**UNIT II**

**Data Communication Interface**

* Asynchronous and Synchronous Transmission
* Line Configuration
* Interfacing

**Data link controls**

* Flow control
* Error detection
* Error control
* HDLC
* Other data link control protocols
* Performance issues

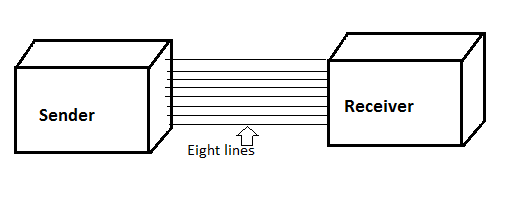
**DATA COMMUNICATION INTERFACE,**

**DATA LINK CONTROL**

* Transmission of binary data across a link can be accomplished in either parallel or serial mode.
* Serial transmission of data means data is transferred over a single patch rather than parallel set of lines.
* With serial transmission signaling elements are sent down the line one at a time.

**Parallel Transmission:**

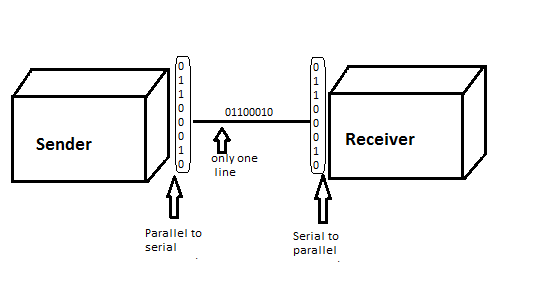
* m/sm for parallel transmission is conceptually simple one.
* Use n wires to send n-bits at one time. Each bit has its own wire , and all n bits of a group can be transmitted with each clock track from one device to another.
* If n=8(data bits)
* Eight wires are bounded in a cable with a connector at each end.



* + - Fig: Parallel transmission
* **Advantage:** Parallel transmission can increase transfer speed by a factor of n over serial transmission.
* **Disadvantage:** Requires n-communication lines just to transmit bit stream.
* -Limited to short distances.

**Serial Transmission**

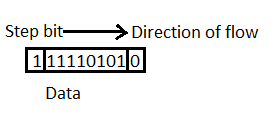
* In serial transmission one bit follows another, so we need only one communication rather than n to transmit data between two communication devices.

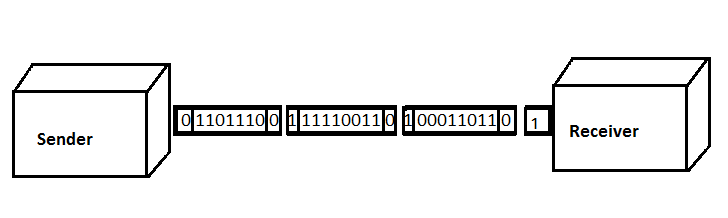


* + - Fig: Serial Transmission

**Asynchronous Transmission**

* In Asynchronous Transmission timing of signal is not important. Information is received and transmitted upon by agreed upon patterns.
* Pattern is based up on grouping of the bit straight into bytes.
* In each group usually 8-bits are sent along the link as a unit.
* Sending system handles each group independently without regard to a timer.
* Without Synchronization receiver can’t use timing to predict when the next group will arrive.
* To alert the receiver the arrival of new group, extra bits were added. We send a bit(0)at beginning and one or more step bits (1’s) at end of each byte.
* There may be gap between each byte. Gap can be represented either by an idle channel or a stream of additional step bits.
* This mechanism is called asynchronous because, at the byte level sender and receiver need not have to be synchronized.

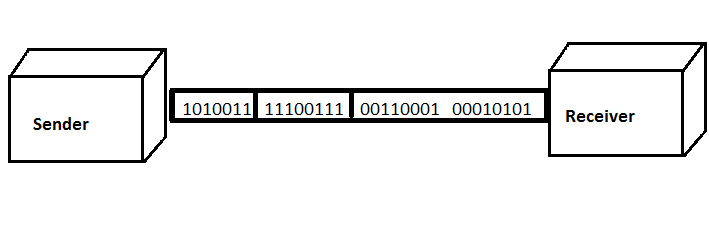




* + - * + Fig: Asynchronous transfer
* **Advantages:** cheap effective units.
* **Disadvantages:** less speed.

**Synchronous Transmission**

* In synchronous transmission bit stream is combined into longer frames which may contain multiple bytes.
* We send(Frames) bits one after another without start/stop bits or gaps.
* It is responsibility of the receiver to group bits into bytes for decoding purpose.

 Fig: Synchronous Transfer

* If sender wishes to send data in separate bursts, gaps between bursts must be filled with a special sequence of 0’s and 1’s,which means idle.
* **Advantages:** speed
* With synchronous transmission there is another level of synchronization required to allow the receiver to determine the beginning and end of a block of data.
* Each block begins with a preamble bit pattern and generally ends with a postamble bit pattern.
* Data plus preamble, postamble and control information is called a frame.
* typically Frames start with preamble called flag which is eight bit long. Same flag is used as postamble.

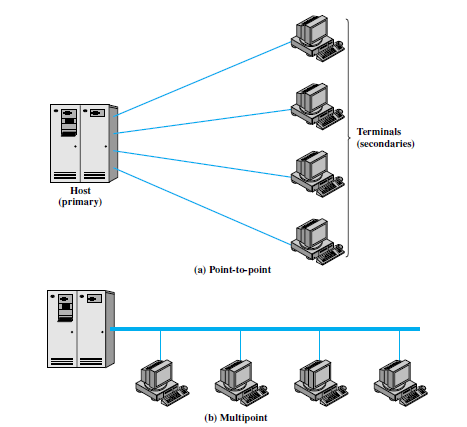
**Line configuration:**

* Two characteristics that distinguish various data link configuration are:

i) Topology ii) Type of link –Half duplex or Full duplex

**Topology:**

* The topology of a data link refers to the physical arrangement of stations on a trans-mission medium.
* If there are only two stations (e.g., a terminal and a computer or two computers), the link is point to point.
* If there are more than two stations, then it is a multipoint topology.
* If each terminal has a point-to-point link to its computer, then the computer must have one I/O port for each terminal.
* Also there is a separate transmission line from the computer to each terminal. In a multipoint configuration, the computer needs only a single I/O port and a single transmission line, which saves costs.



**Fig: Traditional Computer/Terminal Configurations**

**Full Duplex and Half Duplex:**

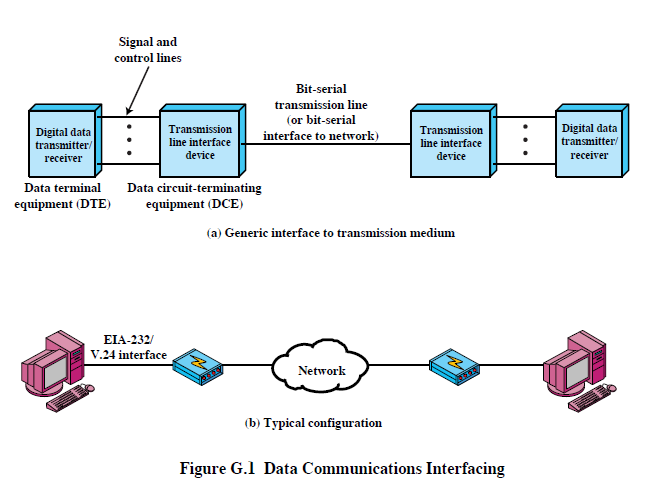
* Data exchanges over a transmission line can be classified as full duplex or half duplex.
* With half-duplex transmission, only one of two stations on a point-to-point link may transmit at a time. This mode is also referred to as *two-way alternate*. This can be compared to a one-lane, two-way bridge.
* In full-duplex transmission, two stations can simultaneously send and receive data from each other. Thus, this mode is known as *two-way simultaneous* and may be compared to a two-lane, two-way bridge. With digital signaling, which requires guided transmission, full-duplex operation usually requires two separate transmission paths (e.g., two twisted pairs), while half duplex requires only one.

**Interfacing:**

* With digital data processing devices have limited data processing capability. Typically, they generate simple digital signal, such as NRZ-L and DCE, across which they can transmit data is limited. Consequently it is rare for such a device to attach directly to a transmission or a networking facility.

**DTE:** (data terminal equipment)- Includes terminals and computers. DTE makes use of transmission system through the mediation of data circuit terminating equipment (DCE). Ex of DCE is modem

* The DCE is responsible for transmitting and receiving bits, one at a time, over
* a transmission medium or network.
* This requires both data and control information to be exchanged. This is done over a set of wires referred to as **interchange circuits**.
* The two DCEs that exchange signals over the transmission line or network must understand each other. That is, the receiver of each must use the same encoding scheme (e.g., Manchester, PSK) and data rate as the transmitter of the other.
* Each DTE-DCE pair must be designed to interact cooperatively.



* To ease the burden on data processing equipment manufacturers and users, standards have been developed that specify the exact nature of the interface between the DTE and the DCE.
* Such an interface has four important characteristics:
  + Mechanical
  + Electrical
  + Functional
  + Procedural

**Mechanical Characteristics:**

* The *mechanical characteristics* pertain to the actual physical connection of the DTE to the DCE.
* the signal and control interchange circuits are bundled into a cable with a terminator connector, male or female, at each end.
* The DTE and DCE must present connectors of opposite genders at one end of the cable, effecting the physical connection.
* This is analogous to the situation for residential electrical power. Power is provided via a socket or wall outlet, and the device to be attached must have the appropriate male connector (two-pronged, two-pronged polarized, or three-pronged) to match the socket.

**Electrical Characteristics:**

* The *electrical characteristics* deals with the voltage levels and timing of voltage changes.
* Both DTE and DCE must use the same code (e.g., NRZ-L), must use the same voltage levels to mean the same things, and must use the same duration of signal elements. These characteristics determine the data rates and distances that can be achieved.

**Functional Characteristics:**

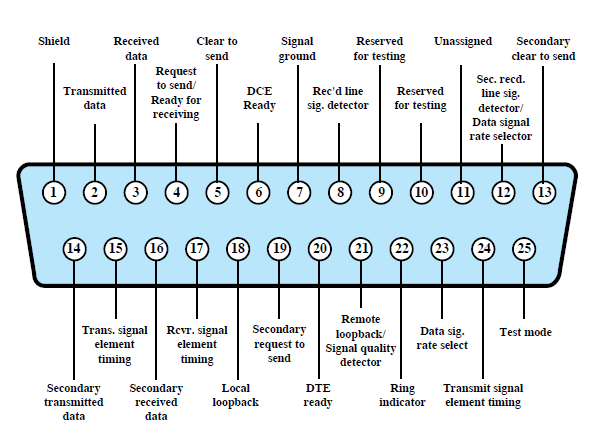
* *Functional characteristics* specify the functions that are performed by assigning meanings to each of the interchange circuits.
* Functions can be classified into the broad categories of data, control, timing, and electrical ground.

**Procedural Characteristics:**

* *Procedural characteristics* specify the sequence of events for transmitting data, based on the functional characteristics of the interface.
* Two main standard interfaces are:
* V.24/EIA-232-F
* ISDN Physical Interface.

**V.24/EIA-232-F**

* One of the most widely used interfaces is specified in the ITU-T standard, V.24. In fact, this standard specifies only the functional and procedural aspects of the interface; V.24 references other standards for the electrical and mechanical aspects.
* In the United States, there is a corresponding specification, virtually identical, that covers all four aspects: EIA-232-F.
* The correspondence is as follows:
  + Mechanical: ISO 2110
  + Electrical: V.28
  + Functional: V.24
  + Procedural V.24
* EIA-232 was first issued by the Electronic Industries Alliance in 1962, as RS-232. It is currently in its sixth revision, EIA-232-F, issued in 1997. The current V.24 and V.28 specifications were issued in 1996 and 1993, respectively.
* This interface is used to connect DTE devices to voice-grade modems for use on public analog telecommunications systems. It is also widely used for many other interconnection applications.

**Mechanical Specifications:**

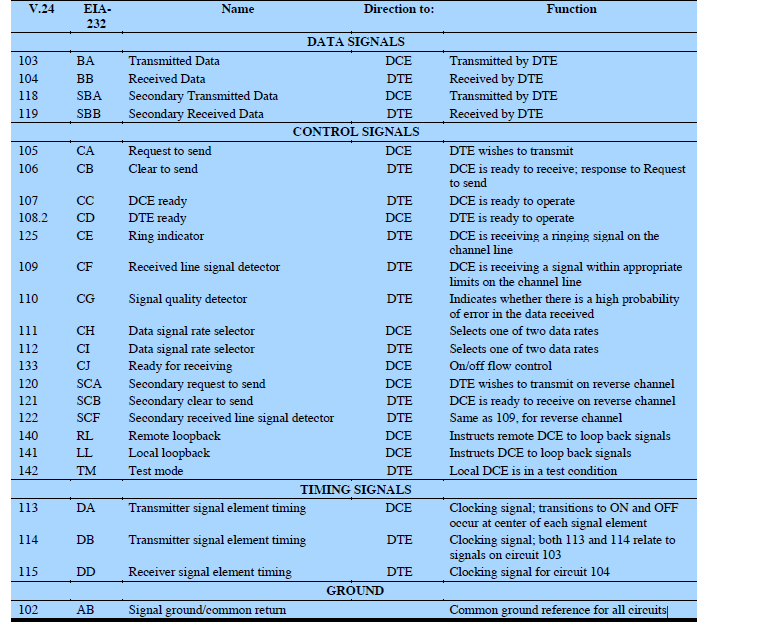
* The mechanical specification for EIA-232-F is illustrated in Figure above. It calls for a 25-pin connector, defined in ISO 2110, with a specific arrangement of leads.
* This connector is the terminating plug or socket on a cable running from a DTE (e.g., terminal) or DCE (e.g., modem).
* Though a 25-wire cable could be used to connect the DTE to the DCE, many applications require far fewer wires.

**Electrical Specifications:**

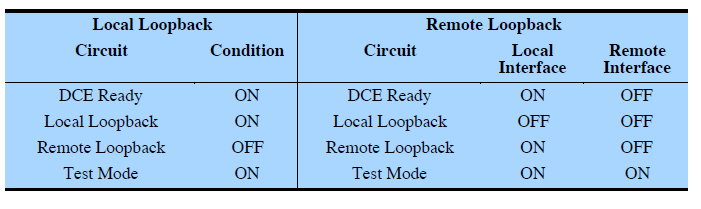
* The electrical specification defines the signaling between DTE and DCE. Digital signaling is used on all interchange circuits. Depending on the function of the interchange circuit, the electrical values are interpreted either as binary data or as control signals.
* The convention specifies that, with respect to a common ground, a voltage more negative than –3 volts is interpreted as binary 1 and a voltage more positive than +3 volts is interpreted as binary 0. This is the NRZ-L code. The interface is rated at a signal rate of <20 kbps and a distance of <15 meters.
* The same voltage levels apply to control signals: a voltage more negative than –3 volts is interpreted as an OFF condition and a voltage more positive than +3 volts is interpreted as an ON condition.

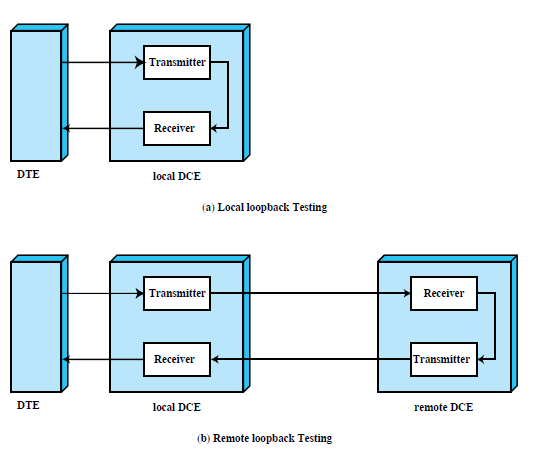
**Functional Specifications:**

* Table below summarizes the functional specification of the interchange circuits
* The circuits can be grouped into the categories of data, control, timing, and ground.
* There is one data circuit in each direction, so full duplex operation is possible. In addition, there are two secondary data circuits that are useful when the device operates in a half-duplex fashion.



* There are 16 control circuits. The first 10 of these listed in Table relate to the transmission of data over the primary channel.
* For asynchronous transmission, six of these circuits are used (105, 106, 107, 108.2, 125, 109).
* In addition to these six circuits, three other control circuits are used in synchronous transmission.
* The Signal Quality Detector circuit is turned ON by the DCE to indicate that the quality of the incoming signal over the telephone line has deteriorated beyond some defined threshold.
* The Data Signal Rate Selector circuits are used to change speeds; either the DTE or DCE may initiate the change.
* Circuit 133 enables a receiver to turn the flow of data on circuit 104 on and off.
* The next three control circuits (120, 121, 122) are used to control the use of the secondary channel, which may be used as a reverse channel or for some other auxiliary purpose.
* The last group of control signals relates to loopback testing. These circuits allow the DTE to cause the DCE to perform a loopback test. These circuits are only valid if the modem or other DCE supports loopback control.
* Local loopback: In the local loopback function, the transmitter output of the modem is connected to the receiver input, disconnecting the modem from the transmission line.
* A stream of data generated by the user device is sent to the modem and looped back to the user device.
* Remote loopback: the local modem is connected to the transmission facility in the usual fashion, and the receiver output of the remote modem is connected to the modem's transmitter input.
* Loopback control is a useful fault isolation tool. For example, suppose that a user at a personal computer is communicating with a server by means of a modem connection and communication suddenly ceases. The problem could be with the local modem, the communications facility, the remote modem, or the remote server.
* A network manager can use loopback tests to isolate the fault.
  + **Table: Loopback Circuit Settings for V.24/EIA-232**

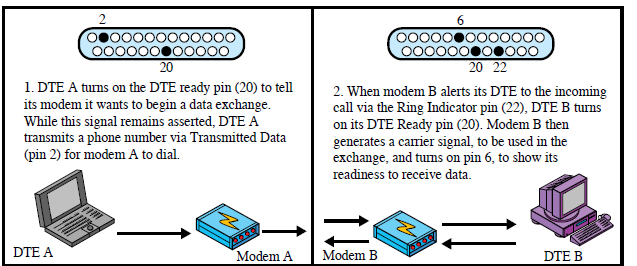


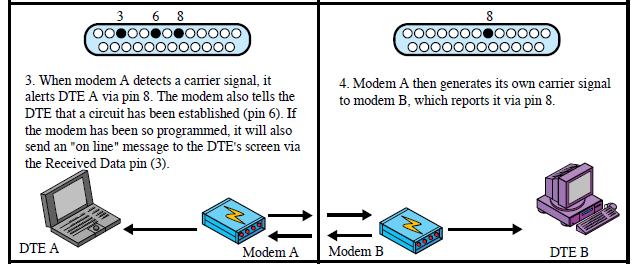


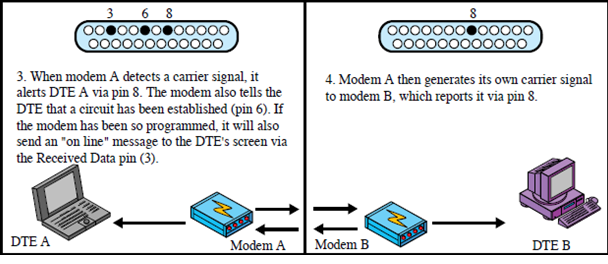
**Fig: Local and Remote loopback**

**Procedural Specifications:**

* The procedural specification defines the sequence in which the various circuits are used for a particular application.
* Examples:
* For connecting two devices over a short distance within a building. It is known as an asynchronous private line modem, or a limited distance modem.
* The limited distance modem accepts digital signals from a DTE, such as a terminal or computer, converts these to analog signals, and then transmits these over a short length of medium, such as twisted pair.
* On the other end of the line is another limited distance modem, which accepts the incoming analog signals, converts them to digital, and passes them on to another terminal or computer.
* The exchange of data is two way.
* The following interchange circuits are actually required:
  + Signal Ground (102)
  + Transmitted Data (103)
  + Received Data (104)
  + Request to Send (105)
  + Clear to Send (106)
  + DCE Ready (107)
  + Received line Signal Detector (109)
* The circuits just listed are sufficient for private line point-to-point modems, but additional circuits are required to use a modem to transmit data over the telephone network.
* In this case, the initiator of a connection must call the destination device over the network. Two additional leads are required:
  + DTE Ready (108.2)
  + Ring Indicator (125)
* With the addition of these two lines, the DTE-modem system can effectively use the telephone network in a way analogous to voice telephone usage.
* Following figures depicts the steps involved in dial-up half-duplex operation.







**ISDN Physical Interface:**

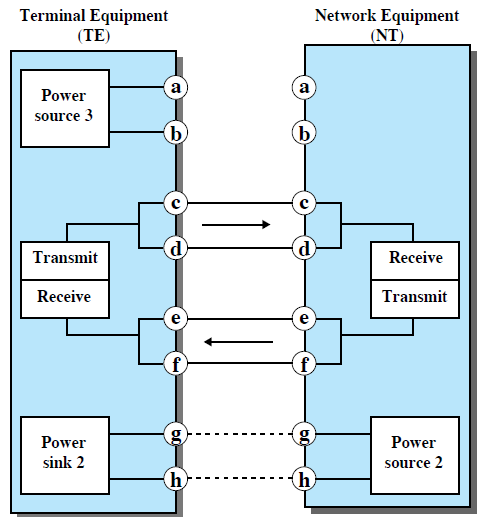
* The wide variety of functions available with V.24/EIA-232 is provided by the use of a large number of interchange circuits. This is a rather expensive way to achieve results.
* An alternative would be to provide fewer circuits but to add more logic at the DTE and DCE interfaces.
* This approach reduces costs of logic circuitry.
* This approach was taken in the X.21 standard for interfacing to public circuit-switched networks, specifying a 15-pin connector.
* More recently, the trend has been carried further with the specification of an 8-pin physical connector to an Integrated Services Digital Network (ISDN).
* ISDN is an all-digital replacement for existing public telephone and analog telecommunications networks.

**Physical Connections:**

* In ISDN terminology, a physical connection is made between terminal equipment (TE) and network-terminating equipment (NT).
* These terms correspond, rather closely, to DTE and DCE, respectively.
* The physical connection, defined in ISO 8877, specifies that the NT and TE cables shall terminate in matching connectors that provide for eight contacts.
* In a typical application, it may be desirable to provide for power transfer from the network side toward the terminal to, for example, maintain a basic telephony service in the event of failure of the locally provided power.
* This power transfer can be accomplished using the same leads used for digital signal transmission (c, d, e, f), or on additional wires, using access leads g-h.
* The remaining two leads are not used in the ISDN configuration but may be useful in other configurations.

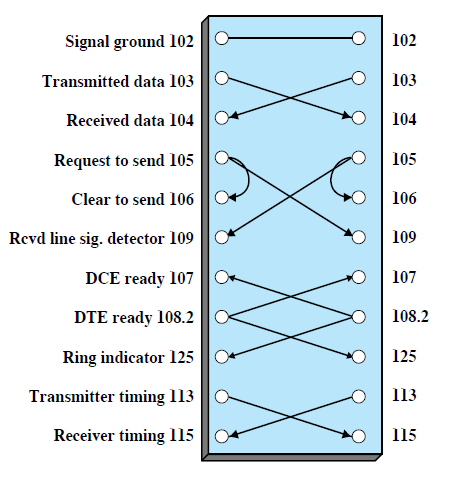
**Electrical connections:**

* The ISDN electrical specification dictates the use of balanced transmission. With *balanced transmission*, signals are carried on a line, such as twisted pair, consisting of two conductors.
* Signals are transmitted as a current that travels down one conductor and returns on the other, the two conductors forming a complete circuit.
* For digital signals, this technique is known as *differential signaling*,1 as the binary value depends on the direction of the voltage difference between the two conductors.
* *Unbalanced transmission*, which is used on older interfaces such as EIA-232, uses a single conductor to carry the signal, with ground providing the return path.
* Figure below shows contact assignments for each of eight lines on both NT and TE sides.
* Two pins are used to provide data transmission in each direction. These contact pins are used to connect twisted pair leads coming from NT and TE devices.
* Because there are no specific functional circuits, transmit/receive circuits are used to carry both data and control signals.
* Control information is framed in the form of messages.
* Specifications provides for capability to transfer power across interface.
* Direction of power depends on application.
* The balanced mode tolerates more, and produces less, noise than unbalanced mode.
* The data encoding format used on the ISDN interface depends on the data rate. For the *basic rate* of 192 kbps, the standard specifies the use of pseudo ternary coding.
* Binary one is represented by the absence of voltage, and binary zero is represented by a positive or negative pulse of 750 mV ±10%.
* For the *primary rate*, there are two options:
* 1.544 Mbps using alternate mark inversion (AMI) with B8ZS (Figure 5.6)
* 2.048 Mbps using AMI with HDB3.



**Null Modems:**

* If the distances between devices are so close as to allow two DTEs to signal each other directly. In this case, the V.24/EIA-232 interchange circuits can still be used, but no DCE equipment is provided.
* For this scheme to work, a null modem is needed, which interconnects leads in such a way as to fool both DTEs into thinking that they are connected to modems.



**Fig: Example of a Null Modem**

**Types of Errors:**

* In digital transmission system,an error occurs when a bit is altered between transmission and reception; i.e. binary 1 is transmitted and binary 0 is received or vice versa.
* Two types of errors can occur i.e. single bit error and burst error.
* Single bit error does not affect nearby bits;it is isolated one.
* A burst error of length B is contiguous sequence of B bits in which the first and last bits and any number of intermediate bits are received in error.

**Error Detection:**

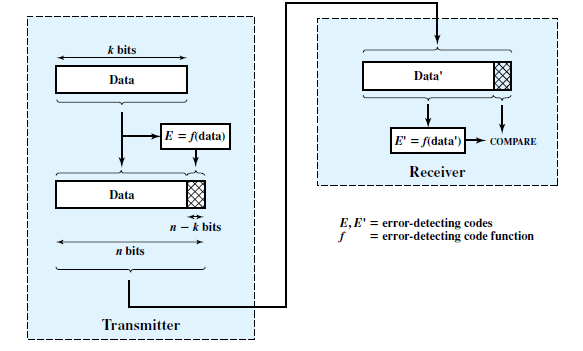
* Regardless of design of transmission system,there will be errors,resulting in change of one or more bits in a transmitted frame.We assume that data are transmitted as one or more contiguous sequence of bits,called frames.
* We define these probabilities w.r.t. errors in transmitted frames.
* Pb:Probability that bit is received with error(BER).
* P1: Probability that frame arrives with no bit errors.
* P2: Probability that, with error-detection also, a frame arrives with undetected errors.
* P3: Probability, with an error undetected also in use, frame arrives with one or more detected bit-errors.
* First consider the case in which no means are taken to detect errors. Then the probability of detected errors(P3) is zero.
* To express the remaining probabilities, assume the probability that any bit is in error is constant and independent for
* each bit. Then we have

P1 = (1 – Pb)F

P2=1-P1

Where *F* is the number of bits per frame.

* The probability that frame arrives with no bit errors decreases when the probability of a single bit error increases.
* The probability that a frame arrives with no bit errors decreases with increasing frame length; the longer the frame, the more bits it has and the higher the probability that one of these is in error.
* This is the kind of result that motivates the use of error-detecting techniques.
* All of these techniques operate on the following principle.



**Fig: Error detection process**

* For a given frame of bits, additional bits that constitute an **error-detecting code** are added by the transmitter.
* This code is calculated as a function of the other transmitted bits.
* For a data block of *k* bits, the error-detecting algorithm yields an error-detecting code of n - k bits, where (n-k)<k.
* The error-detecting code, also referred to as the **check bits**, is appended to the data block to produce a frame of *n* bits, which is then transmitted.
* The receiver separates the incoming frame into the *k* bits of data and (n-k) bits of the error-detecting code.
* The receiver performs the same error-detecting, calculation on the data bits and compares this value with the value of the incoming error-detecting code.
* A detected error occurs if and only if there is a mismatch.

**Parity Check**

* The simplest error-detecting scheme is to append a parity bit to the end of a block of data.
* The value of parity bit is selected so that the character has aneven number of 1s (even parity) or an odd number of 1s (odd parity).
* Note, however, that if two (or any even number) of bits are inverted due to error, an undetected error occurs.
* Typically, even parity is used for synchronous transmission and odd parity for asynchronous transmission.

**Cyclic Redundancy Check (CRC)**

* One of the most common, and one of the most powerful, error-detecting codes is the cyclic redundancy check (CRC).
* Given k-bit block of bits, or message, the transmitter generates an (n-k) bit sequence, known as a frame check sequence (FCS), such that the resulting frame, consisting of *n* bits is exactly divisible by some predetermined number.
* The receiver then divides the incoming frame by that number and, if there is no remainder, assumes there was no error.
* To clarify this, various procedures are there:

i) Modulo 2 Arithmetic.

ii) Polynomials.

iii) Digital Logic.

**Modulo 2 Arithmetic**

* Modulo 2 arithmetic uses binary addition with no carries,which is just the exclusive-OR (XOR) operation.
* Binary subtraction with no carries is also interpreted as the XOR operation.
* For example,

1111 1111 11001

+1010 -0101 \* 11

0101 1010 11001

11001

101011

Now define

T = n-bit frame to be transmitted

D = k-bit block of data, or message, the first k bits of T

F = (n – k)-bit FCS, the last (n – k) bits of T

P = pattern of n - k + 1 bits; this is the predetermined divisor.

* We would like *T*/*P* to have no remainder. It should be clear that

T = 2n-k.D+ F

* That is, by multiplying *D* by 2n-k,we have in effect shifted it to the left by (n-k) bits and padded out the result with zeroes.
* Adding *F* yields the concatenation of *D* and *F*, which is *T.*
* We want *T* to be exactly divisible by *P.*
* Suppose that we divide 2n-k.D by *P:*

2n-k.D =Q+R -----------------------(1)

P P

* There is a quotient and a remainder. Because division is modulo 2, the remainder is always at least one bit shorter than the divisor.
* We will use this remainder as our FCS.

Then T=2n-k.D+R

* Does this *R* satisfy our condition that *T*/*P* has no remainder?
* To see that it does,
* Consider

T = 2n-k.D+R = 2n-k.D + R

P P P P

* Substituting in Equation (1), we have

T = Q + R + R

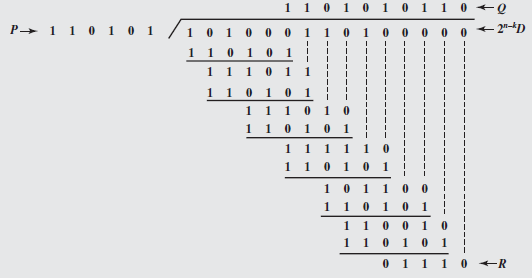
P P P

* Any binary number added to itself modulo 2 yields zero. Thus

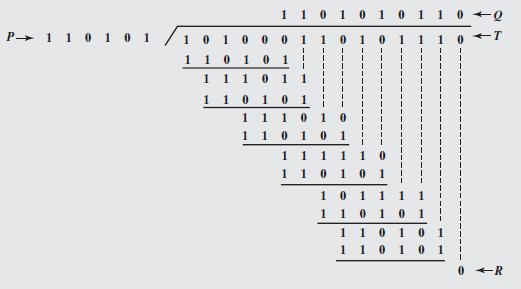
T = Q + R+R = Q

P P

* Thus, there is no remainder, and therefore *T* is exactly divisible by *P.*
* Thus, the FCS is easilygenerated:
* Simply divide 2n-k.D by *P* and use (n-k) bit remainder as the FCS.
* On reception, the receiver will divide *T* by *P* and will get no remainder if there have been no errors.
* **EXAMPLE:**
* Given
  + Message D = 1010001101 (10 bits)
  + Pattern P = 110101 (6 bits)
  + FCS R = to be calculated (5 bits)
* Thus, n = 15, k = 10, (n–k)= 5.
* The message is multiplied by 25 , yielding 101000110100000.
* This product is divided by *P*:



* The remainder is added to 25 .D to give T=101000110101110 which is transmitted.
* If there are no errors, the receiver receives *T* intact. The received frame is divided by *P.*



* Because there is no remainder, it is assumed that there have been no errors.
* The pattern *P* is chosen to be one bit longer than the desired FCS, and exact bit pattern chosen depends on the type of errors expected.
* At minimum, both the high- and low-order bits of *P* must be 1.

**Polynomials**

* A second way of viewing the CRC process is to express all values as polynomials in a dummy variable *X,* with binary coefficients.
* The coefficients correspond to the bits in the binary number.  
  Thus, for D = 110011 we have
* D(X) = X5+X4+X+1

and for P = 11001 we have P(X)= X4+X3+1

* Arithmetic operations are again modulo 2.The CRC process can now be described as

Xn-k.D(X) = Q(X) + R(X)

P(X) P(X)

T(X) = Xn-k.D(X) + R(X)

**EXAMPLE:**

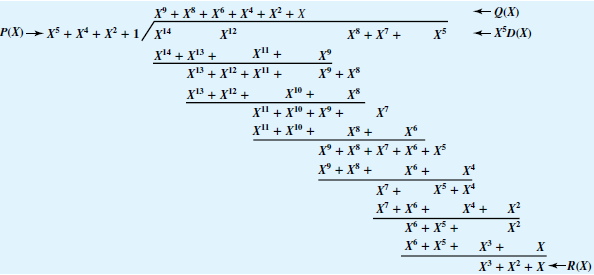
* Using the preceding example, for D = 1010001101,we have

D(X) = X9+X7+X3 +X2+1.

For P = 110101,we have P(X) = X5+X4+X2+1.

* We should end up with R = 01110 which corresponds to

R(X) = X3 +X2+X.



* An error *E*(*X*) will only be undetectable if it is divisible by *P*(*X*).
* Following errors are not divisible by a suitably chosen *P*(*X*) and hence are detectable:
  + All single-bit errors, if *P*(*X*) has more than one nonzero term.
  + All double-bit errors, as long as *P*(*X*) is a special type of polynomial, called a primitive polynomial.
* (iii)Any odd number of errors.
* (iv)Any burst error for which the length of the burst is less than or equal to n – k (FCS).
* Four versions of *P*(*X*) are widely used

CRC-12 = x12+x11+x3+x2+x+1

CRC-16 = x16+x15+x2+1

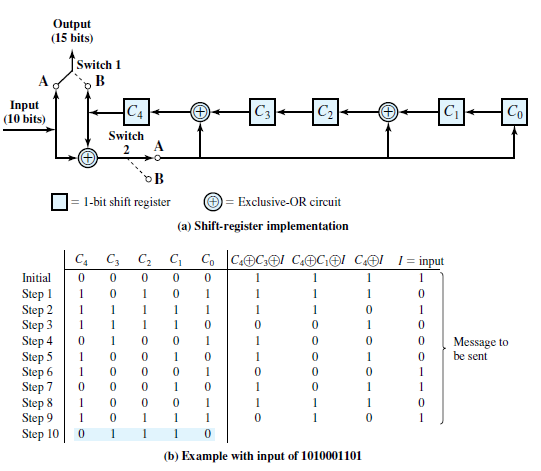
CRC-CCITT = x16+ x12+x5+1

CRC-32 = x32+x26+x23+x22+ x16+ x12+x11+x10+x8+x7+x5+x4+ x2+x+1

* The CRC-12 system is used for transmission of streams of 6-bit characters and generates a 12-bit FCS.
* Both CRC-16 and CRC-CCITT are used for 8-bit characters,
* in the United States and Europe, respectively, and both result in a 16-bit FCS.
* CRC-32 is specified as an option in some point-to-point synchronous transmission standards and is used in IEEE 802 LAN standards.
* **Digital Logic**
* The CRC process can be represented by, and indeed implemented as, a dividing circuit consisting of XOR gates and a shift register.
* The shift register is a string of 1-bit storage devices.
* Each device has an output line, which indicates the value currently stored, and an input line.
* At clock times, the value in the storage device is replaced by the value indicated by its input line.
* The entire register is clocked simultaneously, causing a 1-bit
* shift along the entire register.
* The circuit is implemented as follows:
* The register contains n-k bits, equal to the length of the FCS.
* There are up to n-k XOR gates.
* The presence or absence of XOR gate corresponds to the presence or absence of a term in the divisor polynomial, *P*(*X*), excluding the terms 1 and Xn-k.

Data D = 1010001101 D(X) = x9+x7+x3+x2+1

Divisor P = 110101 P(X) = x5+x4+ x2+1



**Fig: Circuit with shift registers for dividing by the polynomial**

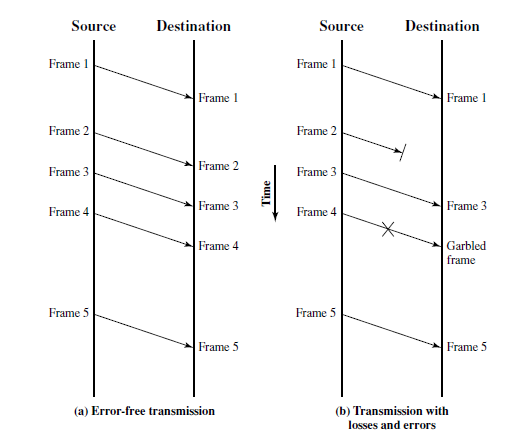
**X5 + X4 + X2 + 1**

**Data Link Controls**

* To send data over data communication link, control logic is added above physical interfacing called data link control or data link control protocol.
* When data link control protocol is used transmission medium between systems is referred to a data link.
* To see the need for data link control, we list some of the requirements and objectives for effective data communication between two directly connected transmitting-receiving stations:
* **Frame synchronization:** Data are sent in blocks called frames. The beginning and end of each frame must be recognizable. We briefly introduced this topic with the discussion of synchronous frames.
  + **Flow control:** The sending station must not send frames at a rate faster than the receiving station can absorb them.
  + **Error control:** Bit errors introduced by the transmission system should be corrected.
  + **Addressing:** On a shared link, such as a local area network (LAN), the identity of the two stations involved in a transmission must be specified.
* **Control and data on same link:** It is usually not desirable to have a physically separate communications path for control information. Accordingly, the receiver must be able to distinguish control information from the data being transmitted.
* **Link management:** The initiation, maintenance, and termination of a sustained data exchange require a fair amount of coordination and cooperation among stations. Procedures for the management of this exchange are required.
* None of these requirements is satisfied by the techniques
* a data link protocol satisfies these requirements.
* Two key mechanisms that are part of data link control:

flow control and error control.

* **Flow Control**
* Flow control is a technique for assuring that a transmitting entity does not overwhelm a receiving entity with data.
* The receiving entity typically allocates a data buffer of some maximum length for a transfer.
* When data are received, the receiver must do a certain amount of processing before passing the data to the higher-level software. In the absence of flow control, the receiver’s buffer may fill up and overflow while it is processing old data.
* The model we will see is a vertical-time sequence diagram, showing time dependencies and illustrating the correct send-receive relationship.
* Each arrow represents a single frame transiting a data link between two stations.The data are sent in a sequence of frames, with each frame containing a portion of the data and some control information.
* **Transmission time:** The time it takes for a station to emit all of the bits of a frame onto the medium is the transmission time. This is proportional to the length of the frame.
* **Propagation time:** The propagation time is the time it takes for a bit to traverse the link between source and destination.



**Fig: Model of Frame Transmission**

**Stop-and-Wait Flow Control**

* A source entity transmits a frame. After the destination entity receives the frame, it indicates its willingness to accept another frame by sending back an acknowledgment to the frame just received.
* The source must wait until it receives the acknowledgment before sending the next frame. The destination can thus stop the flow of data simply by withholding acknowledgment.
* A source will break up a large block of data into smaller blocks and transmit the data in many frames. This is done for the following reasons:
* The buffer size of the receiver may be limited.
  + The longer the transmission, the more likely that there will be an error, necessitating retransmission of the entire frame. With smaller frames, errors are detected sooner, and a smaller amount of data needs to be retransmitted.
  + On a shared medium, such as a LAN, it is usually desirable not to permit one station to occupy the medium for an extended period, thus causing long delays at the other sending stations.
* With the use of multiple frames for a single message, the stop-and-wait procedure may be inadequate. The essence of the problem is that only one frame at a time can be in transit. To explain we first define the **bit length of a link** as follows:



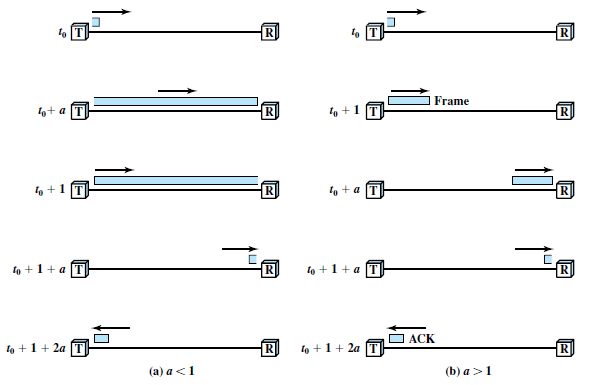
B= length of the link in bits; this is the number of bits present on the link at an instance in time when a stream of bits fully occupies the link

V = velocity of propagation, in m/s

d = length, or distance, of the link in meters

R = data rate of the link, in bps

* In situations where the bit length of the link is greater than the frame length, serious inefficiencies result.



**Fig: Stop and wait link utilization**

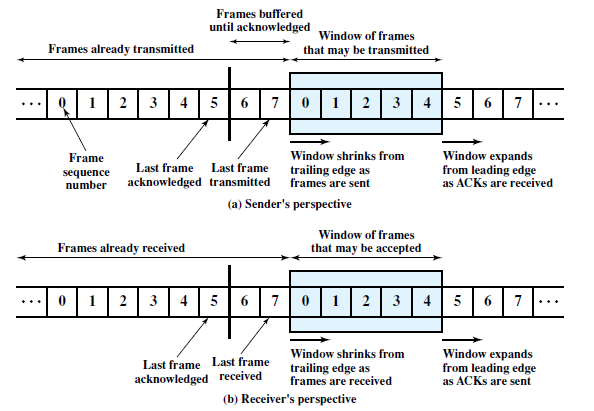
we can express *a* as

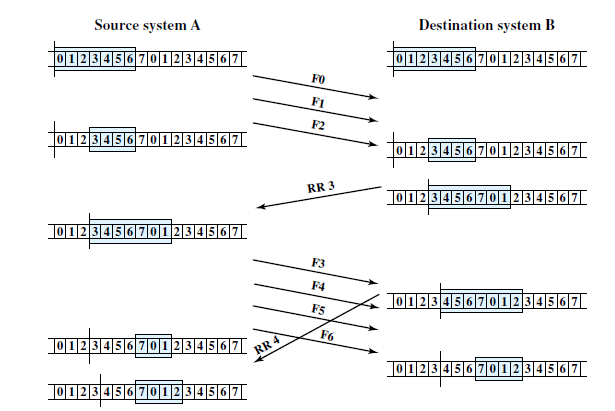


where *L* is the number of bits in the frame (length of the frame in bits).

* When *a* is less than 1, the propagation time is less than the transmission time. In this case, the frame is sufficiently long that the first bits of the frame have arrived at the destination before the source has completed the transmission of the frame.
* When *a* is greater than 1, the propagation time is greater than the transmission time. In this case, the sender completes transmission of the entire frame before the leading bits of that frame arrive at the receiver.

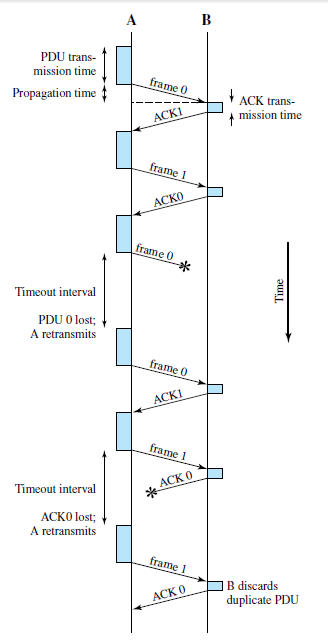
**SLIDING AND FLOW CONTROL**

* Efficiency of transmission can be greatly improved by allowing multiple frames to be in transit at the same time.
* Let us examine how this might work for two stations, A and B, connected via a full-duplex link
* Station B allocates buffer space for W frames. Thus, B can accept W frames, and A is allowed to send W frames without waiting for any acknowledgments
* To keep track of which frames have been acknowledged, each is labeled with a sequence number B acknowledges a frame by sending an acknowledgment that includes the sequence number of the next frame expected
* This acknowledgment also implicitly announces that B is prepared to receive the next W frames, beginning with the number specified
* This scheme can also be used to acknowledge multiple frames
* B could receive frames 2, 3, and 4 but withhold acknowledgement until frame 4 has arrived. By then returning an acknowledgment with sequence number 5, B acknowledges frames 2, 3, and 4 at one time.
* A maintains a list of sequence numbers that it is allowed to send, and B maintains a list of sequence numbers that it is prepared to receive.
* Each of these lists can be thought of as a window of frames.
* The operation is referred to as sliding-window flow control.
* Several additional comments need to be made. Because the sequence number to be used occupies a field in the frame, it is of bounded size
* For example, for a 3-bit field, the sequence number can range from 0 to 7.Accordingly,frames are numbered modulo 8;that is, after sequence number 7,the next number is 0
* For a k-bit field the range of sequence numbers is 0 to 2k-1 and frames are numbered modulo 2k.Maximum window size is 2k-1.
* Figure shows a way of depicting sliding-window process. It assumes 3-bit sequence no, so frames are numbered sequence from 0-7 and then same numbers are reused for subsequent frames.
  + 
* Shaded rectangles indicates frames that may be sent
* Each time a frame is sent, the shaded window shrinks; each time an acknowledgement is recorded, shaded window grows
* Frames between vertical bar and shaded window have been sent but not yet acknowledged
* Sender must buffer these frames in case they need to be retransmitted.
* Example is shown below



* Sliding window flow control is potentially much more efficient than stop-and-wait flow control.
* As with sliding window flow control, transmission link is treated as pipeline that may be filled with frames in transit.
* In stop-and-wait, only one frame may be in the pipe at a time.
* Example assumes a 3-bit sequence number field and a maximum window size of seven frames.
* Initially, A and B have windows indicating that A may transmit seven frames, beginning with frame 0 (F0).
* After transmitting three frames (F0, F1, F2) without acknowledgment, A has shrunk its window to four frames and maintains a copy of the three transmitted frames. The window indicates that A may transmit four frames, beginning with frame number 3.
* B then transmits an RR (receive ready) 3,which means “I have received all frames up through frame number 2 and am ready to receive frame number 3 and am prepared to receive seven frames, beginning with frame number 3.”
* A may discard the buffered frames as acknowledgment has been received for them.
* A Proceeds to transmit frames 3, 4, 5, and 6.
* B returns RR 4, which acknowledges F3, and allows transmission of F4.
* By the time this RR reaches A, it has already transmitted F4, F5, and F6, and therefore A may only open its window to permit sending four frames beginning with F7.
* The mechanism so far described provides a form of flow control.
* Most protocols also allow a station to cut off the flow of frames from the other side by sending a Receive Not Ready (RNR) message.
* RNR 5 means “I have received all frames up through number 4 but am unable to accept any more.”
* We have discussed transmission in one direction only. If two stations exchange data, each needs to maintain two windows, one for transmit and one for receive, and each side needs to send the data and acknowledgments to the other.
* To support this requirement, a feature known as piggybacking is provided.
* Each data frame includes a field that holds the sequence number of that frame plus a field that holds the sequence number used for acknowledgment.
* If a station has data to send and an acknowledgment to send, it sends both together in one frame, saving communication capacity.
* If a station has an acknowledgment but no data to send,it sends a separate acknowledgment frame.
* If a station has data to send but no new acknowledgment to send, it must repeat the last acknowledgment sequence number that it sent.
* Receiving stations ignores repeated acknowledgment of frames.
* ERROR CONTROL
* Error control refers to mechanisms to detect and correct errors that occur in the transmission of frames.
* data are sent as a sequence of frames;
* frames arrive in the same order in which they are sent and potentially variable amount of delay before reception.
* There is the possibility of two types of errors:
* **Lost frame:** A frame fails to arrive at the other side.For example,a noise burst may damage a frame to the extent that the receiver is not aware that a frame has been transmitted.
* **Damaged frame:** Some of the frame bits are in error (have been altered during transmission).
* Error detection techniques are based on the following aspects:
* **Error detection:**
* **Positive acknowledgment:** The destination returns a positive acknowledgment for error-free frames.
* **Retransmission after timeout:** The source retransmits a frame that has not been acknowledged after timeout.
* **Negative acknowledgment and retransmission:** The destination returns a negative acknowledgment to frames and the source retransmits such frames.
* Collectively, these mechanisms are all referred to as automatic repeat request (ARQ).
* Three versions of ARQ have been standardized.

**1)Stop-and-wait ARQ:**

* This is based on the stop-and-wait flow control technique.
* The source station transmits a single frame and then must await an acknowledgment (ACK).
* No other data frames are sent until the destination station’s reply arrives in the form of an acknowledgment at the source station.
* Two sorts of errors could occur.
* I)The frame that arrives at the destination is damaged.
* The receiver detects this by using the error-detection technique referred to earlier and discards the frame.
* To overcome this, the source station is equipped with a timer.
* After a frame is transmitted, the source station waits for an acknowledgment. If no acknowledgment is received by the time that the timer expires, then the same frame is sent again.
* For this, source station should maintain a copy of a transmitted frame until an acknowledgment is received.
* II)Acknowledgment is damaged.
* Station A sends a frame.
* The frame is received correctly by station B.
* Station B responds with an acknowledgment (ACK).
* The ACK is damaged in transit and is not recognizable by A.
* Now ,after time out at A,A will resend the same frame.
* This will make duplicate copies of frames at the receiver side.
* B has accepted two copies of the same frame as if they were separate.
* To avoid this problem,
* Frames are alternately labeled with 0 or 1, and positive acknowledgments are of the form ACK0 and ACK1.
* According to sliding window flow control ACK0-means received frame 1 and ready to receive frame 0.
* Figure shows example of Stop and wait ARQ
* 
  + - **Fig: Stop and wait ARQ**
* Principle advantage of Stop and wait ARQ is its simplicity and disadvantage is that its inefficient mechanism.
* Sliding window flow control technique can be adapted to provide efficient line use.

**2)Go Back N ARQ:**

* Error control based on sliding-window flow control that is most commonly used is called go-back-N ARQ.
* In this method, a station may send a series of frames sequentially numbered modulo some maximum value.
* The number of unacknowledged frames outstanding is determined by window size, using the sliding window flow control technique.
* While no errors occur, the destination will acknowledge incoming frames as usual.
* If the destination station detects an error in a frame, it may send a negative acknowledgment for that frame (REJ=reject) for the frame.
* The destination station will discard that frame and all future incoming frames until the frame in error is correctly received.
* Thus, the source station, when it receives a REJ, must retransmit the frame in error plus all succeeding frames that were transmitted in the interim.
* Example
  + Suppose that station A is sending frames to station B.
  + After each transmission, A sets an acknowledgment timer for the frame just transmitted.
  + Suppose that B has previously successfully received frame and A has just transmitted frame i.
* The go-back-N technique takes into account the following contingencies:

**Damaged frame:**

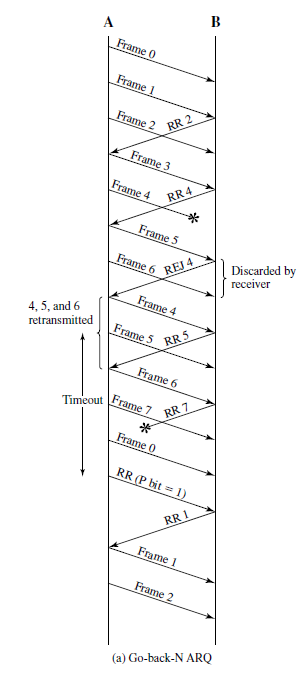
* If the received frame is invalid (i.e., B detects an error, or the frame is so damaged that B does not even perceive that it has received a frame)
* B discards the frame and takes no further action as the result of that frame. There are two sub cases:
  + Within a reasonable period of time, A subsequently sends frame(i+1)
  + B receives frame (i+1) out of order and sends a REJ i.
  + A must retransmit frame i and all subsequent frames.
  + A does not soon send additional frames. B receives nothing and returns neither an RR nor a REJ.
  + When A’s timer expires, it transmits an RR frame that includes a bit known as the P bit, which is set to 1.
  + B interprets the RR frame with a P bit of 1 as a command that must be acknowledged by sending an RR indicating the next frame that it expects, which is frame i.
    - When A receives the RR, it retransmits frame i. Alternatively, A could just retransmit frame i when its timer expires.

**Damaged RR:**

* There are two sub cases:
  + B receives frame i and sends RR (i+1) which suffers an error in transit.
  + Because acknowledgments are cumulative (e.g., RR 6 means that all frames through 5 are acknowledged),it may be that A will receive a subsequent RR to a subsequent frame and that it will arrive before the timer associated with frame i expires.
  + If A’s timer expires, it transmits an RR command as in Case 1b.
  + It sets another timer, called the P-bit timer.
  + If B fails to respond to the RR command, or if its response suffers an error in transit, then A’s P-bit timer will expire.
  + At this point, A will try again by issuing a new RR command and restarting the P-bit timer.
  + This procedure is tried for a number of iterations. If A fails to obtain an acknowledgment after some maximum number of attempts, it initiates a reset procedure.

**Damaged REJ:**

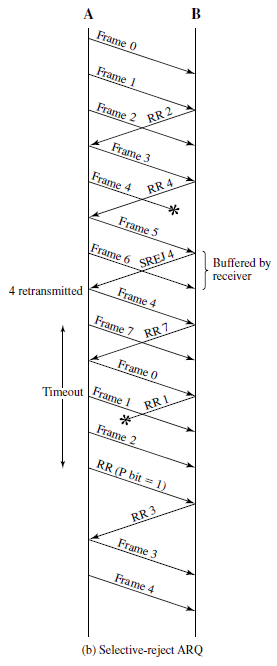
* If a REJ is lost, this is equivalent to Case 1b



* When frame 5 is received out of order, B sends a SREJ 4, indicating that frame 4 has not been received
* At this point B continues to accept incoming frame and buffers them until frame 4 is received
* B can place all frames in order for delivery to higher-layer software
* Selective regret is more efficient than go back N because it minimizes amount of retransmission
* Receiver must maintain buffer large enough
* Mostly useful in satellite link because of long propagation delay.

**3)Selective-Reject ARQ**

* With selective-reject ARQ, the only frames retransmitted are those that receive a negative acknowledgment.
* In this case called SREJ, or those that time out.
* Figure illustrates this scheme



**HIGH-LEVEL DATA LINK CONTROL (HDLC)**

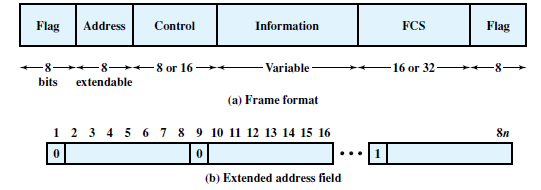
* The most important data link control protocol is HDLC (ISO 3009, ISO 4335). Not only is HDLC widely used, but it is the basis for many other important data link control protocols, which use the same or similar formats and the same mechanisms as employed in HDLC.

***Basic Characteristics***

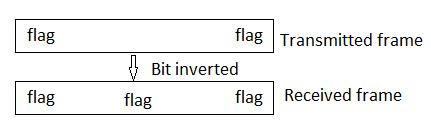
* HDLC defines three types of stations, two link configurations, and three data transfer modes of operation. The three station types are
  + - * Primary station: Responsible for controlling the operation of the link. Frames issued by the primary are called commands.
      * Secondary station: Operates under the control of the primary station. Frames issued by a secondary are called responses. The primary maintains a separate logical link with each secondary station on the line.
      * Combined station: Combines the features of primary and secondary. A combined station may issue both commands and responses.
* The two link configurations are
  + - * Unbalanced configuration: Consists of one primary and one or more secondary stations and supports both full-duplex and half-duplex transmission.
      * Balanced configuration: Consists of two combined stations and supports both full-duplex and half-duplex transmission.
* The three data transfer modes are
  + - * Normal response mode (NRM): Used with an unbalanced configuration. The primary may initiate data transfer to a secondary, but a secondary may only transmit data in response to a command from the primary.
      * Asynchronous balanced mode (ABM): Used with a balanced configuration. Either combined station may initiate transmission without receiving permission from the other combined station.
      * Asynchronous response mode (ARM): Used with an unbalanced configuration. The secondary may initiate transmission without explicit permission of the primary.
* The primary still retains responsibility for the line, including initialization, error recovery, and logical disconnection.
* NRM is used on multidrop lines, in which a number of terminals are connected to a host computer.
* The computer polls each terminal for input.
* NRM is also sometimes used on point-to-point links.
* 2) ABM is the most widely used of the three modes; it makes more efficient use of a full-duplex point-to-point link because there is no polling overhead.
* 3) ARM is rarely used; it is applicable to some special situations in which a secondary may need to initiate transmission.

**Frame Structure**

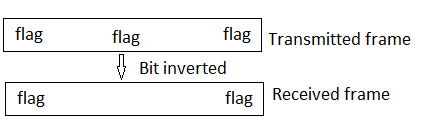
* HDLC uses synchronous transmission. All transmissions are in the form of frames, and a single frame format suffices for all types of data and control exchanges.
* Figure shows the structure of the HDLC frame. The flag, address, and control fields that precede the information field are known as a header.
* The FCS and flag fields following the data field are referred to as a trailer.



* Flag Fields: Flag fields delimit the frame at both ends with the unique pattern 01111110.
* A single flag may be used as the closing flag for one frame and the opening flag for the next.
* On both sides of the user-network interface, receivers are continuously hunting for the flag sequence to synchronize on the start of a frame.
* While receiving a frame, a station continues to hunt for that sequence to determine the end of the frame.
* There is no assurance that the pattern 01111110 will not appear somewhere inside the frame, thus destroying synchronization.
* To avoid this problem, a procedure known as bit stuffing is used.
* 1 bit error may merge or split two frames.



**Fig: An inverted bit splits a frame in two**

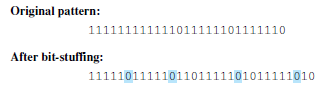
****

**Fig: An inverted bit merges two frames**

**Address Field:** The address field identifies the secondary station that transmitted or is to receive the frame. This field is not needed for point-to-point links but is always included for the sake of uniformity.

* The address field is usually 8 bits long but, by prior agreement, an extended format may be used in which the actual address length is a multiple of 7 bits.
* The leftmost bit of each octet is 1 or 0 according as it is or is not the last octet of the address field.
* The remaining 7 bits of each octet form part of the address.
* Between the starting and ending flags, the transmitter inserts an extra 0 bit after each occurrence of five 1s in the frame.
* After detecting a starting flag, the receiver monitors the bit stream. When a pattern of five 1s appears, the sixth bit is examined.
* If this bit is 0, it is deleted. If the sixth bit is a 1 and the seventh bit is a 0, the combination is accepted as a flag.
* If the sixth and seventh bits are both 1, the sender is indicating an abort condition.
* With the use of bit stuffing, arbitrary bit patterns can be inserted into the data field of the frame. This property is known as **data transparency.**

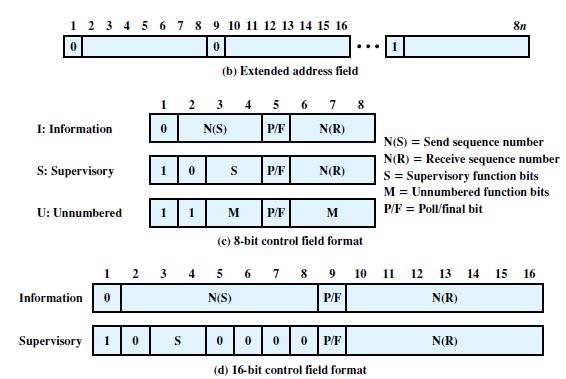
Example:



**Fig: Bit Stuffing**

**Control Field:** HDLC defines three types of frames, each with a different control field format.

Information frames(I-frames) carry the data to be transmitted for the user (the logic above HDLC that is using HDLC). Additionally, flow and error control data, using the ARQ mechanism, are piggybacked on an information frame.

****

* Supervisory frames (S-frames) provide the ARQ mechanism when piggybacking is not used.
* Unnumbered frames (U-frames) provide supplemental link control functions.
* The first one or two bits of the control field serves to identify the frame type.
* All of the control field formats contain the poll/final (P/F) bit.
* Its use depends on context. Typically, in command frames, it is referred to as the P bit and is set to 1 to solicit (poll) a response frame from the peer HDLC entity.
* In response frames, it is referred to as the F bit and is set to 1 to indicate the response frame transmitted as a result of a soliciting command.

**Information Field:** The information field is present only in I-frames and some U frames.

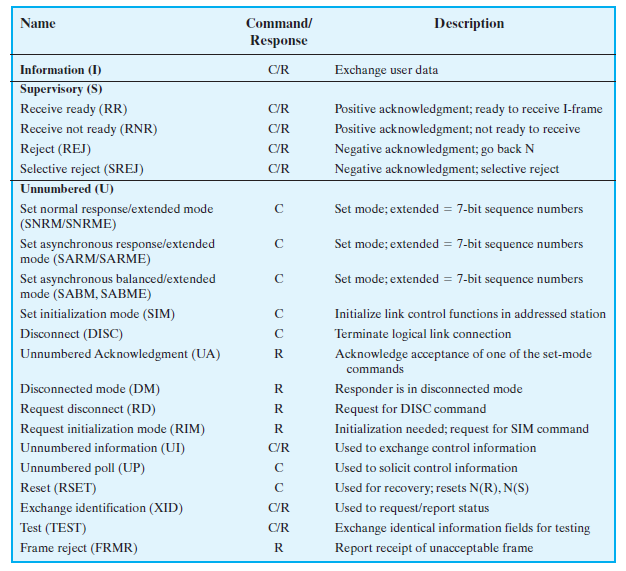
* The field can contain any sequence of bits but must consist of an integral number of octets.
* The length of the information field is variable up to some system defined maximum.

**Frame Check Sequence Field:** The frame check sequence (FCS) is an error detecting code calculated from the remaining bits of the frame, exclusive of flags.

* The normal code is the 16-bit CRC-CCITT defined in Section 6.3.
* An optional 32-bit FCS, using CRC-32, may be employed if the frame length or the line reliability dictates this choice.

**Operation**

* HDLC operation consists of the exchange of I-frames, S-frames, and U-frames between two stations.
* The various commands and responses defined for these frame types are as follows:

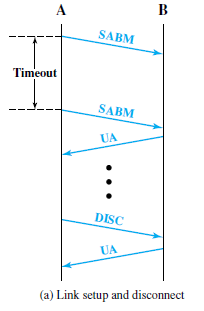
****

* The operation of HDLC involves three phases.
* **Initialization:** one side or another initializes the data link so that frames may be exchanged in an orderly fashion.
* Exchange of data and the control information
* Termination of the operation.
* **Initialization:** Either side may request initialization by issuing one of the six set mode commands. This command serves three purposes:
* It signals the other side that initialization is requested.
* It specifies which of the three modes (NRM, ABM, ARM) is requested.
* It specifies whether 3- or 7-bit sequence numbers are to be used.
  + If the other side accepts this request, then the HDLC module on that end transmits an unnumbered acknowledged (UA) frame back to the initiating side. If the request is rejected, then a disconnected mode (DM) frame is sent.

**Data Transfer:** When the initialization has been requested and accepted, then a logical connection is established.

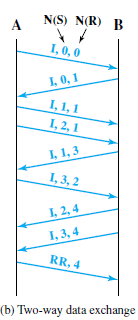
* Both sides may begin to send user data in I frames, starting with sequence number 0.
* The N(S) and N(R) fields of the I-frame are sequence numbers that support flow control and error control.
* An HDLC module sending a sequence of I-frames will number them sequentially, modulo 8 or 128, depending on whether 3- or 7-bit sequence numbers are used, and place the sequence number in N(S).
* N(R) is the acknowledgment for I-frames received; it enables the HDLC module to indicate which number I-frame it expects to receive next.
* S-frames are also used for flow control and error control.
* The receive ready (RR) frame acknowledges the last I-frame received by indicating the next I-frame expected.
* Receive not ready (RNR) acknowledges an I-frame, as with RR, but also asks the peer entity to suspend transmission of I-frames.
* When the entity that issued RNR is again ready, it sends an RR.
* REJ initiates the go-back-NARQ.
* It indicates that the last I-frame received has been rejected and that retransmission of all I-frames beginning with number N(R) is required.
  + Disconnect: Either HDLC module can initiate a disconnect, either on its own initiative if there is some sort of fault, or at the request of its higher-layer user.
  + HDLC issues a disconnect by sending a disconnect (DISC) frame.
  + The remote entity must accept the disconnect by replying with a UA and informing its layer 3 user that the connection has been terminated.
  + Examples of operation:

**1.** The HDLC protocol entity for one side issues an **SABM** command to the other side and starts a timer.



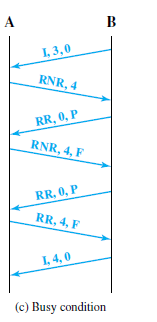
* + On other side, upon receiving the SABM, returns a UA response and sets local variables and counters to their initial values.
  + The initiating entity receives the UA response, sets its variables and counters, and stops the timer.
  + The logical connection is now active.
  + The same figure shows the disconnect procedure.
  + One side issues a DISC command, and the other responds with a UA response.

1. Figure illustrates the **full-duplex exchange** of I-frames.



* + When an entity sends a number of I-frames in a row with no incoming data, then receives sequence number is simply repeated (e.g., I,1,1; I,2.1 in the A-to-B direction).
* When an entity receives a number of I-frames in a row with no outgoing frames, then they receive sequence number in the next outgoing frame must reflect the cumulative activity (e.g., I,1,3 in the B-to-A direction).
* Note that, in addition to I-frames, data exchange may involve supervisory frames.

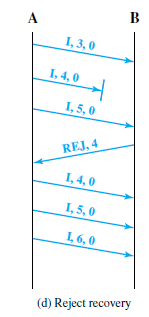
**3.Busy condition** may arise because an HDLC entity is not able to process I-frames as fast as they are arriving, or the intended user is not able to accept data as fast as they arrive in I-frames.



* In either case, the entities receive buffer fills up and it must halt the incoming flow of I-frames, using an RNR command.
* In this example, A issues an RNR, which requires B to halt transmission of I-frames.
* The station receiving the RNR will usually poll the busy station at some periodic interval by sending an RR with the P bit set.

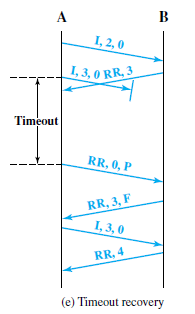
This requires the other side to respond with either an RR or an RNR. When the busy condition has cleared, A returns an RR, and I-frame transmission from B can resume.

**4.Reject Recovery:**



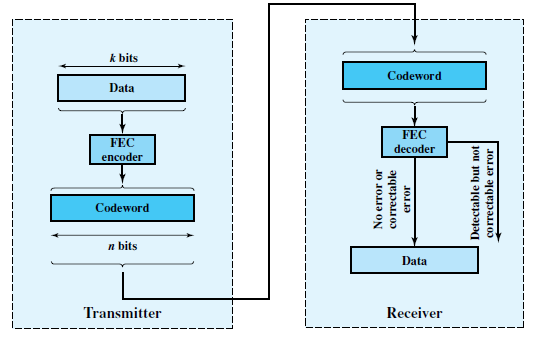
* A transmits I-frames numbered 3, 4, and 5. Number 4 suffers an error and is lost.
* When B receives I-frame number 5, it discards this frame because it is out of order and sends an REJ with an N(R) of 4.
* This causes A to initiate retransmission of I-frames previously sent, beginning with frame 4.

**5.Timeout Recovery**:



* A transmits I-frame number 3 as the last in a sequence of I-frames.
* The frame suffers an error.
* B detects the error and discards it.
* B cannot send an REJ, because there is no way to know if this was an I-frame.
* As the timer expires, A initiate’s recovery action.
* This is usually done by polling the other side with an RR command with the P bit set.

**Error Detection:**



**Fig: Error correction process**

* On the transmission end, each *k*-bit block of data is mapped into an *n*-bit block (n>k) called a **codeword**, using an FEC (forward error correction) encoder.
* The codeword is then transmitted. During transmission, the signal is subject to impairments, which may produce bit errors in the signal.
* At the receiver, the incoming signal is demodulated to produce a bit string that is similar to the original codeword but may contain errors.
* This block is passed through an FEC decoder, with one of four possible outcomes:
* If there are no bit errors, the input to the FEC decoder is identical to the original codeword, and the decoder produces the original data block as output.
* For certain error patterns, it is possible for the decoder to detect and correct those errors. Thus, even though the incoming data block differs from the transmitted codeword, the FEC decoder is able to map this block into the original data block.
* For certain error patterns, the decoder can detect but not correct the errors. In this case, the decode simply reports an uncorrectable error.
* For certain, typically rare, error patterns, the decoder does not detect that any errors have occurred and maps the incoming *n*-bit data block into a *k*-bit block that differs from the original *k*-bit block.
* The error-correcting code follows the same general layout as for error-detecting codes.
* The FEC algorithm takes as input a *k*-bit block and adds (n-k) bits of check bits to that block to produce an *n*-bit block of data.
* For some FEC algorithms, the FEC algorithm maps the *k*-bit input into an *n*-bit codeword in such a way that the original *k* bits do not appear in the codeword.

**Block Code Principles**

* To begin, we define a term that shall be of use to us. The **Hamming distance d(v1,v2)** between two *n*-bit binary sequences v1 and v2 is the number of bits in which v1 and v2 disagree.
* For example, if v1 = 011011, **v**2 = 110001

V1: 011011

V2: 110001

101010 No. of 1’s is 3

* Then d(**v**1 , **v**2) = 3
* Now let us consider the block code technique for error correction.
* Suppose we wish to transmit blocks of data of length *k* bits. Instead of transmitting each block as *k* bits, we map each *k*-bit sequence into a unique *n*-bit codeword.