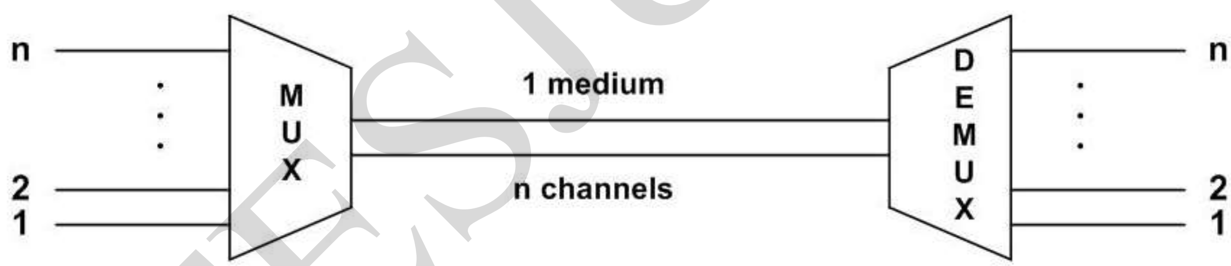
# DATA LINK LAYER

## Multiplexing

### Introduction

It has been observed that most of the individual data-communicating devices typically require modest data rate. But, communication media usually have much higher bandwidth. As a consequence, two communicating stations do not utilize the full capacity of a data link. Moreover, when many nodes compete to access the network, some efficient techniques for utilizing the data link are very essential. When the bandwidth of a medium is greater than individual signals to be transmitted through the channel, a medium can be shared by more than one channel of signals. The process of making the most effective use of the available channel capacity is called **Multiplexing**. For efficiency, the channel capacity can be shared among a number of communicating stations just like a large water pipe can carry water to several separate houses at once. Most common use of multiplexing is in long-haul communication using coaxial cable, microwave and optical fibre.

Figure 2.7.1 depicts the functioning of multiplexing functions in general. The **multiplexer** is connected to the **demultiplexer** by a single data link. The multiplexer combines (multiplexes) data from these 'n' input lines and transmits them through the high capacity data link, which is being demultiplexed at the other end and is delivered to the appropriate output lines. Thus, **Multiplexing** can also be defined as a technique that allows simultaneous transmission of multiple signals across a single data link.




By **Multiplexing** different message signals can share a single transmission media (The media can be guided or unguided). All they need is they should either differ in their frequency slot or wavelength slot or in time slot.

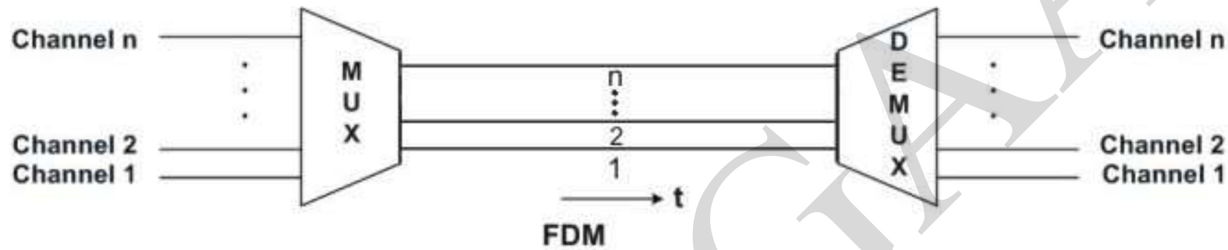
When two communicating nodes are connected through a media, it generally happens that bandwidth of media is several times greater than that of the communicating nodes. Transfer of a single signal at a time is both slow and expensive. The whole capacity of the link is not being utilized in this case. This link can be further exploited by sending several signals combined into one. This combining of signals into one is called multiplexing.

### Multiplexing is of three types :- FDM, WDM and TDM

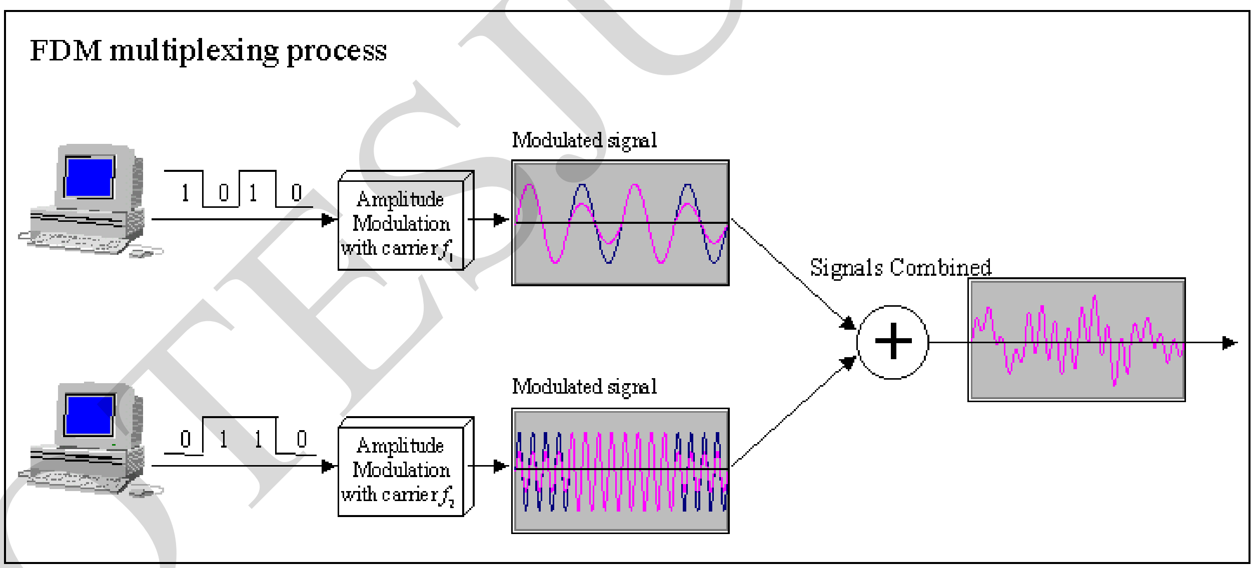
# 1. Frequency-Division Multiplexing (FDM)

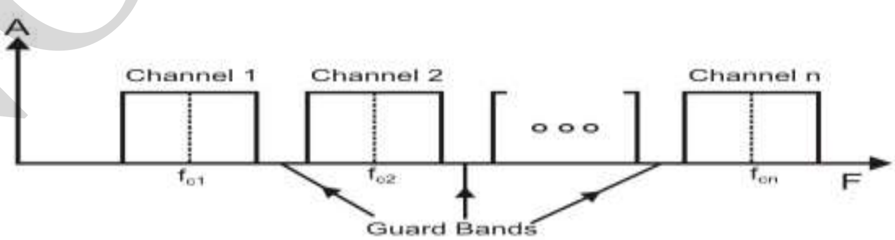
In frequency division multiplexing, the available bandwidth of a single physical medium is subdivided into several independent frequency channels. Independent message signals are translated into different frequency bands using modulation techniques, which are combined in the multiplexer, to a composite signal. The resulting signal is then transmitted along the single channel by electromagnetic means.

Basic approach is to divide the available bandwidth of a single physical medium into a number of smaller, independent frequency channels. Using modulation, independent message signals are translated into different frequency bands. All the modulated signals are combined in a linear summing circuit to form a composite signal for transmission.




At the receiving end the signal is applied to a bank of band-pass filters, which separates individual frequency channels. The band pass filter outputs are then demodulated and distributed to different output channels.



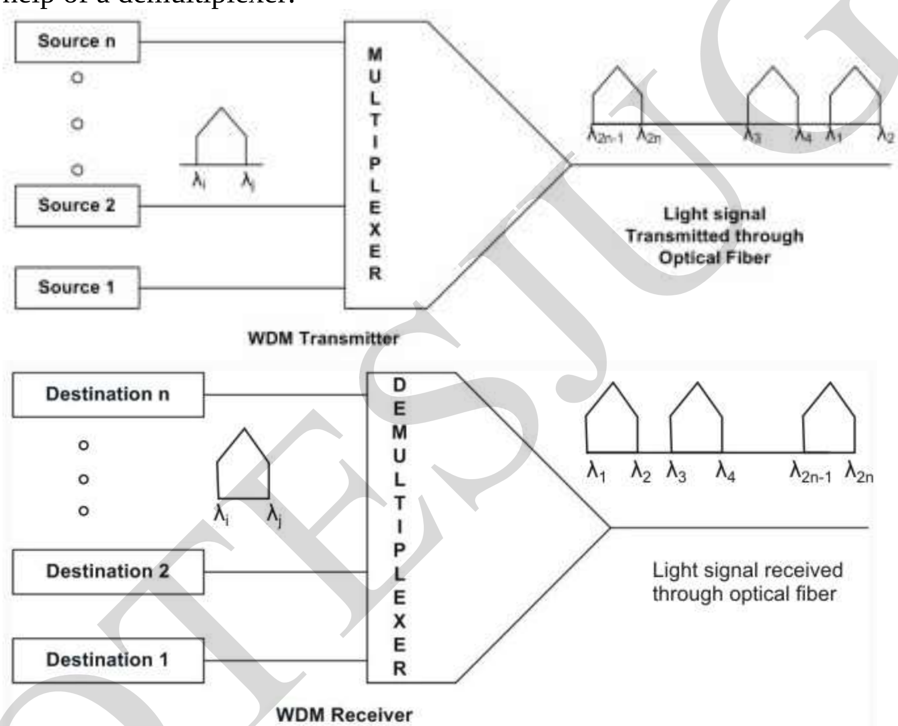



If the channels are very close to one other, it leads to inter-channel cross talk. Channels must be separated by strips of unused bandwidth to prevent inter-channel cross talk. These unused channels between each successive channel are known as **guard band**s.

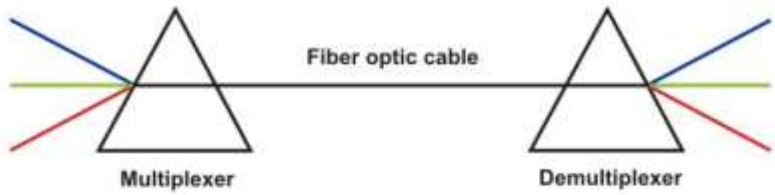
FDM are commonly used in radio broadcasts and TV networks.

## 2.Wavelength-Division Multiplexing

Wavelength-division multiplexing (WDM) is conceptually same as the FDM, except that the multiplexing and demultiplexing involves light signals transmitted through fibre-optic channels. The idea is the same: we are combining different frequency signals. However, the difference is that the frequencies are very high. It is designed to utilize the high data rate capability of fibre-optic cable. Very narrow band of light signal from different source are combined to make a wider band of light. At the receiver the signals are separated with the help of a demultiplexer.




Multiplexing and demultiplexing of light signals can be done with the help of a **prism** as shown in Fig.
One prism performs the role of a multiplexer by combining lights having different frequencies from different sources. The composite signal can be transmitted through an optical fibre cable over long distances, if required. At the other end of the optical fibre cable the composite signal is applied to another prism to do the reverse operation, the function of a demultiplexer.



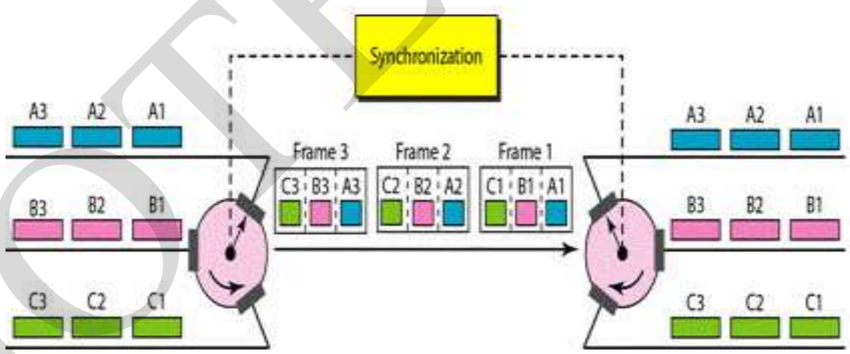

## 3. TIME DIVISION MULTIPLEXING

A type of multiplexing where two or more channels of information are transmitted over the same media by allocating a different time interval ("slot" or "slice") for the transmission of each channel. The channels take turns to use the media.

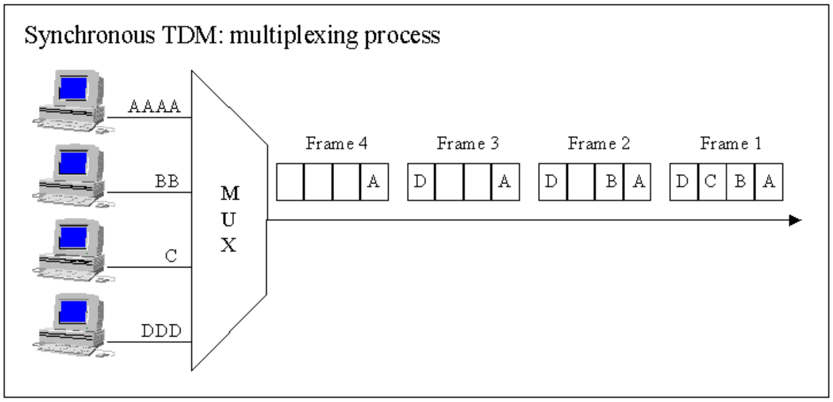
The main reason to use TDM is to take advantage of existing transmission lines. It would be very expensive if each low-bit-rate stream were assigned a costly physical channel (say, an entire fiber optic line) that extended over a long distance.

### *Interleaving*

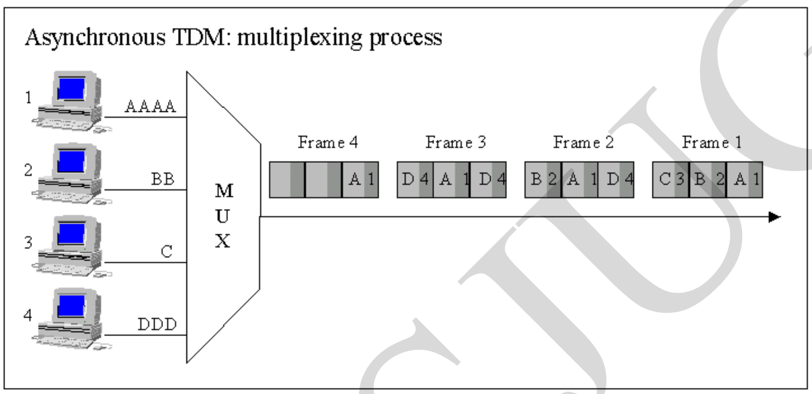
- TDM can be visualized as two fast-rotating switches, one on the multiplexing side and the other on the demultiplexing side.
- The switches are synchronized and rotate at the same speed, but in opposite directions.
- On the multiplexing side, as the switch opens in front of a connection, that connection has the opportunity to send a unit onto the path. This process is called **interleaving.**
- On the demultiplexing side, as the switch opens in front of a connection, that connection has the opportunity to receive a unit from the path.




**Synchronous TDM:** Time slots are preassigned and are fixed. Each source is given it's time slot at every turn due to it. This turn may be once per cycle, or several turns per cycle, if it has a high data transfer rate, or may be once in a no. of cycles if it is slow. This slot is given even if the source is not ready with data. So this slot is transmitted empty.




**Asynchronous TDM/ Statistical TDM:** In this method, slots are not fixed. They are allotted dynamically depending on speed of sources, and whether they are ready for transmission.




## <u>ERROR DETECTION and CORRECTION</u>

# Introduction

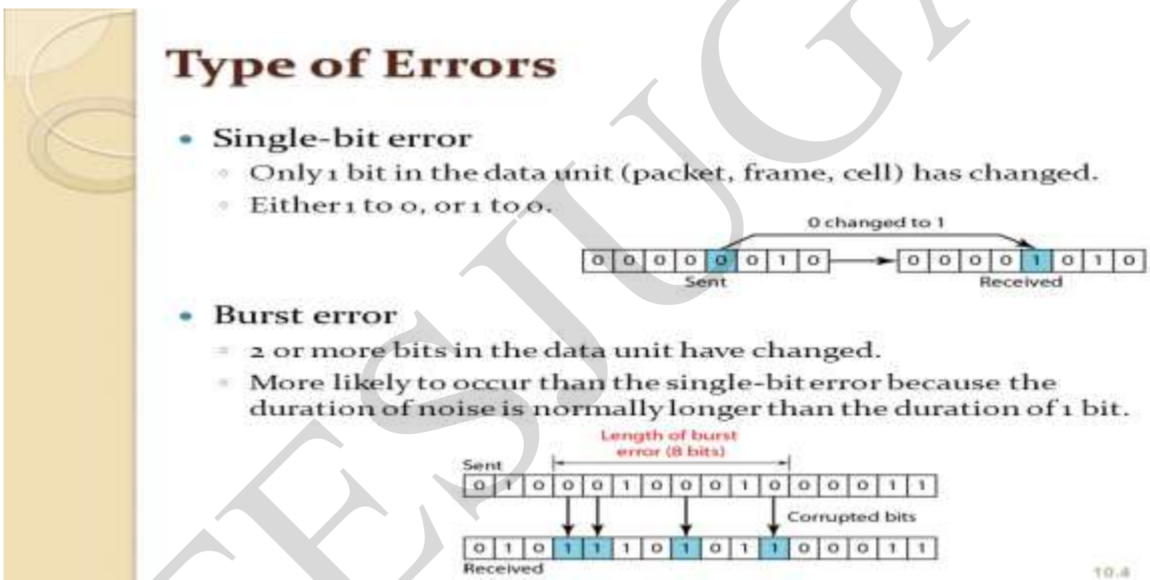
- Environmental interference and physical defects in the communication medium can cause random bit errors during data transmission. Error coding is a method of detecting and correcting these errors to ensure information is transferred intact from its source to its destination.
- Error coding uses mathematical formulas to encode data bits at the source into longer bit words for transmission. The "code word" can then be decoded at the destination to retrieve the information.
- The extra bits in the code word provide *redundancy* that, according to the coding scheme used, will allow the destination to use the decoding process to determine

if the communication medium introduced errors and in some cases correct them so that the data need not be retransmitted.

- Different error coding schemes are chosen depending on the types of errors expected, the communication medium's expected error rate, and whether or not data retransmission is possible.

There are two basic strategies for dealing with errors. One way is to include enough redundant information (extra bits are introduced into the data stream at the transmitter on a regular and logical basis) along with each block of data sent to enable the receiver to deduce what the transmitted character must have been.

The other way is to include only enough redundancy to allow the receiver to deduce that error has occurred, but not which error has occurred and the receiver asks for a retransmission. The former strategy uses **Error-Correcting Codes** and latter uses **Error-detecting Codes**.




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

## Error Detecting Methods

Basic approach used for error detection is the use of redundancy, where additional bits are added to facilitate detection and correction of errors.

**Popular techniques are:**

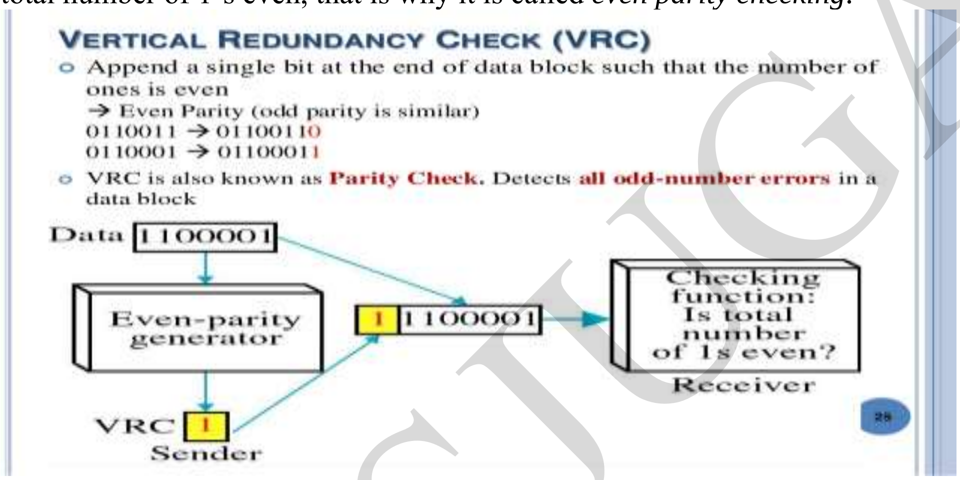
**1) Simple Parity check / Vertical redundancy check (VRC)**
**2) Two-dimensional Parity check / Longitudinal redundancy check(LRC)**

**3) Cyclic redundancy check (CRC)**
**4) Checksum**

## 1. Simple Parity Checking or One-dimension Parity Check

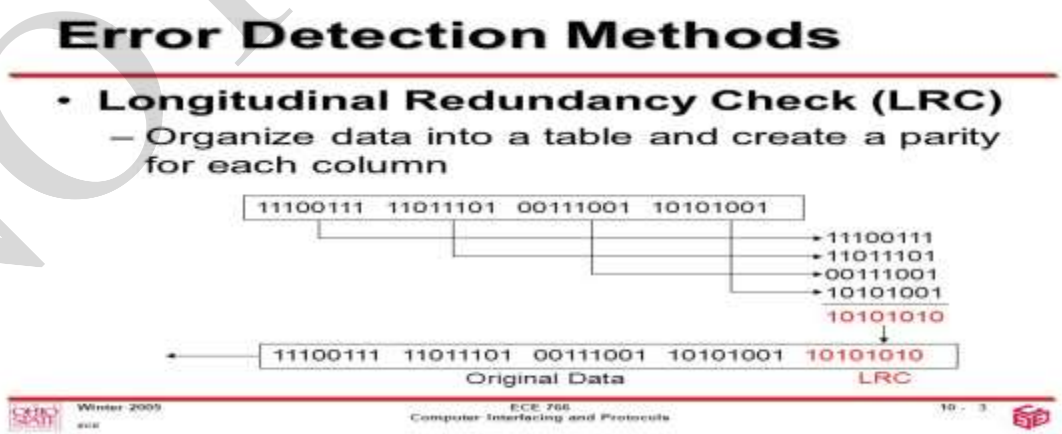
The most common and least expensive mechanism for error- detection is the simple parity check. In this technique, a redundant bit called **parity bit**, is appended to every data unit so that the number of 1s in the unit (including the parity becomes even).

Blocks of data from the source are subjected to a check bit or *Parity bit* generator form, where a parity of 1 is added to the block if it contains an odd number of 1's (ON bits) and 0 is added if it contains an even number of 1's. At the receiving end the parity bit is computed from the received data bits and compared with the received parity bit. This scheme makes the total number of 1's even, that is why it is called *even parity checking*.




## 2. Longitudinal Redundancy Check(LRC)

This method organizes the block of bits in the form of a table. Parity check bits are calculated for each column, which is equivalent to a simple parity check bit. The parity code generated is sent along with the data. At the receiving end these are compared with the parity bits calculated on the received data.



## 3. Checksum

In checksum error detection scheme, the data is divided into k segments each of m bits. At the sender's end the segments are added using 1's complement arithmetic to get the sum.

The sum is complemented to get the checksum. The checksum segment is sent along with the data segments. At the receiver's end, all received segments are added using 1's complement arithmetic to get the sum. The sum is complemented. If the result is zero, the received data is accepted; otherwise discarded.

**Eg:-** Suppose the following block of 16 bits is to be sent with segment of 8 bits.

10101001 00111001

Adding the numbers
10101101
00111001
-------------
11100010  (sum)

**00011101 (Checksum bits)** (complement of sum)

The codeword sent is 10101001 00111001 00011101

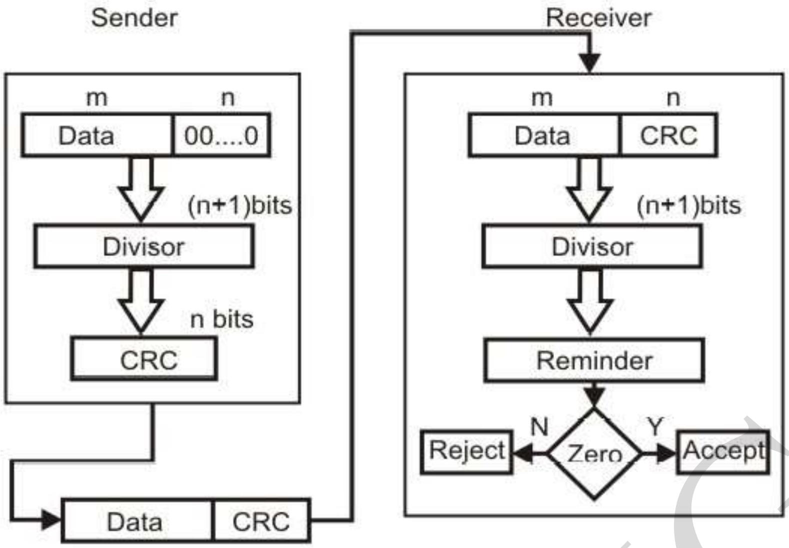
## 4. Cyclic Redundancy Check (CRC)

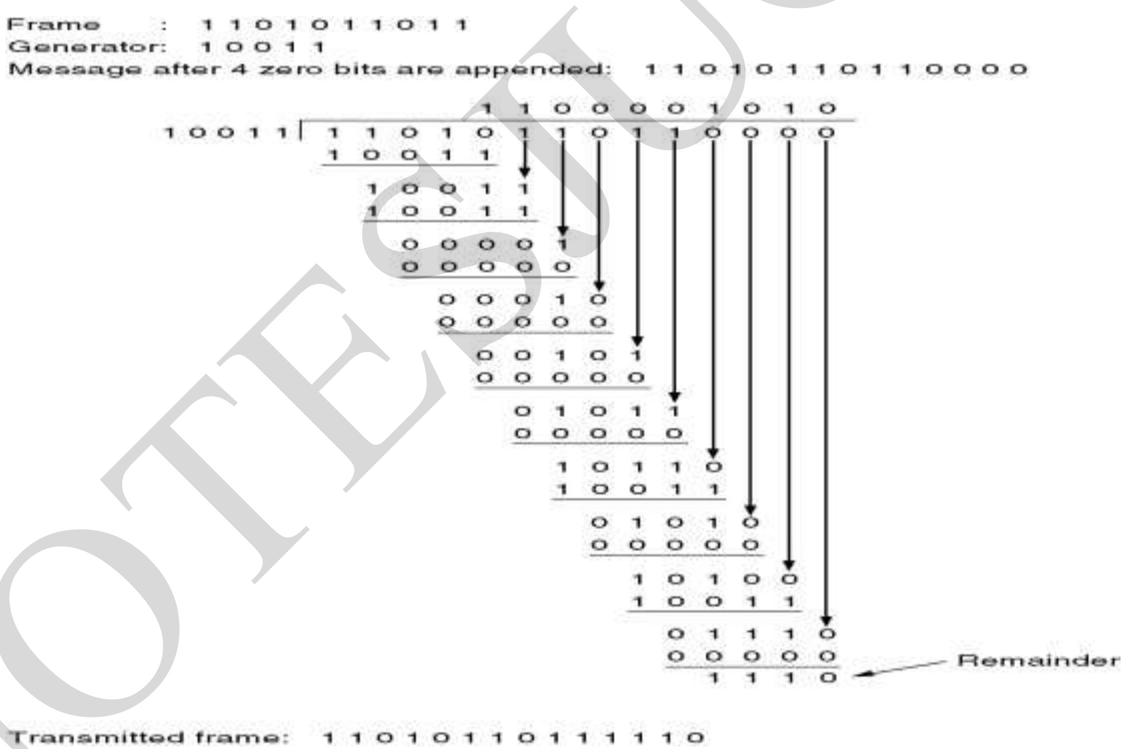
The Cyclic Redundancy Check is the most powerful and easy to implement technique. CRC is based on binary division. In CRC, a sequence of redundant bits, called **cyclic redundancy check bits**, are appended to the end of data unit so that the resulting data unit becomes exactly divisible by a predetermined binary number.
At the destination, the incoming data unit is divided by the same number. If at this step there is no remainder, the data unit is assumed to be correct and is therefore accepted. A remainder indicates that the data unit has been damaged in transit and therefore must be rejected. The generalized technique can be explained as follows.

If a *k* bit message is to be transmitted, the transmitter generates an *r*-bits sequence, so that the *(k+r)* bits are actually being transmitted. Now this *r*-bits is generated by dividing the original number, appended by *r* zeros, by *a* predetermined number.

This number, which is *(r+1)* bit in length, can also be considered as the coefficients of a polynomial, called G*enerator Polynomial.* The remainder of this division process generates the *r*-bits which are called as the CRC bits. The CRC bits are one less than the divisor.

On receiving the packet, the receiver divides the *(k+r)* bit frame by the same predetermined number and if it produces no remainder, it can be assumed that no error has occurred during the transmission. Operations at both the sender and receiver end are shown in Fig.




Commonly used divisor polynomials are:

CRC-16 = X16 + X15 + X2 + 1

CRC-CCITT = X16 + X12 + X5 + 1

CRC-32 = X32 + X26 + X23 + X22 + X16 + X12 + X11 + X10 + X8 + X7 + X5 + X4 + X2 + 1

**Performance**
CRC is a very effective error detection technique. If the divisor is chosen according to the previously mentioned rules, its performance can be summarized as follows:

CRC can detect all single-bit errors
CRC can detect all double-bit errors (three 1's)
CRC can detect any odd number of errors (X+1)
CRC can detect all burst errors of less than the degree of the polynomial.
CRC detects most of the larger burst errors with a high probability.

## ERROR CORRECTION

Error correction can be done in two ways :- first have the sender retransmit the data which is also called **backward error correction.**
Second, apply an error correction method at receiver, also called as **forward error correction.**

Error correcting codes are however more sophisticated than the error detecting codes. Hence the error correction is limited to one, two, or fewer bit errors.

### Hamming Method of Error Correction(HEC)

This scheme is due to R.Hamming, a well-known researcher in the field of telecommunications. Like any other scheme, Hamming Error Correcting Code (HEC) augments redundancy bits using which it identifies which bit is in error.

The message to be transmitted along with the redundancy bits is m + r and the receiver can be in one of the following states.

(i) No error, i.e., none of m + r bits is in error
**(ii)** the first bit is in error
**(iii)**the second bit is in error and so on.

This implies that $2^r \geq (m+r)+1$, where $2^r$ denotes the number of possible configurations of redundancy bits and each configuration represents a state of the receiver, and the number of such states is (m+r)+ 1

HEC follows even parity scheme.

Suppose the message to be transmitted is 1 0 0 1 1 0 1 0. Since m = 8, r = 4. We associate four redundancy bits r1, r2, r4, r8 which we compute using the values present in the following bit positions.

The value of r1 is '1' if the number of 1's in the following bit positions including the bit r1 is even. Similarly, for r2 and other redundancy bits.

r1 = 1, 3, 5, 7, 9, 11
r2 = 2, 3, 6, 7, 10, 11
r4 = 4, 5, 6, 7, 12
r8 = 8, 9, 10, 11, 12

| Bit Positions | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Message | 1 | 0 | 0 | 1 | | 1 | 0 | 1 | | 0 | | |

For the above message the values of r1 = 1, r2 = 1, r4 = 1 and r8 = 0.
The message to be transmitted is 1 0 0 1 0 1 0 1 1 0 1 1
On receiving the message, the receiver computes the values of four redundancy bits r1, r2, r4, r8 to ascertain whether there was an error in the data transmission and these bits also help us in identifying the bit in error.

Suppose there is no error in transmission. Then the received message is 1 0 0 1 0 1 0 1 1 0 1 1 .
It is easy to see that r1 = 0, r2 = 0, r4 = 0 and r8 = 0. This corresponds to the decimal value '0' which means there is no error in the transmission.

Suppose 3rd bit is in error. Then the received message is 1 0 0 1 0 1 0 1 1 11 1

On computing redundancy bits, we get r1 = 1, r2 = 1, r4 = 0 and r8 = 0. This corresponds to the decimal value '3' which means 3rd bit is in error and the receiver changes the third bit to get the actual message.

## FUNCTIONS OF DATA LINK LAYER

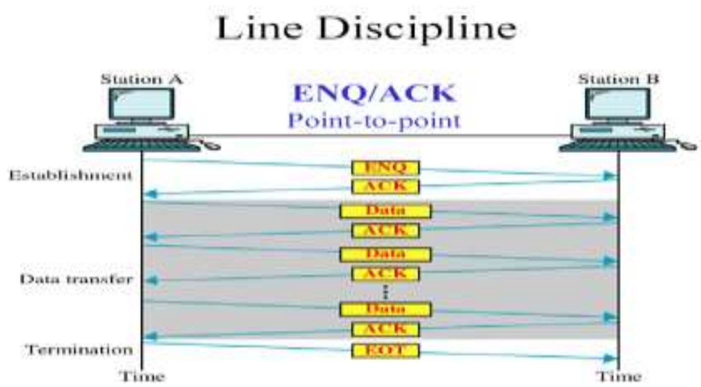
The most important functions of Data Link layer are **line control**, **error control** and **flow control**. Collectively, these functions are known as **data link control.**

**Line Discipline or Line control is of 2 categories –**

- **ENQ/ACK**
- **POLL/SELECT**

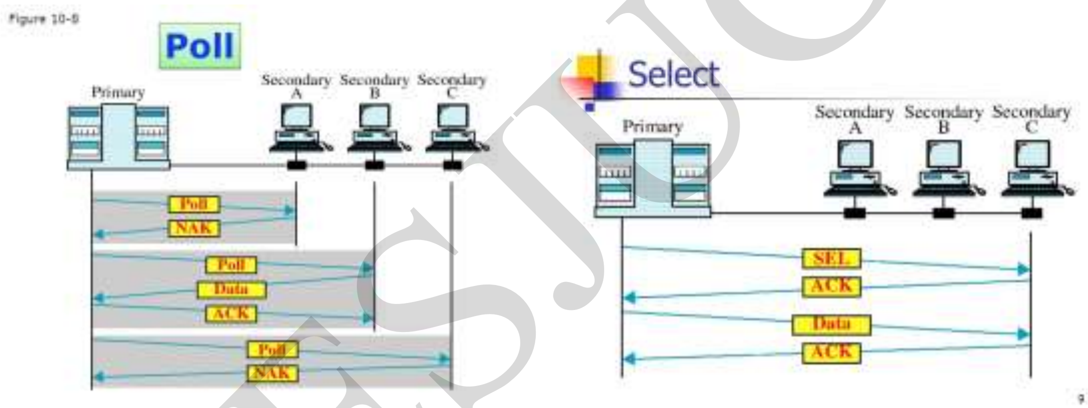
**ENQ/ACK: (Enquiry/Acknowledgement)**

- It is used in systems when there is a dedicated link between two devices.
- The sender first transmits a frame (ENQ) to the receiver to check whether ready to receive data or not.
- The receiver sends (ACK) if ready or NAK if not ready.
- If neither received, then sender sends ACK again. (makes 3 attempts)
- If ACK is received, the sender sends data, and finishes with a EOT (End of transmission) frame.



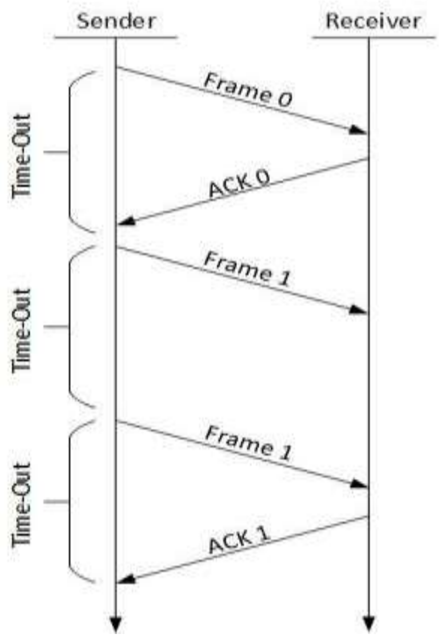

**POLL/SELECT:**

- This method works in topologies where one device is primary and the other devices are secondary.
- If the primary wants to receive data, it asks the secondary devices if they have anything to send, this is called **polling**.
- If the primary wants to send data, it tells the target secondary device to get ready to receive, this function is called **selecting.**




**Flow Control** is a technique so that transmitter and receiver with different speed characteristics can communicate with each other. Flow control ensures that a transmitting station, such as a server with higher processing capability, does not overwhelm a receiving station, such as a desktop system, with lesser processing capability.

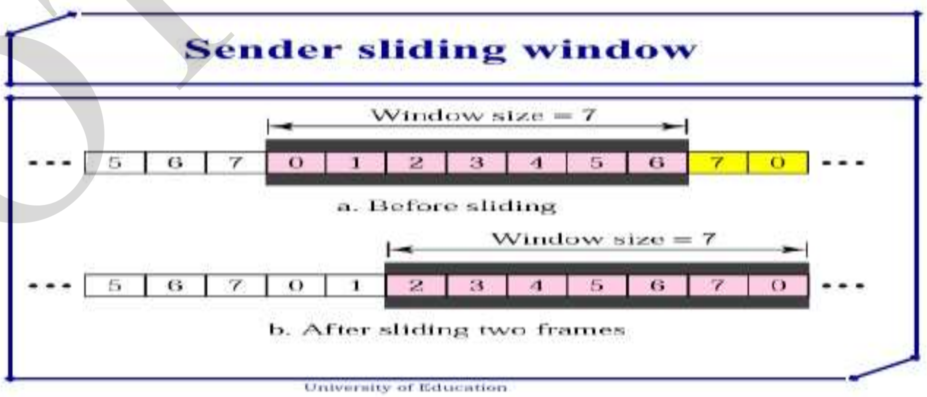
There are two methods developed for flow control namely **Stop-and-wait and Sliding-window.**
Stop-and-wait is also known as Request/reply sometimes. Request/reply (Stop-and-wait) flow control requires each data packet to be acknowledged by the remote host before the next packet is sent.

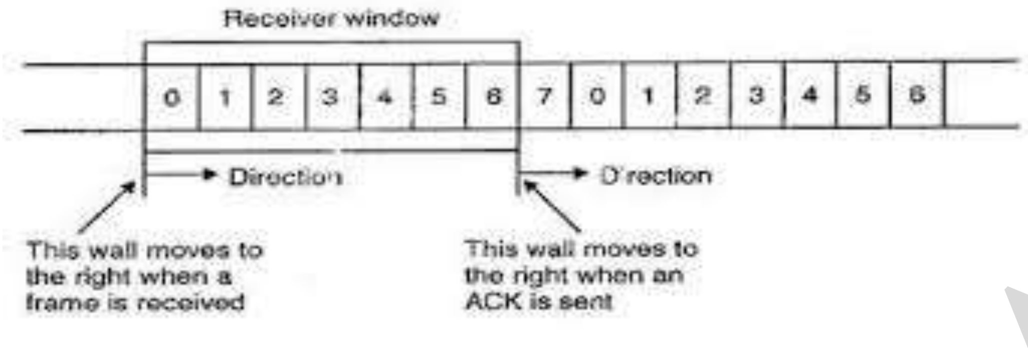



Major drawback of Stop-and-Wait Flow Control is that only one frame can be in transmission at a time, this leads to inefficiency.

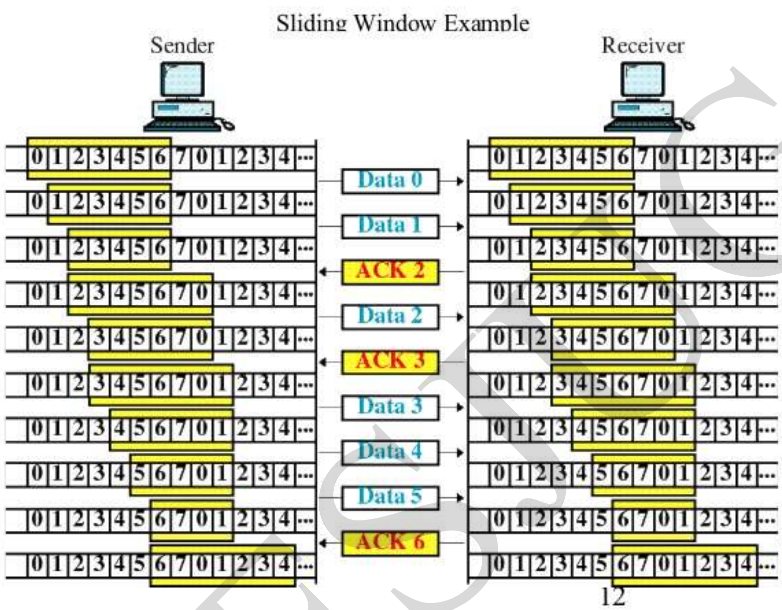
## SLIDING WINDOW METHOD OF FLOW CONTROL

- In this method, the sender can send multiple frames before needing an acknowledgement.
- The receiver acknowledges some of the frames, using a single ACK to confirm the receipt of multiple frames.
- Sliding window refers to imaginary boxes at both sender and receiver. This window can hold frames and provides an upper limit to the number of frames that can be transmitted.
- To keep track, frames are numbered 0 to n-1. (eg: if n=8, frames are 0,1,2,3,4,5,6,7). The size of window is n-1.
- When the receiver sends an ACK, it includes the number of the next frame it expects.



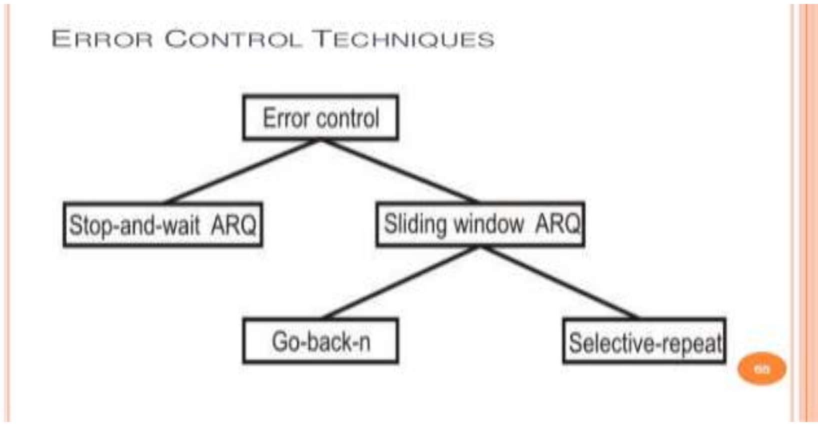



Sliding window on receiver side



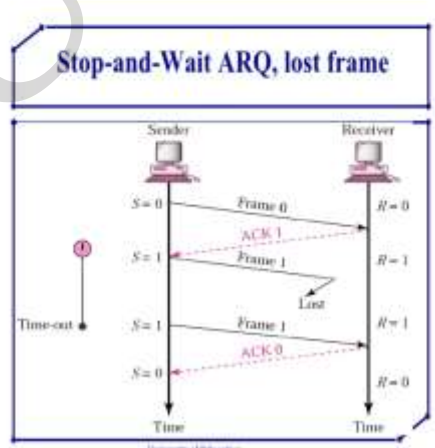

## ERROR CONTROL IN DATA LINK LAYER

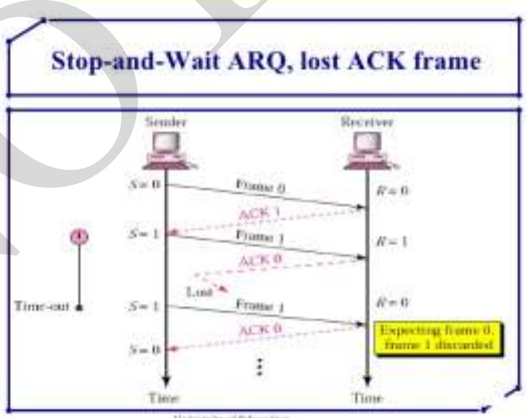
When an error is detected in a message, the receiver sends a request to the transmitter to retransmit the ill-fated message or packet. The most popular retransmission scheme is known as Automatic-Repeat-Request (ARQ).

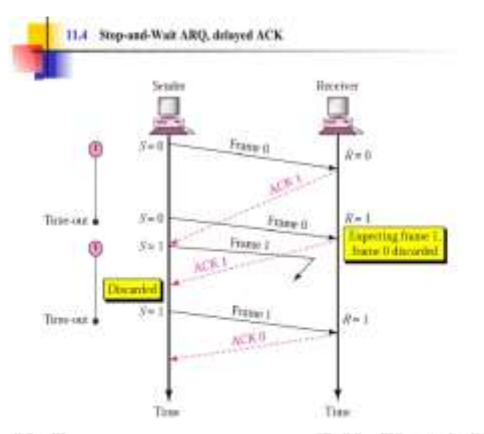



## Stop-and-Wait ARQ

- In Stop and wait ARQ, the sender keeps a copy of the frame transmitted, until an acknowledgement is recd. Both data frame and ACK frames are numbered alternatively 0 and 1.
- If error detected, NAK frame returned, and the last frame transmitted is again sent.
- The sender keeps a timer, if acknowledgement is not received within a time period, the last frame sent is again sent.

**Go-back-N ARQ**

- The most popular ARQ protocol is the go-back-N ARQ, where the sender sends the frames continuously without waiting for acknowledgement.
- That is why it is also called as *continuous ARQ*. As the receiver receives the frames, it keeps on sending ACKs or a NAK, in case a frame is incorrectly received.
- When the sender receives a NAK, it retransmits the frame in error plus all the succeeding frames. Hence, the name of the protocol is go-back-N ARQ.
- If a frame is lost, the receiver sends NAK after receiving the next frame.
- In case there is long delay before sending the NAK, the sender will resend the lost frame after its timer times out.
- If the ACK frame sent by the receiver is lost, the sender resends the frames after its timer times out.

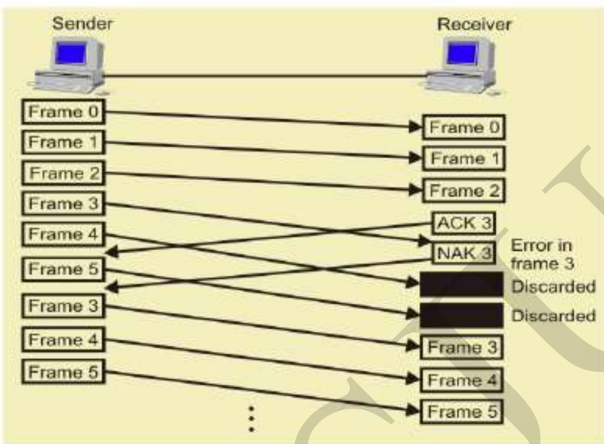



**Figure 3.3.9** Frames in error in go-Back-N ARQ

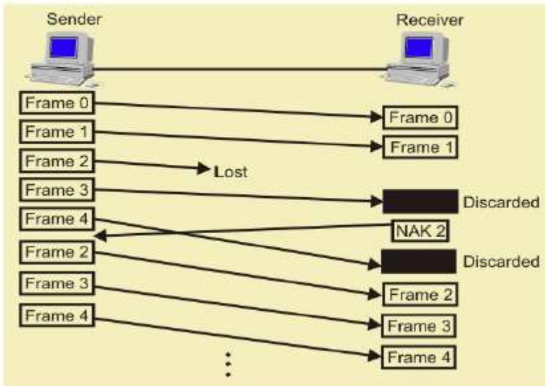



**Figure 3.3.10** Lost Frames in Go-Back-N ARQ

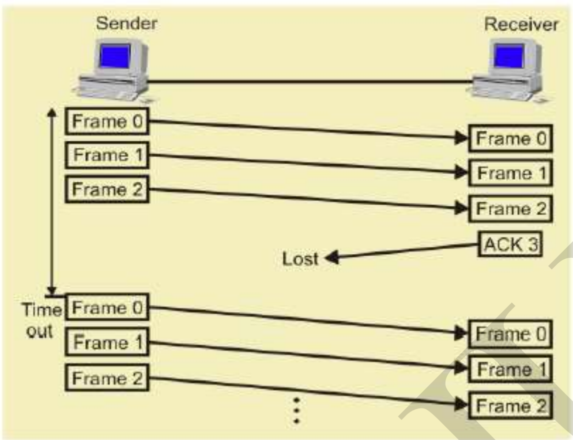



**Figure 3.3.11** Lost ACK in Go-Back-N ARQ

## Selective-Repeat ARQ

The selective-repetitive ARQ scheme retransmits only those for which NAKs are received or for which timer has expired, This is the most efficient among the ARQ schemes, but the sender must be more complex so that it can send out-of-order frames.

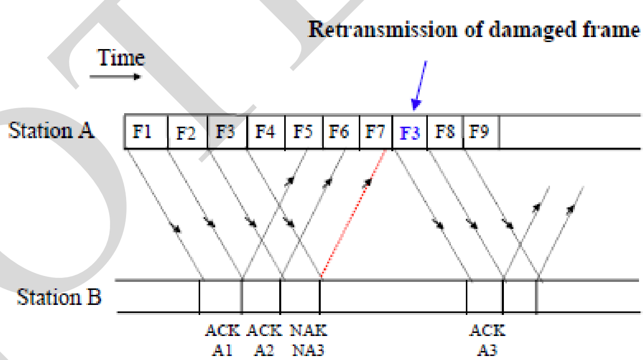
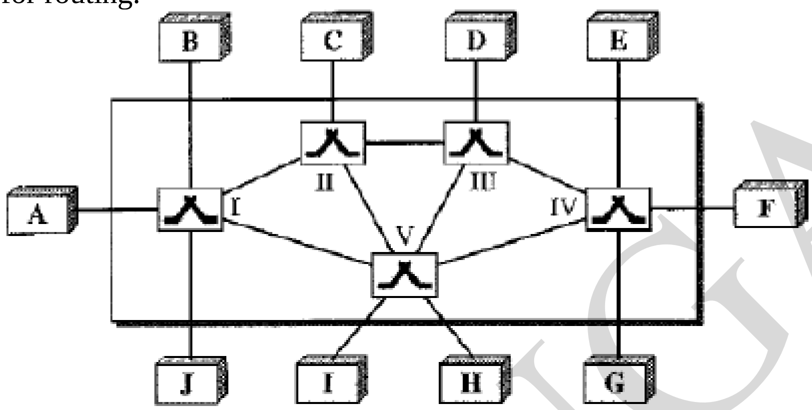



**Figure 3.3.12** Selective-repeat Reject

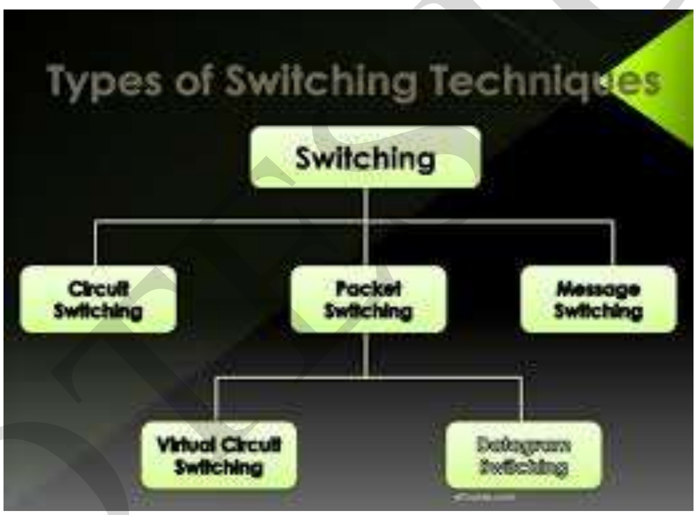
## *Switching*

A network is a set of connected devices. Whenever we have multiple devices, we have the problem of how to connect them to make one-to-one communication possible.

A solution is switching. A switched network consists of a series of interlinked nodes, called switches. Switches are devices capable of creating temporary connections between two or more devices linked to the switch. In a switched network, some of these nodes are connected to the end systems (computers or telephones, for example). Others are used only for routing.




The end systems (communicating devices) are labeled A, B, C, D, and so on, and the switches are labeled I, II, III, IV, and V. Each switch is connected to multiple links.




### Circuit Switching

A circuit-switched network consists of a set of switches connected by physical links. A connection between two stations is a dedicated path made of one or more links. However, each connection uses only one dedicated channel on each link.

The actual communication in a circuit-switched network requires three phases: connection setup, data transfer, and connection teardown.

Setup Phase : Before the systems can communicate, a dedicated circuit (combination of channels in links) needs to be established. The end systems are normally connected through dedicated lines to the switches, so connection setup means creating dedicated channels between the switches.
Data Transfer Phase : After the establishment of the dedicated circuit (channels), the two systems can transfer data.
Teardown Phase : When one of the systems needs to disconnect, a signal is sent to each switch to release the resources.

**PACKET SWITCHING**
In a Computer Network, the communication between two ends is done in blocks of data called packets. So instead of continuous communication the exchange takes place in the form of individual packets between the two computers.

**2 approaches are followed:**
**Datagram approach**
**Virtual Circuit approach**

In a **datagram network,** each packet is treated independently of all others. Even if a packet is part of a multi packet transmission, the network treats it as though it existed alone. Packets in this approach are referred to as datagrams.

This approach can cause the datagrams of a transmission to arrive at their destination out of order with different delays between them packets. It is the responsibility of an upper layer protocol to reorder the datagrams or ask for lost datagrams before passing them on to the application.

The datagram networks are sometimes referred to as connectionless networks.

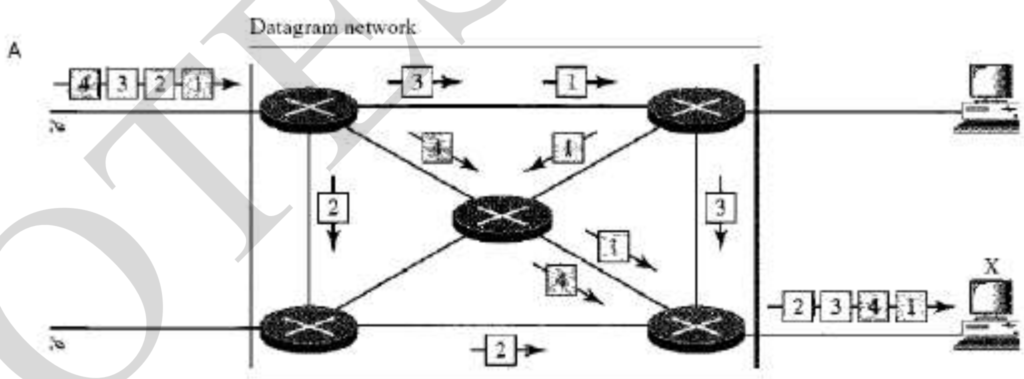



Fig: *A datagram network with four switches (routers)*

**VIRTUAL-CIRCUIT NETWORKS**:
A virtual-circuit network is a cross between a circuit-switched network and a datagram network. It has some characteristics of both.

- In this method, a single route is chosen between sender and receiver at the beginning of the session. When data is sent, all packets of the transmission travel along that route only.
- It is also known as **connection oriented approach**.
- Virtual circuit transmission is implemented in two formats:
    - SVC (Switched Virtual Circuit)
    - PVC (Permanent Virtual Circuit)

SVC (Switched Virtual Circuit)
In this method, a virtual circuit is created whenever it is needed and exists only for the duration of that specific exchange.

PVC (Permanent Virtual Circuit)
In this method, the same virtual circuit is used between two users for transmission on a permanent basis.

**MESSAGE SWITCHING:**
It is also known as Store and Forward Switching.
In this method, a node receives a message, stores it until the appropriate route is free and then sends it.
The messages are stored and relayed from the secondary storage.