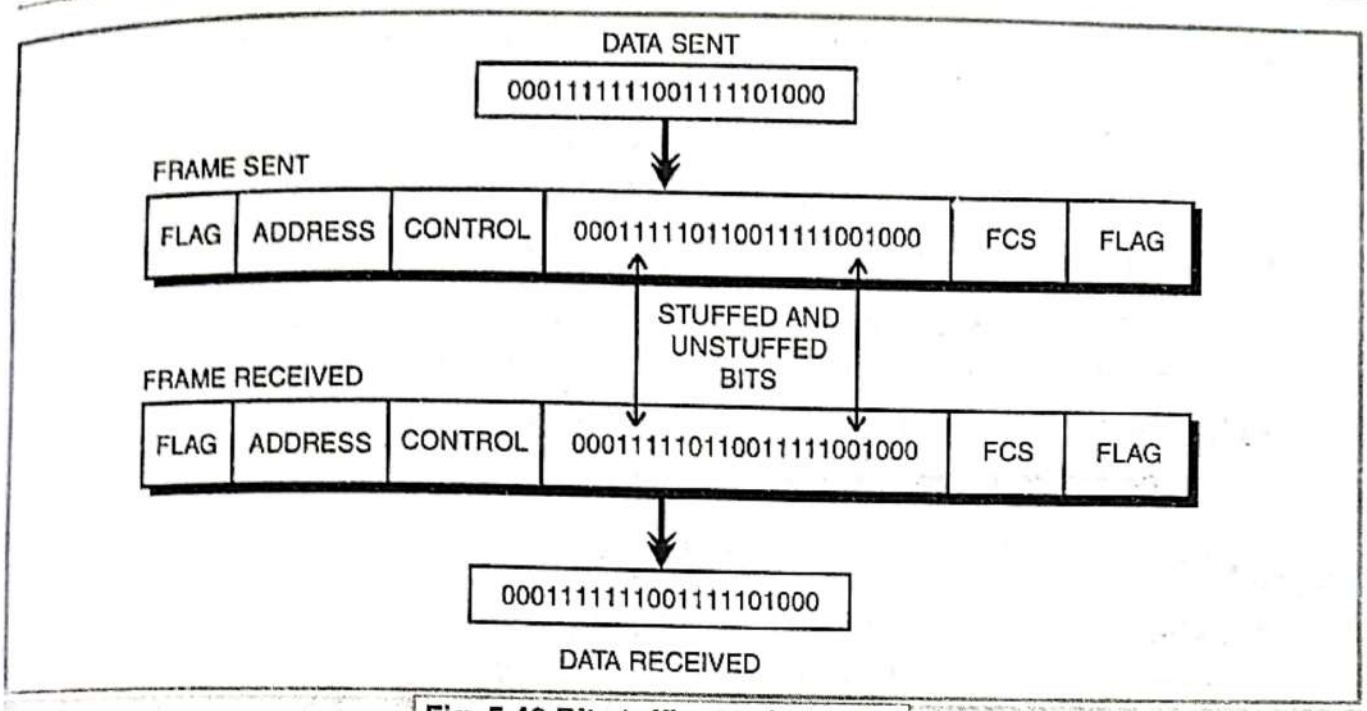


Fig. 5.46 Bit stuffing and removal

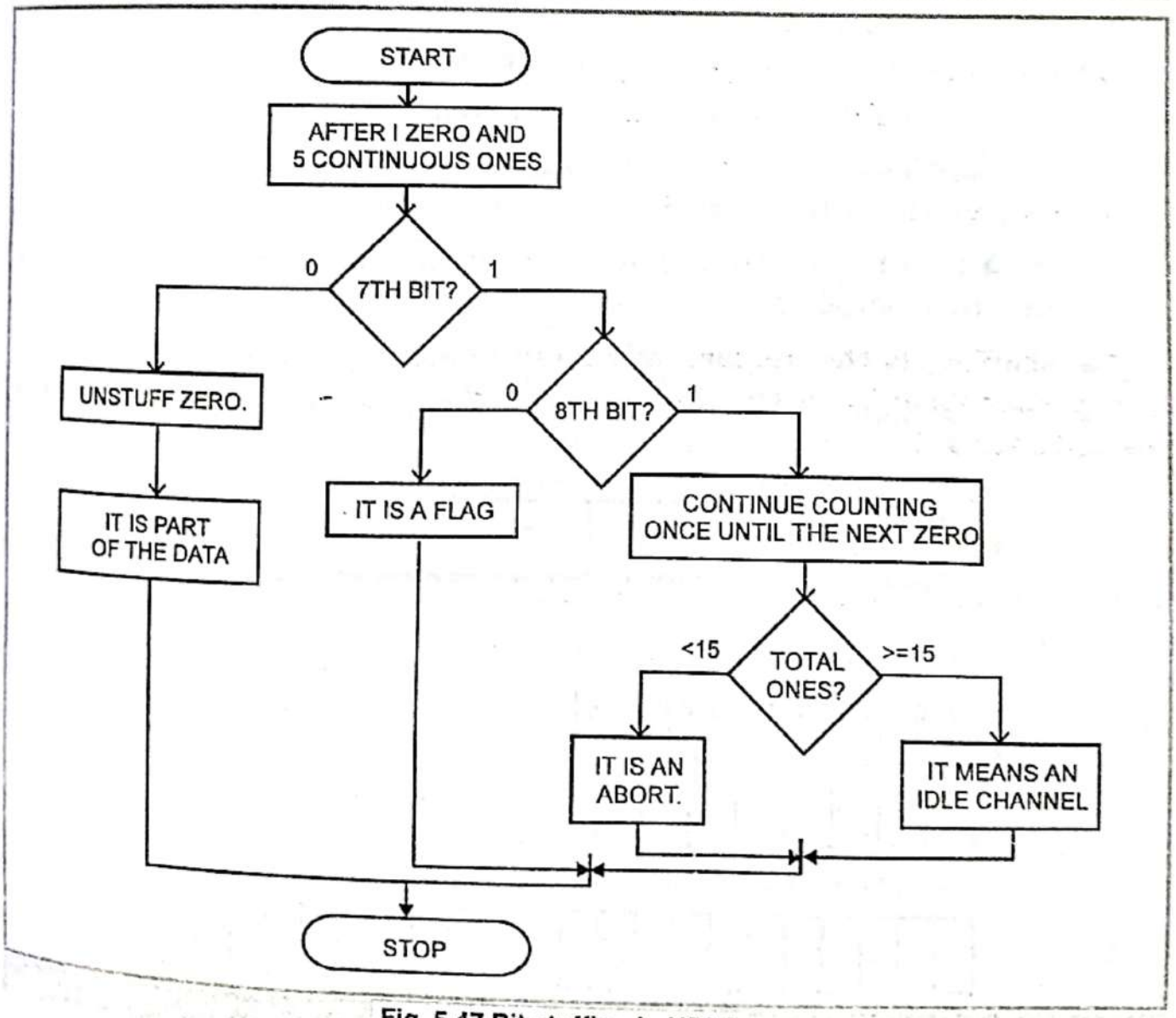


Fig. 5.47 Bit stuffing in HDLC

- When it finds five consecutive 1s after a 0, it checks the next (seventh) bit. If the seventh bit is a 0, the receiver recognizes it as a stuffed bit, discards it, and resets its counter. If the seventh bit is a 1 the receiver checks the eighth bit.
- If the eighth bit is a 0, the sequence is recognized as a flag and treated accordingly.
- If the eighth bit is another 1, the receiver continues counting.
- A total of 7 to 14 consecutive 1s indicates an abort. A total of 15 or more 1s indicates an idle channel.

(ii) Address Field

- The second field of an HDLC frame contains the address of the secondary station that is either the originator or destination of the frame (or the station acting as secondary in the case of combined stations.)
- If a primary station creates a frame, it contains a to address.
- If a secondary creates the frame, it contains a from address.
- An **address field** can be one byte or several bytes long, depending on the needs of the network.
- One byte can identify up to 128 stations (because one bit is used for another purpose).
- Larger network require multiple-byte address fields.
- Figure 5.48 shows the address field in relation to the rest of the frame.

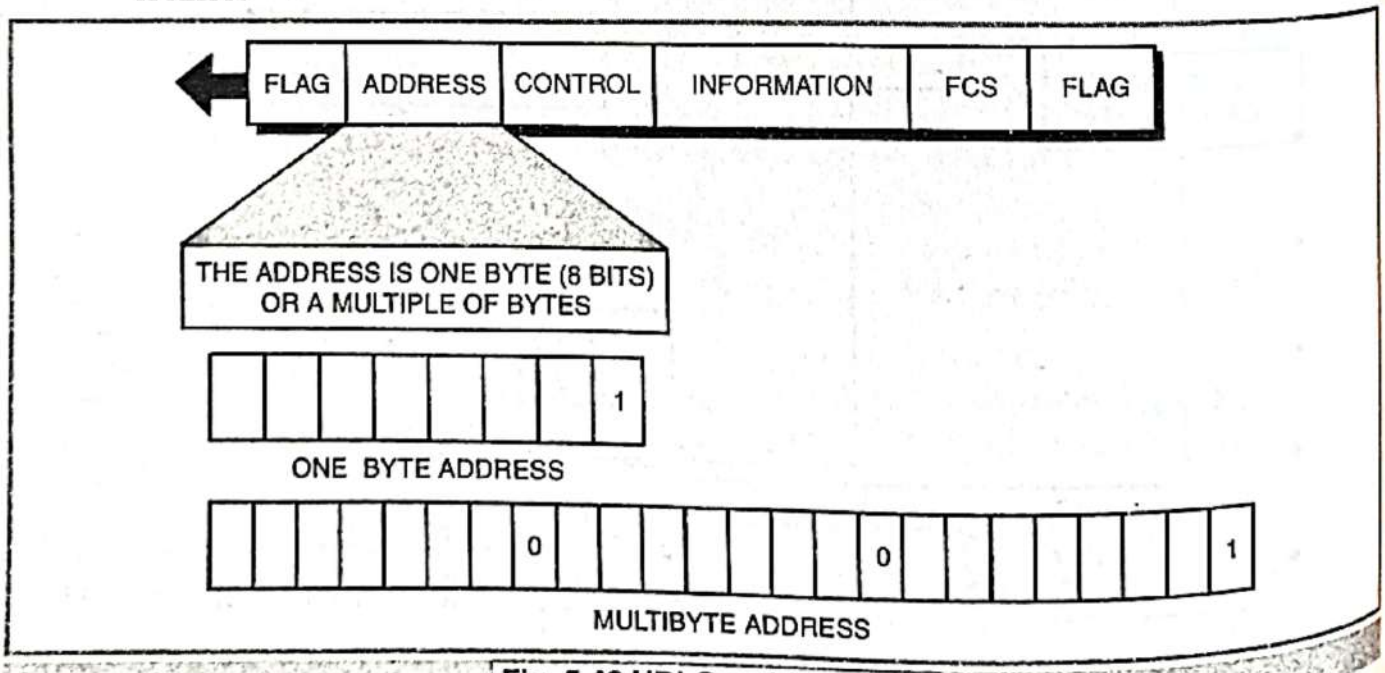


Fig. 5.48 HDLC caddress field

- If the address field is only one byte, the last bit is always a1. If the address is more than one byte, all bytes but the last one will end with 0; only the last will end with 1.

Ending each intermediate byte with 0 indicates to the receiver that there are more address bytes to come.

(iii) Control Field

- The control field is a one-or-two-byte segment of the frame used for flow management.
- We first discuss the one-byte case and then the two-byte case, called the extended mode.
- Control fields differ depending on frame type. If the first bit of the control field is 0, the frame is an 1-frame.
- If the first bit is a 1 and the second bit is 0, it is an S-frame. if both the first and second bits are 1s, it is a U-frame.
- The control fields of all three types of frames contain a bit called the **poll/final (P/F) bit** (discussed below).
- An I-frame contains two 3-bit flow and error control sequences, called N(S) and N (R), flanking the P/F bit. N(S) specifies the number of the frame being sent (its own identifying number).
- N(R) indicates the number of the frame expected in return in a two-way exchange; thus N(R) is the acknowledgement field.
- If the last frame received was error-free, the N(R) number will be that of the next frame in the squence.
- If the last frame was not received correctly, the N(R) number will be the number of the damaged frame, indicating the need for its retransmission.
- The control field of an S-frame contains an N(R) field but not an N(S) field.
- S-frames are used to return N(R) when the receiver does not have data of its own to send.
- Otherwise the acknowledgement is contained in the control field of an I-frame (above).
- S-frames do not transmit data and so do not require N(S) fields to identify them.
- The two bits preceding the P/F bit in an S-frame are used to carry coded flow and error control information, which we will discuss later in this chapter.

- U-frames have neither, N(S) nor N(R) fields, and are not designed for user data exchange or acknowledgement.
- Instead, U-frames have two code fields, one two bits and the other three, flanking the P/F bit. These codes are used to identify the type of U-frame and its function (e.g. establishing the mode of an exchange).
- The control fields of all three types of frames are shown in figure.5.49

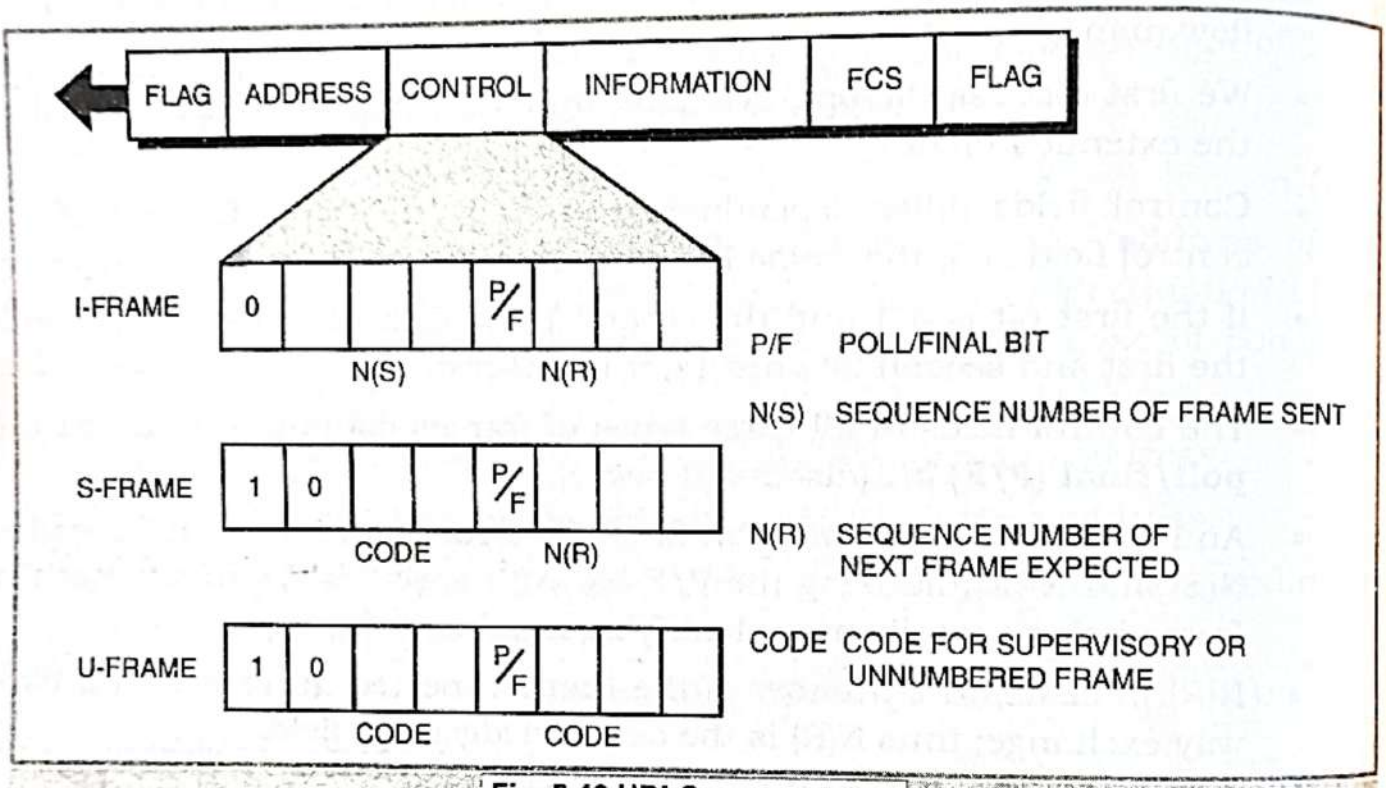


Fig. 5.49 HDLC control fields

- Figure Above shows the control field in the extended mode.

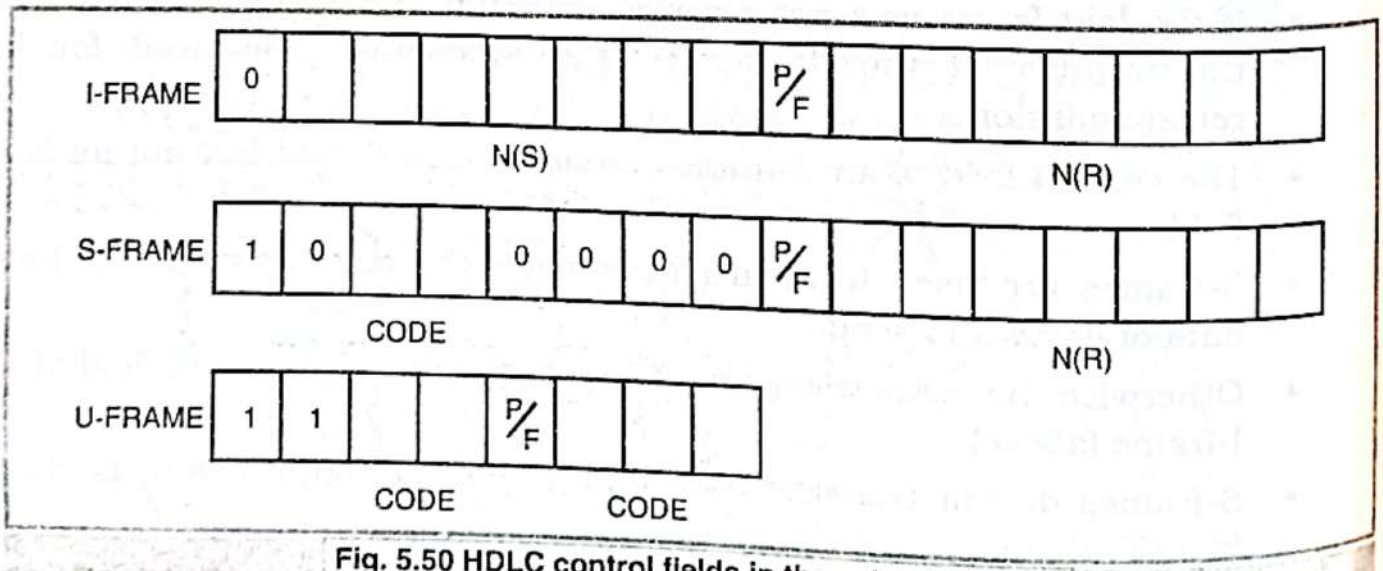


Fig. 5.50 HDLC control fields in the extended mode

- Note that in the extended mode, the control field in the I-frame and S-frame is two bytes long to allow seven bits for the sending and

receiving sequence number (the sequence number is between 0 and 127).

- However, the control field in the U-frame is still one byte.
- 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);

(iv)Information Field

- The information field contains the user's data in an I-frame, and network management information in a U-frame.
- Its length can very from one network to another but is always fixed within each network.
- An S-frame has no information filed.
- As we have seen in the several cases above, it is often possible to include flow, error, and other control information in an I-frame that also contains data.
- For example, in a two-way exchange of data(either half-or-full-duplex), station 2 can acknowledge.

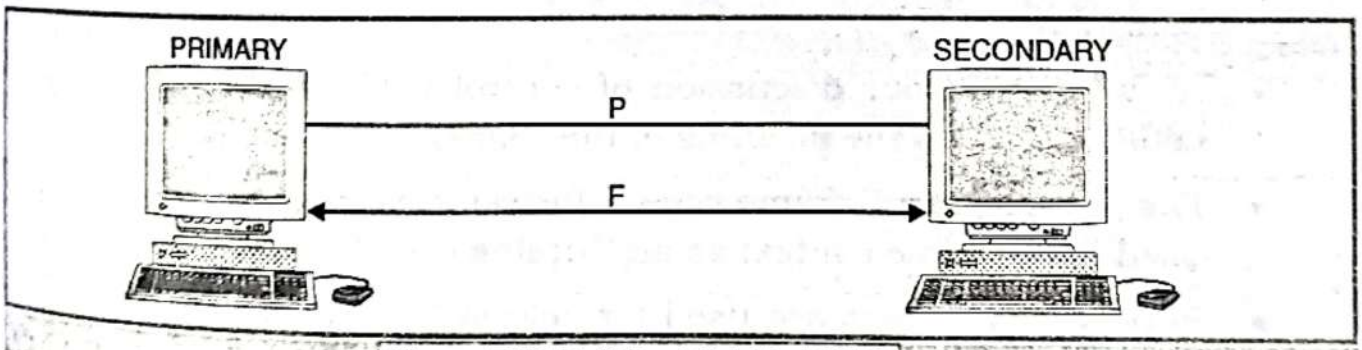


Figure 5.51 Poll/final field in HDLC

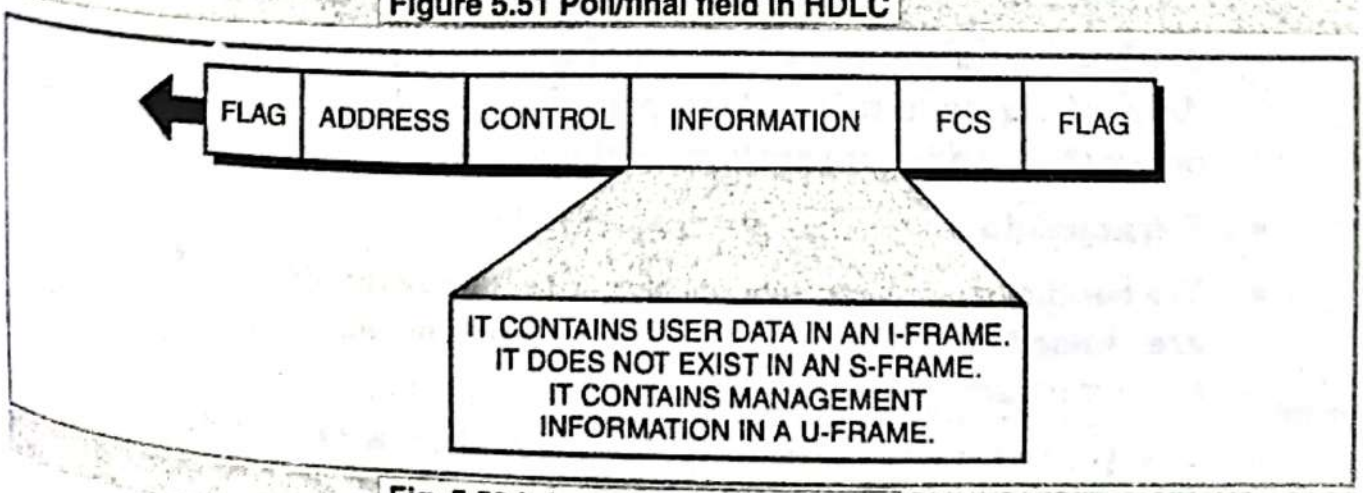


Fig. 5.52 Information field in HDLC

- Receipt of data from station I in the control field of its own data frame rather then sending a separate frame just for the acknowledgment.
- Combining data to be sent with control information this way is called **piggybacking.**

Piggybacking means combining data to be sent and acknowledgement of the frame received in one single frame.

(v) FCS Field

- The **Frame Check Sequence (FCS)** is HDLC's error detection field.
- It can contain either a two-or four-byte CRC

More about Frames

- Of the three frames used by HDLC, the I-frame is the most straightforward.
- I-frames are designed for user information transport and piggybacked acknowledgements and nothing else.
- For this reason, the range of variation in I-frames is small-all differences relate either to the data (content and CRC), to the identifying number of the frame, or to the acknowledgement of received frames (ACK or NAK)
- S-frames and U-frames, however, contain subfields within their control fields.
- As we saw in our discussion of control fields, these subfields carry codes that alter the meaning of the frame.
- For example, an S-frame coded for selective-reject (SREJ) cannot be used in the same context as an S-frame coded for receive ready (RR).
- Supervisory frames are used for acknowledgement, flow control, and error control whenever piggybacking that information onto on-I-frame is either impossible or inappropriate (when the station either has no data of its own to send or needs to send a command or response other than an acknowledgement).
- S-frames do not have information fields.
- Yet each one carries messages to the receiving station. These message are based on the type of the S-frame and the context of the transmission.
- The type of each S-frame is determined by a two-bit code set into its control field just before the P/F bit. There are four types of S-frames:

receive ready (RR), receive not ready (RNR), reject (REJ), and selective-reject (SREJ)

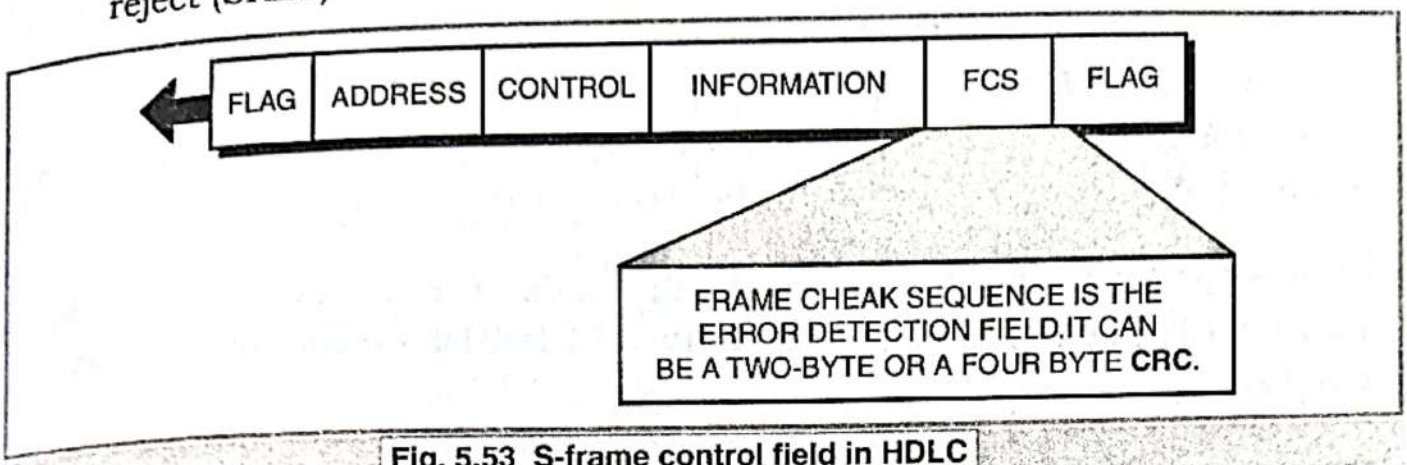


Fig. 5.53 S-frame control field in HDLC

SOLVED PROBLEMS

Example 1. Find hamming distance of data byte 10110101 and 11001011.

Solution.

1	0	1	1	0	1	0	1	
	↕	↕	↕	↕	↕	↕	↕	difference in
1	1	0	0	1	0	1	1	bit positions

As there are six bits prosition differeing in both data bytes, hence hamming distance is 6.

Example 2. Find the parity bits for following data bytes using odd and even parity schemes

(*a*) 10110100, (*b*) 11000101, (*c*) 00100011, (*d*) 10101110

Solution.

(*a*) For Data byte 10110100

no. of 1's = 4 (even)

So parity bit is '0' in even parity scheme and '1' in odd parity scheme,

(*b*) for data byte 11000101 p

no. of '1' s = 4 (even)

Hence parity bit is '0' in even parity scheme and '1' in odd parity scheme.

(*c*) For data byte 00100011

no. of 1's = 3 (odd)

Hence parity bit is '1' in even parity scheme and '0' in odd parity scheme.

Example 3. Write ASCII code of word 'CHECK' using odd parity scheme.

Solution.

Bit Position→	Parity bit	7	6	5	4	3	2	1
C	0	1	0	0	0	0	1	1
H	1	1	0	0	1	0	0	0
C	0	1	0	0	0	0	1	1
K	1	1	0	0	1	0	1	1

Example 4. Find block check code for following data bytes. 11000011, 10101000, 10101111, 10010010, 01001001 using odd parity.

Solution :

Row Parity bit ↓	B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0
1	1	1	0	0	0	0	1	1
0	1	0	1	0	1	0	0	0
1	1	0	1	0	1	1	1	1
0	1	0	0	1	0	0	1	0
0	0	1	0	0	1	0	0	1
0	0	1	0	0	1	0	0	1
Column → Parity bits 0	1	1	1	0	0	0	0	0

Example 5. Find the checksum code for following message 01100010, 01001001, 10101101, 11010001

Solution.

	0	1	1	0	0	0	1	0	TO get code first add
	0	1	0	0	1	0	0	1	all databytes using 1's
	1	0	1	0	1	1	0	1	complement arithmatic
	1	1	0	1	0	0	0	1	the is find its one's
1	0	0	1	0	1	0	0	1	complement
	0	0	1	0	1	0	1	0	

1's complement of 00101010 is 11010101 which is the required checksum code.

Example 6. Generate the checksum bytes as per the transport protocol for the following data bytes-10101100,11101000, 01110100, 10001011

Solution.

	A_0	= 0 0 0 0 0 0 0 0,	B_0	= 0 0 0 0 0 0 0 0
Transmitting end	x	= 0 0 0 0 0 0 0 0,	y	= 0 0 0 0 0 0 0 0
Date bytes	A_0	= 0 0 0 0 0 0 0 0,	B_0	= 0 0 0 0 0 0 0 0
10101100	D_1	= 1 0 1 0 1 1 0 0,	A_1	= 1 0 1 0 1 1 0 0
	A_1	= 1 0 1 0 1 1 0 0,	B_1	= 1 0 1 0 1 1 0 0
11101000	D_2	= 1 1 1 0 1 0 0 0,	A_2	= 1 0 0 1 0 1 0 0
	A_2	= 1 0 0 1 0 1 0 0,	B_2	= 1 0 0 0 0 0 0
01101100	D_3	= 0 1 1 0 1 1 0 0,	A_3	= 0 0 0 0 0 0 0 0
	A_3	= 0 0 0 0 0 0 0 0,	B_3	= 0 1 0 0 0 0 0 0
10001011	D_4	= 1 0 0 0 1 0 1 1,	A_4	= 1 0 0 0 1 0 1 1
	A_4	= 1 0 0 0 1 0 1 1,	B_4	= 1 1 0 0 1 0 1 1
	x	= 0 0 0 0 0 0 0 0,	A_5	= 1 0 0 0 1 0 1 1
	A_5	= 1 0 0 0 1 0 1 1,	B_5	= 0 1 0 1 0 1 1 0
	y	= 0 0 0 0 0 0 0 0,	A_6	= 1 0 0 0 1 0 1 1
	A_6	= 1 0 0 0 1 0 1 1,	B_6	= 1 1 0 0 0 0 1
	x	= $A_6 - B_6$	Y	= $B_6 - 2A_6$
		= 0 0 0 1 0 1 1		= 1 1 1 0 0 0 0 1
		− 1 1 1 0 0 0 0 1		− 0 0 0 1 0 1 1 0
	x	= 1 0 1 0 1 0 1 0	y	= 1 1 0 0 1 0 1 1

The required check sum bytes are 10101010 and 11001011.

Example 7. Check whether the following bytes have any error. The first two bytes are the transport protocol checksum bytes. 00101111, 10100011, 10101011, 10001001, 11010011

Solution. At receiving end

x = 0 0 1 0 1 1 1 1, y = 1 0 1 0 0 0 1 1

A_0 = 0 0 0 0 0 0 0 0, B_0 = 0 0 0 0 0 0 0 0

↓ Data Bytes	A_0	= 0 0 0 0 0 0 0 0	B_0	= 0 0 0 0 0 0 0 0
10101011	D_1	= 1 0 1 0 1 0 1 1,	A_1	= 1 0 1 0 1 0 1 1
	A_1	= 1 0 1 0 1 0 1 1	B_1	= 1 0 1 0 1 0 1 1
10 0 01001	D_2	= 1 0 0 0 1 0 0 1,	A_2	= 0 0 1 1 1 0 1 0 0
	A_2	= 0 0 1 1 0 1 0 0,	B_2	= 1 1 0 1 1 1 1
11010011	D_3	= 1 1 0 1 0 0 1 1,	A_3	= 1 1 1 0 0 1 1 1

A_3 = 1 1 1 0 0 1 1 1, B_3 = 1 1 0 0 0 1 1 0
x = 0 0 1 0 1 1 1 1, A_4 = 0 0 0 1 0 0 0 0
y = 1 0 1 0 0 0 1 1, A_5 = 1 0 1 1 0 0 1 1
A_5 = 1 0 1 1 0 0 1 1, B_5 = 1 0 0 0 1 0 0 1

As A_5 and B_5 are not zero so the transmitted data contains error.

Example 8 Generate cyclie redundance code for the data word 101101011 using divisor 10011.

Solution. 10101010 1

```
10011 )1011010110000(
  ⊕ 10011
      0010110
    ⊕ 10011
      0010111
    ⊕ 10011
          0010000
      ⊕ 10011
            010010
        ⊕ 10110
                00110
      ⊕ 0010011
            10110
        0001100   Remainder
1 0 1 1 0 1 0 1 1 0 0 0 0
            1 1 0 0
1 0 1 1 0 1 0 1 1 1 1 0          Code word
```

Hence the required codeword is 1011010111100

Example 9. A CRC code is transmitted as 1001011100101. Check if there are some errors using divisor as 10101.

Solution.

```
        101100000
10101)1001011100101( Quotient
      ⊕ 10101      Dividend
          0011111
      ⊕ 10101
            010101
          10101
                0000000101      Remainder
```

As the reminder is non zero so there are some errors in code.

Example 11. A CRC code was generated using the divisor as 10101. the code received at transmitting end is 1001011110101 then check her it is correct or not.

Solution.

```
      101100001
10101)1001011110101( Quotient
    ⊕ 10101        Dividend
        0011111
      ⊕ 10101
        010101
      ⊕ 10101
          0000010101
      ⊕     10101
                00000        Remainder
```

As zero remainder indicates that the code transmitteed in correct.

Example 11. The following bit stream is encoded using logitudinal and vertical redundancy check and even parity. Correct the error if any. 11000011, 11110011, 11110010, 00001010, 10101010, 00101011, 10100011. 01001011, 11100001

Consider LSB as parity bit

Solution.

As seen from block that the parity bits are wron for 3rd row and 2nd column. So the bit at their crossection indicated as '1' is wrong. It should be '0' shown encricled in fig.

	1	2	3	4	5	6	7	8
1	1	1	0	0	0	0	1	1
2	1	1	1	1	0	0	1	1
3	1	1	1	1	0	0	1	0
4	0	0	0	0	1	0	1	0
5	1	0	1	0	1	0	1	0
6	0	0	1	0	1	0	1	1
7	1	0	1	0	0	0	1	1
8	0	1	0	0	1	0	1	1
	1	1	1	0	0	0	0	1

EVEN PARITY

EVEN PARITY BITS

Example 12. Find a 7 bit hamming code for the data bits 1011 using odd parity scheme.

Solution.

P_1	P_2	P_3	P_4	P_5	P_6	P_7
		1		0	1	1

in bits P_1 D_3 D_6 D_7 P_1 should be '1' for odd parity

in bits P_2 D_3 D_6 D_7 P_2 should be '0'

in bits P_4 D_5 D_6 D_7 P_4 should be '1'

Hence the required code is 1011011

Example 13. A Hamming code was transmitted as 0111101. Check it and correct if necessary. The parity bit is taken using even parity scheme. What is the correct data?

Solution.

P_1	P_2	P_3	P_4	P_5	P_6	P_7
1	1	1	1	0	1	1

For fist group P_1 D_3 D_5 D_7 P_1 is wrong.

For second group $P_2D_3D_6D_7$ P_2 is correct

For third group $P_4D_5D_6D_7$ P_4 is wrong

Hence the bit common to group 1& 3 is wrong i.e. D_5 should be 1. So the data 1111 was transmitted wrongly as 1011.

SUMMARY

- In asynchronous transmission the data block is transmitted byte by byte. Start and stop bits are prefixed and suffixed to each byte. In synchronous transmission data is transmitted in form of frames. Flags are appended to identify the frame at its starting end. In this transmission clock is required.
- In parity check method one or more parity bits are added to data word to obtained code word. In block sum check and checksum method the sum of bits of data words are used to obtain their sum and finally the code word. The transport checksum method is used with transport protocols. In this method two code words are generated and checked. CRC method is best for multiple error detection. It is powerful method with least chances of errors remain undetected.
- In forward error correction methods the errors are detected and corrected by inverting the bits at location of errors. In reverse error correction method once the error is detected its sends request to

sender to retransmit that data block in which error has been detected. The forward error correction methods are parity method, Hamming code and convolutione code while the reverse error correction methods includs stop and wait machanism, go-back-N mechanism and selective repeat mechanism.

- Interfacing means defining various specifications for connection standards (interfaces) Various standard for physical medium interface are ISO 2110, ISO-4902, ISO 4903, V.26, V.27, V2d, RS-232-C, EIA-232 D, Rs-422A etc.
- The flow control, error control and line management functions during data transmission are combined known as data link control. The rules and procedures used for data link control are known as data link protocols. The first task of data link control is to design a frame format. The general format of frame contains three parts. Header consists of Flag, Address and control bits. Data bits contain the actual data Trailer consists of FCS and flag. Three types of frames are fixed format-fixed length, fixed format-variable length and variable format-variable length.
- Flow control is concerned with controlling the rate of transmission of frames. Commonly used flow control mechanisms are X-ON/X-OFF mechanism, Stop and wait mechanism and sliding window mechanism. Error control is related with recovery of errors in data. It can be a manual mode or ARQ mode. Two types of ARQ are idle RQ and continuous RQ modes. Stop and wait is an idle RQ while Go back-n and selective repeat are continuous RQ methods. Data link management is done in five phases named as connect, link establishment, data transfer, termination and clear.
- Two types of data link control protocols are character oriented and Bit oriented protocols. The grouping of signals coming from several channels without mixing is known as multiplexing. Two basic multiplexing methods are time division multiplexing and frequency division multiplexing. FDM is based on analog transmission principles while TDM is based on digital transmission principles.

EXERCISE

1. FILL UP THE BLANKS

1. The two modes of transmission are.....and.....

2. In asynchronous transmission.....is transmitted a time.
3. Start bit and stop bits are used in.....transmission.
4. STX and ETX characters are used in.....synchronization.
5. STX stands for.....
6. ETX stands for.....
7. When one frame is sent at a time then such transmission is knwon as.....
8. The flag identifies start of a.....
9. Two types of errors are.....and.....
10. The errors which occur due to error in data bits in known as.....
11. Missing block of data due to delivery to wrong destination is known as.....
12. The block of data bits with check bits is known as.....
13. The number of disagreement between two code words is known as.....
14. Error bursts can be detected by.....method.
15. The parity check method and check sum method are used to detec.....errors.
16. The method used for error detection in transport protocol of ISO model is knwon as.....
17. The TPC method generates.....code words.
18. The method in which errors are detected and corrected by receiver it self is known as.....
19. The method in which the receiver requests the transmitter to retransmit the data is known as.....
20. Hamming code method is a.....method.
21. State diagram and Trellis diagrams are used in.....method.
22. REJ and SREJ are used for.....acknowledgement.
23. RS-232-C and EIA 232-D are.....interfaces.
24. Centronics interface is used to connect.....to.....
25. To define various specifications for physical medium is known as.....
26. The error control and flow control functions are known as .....
27. the data link control includes link management, error control and .....
28. Formatting the frame is task of ..... .

29. Each frame contains Header, Trailer and .....
30. The address of receiving device is included in ..... field of .....
31. The data link control information's are contained in ..... and ..... field.
32. HDLC stands for .....
33. Variable format-variable length is a type of .....
34. X-ON/X-OFF mechanism is a ..... mechanism.
35. ACK is used for ..... acknowledgement.
36. NAK is used for ..... acknowledgement.
37. ACK and NAK are used in ..... mechanism.
38. RR & RNR are used in ..... mechanism.
39. The minimum transmission rate is found in ..... mechanism.
40. The stop and wait mechanism is very ..... from the link utilization point of view.
41. If the window size is 1 then sidings window mechanism is similar to ..... mechanism.
42. A frame can be completely lost without knowledge of sender and receiver in ..... mechanism.
43. the sequence numbering of frame is used in ..... mechanism.
44. Two types of errors are ..... and .....
45. Two types of data link control protocols are ..... and .....

II. SHORT ANSWER TYPE QUESTIONS

1. What are various modes of data transmission ?
2. What are the two types of data transmission ?
3. What do you mean by sybchronous transmission ?
4. Define asynchronous trnasmission ?
5. What do you mean by synchronizations ?
6. What are the two types of errors ?
7. Define contents integrity errors ?
8. Define flow integrity errors ?
9. What is a check bit ?
10. What do you mean by data word and code word ?
11. What do you mean by hamming distance ?

12. What do you mean by weight of a codeword ?
13. What is a code set ?
14. Name any two error detection techniques ?
15. What do you mean by parity bit ?
16. What is a block parity ?
17. Define Forward eror correction ?
18. What is meant by Reverse error correction ?

III. VERY SHORT ANSWER TYPE QUESTIONS

1. Define data link control ?
2. What is error control ?
3. What do you mean by flow control ?
4. What are the 3 essential parts of a frame ?
5. How many types of frame are there ? Name them.
6. Name various types of flow control mechanisms.
7. Name various types of errors?
8. Name various types of error control mechanisms?
9. Define ACK, NAK, RR, RNR and ENQ.
10. What do you mean by challenging ?
11. What is link utilization efficiency ?
12. What do you mean by window size ?
13. What is ARQ ?
14. What are the phases of Data link management ?
15. What is data link protocol ?
16. How many types of data link protocols are there ?
17. Draw a 'Kermit' frame format?
18. What is multiplexing ?
19. How many types of multiplexing are there ?
20. What are the main drawbacks of TDM ?

IV. SHORT ANSWER TYPE QUESTIONS

1. Explain the terms data link control, error control and flow control?
2. What do you mean by frame designing ? Discuss with an example?

3. Explain X-ON/X-OFF flow control mechanism?
4. Explain sliding window flow control mechanism in brief.
5. Discuss link utilization in stop and wait mechanism.
6. Explain stop and wait mechanism.?
7. Explain Data link management functions?
8. Discuss bit oriented protocols in brief?
9. Explain frequency division multiplexing?
10. Discuss time division multiplexing in brief?
11. Explain BSC frame format?
12. What are idle RQ and continuous RQ mechanism?

V. LONG ANSWER TYPE QUESTIONS

1. Discuss various error control mechanisms?
2. Explain various flow control mechanisms in brief?
3. Define multiplexing. Explain their types?
4. Explain Go-back-N and selective retransmission mechanisms in sliding window flow control along with examples?
5. What are the main draw backs of stop and wait mechanism? How they are over come in sliding window mechanism ?
6. What do you mean by data link control protocol ? Explain character oriented protocols in detail.
7. Explain HDLC and LLC protocols in brief along with their frame formats.
8. In stop and wait mechanism. If the probability of receiving a data frame with errors is P, then shown that the effective link utilization of given by Also assume the acknowledgements are always error free?
9. Indicate the frame exchange for following actions in stop and wait flow control with +ve and -ve acknowledgements?
   (a) A sends a data frame.
   (b) B detects error and indicates it to A.
   (c) A retransmits frame.
   (d) D receives frame but asks A to wait.
   (e) B asks A to send next frame.
   (f) A Sends a frame but frame is lost.

(*g*) A challenges B

(*h*) B responds

(*i*) A retransmits the frame

(j) B acknowledges it

10. Taking the window size 7 indicate frame exchanges for following actions.

(*a*) A sends frames 0, 1, 2, 3 to B

(*b*) B acknowledges the frames 0, 1

(*c*) A sends frames 4, 5, 6

(*d*) B rejects frame 3 and asks to retransmit from frame 3 on wards.

(*e*) A transmits frame 3, 4, 5, 6, 7

(*f*) B acknowledges 3, 4, 5, 6, 7

(*g*) A transmits 0, 1

(*h*) B acknowledges 0, 1

How many bits are used in above mechanism for frame sequencing ?

VI. NUMERICAL PROBLEMS

1. A channel is operating at 1200 bps and propagation delay is 20 ms. If frame size is 1200 bits then calculate link utilization efficiency.

2. A channel is operating at 4800 bps and the propagation delay is 20 ms. What should be the minimum frame size for stop and wait flow control to get 50 percent link utilization efficiency.

3. If the frame size is 960 bits on a channel operating at 960 kbps then what is the maximum link utilization for stop and flow control mechanism and sliding window flow control with window size of 7. Propagation delay is 270 ms.

4. Calculate link utilization efficiency for stop and wait flow control mechanism if the frame size is 2400 bits, bit rate is 4800 bps, distance between devices is 2000 km and speed of propagation over transmission media is 200000 km/s.

5. Calculate link utilization efficiency if

(*a*) bit rate = 19.2 kbps, Frame size = 960 bits

Window size = 2, propagation time = 0.06 s

(*b*) bit rate = 19.2 kbps, frame size = 960 bits

window size = 7, propagation time = 0.06s

what is the minimum window size for 100 percent link utilization.

6. A series of 1000 bit frames is to be transmitted using an idle RQ protocol. Determine the link utilization for the following types of data link assuming a data transmission rate of (*a*) 1 Kbps and (*b*) 1 mbps. The speed of propagation of transmission media is 2×10^8 m/s and bit error rate is negligible.

(*i*) a cable of length 1 km.

(*ii*) a line of length 200 km.

(*iii*) a satellite link of 50000 km.

7. Calculate link utilization efficiency for continuous RQ protocol based link where frame size = 1000 bits, speed of propagation is 2×10^8 m/s and

(*a*) 1 Km link of 1 Mbps and window size = 2

(*b*) 10 km link of 200 Mbps and window size = 7

(*c*) 50000 km link of 2 Mbps and window size = 127

ANSWERS

I. FILL UP THE BLANKS

1. data link control 2. flow control 3. data link control 4. Data fields 5. Address, Header 6. Header and Trailer 7. High level data link control 8. frame 9. Flow control 10. Positive 11. negative 12. stop and wait flow control 13. sliding window flow control 14. stop and wait 15. inefficiency 16. stop and wait 17. stop and wait 18. sliding window 19. content errors, flow integrity errors 20. character oriented, bit oriented.

□□□□□