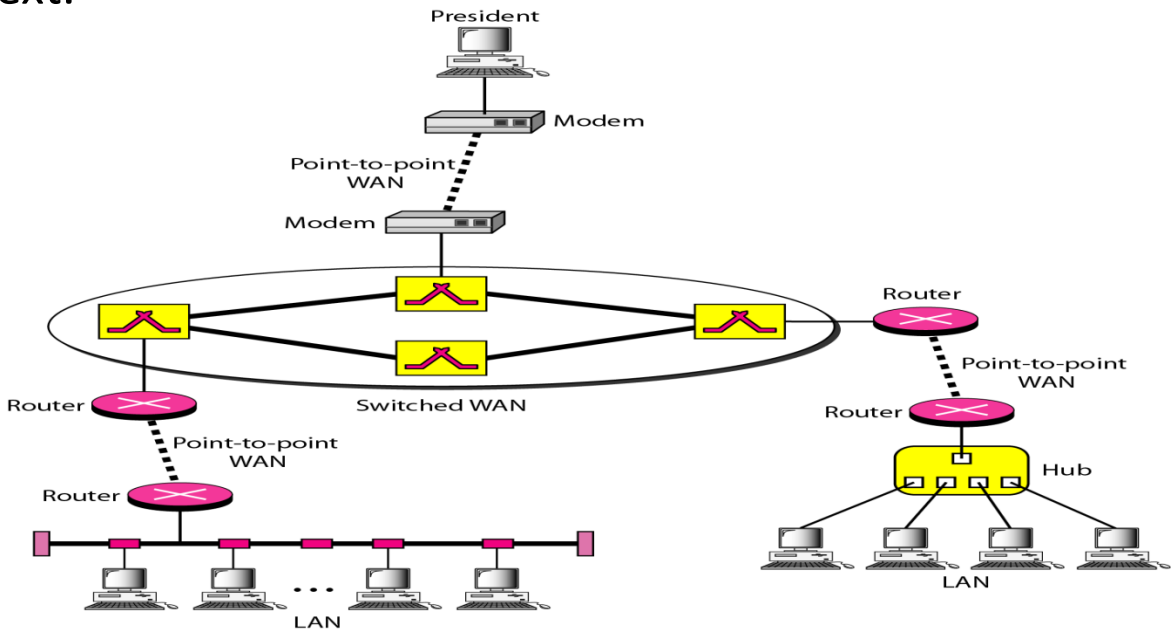
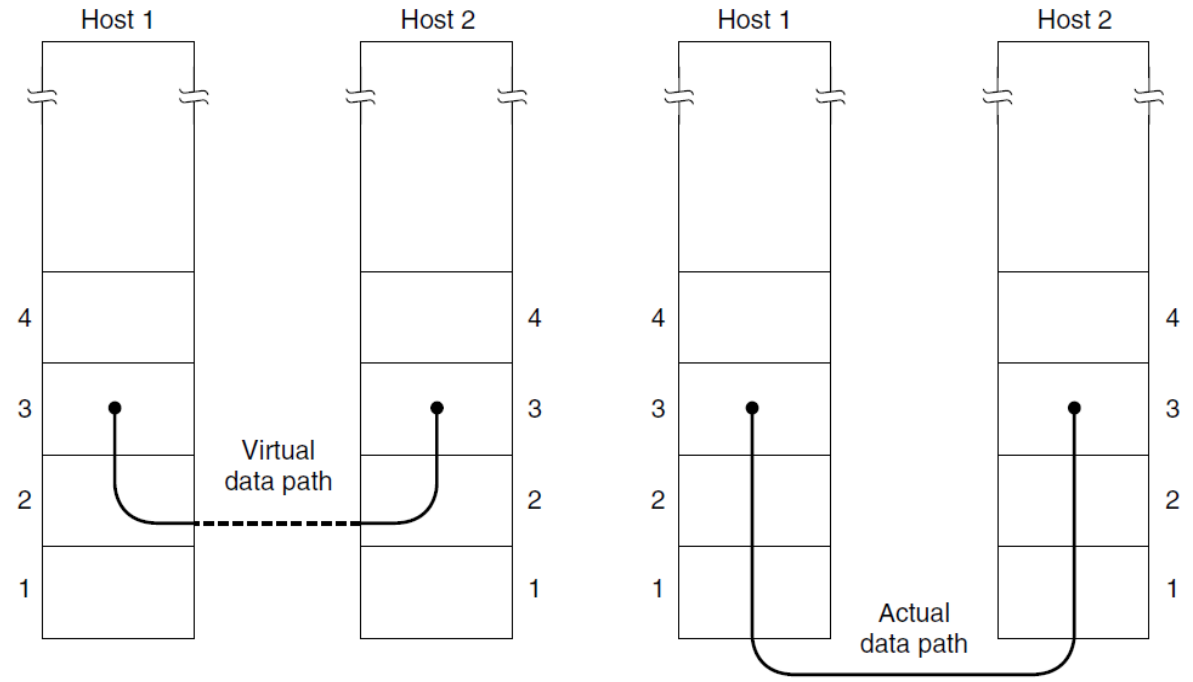
Data Link Layer

- The data link layer is responsible for moving frames from one hop(node) to the next.



- <u>Design Issues:</u>
- Framing, Physical addressing, error control, flow control, access control.

- **Services Provided to the Network Layer**
- The function of the data link layer is to provide services to the network layer.
- The principal service is transferring data from the network layer on the source machine to the network layer on the destination machine.



- Three different services are offered by data link layer to network layer.
- These services differ from protocol to protocol.

- **Unacknowledged connectionless service:** Unacknowledged connectionless service consists of having the source machine send independent frames to the destination machine without having the destination machine acknowledge them.
- No logical connection is established beforehand or released afterward.
- If a frame is lost due to noise on the line, no attempt is made to detect the loss or recover from it in the data link layer.
- This class of service is appropriate when the error rate is very low on reliable transmission media, so recovery is left to higher layers.
- It is also appropriate for real-time traffic, such as **voice**, in which late data are worse than bad data.
- **Ethernet** is a good example of a data link layer that provides this class of service.

- **Acknowledged connectionless service:** When this service is offered, there are still no logical connections used, but each frame sent is individually acknowledged.
- In this way, the sender knows whether a frame has arrived correctly or been lost.

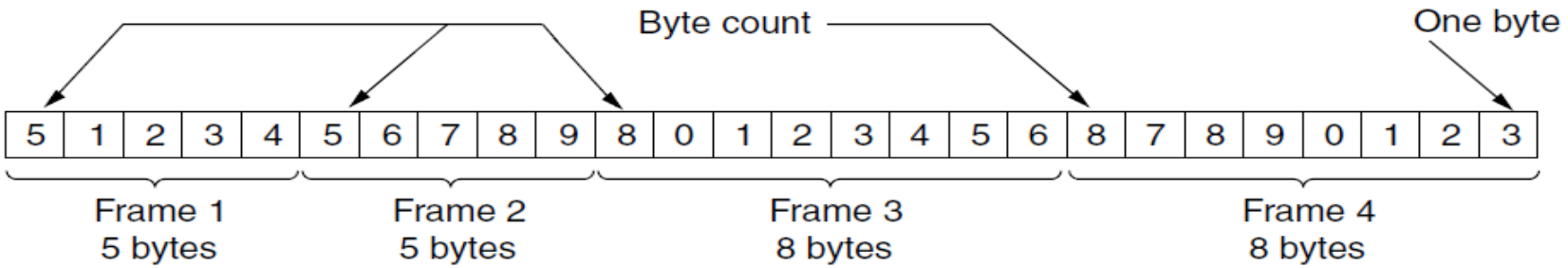
- This service is useful over unreliable channels, such as wireless systems. 802.11 (WiFi) is a good example of this class of service.

- **Acknowledged connection-oriented service:** With this service, the source and destination machines establish a connection before any data are transferred.
- Each frame sent over the connection is numbered, and the data link layer guarantees that each frame sent is indeed received.
- Furthermore, it guarantees that each frame is received exactly once and that all frames are received in the right order.
- It is appropriate over long, unreliable links such as a satellite channel or a long-distance telephone circuit.

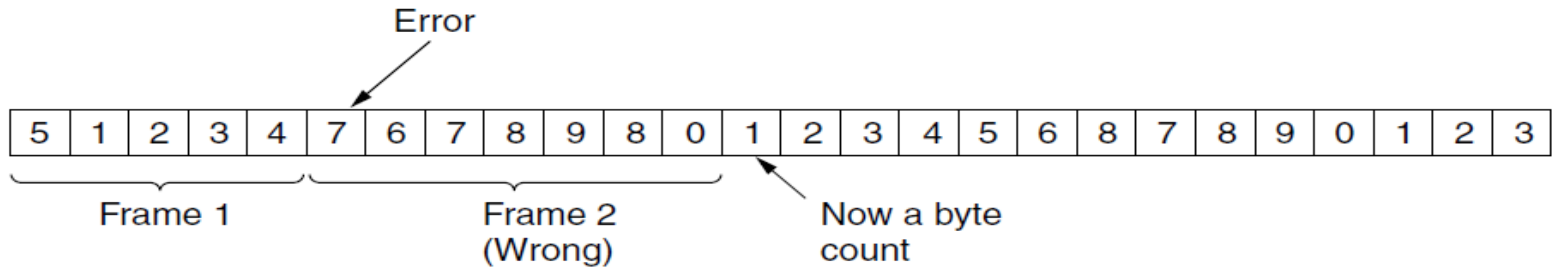
- **Framing**
- In order to provide error control and flow control mechanisms, the data at data link layer is converted into manageable data units called as frames.
- Breaking up the bit stream into frames is more difficult than it at first appears.
- A good design must make it easy for a receiver to find the start of new frames while using little of the channel bandwidth. Four methods exist.

1. Byte count.
2. Flag bytes with byte stuffing.
3. Flag bits with bit stuffing.
4. Physical layer coding violations.

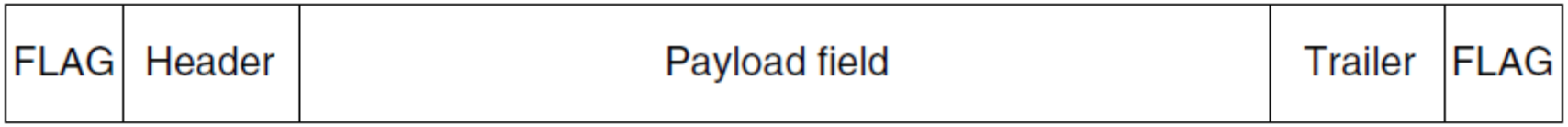
- **Byte count.**



- This method uses a field in the header to specify the number of bytes in the frame.
- When the data link layer at the destination sees the byte count, it knows how many bytes follow and hence where the end of the frame is.
- The trouble with this algorithm is that the count can be garbled by a transmission error.
- For example, if the byte count of 5 in the second frame becomes a 7 due to a single bit flip, the destination will get out of synchronization. It will then be unable to locate the correct start of the next frame.

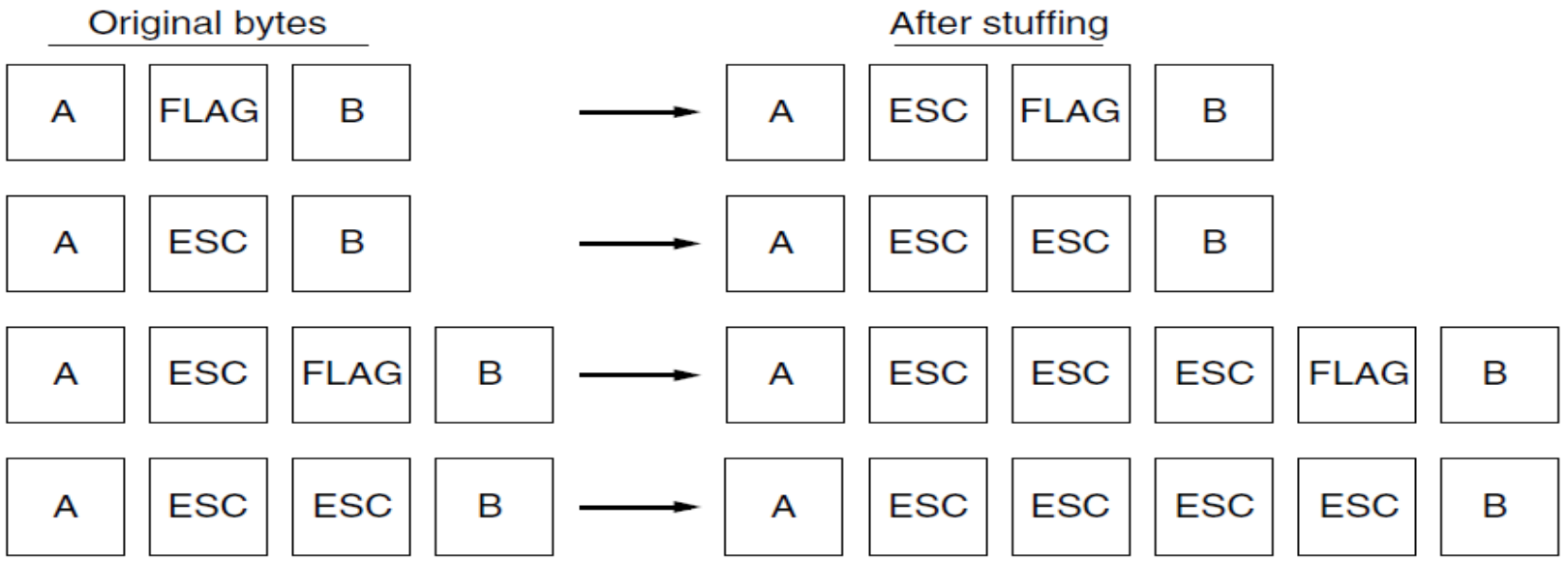


- **Flag bytes with byte stuffing.**
- The second framing method gets around the problem of resynchronization after an error by having each frame start and end with special bytes.
- Often the same byte, called a **flag byte**, is used as both the starting and ending delimiter.
- This byte is called as FLAG.



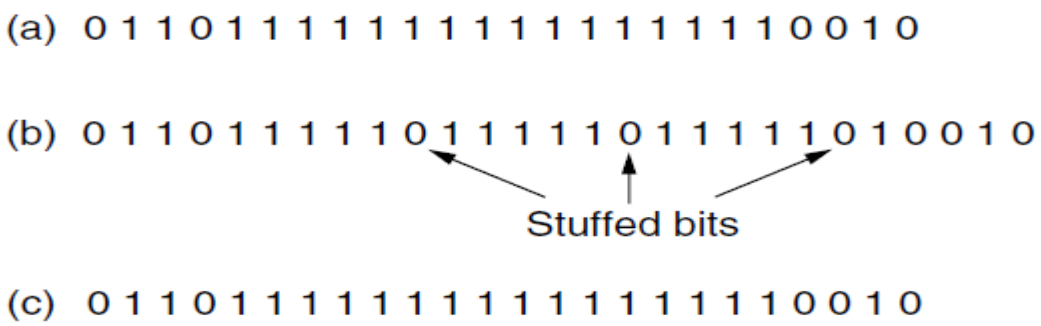
- However, there is a still a problem we have to solve. It may happen that the flag byte occurs in the data, especially when binary data such as photographs or songs are being transmitted. This situation would interfere with the framing.
- One way to solve this problem is to have the sender's data link layer insert a special escape byte (ESC) just before each ''accidental'' flag byte in the data.

- Thus, a framing flag byte can be distinguished from one in the data by the absence or presence of an escape byte before it.
- The data link layer on the receiving end removes the escape bytes before giving the data to the network layer.
- This technique is called **byte stuffing**.



- **Flag bits with bit stuffing.**
- The third method of delimiting the bit stream gets around a disadvantage of byte stuffing, which is that it is tied to the use of 8-bit bytes.

- Framing can be also be done at the bit level, so frames can contain an arbitrary number of bits made up of units of any size.
- Each frame begins and ends with a special bit pattern, 01111110 or 0x7E in hexadecimal. This pattern is a flag byte.
- Whenever the sender's data link layer encounters five consecutive 1s in the data, it automatically stuffs a 0 bit into the outgoing bit stream. This **bit stuffing** is analogous to byte stuffing.
- When the receiver sees five consecutive incoming 1 bits, followed by a 0 bit, it automatically destuffs (i.e., deletes) the 0 bit.



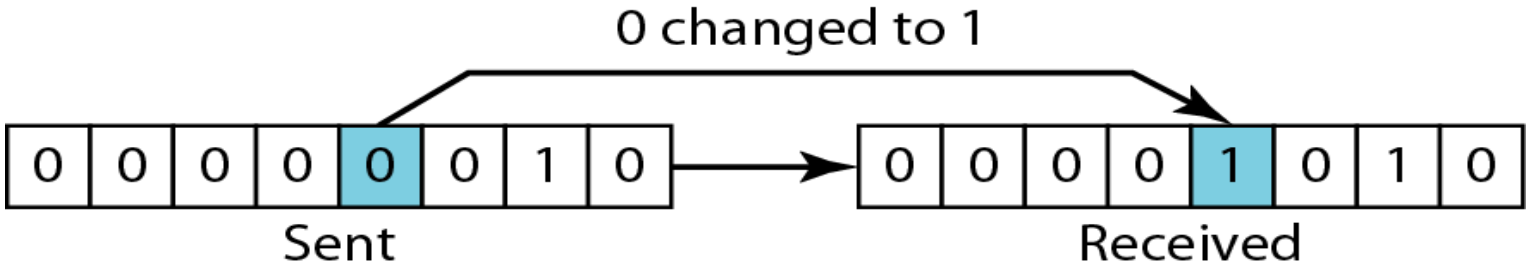
Bit stuffing. (a) The original data. (b) The data as they appear on the line. (c) The data as they are stored in the receiver's memory after destuffing.

- If the user data contain the flag pattern, 01111110, this flag is transmitted as 011111010 but stored in the receiver's memory as 01111110.

- **Physical layer coding violations.**
- The last method of framing is to use a shortcut from the physical layer.
- Encoding of bits as signals using manchester encoding/differential manchester encoding techniques often includes redundancy to help the receiver.
- This redundancy means that some signals will not occur in regular data.
- Reserved signals are used to indicate the start and end of frames.
- In effect, we are using ‘‘coding violations’’ to delimit frames.
- The beauty of this scheme is that, because they are reserved signals, it is easy to find the start and end of frames and there is no need to stuff the data.

- **Error Control**
- Data link layer ensures frames are transmitted from network layer of the source to the network layer of the destination(hop to hop) without errors.
- It also ensures frames are not lost.
- An error in data transmission results in inversion of a bit, a 1 inverted as 0 or a 0 inverted as a 1.

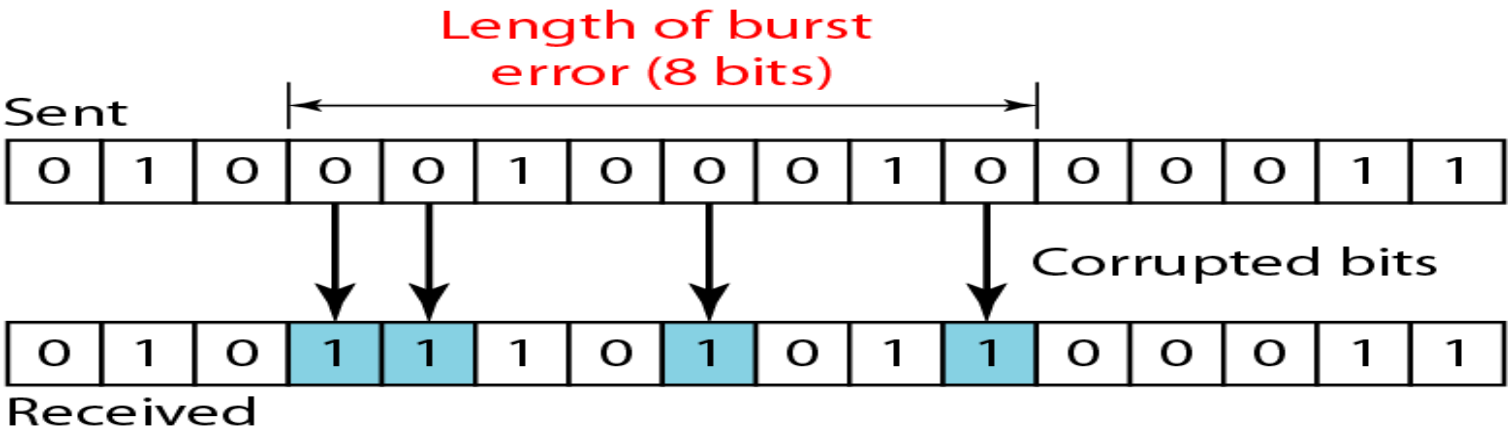
- *Types of Errors*
- **Single-Bit Error.** The term *single-bit error* means that only 1 bit of a given data unit (such as a byte, character, or packet) is changed from 1 to 0 or from 0 to 1.



- Single-bit errors are the least likely type of error in serial data transmission.

- Imagine data sent at 1 Mbps. This means that each bit lasts only 1 µs.
- For a single-bit error to occur, the noise must have a duration of only 1 µs, which is very rare; noise normally lasts much longer than this.

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



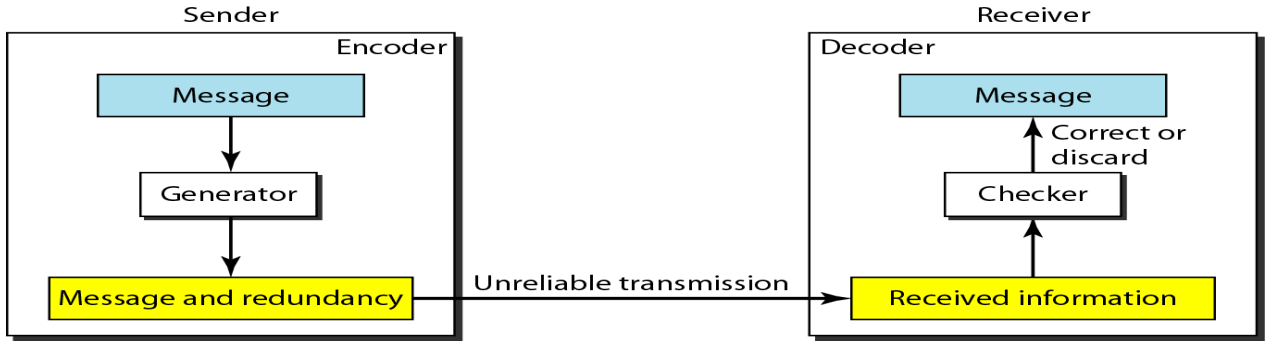
- In this case, 0100010001000011 was sent, but 0101110101100011 was received.
- Note that a burst error does not necessarily mean that the errors occur in consecutive bits. The length of the burst is measured from the first corrupted bit to the last corrupted bit.

- A burst error is more likely to occur than a single-bit error. The duration of noise is normally longer than the duration of 1 bit, which means that when noise affects data, it affects a set of bits.
- The number of bits affected depends on the data rate and duration of noise.
- The noise duration is 10 ms. If the data rate is 1 Kbps, how many bits will be effected?
- For the same noise duration, if the data rate is 1 Mbps, how many bits will be effected?
- Errors are controlled (avoided) either by error detecting codes or error correcting codes.
- To be able to detect or correct errors, we need to send some extra bits with our data. These bits are called **redundant** bits.
- These redundant bits are added by the sender and removed by the receiver.
- Their presence allows the receiver to detect or correct corrupted bits.

- **Detection Versus Correction.**
- In **error detection**, we are looking only to see if any error has occurred. The answer is a simple yes or no.
- We are not even interested in the number of errors. A single-bit error is the same for us as a burst error.
- When error detecting schemes are used, the receiver asks the sender to resend the message if it detects the occurrence of an error.
- These techniques require less number of redundant bits.
- Resending is repeated until a message arrives that the receiver believes is error-free.

- The correction of errors is more difficult than the detection. **In error correction** we need to know the exact number of bits that are corrupted and more importantly, their location in the message.
- If we need to correct one single error in an 8-bit data unit, we need to consider eight possible error locations.
- If we need to correct two errors in a data unit of the same size, we need to consider 28 possibilities.

- These techniques require more number of redundant bits.
- The receiver tries to guess the message by using redundant bits.
- **Coding.**
- Most error detecting and error correcting codes(schemes) work by considering data bits as blocks of data.
- Message bits are divided into blocks, each of *k* bits, called datawords.
- *r* redundant bits are added to each block to make the length $n = k + r$.
- The resulting *n-bit* blocks are called codewords.
- Redundancy is achieved through various coding schemes.
- The sender adds redundant bits through a process that creates a relationship between the redundant bits and the actual data bits.
- The receiver checks the relationships between the two sets of bits to detect or correct the errors.



- **Error Detecting Codes**

1) **Parity Codes**
2) **Cyclic Redundancy Checks (CRC)**

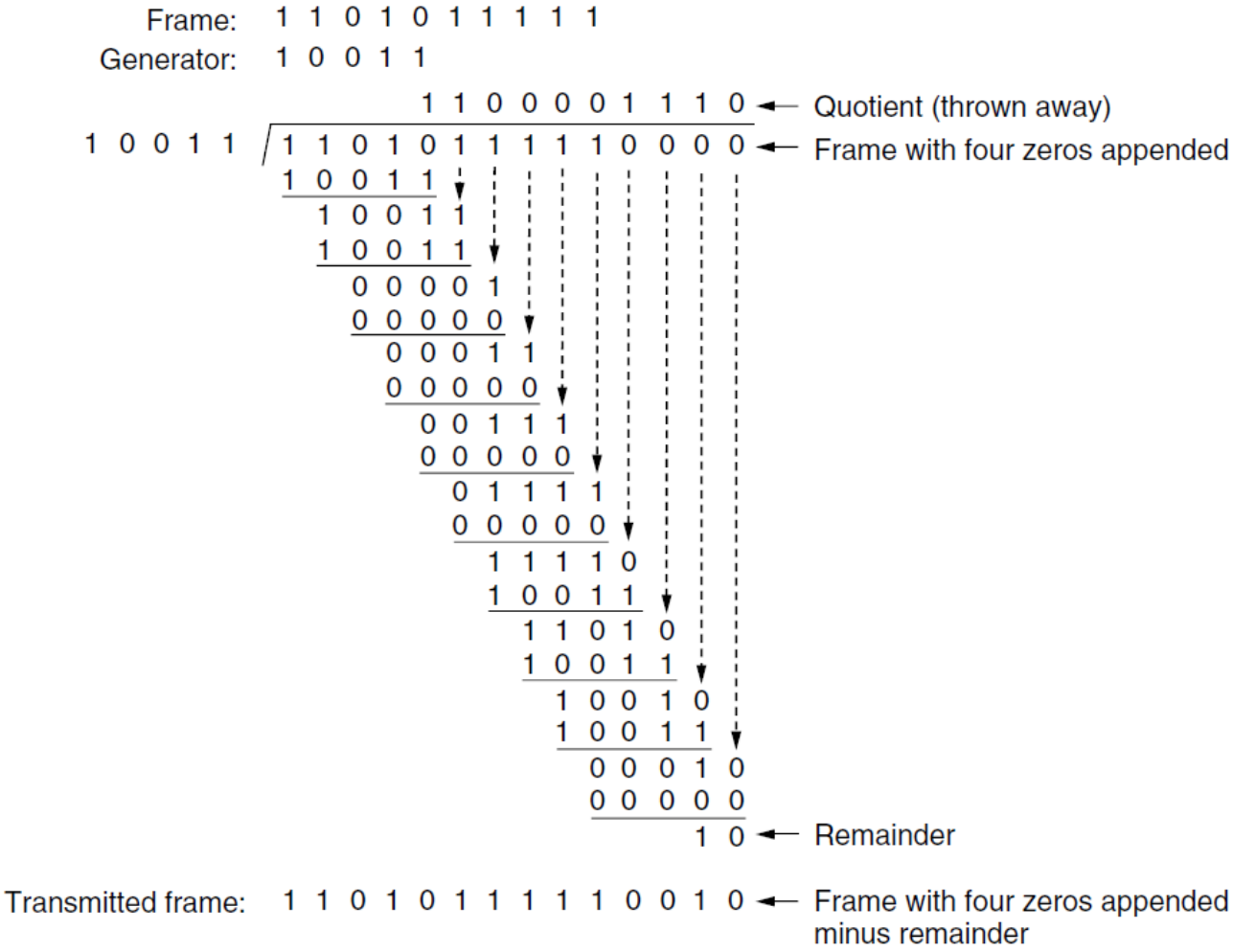
- **Parity Codes**. A single **parity bit** is appended to the data.
- The parity bit is chosen so that the number of 1 bits in the codeword is even (or odd).

- **CRC** (**Cyclic Redundancy Check**). Also known as a **polynomial code**.
- Polynomial codes are based upon treating bit strings as representations of polynomials with coefficients of 0 and 1 only.
- A *k*-bit frame is regarded as the coefficient list for a polynomial with *k* terms, ranging from x^{k-1} to x^0. Such a polynomial is said to be of degree $k - 1$.
- The high-order (leftmost) bit is the coefficient of x^{k-1}, the next bit is the coefficient of x^{k-2}, and so on.

- For example, 110001 has 6 bits and thus represents a six-term polynomial with coefficients 1, 1, 0, 0, 0, and 1: $1x^5 + 1x^4 + 0x^3 + 0x^2 + 0x^1 + 1x^0$.
- Polynomial arithmetic is done modulo 2, according to the rules of algebraic field theory. It does not have carries for addition or borrows for subtraction. Both addition and subtraction are identical to exclusive OR.
- When the polynomial code method is employed, the sender and receiver must agree upon a **generator polynomial**, *G*(*x*), in advance.
- Both the high- and low order bits of the generator must be 1.
- To compute the CRC for some frame with *m* bits corresponding to the polynomial *M*(*x*), the frame must be longer than the generator polynomial.
- The idea is to append a CRC to the end of the frame in such a way that the polynomial represented by the checksummed frame is divisible by *G*(*x*).
- When the receiver gets the checksummed frame, it tries dividing it by *G*(*x*). If there is a remainder, there has been a transmission error.

- The algorithm for computing the CRC is as follows:
- 1. Let *r* be the degree of *G*(*x*). Append *r* zero bits to the low-order end of the frame so it now contains *m* + *r* bits and corresponds to the polynomial x^r *M*(*x*).
- 2. Divide the bit string corresponding to *G*(*x*) into the bit string corresponding to x^r *M*(*x*), using modulo 2 division.
- 3. Subtract the remainder (which is always *r* or fewer bits) from the bit string corresponding to x^r *M*(*x*) using modulo 2 subtraction. The result is the checksummed frame to be transmitted. Call its polynomial *T*(*x*).

- Calculate the CRC for the frame 1101011111 using the generator $G(x) = x^4 + x + 1$.



- Data word to be sent – 100100. Key – 1101. Calculate CRC
- The frame received by the receiver is 10111101100. Using the generator polynomial $G(x) = x^3 + 1$, verify if the frame is correct or damaged.

- What kinds of errors will be detected?
- Imagine that a transmission error occurs, so that instead of the bit string for $T(x)$ arriving, $T(x) + E(x)$ arrives.
- Each 1 bit in $E(x)$ corresponds to a bit that has been inverted.
- Upon receiving the checksummed frame, the receiver divides it by $G(x)$; that is, it computes $[T(x) + E(x)]/G(x)$.
- $T(x)/G(x)$ is 0, so the result of the computation is simply $E(x)/G(x)$.
- Those errors that happen to correspond to polynomials containing $G(x)$ as a factor will slip by; all other errors will be caught.
- If there has been a single-bit error, $E(x) = x^i$, where i determines which bit is in error.
- If $G(x)$ contains two or more terms, it will never divide into $E(x)$, so all single-bit errors will be detected.

- If there have been two isolated single-bit errors, $E(x) = x^i + x^j$, where $i > j$.
- Alternatively, this can be written as $E(x) = x^j(x^{i-j} + 1)$.
- If we assume that $G(x)$ is not divisible by x, a sufficient condition for all double errors to be detected is that $G(x)$ does not divide $x^k + 1$ for any k up to the maximum value of $i - j$ (i.e.,up to the maximum frame length).
- For example, $x^{15} + x^{14} + 1$ will not divide $x^k + 1$ for any value of k below 32,768.
- Certain polynomials have become international standards.

Standard polynomials

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

- Among other desirable properties, CRC-32 has the property that it detects all bursts of length 32 or less and all bursts affecting an odd number of bits.

- **Error Correcting Codes**
- **Hamming Code.**
- Here, to each group of m information bits, r parity bits are added denoted as P_1, P_2,……. P_k located at positions 2^{k-1} from the lowest order bit to form n=m+r bit codeword.
- The number of parity bits are determined from the two equations for the lowest possible r. n=m+r, n= 2^r-1.
- For m=4, r= 3, n=7 and the code is denoted as C(7,4).
- The placement of parity bits is given by $D_7 D_6\ D_5\ P_4 D_3 P_2 P_1$.

- Each of the parity bit values are calculated from the below demonstration.

111	110	101	100	011	010	001
D_7	D_6	D_5	P_4	D_3	P_2	P_1

- P_1 carries the even parity of bits P_1, D_3, D_5, D_7. P_2 carries the even parity of bits P_2, D_3,D_6, D_7. P_4 carries the even parity of bits P_4,D_5,D_6, D_7.
- Encode data bits 1101 into the 7 bit even parity hamming code.
- The message below coded in the 7 bit hamming code is received through a noisy channel. Decode the message. 1001001,0111001,1110110,0011011.
- **Error positions and their corresponding values.**

C_3	C_2	C_1
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

- $C_1 = P_1 \bmod D_3 \bmod D_5 \bmod D_7$
- $C_2 = P_2 \bmod D_3 \bmod D_6 \bmod D_7$
- $C_3 = P_4 \bmod D_5 \bmod D_6 \bmod D_7$

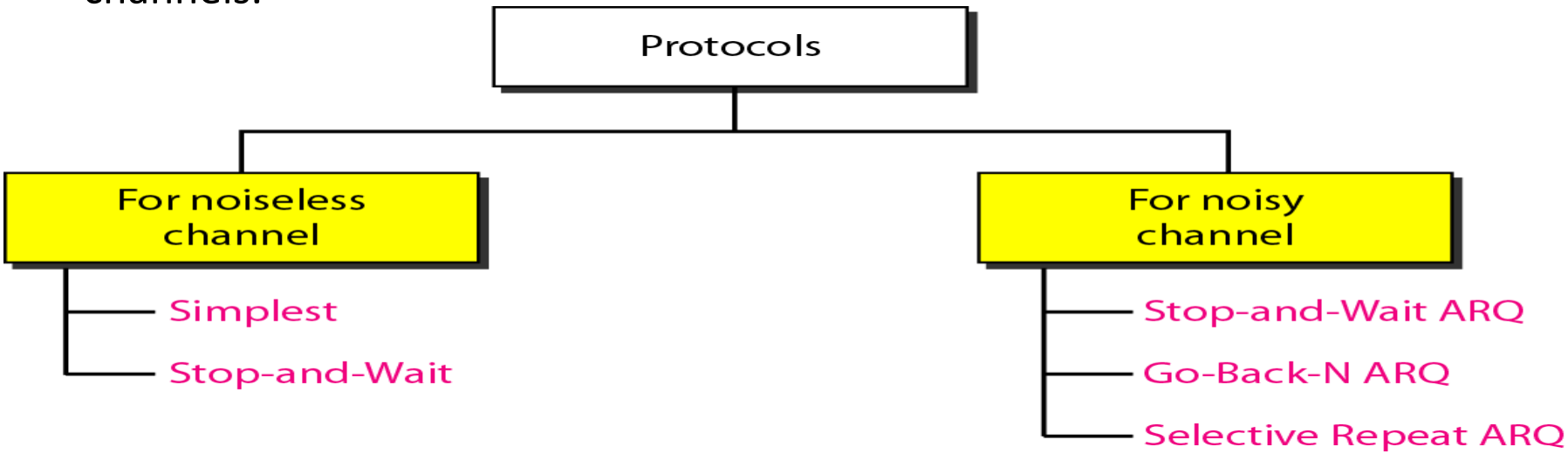
- **Haming code C(7,4) for even parity**

Datawords	Codewords	Datawords	Codewords
0000		1000	
0001		1001	
0010		1010	
0011		1011	
0100		1100	
0101		1101	
0110		1110	
0111		1111	

- The minimum distance between any two codewords is 3.
- A Hamming code with d_{min}=3 can detect upto two errors and correct one single error.

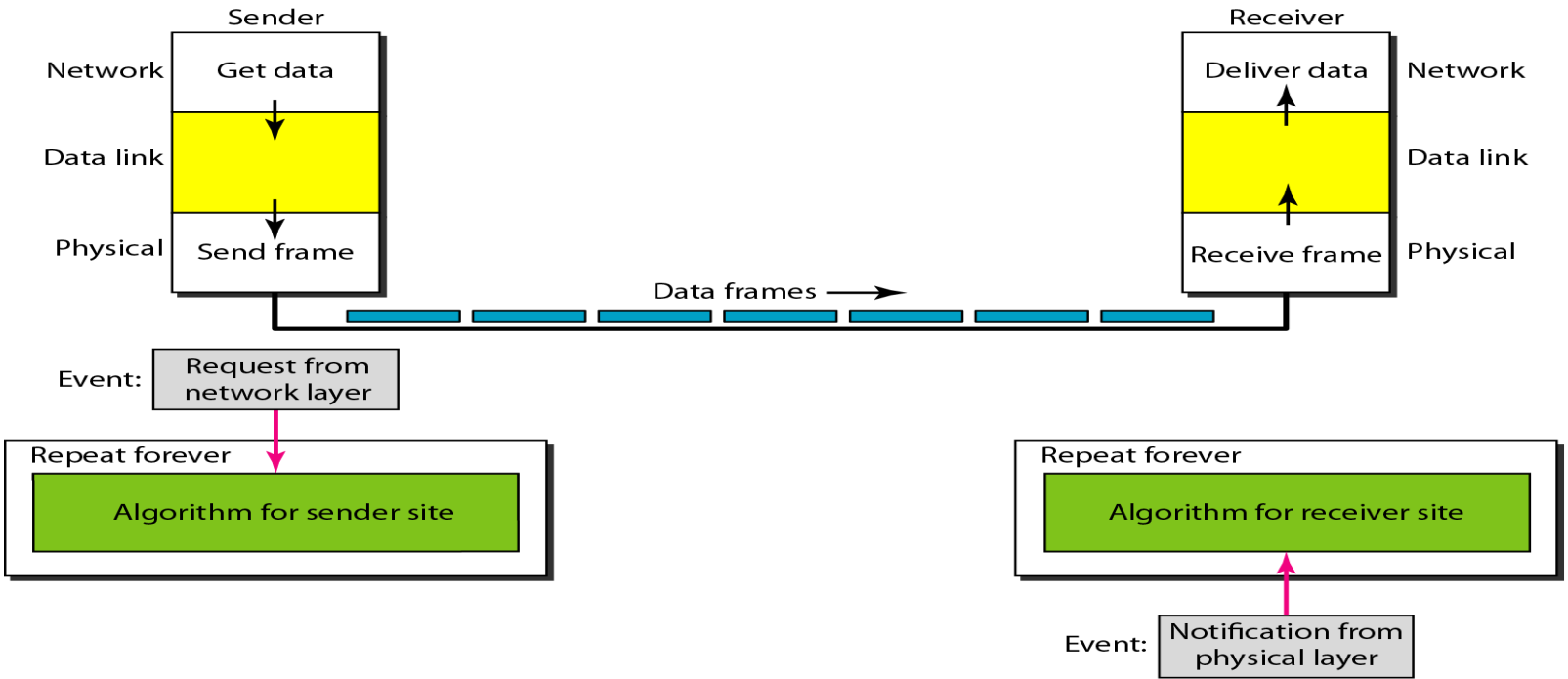
- **FLOW AND ERROR CONTROL**
- The most important responsibilities of the data link layer are flow control and error control.
- Collectively, these functions are known as data link control.
- **Flow Control**
- Flow control coordinates the amount of data that can be sent before receiving an acknowledgment and is one of the most important duties of the data link layer.
- In most protocols, flow control is a set of procedures that tells the sender how much data it can transmit before it must wait for an acknowledgment from the receiver.
- The receiving device must be able to inform the sending device before those limits are reached and to request that the transmitting device send fewer frames or stop temporarily.
- Incoming data must be checked and processed before they can be used.
- For this reason, each receiving device has a block of memory, called a *buffer,* reserved for storing incoming data until they are processed.

- **Error Control**
- In the data link layer, the term *error control* refers primarily to methods of error detection and retransmission.
- It allows the receiver to inform the sender of any frames lost or damaged in transmission and coordinates the retransmission of those frames by the sender.
- **DATA LINK PROTOCOLS**
- Data link layer combines framing, flow control, and error control to achieve the delivery of data from one node to another.
- These protocols are divided based on those that can be used for noiseless (error-free) channels and those that can be used for noisy (error-creating) channels.



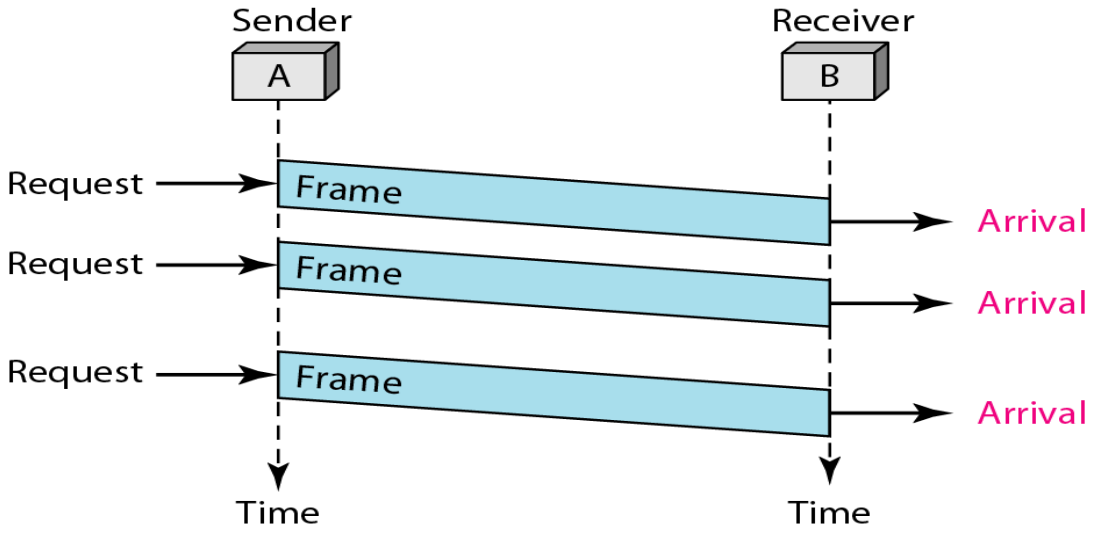
- **NOISELESS CHANNELS**
- **Simplest Protocol**
- *Assumptions:* No frames are lost, duplicated, or corrupted, sender and receiver have infinite buffer and processing capabilities.
- No flow control or error control procedures are required.
- It is a unidirectional protocol in which data frames are traveling in only one direction-from the sender to receiver.

- *Design*

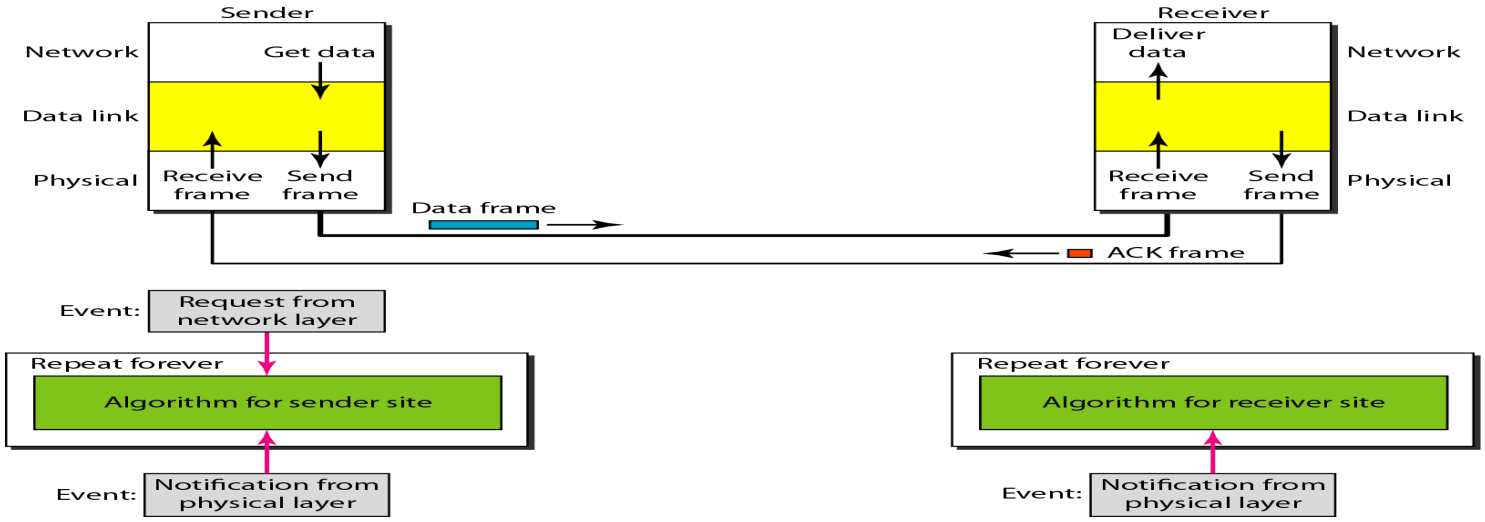


- The data link layer at the sender site gets data from its network layer, makes a frame out of the data, and sends it.
- The data link layer at the receiver site receives a frame from its physical layer, extracts data from the frame, and delivers the data to its network layer.
- The procedure at the sender site is constantly running; there is no action until there is a request from the network layer.
- The procedure at the receiver site is also constantly running, but there is no action until notification from the physical layer arrives.

- *Flow diagram*

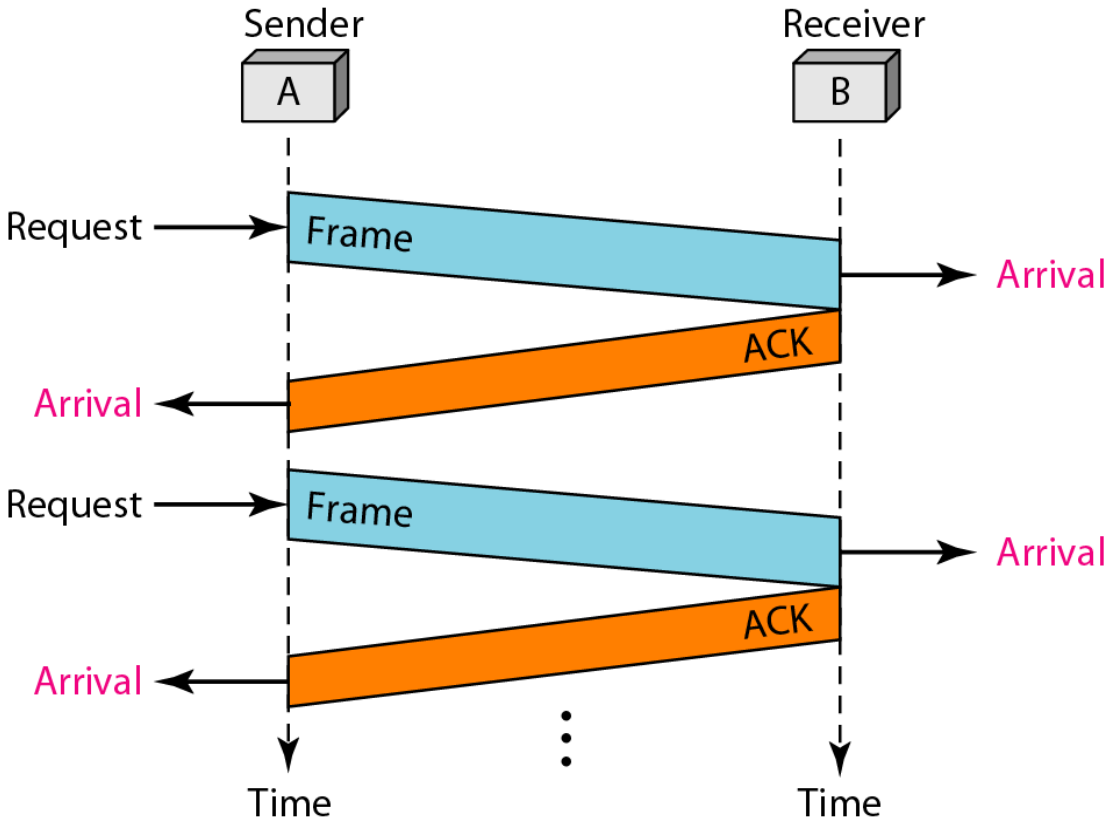


- **Stop-and-Wait Protocol**
- *Assumptions:* No frames are lost, duplicated, or corrupted, sender has infinite buffer and processing capabilities but receiver is finite.
- Normally, the receiver is slow and does not have enough storage space, especially if it is receiving data from many sources.
- There must be feedback from the receiver to the sender.
- In the Stop-and-Wait Protocol sender sends one frame, stops until it receives confirmation from the receiver and then sends the next frame.
- We still have unidirectional communication for data frames, but auxiliary ACK frames travel from the other direction.
- *Design*



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

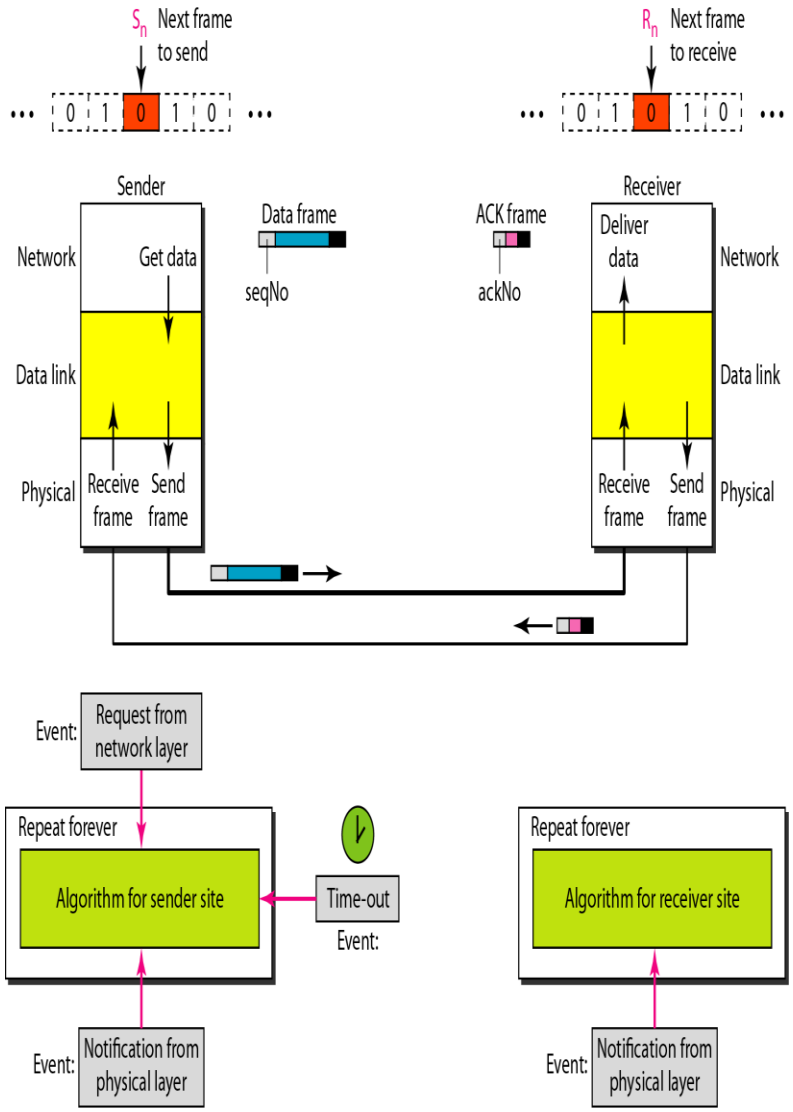
- *Flow diagram*



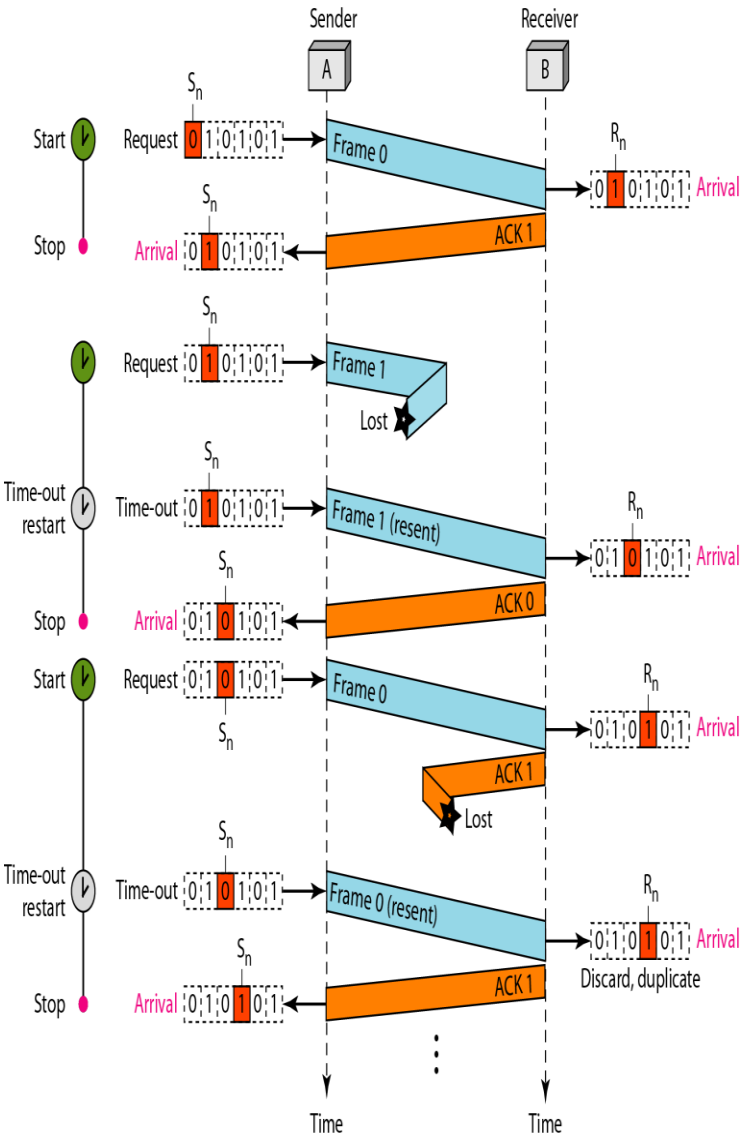
- **NOISY CHANNELS**
- Although the Stop-and-Wait Protocol gives us an idea of how to add flow control to its predecessor, noiseless channels are nonexistent.
- **Sliding Window Protocols.**
- **Stop-and-Wait Automatic Repeat Request(Stop and Wait ARQ)**
- *Assumptions:* Frames can be lost, duplicated, or corrupted, sender and receiver have finite buffer and processing capabilities.
- To detect and correct corrupted frames, we need to add redundancy bits to our data frame. When the frame arrives at the receiver site, it is checked and if it is corrupted, it is silently discarded. The detection of errors in this protocol is manifested by the silence of the receiver.
- Lost frames are more difficult to handle than corrupted ones. The solution is to number(sequence numbers) the frames. When the receiver receives a data frame that is out of order, this means that frames were either lost or duplicated.
- If the receiver does not respond when there is an error, how can the sender know which frame to resend?
- To remedy this problem, the sender keeps a copy of the sent frame. At the same time, it starts a timer.

- If the timer expires and there is no ACK for the sent frame, the frame is resent, the copy is held, and the timer is restarted.
- Since an ACK frame can also be corrupted and lost, it too needs redundancy bits and a sequence number. The ACK frame for this protocol has a sequence number field.
- In this protocol, the sender simply discards a corrupted ACK frame or ignores an out-of-order one.
- One important consideration is the range of the sequence numbers.
- For example, if we decide that the field is *m* bits long, the sequence numbers start from 0, go to 2^m- 1, and then are repeated.
- Since the sequence numbers must be suitable for both data frames and ACK frames, the acknowledgment numbers always announce the sequence number of the next frame expected by the receiver.
- One bit sequence numbers are used.

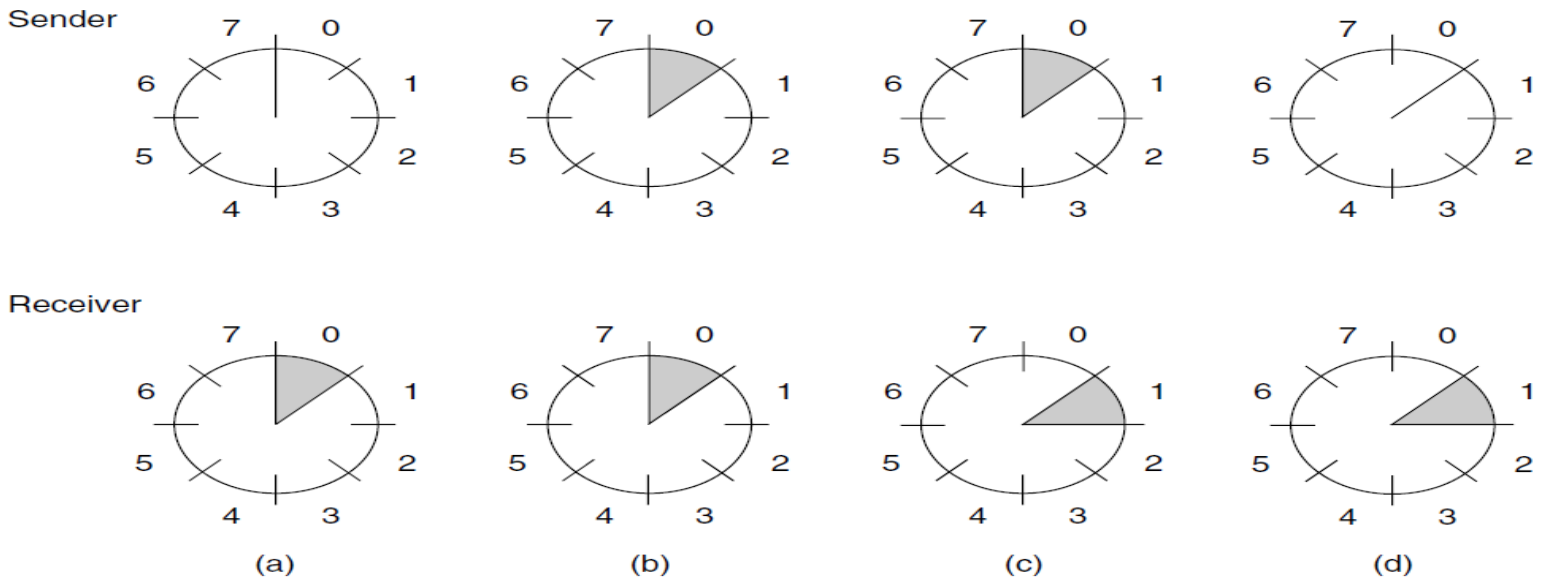
- *Design*



Flow Diagram

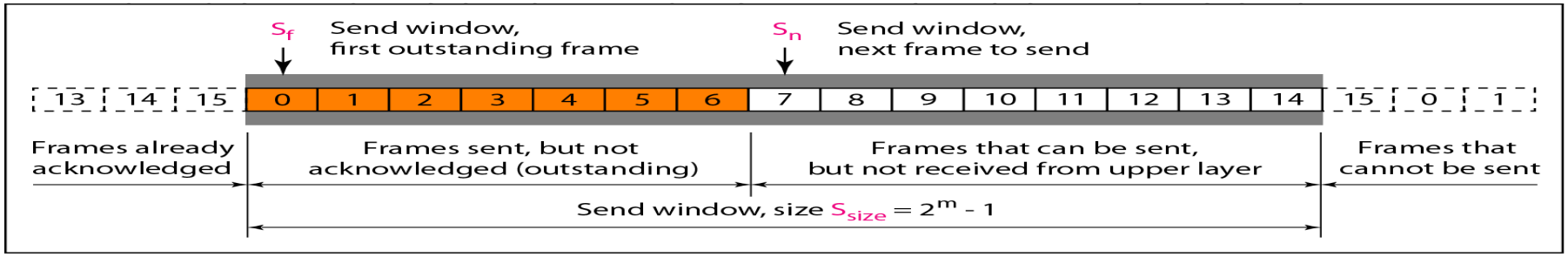


- **Stop and Wait ARQ** is a sliding window protocol.
- The essence of all sliding window protocols is that at any instant of time, the sender maintains a set of sequence numbers corresponding to frames that are sent but not yet acknowledged. These frames are said to fall within the **sending window**.
- Similarly, the receiver also maintains a **receiving window** corresponding to the set of frames it is permitted to accept.
- For stop and wait ARQ both the sender and receiver window sizes are 1.

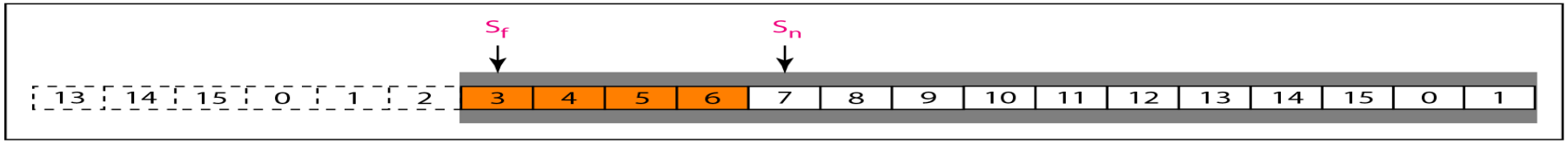


A sliding window of size 1, with a 3-bit sequence number. (a) Initially. (b) After the first frame has been sent. (c) After the first frame has been received. (d) After the first acknowledgement has been received.

- **Go-Back-N Automatic Repeat Request (Go-Back-N ARQ)**
- Sender window size is greater than 1, but the receiver window size is 1.
- In this protocol several frames are sent before receiving acknowledgments; we keep a copy of these frames until the acknowledgments arrive.
- m bit sequence numbers are used ranging from 0 to 2^m- 1.
- For example, if *m* is 4, the only sequence numbers are 0 through 15 inclusive. However, we can repeat the sequence. So the sequence numbers are

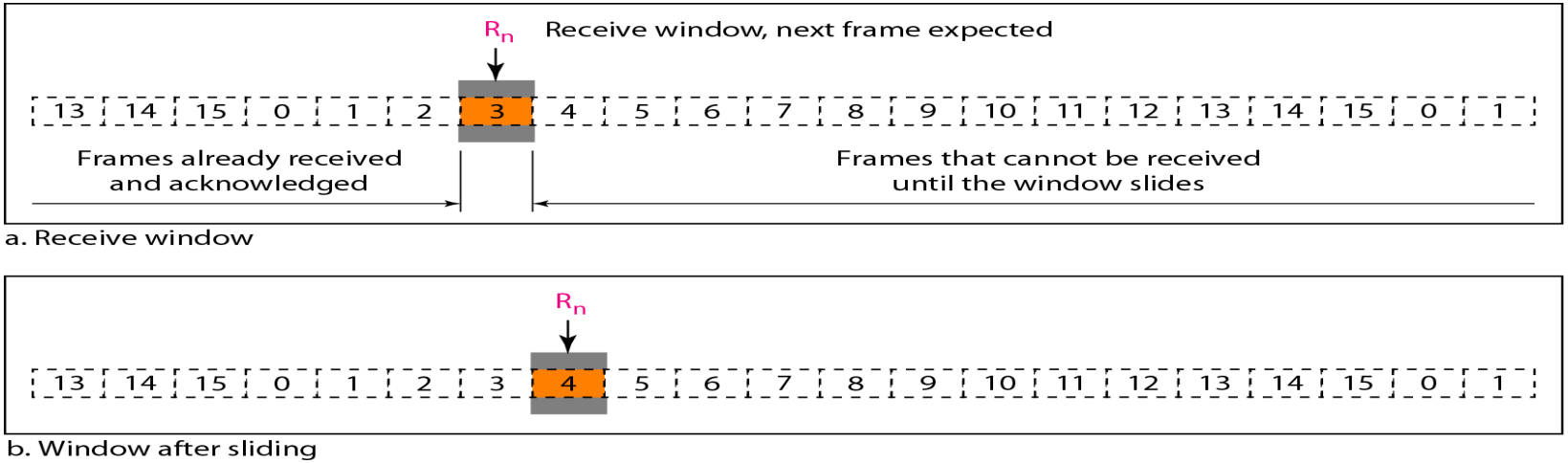


a. Send window before sliding



b. Send window after sliding

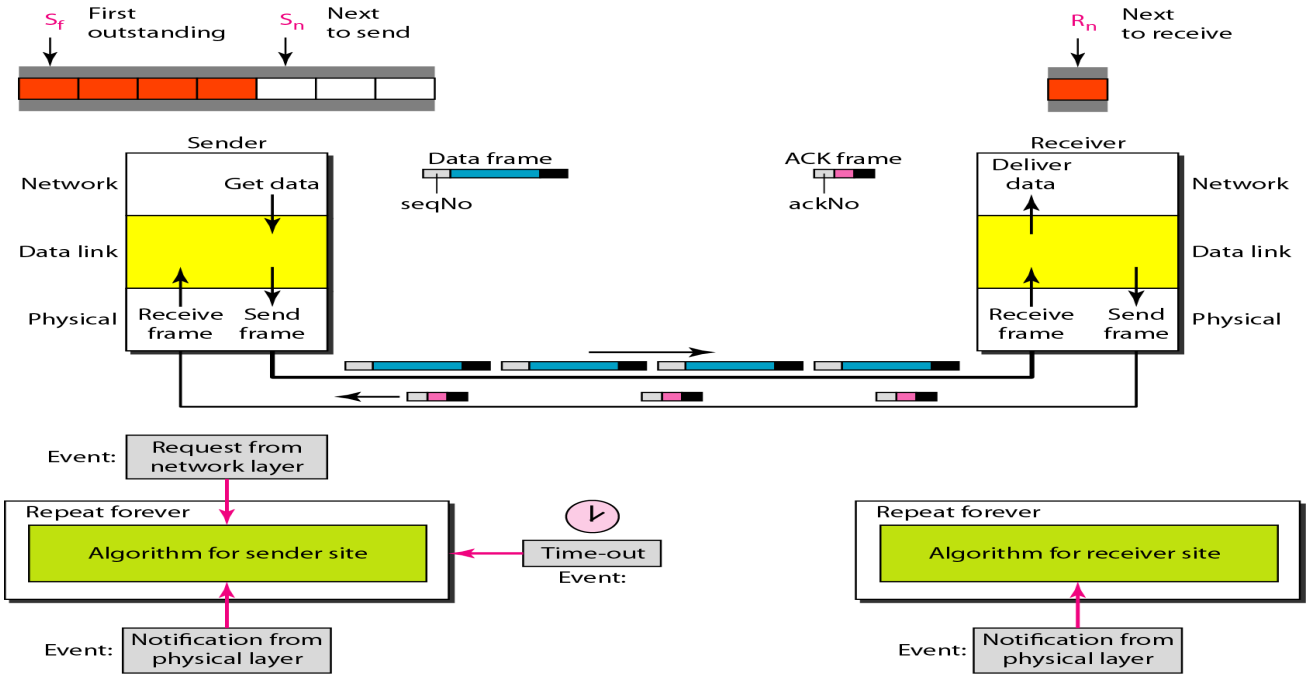
- The variable *Sf* defines the sequence number of the first (oldest) outstanding frame. The variable *Sn* holds the sequence number that will be assigned to the next frame to be sent. Finally, the variable Ssize defines the size of the window, which is fixed in our protocol.
- *Receive window*



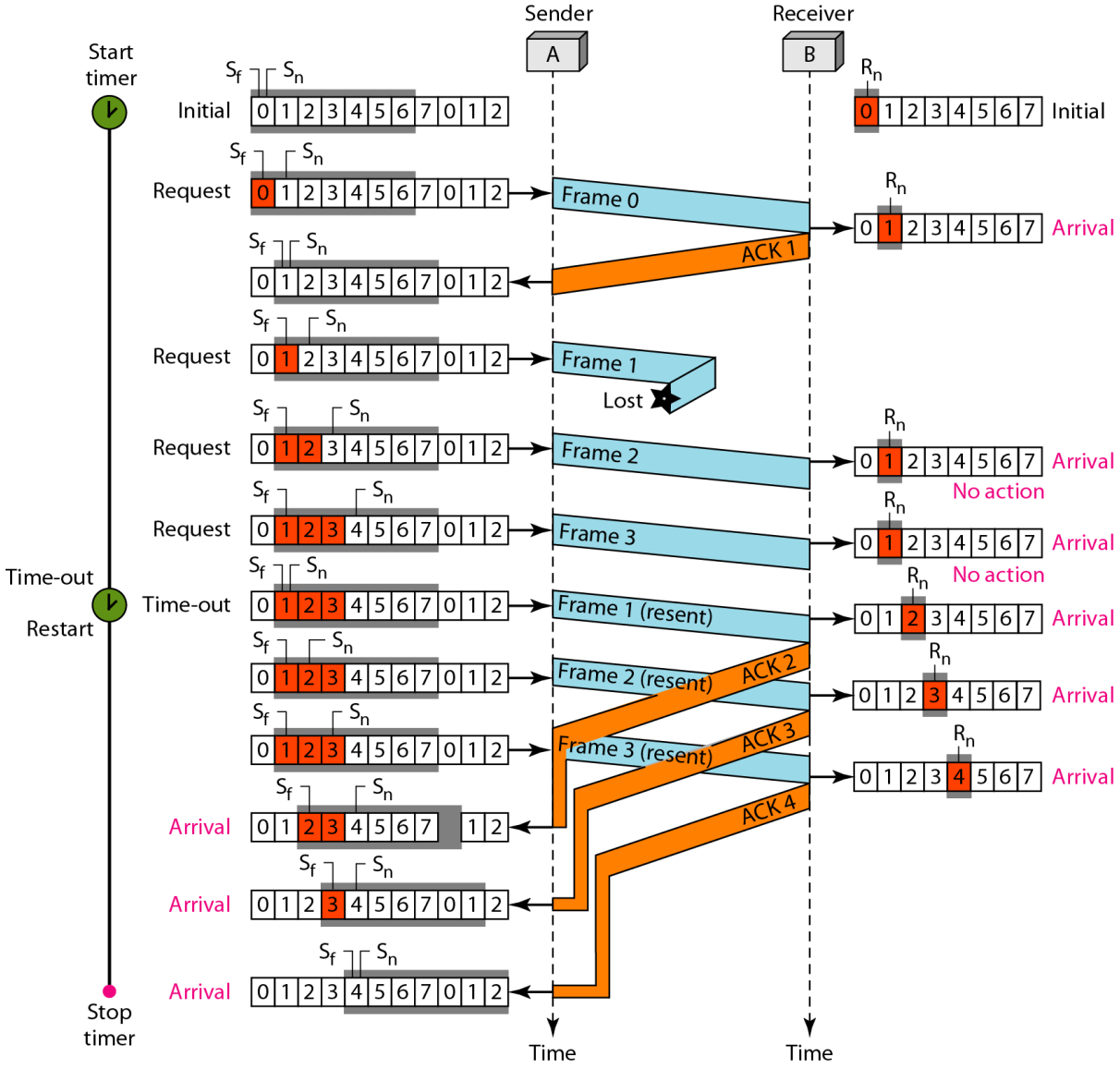
- The size of the receive window is always I. The receiver is always looking for the arrival of a specific frame. Any frame arriving out of order is discarded and needs to be resent.
- Note that we need only one variable *Rn* (receive window, next frame expected) to define this abstraction. Only a frame with a sequence number matching the value of *Rn* is accepted and acknowledged.

- The sender maintains only one timer for the first outstanding frame in the window. When the timer expires, the sender resends all outstanding frames.
- The receiver sends a positive acknowledgment if a frame has arrived safe and sound and in order.
- If a frame is damaged or is received out of order, the receiver is silent and will discard all subsequent frames until it receives the one it is expecting.

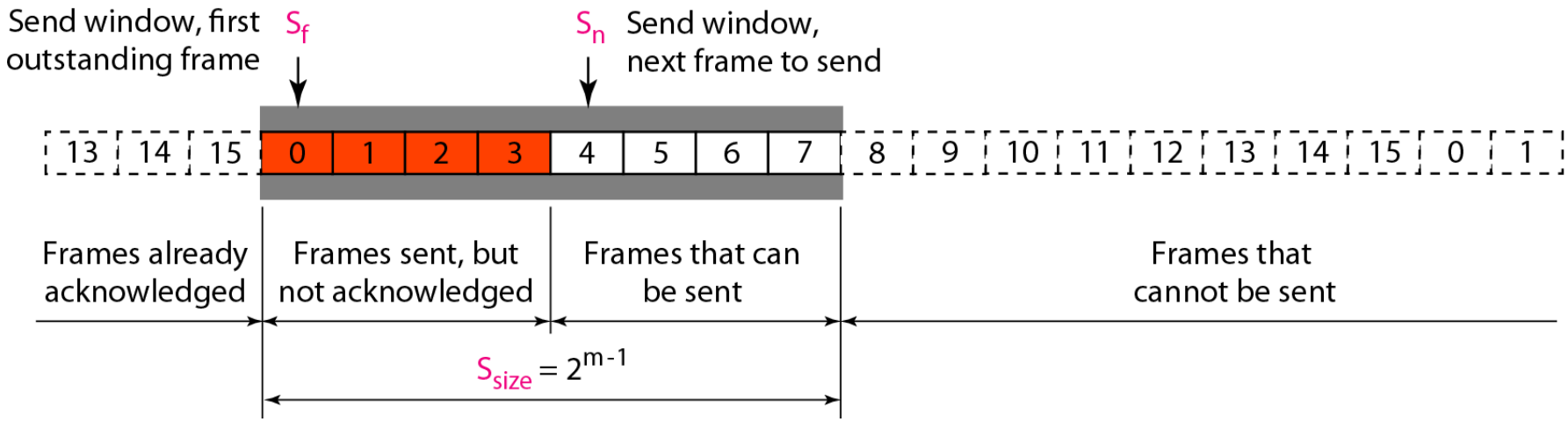
- *Design of Go-Back-N ARQ*



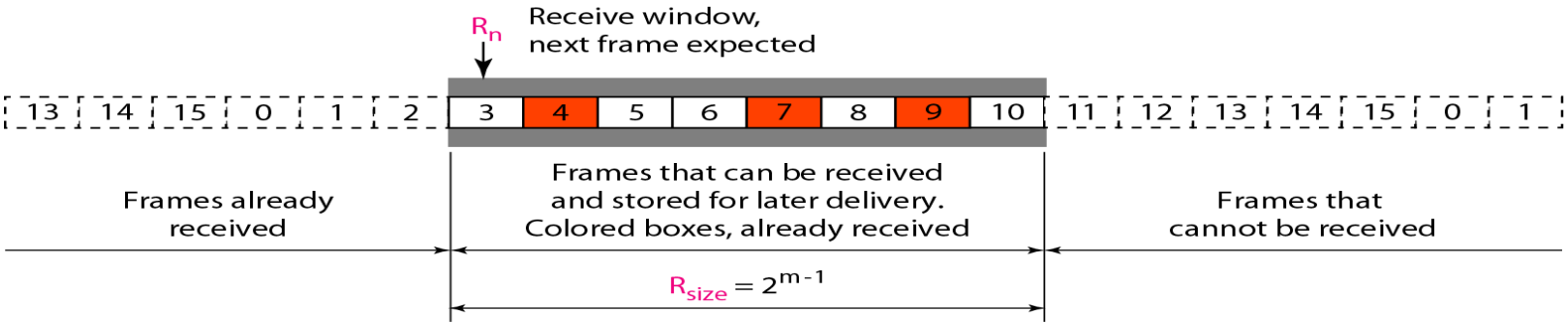
- *Flow diagram*



- **Selective Repeat Automatic Repeat Request (Selective Repeat ARQ)**
- *Go-Back-N* ARQ is very inefficient for a noisy link.
- In selective Repeat ARQ sender window size is greater than 1 and receiver window size is greater than 1.
- It is more efficient for noisy links, but the processing at the receiver is more complex.
- The size of the send window is much smaller, it is 2^{m-1}. Second, the receive window is the same size as the send window.
- For example, if *m* = 4, the sequence numbers go from 0 to 15, but the size of the window is just 8 (it is 15 in the *Go-Back-N* Protocol).

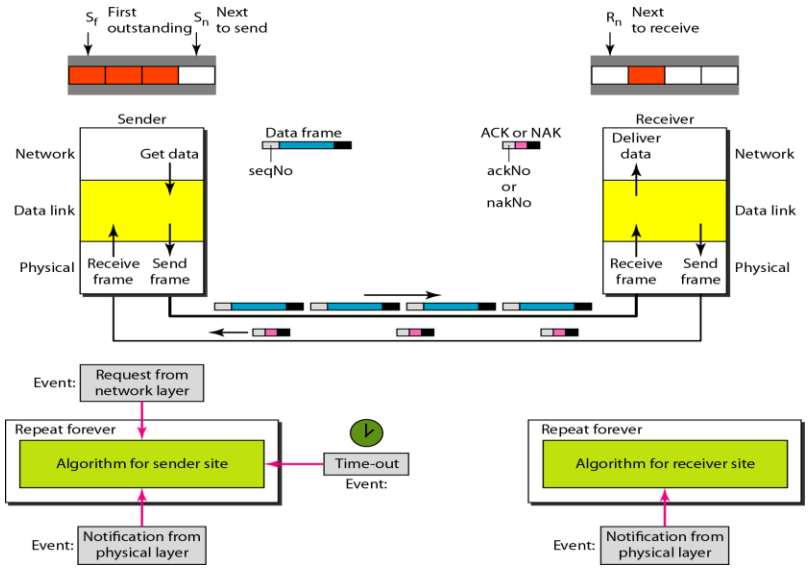


- The receive window in Selective Repeat is totally different from the one in Go-Back-N.

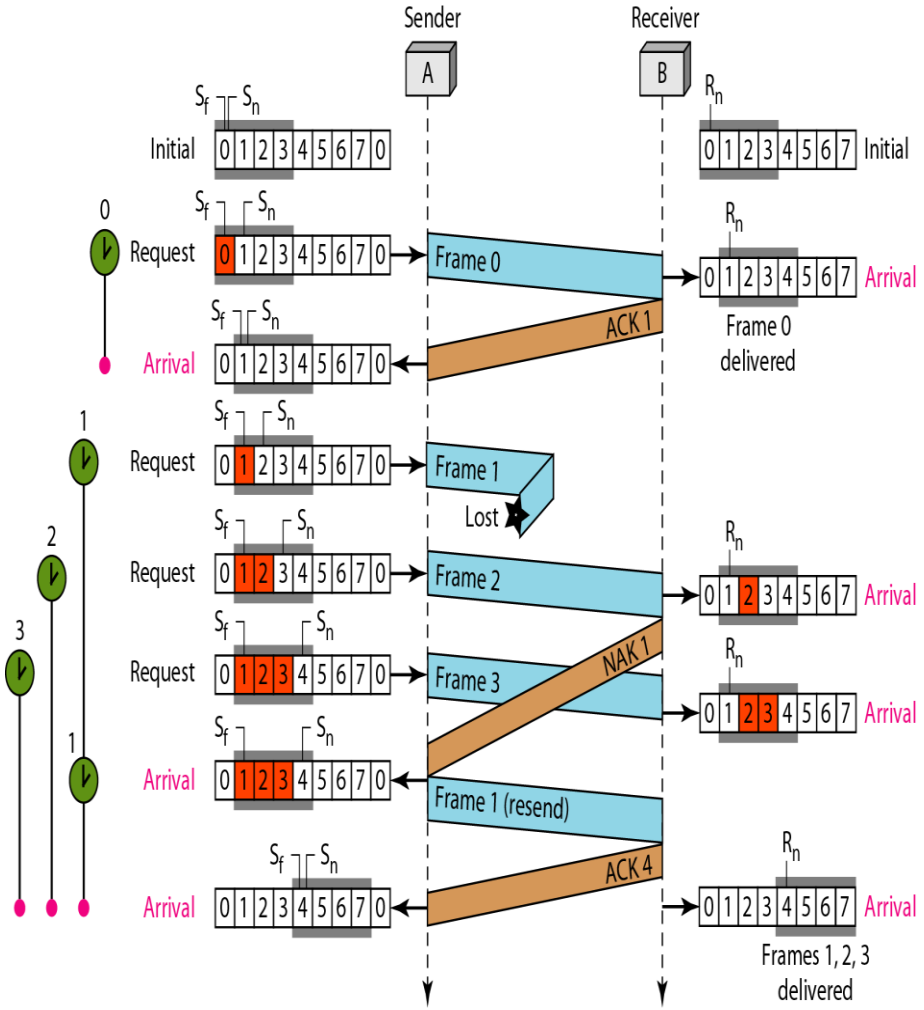


- The Selective Repeat Protocol allows as many frames as the size of the receive window to arrive out of order and be kept until there is a set of in-order frames to be delivered to the network layer.
- Also when frames are damaged/lost Negative Acknowledgment (NAK) is sent from receiver.

- *Design*

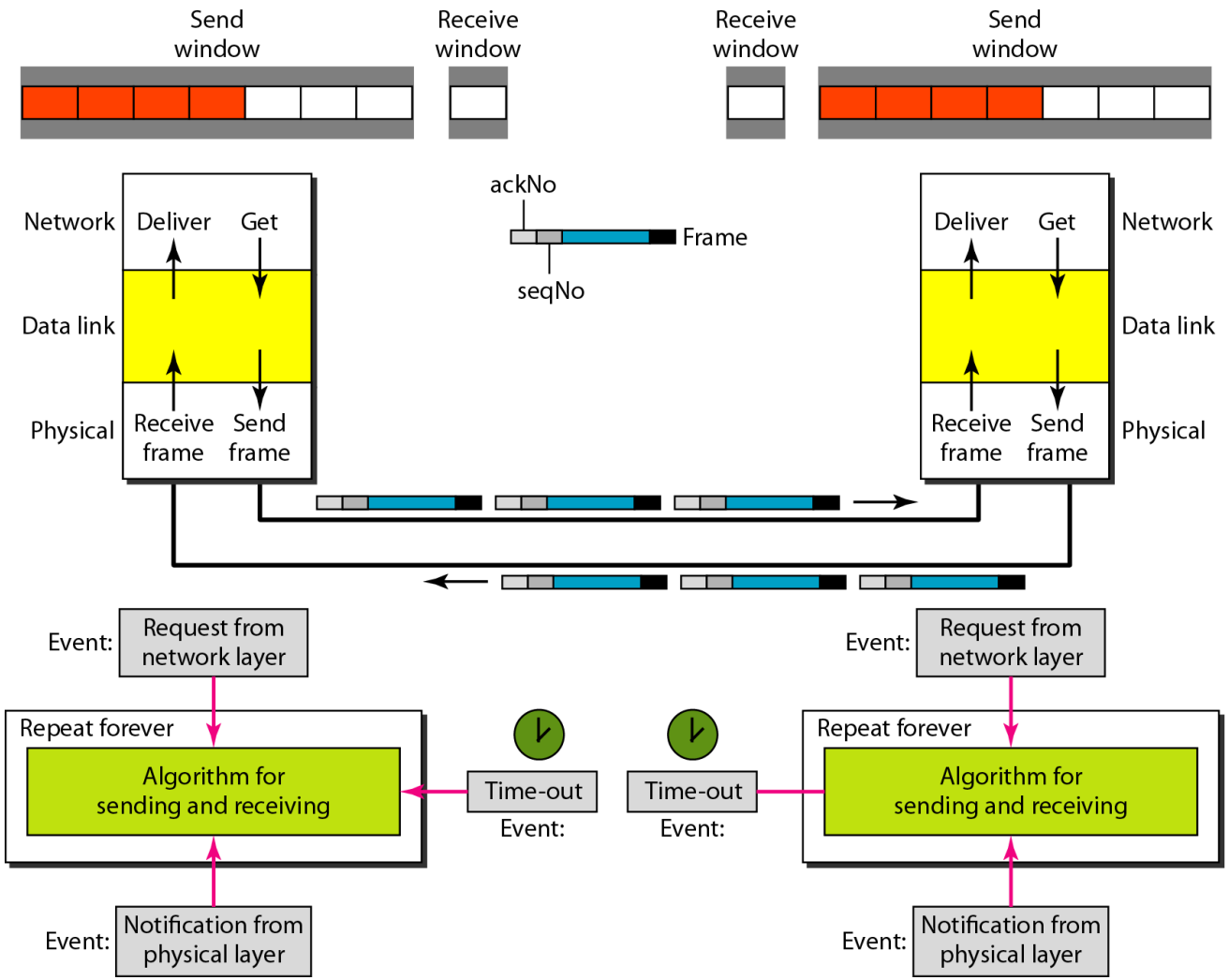


Flow diagram



- **Piggybacking**
- The three protocols we discussed in this section are all unidirectional: data frames flow in only one direction although control information such as ACK and NAK frames can travel in the other direction.
- In real life, data frames are normally flowing in both directions: from node A to node B and from node B to node A. This means that the control information also needs to flow in both directions.
- A technique called **piggybacking** is used to improve the efficiency of the bidirectional protocols.
- When a frame is carrying data from A to B, it can also carry control information about arrived (or lost) frames from B; when a frame is carrying data from B to A, it can also carry control information about the arrived (or lost) frames from A.

- *Design of piggybacking in Go-Back-N ARQ*



- **HDLC (High-level Data Link Control)**
- HDLC is a bit-oriented protocol for communication over point-to-point and multipoint links.
- It implements the ARQ mechanisms.

- **Configurations and Transfer Modes**
- HDLC provides two common transfer modes that can be used in different configurations: normal response mode (NRM) and asynchronous balanced mode (ABM).

- *Normal Response Mode*
- In normal response mode (NRM), the station configuration is unbalanced.
- We have one primary station and multiple secondary stations. A primary station can send commands; a secondary station can only respond.
- The NRM is used for both point-to-point and multiple-point links.

a. Point-to-point

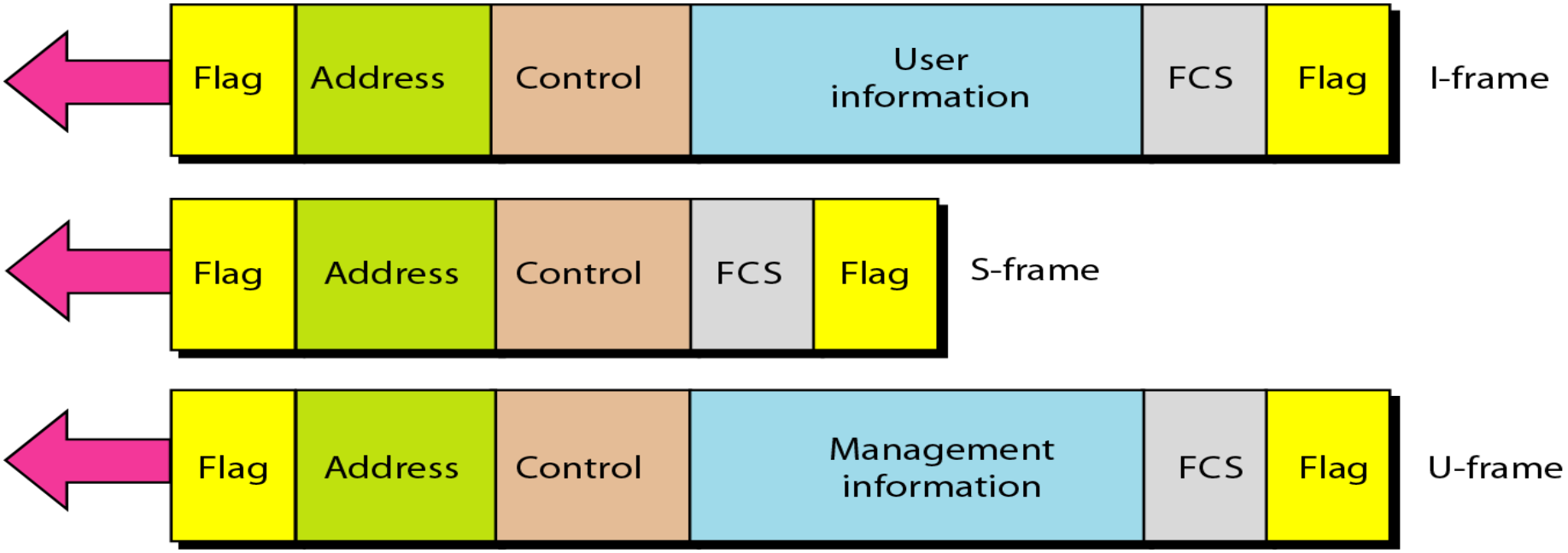
b. Multipoint

- *Asynchronous Balanced Mode*
- In asynchronous balanced mode (ABM), the configuration is balanced.
- The link is point-to-point, and each station can function as a primary and a secondary (acting as peers).

- **Frames**
- HDLC defines three types of frames: information frames (I-frames), supervisory frames (S-frames), and unnumbered frames (U-frames).
- Each type of frame serves as an envelope for the transmission of a different type of message.
- I-frames are used to transport user data and control information relating to user data (piggybacking).
- S-frames are used only to transport control information.
- U-frames are reserved for system management. Information carried by U-frames is intended for managing the link itself.

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

- *HDLC frames*



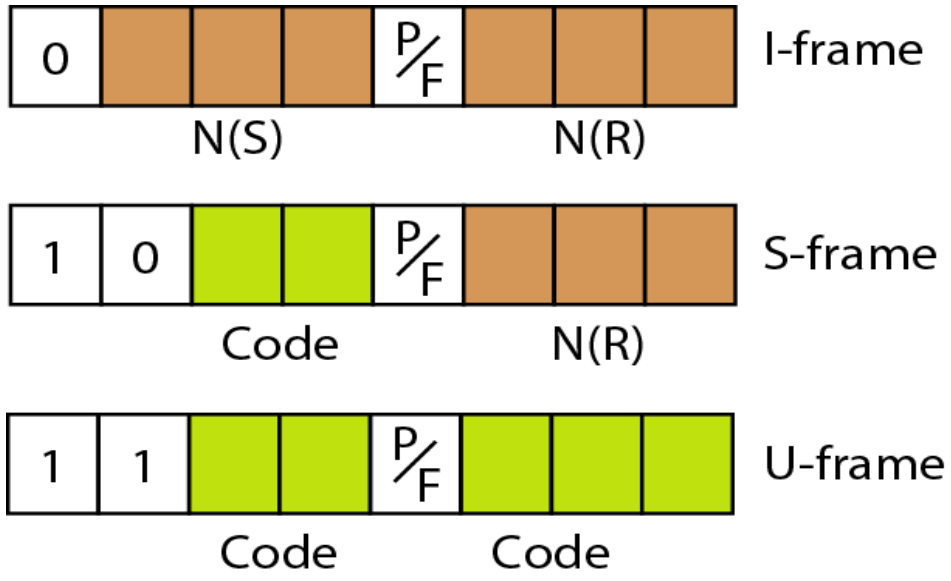
- *Fields*
- **Flag field.** The flag field of an HDLC frame is an 8-bit sequence with the bit pattern 01111110 that identifies both the beginning and the end of a frame and serves as a synchronization pattern for the receiver.

- **Address field.** The second field of an HDLC frame contains the address of the secondary station.
- If a primary station created the frame, it contains a *to* address. If a secondary creates the frame, it contains *a from* address.
- An address field can be 1 byte or several bytes long, depending on the needs of the network. One byte can identify up to 128 stations (1 bit is used for another purpose).
- Larger networks require multiple-byte address fields.
- If the address field is only 1 byte, the last bit is always a 1. If the address is more than 1 byte, all bytes but the last one will end with 0; only the last will end with 1.
- Ending each intermediate byte with 0 indicates to the receiver that there are more address bytes to come.

- **Control field.** The control field is a 1- or 2-byte segment of the frame used for flow and error control. The interpretation of bits in this field depends on the frame type.

- **Information field.** The information field contains the user's data from the network layer or management information. Its length can vary from one network to another.
- **FCS field.** The frame check sequence (FCS) is the HDLC error detection field. It can contain either a 2- or 4-byte ITU-T CRC.

- **Control Field**
- The control field determines the type of frame and defines its functionality.



- *Control Field for I-Frames*
- I-frames are designed to carry user data from the network layer. In addition, they can include flow and error control information (piggybacking).
- The first bit defines the type. If the first bit of the control field is 0, this means the frame is an I-frame.
- The next 3 bits, called *N(S),* define the sequence number of the frame. Note that with 3 bits, we can define a sequence number between 0 and 7.
- The last 3 bits, called *N(R),* correspond to the acknowledgment number when piggybacking is used.
- The single bit between *N(S)* and *N(R)* is called the *P/F* bit. It has meaning only when it is set (bit = 1) and can mean poll or final.
- It means *poll* when the frame is sent by a primary station to a secondary.
- It means *final* when the frame is sent by a secondary to a primary.

- *Control Field for S-Frames*
- Supervisory frames are used for flow and error control whenever piggybacking is either impossible or inappropriate. S-frames do not have information fields.
- If the first 2 bits of the control field is 10, this means the frame is an S-frame.
- The last 3 bits, called *N(R),* corresponds to the acknowledgment number (ACK) or negative acknowledgment number (NAK) depending on the type of S-frame.
- The 2 bits called code is used to define the type of S-frame itself. With 2 bits, we can have four types of S-frames, as described below:
- **Receive ready (RR).** If the value of the code subfield is 00, it is an RR S-frame. This kind of frame acknowledges the receipt of a safe and sound frame or group of frames. In this case, the value *N(R)* field defines the acknowledgment number.
- **Receive not ready (RNR).** If the value of the code subfield is 10, it is an RNR S-frame. This kind of frame is an RR frame with additional functions. It acknowledges the receipt of a frame or group of frames, and it announces that the receiver is busy and cannot receive more frames. The value of *NCR)* is the acknowledgment number.

- **Reject (REJ).** If the value of the code subfield is 01, it is a REJ S-frame. It is a NAK that can be used in *Go-Back-N* ARQ to improve the efficiency of the process by informing the sender, before the sender time expires, that the last frame is lost or damaged. The value of *NCR)* is the negative acknowledgment number.
- **Selective reject (SREJ).** If the value of the code subfield is 11, it is an SREJ S-frame. This is a NAK frame used in Selective Repeat ARQ. Note that the HDLC Protocol uses the term *selective reject* instead of *selective repeat.* The value of *N(R)* is the negative acknowledgment number.

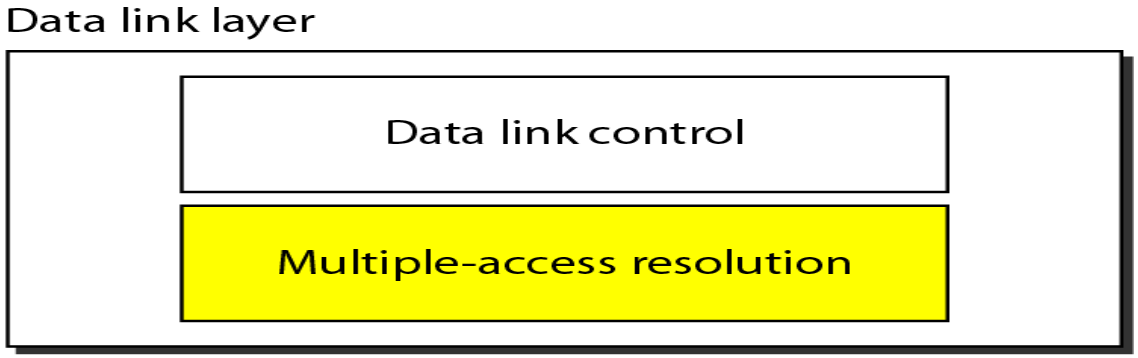
- *Control Field for U-Frames*
- Unnumbered frames are used to exchange session management and control information between connected devices.
- Unlike S-frames, U-frames contain an information field, but one used for system management information, not user data.
- U-frame codes are divided into two sections: a 2-bit prefix before the P/F bit and a 3-bit suffix after the P/F bit.

- Together, these two segments (5 bits) can be used to create up to 32 different types of U-frames. Some of the more common types are shown in Table.

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

THE MEDIUM ACCESS CONTROL SUBLAYER

- MAC Sub layer deals with broadcast links and their protocols.



- In any broadcast network, access control is the main issue.
- The MAC sublayer is especially important in LANs, particularly wireless ones because wireless is naturally a broadcast channel.

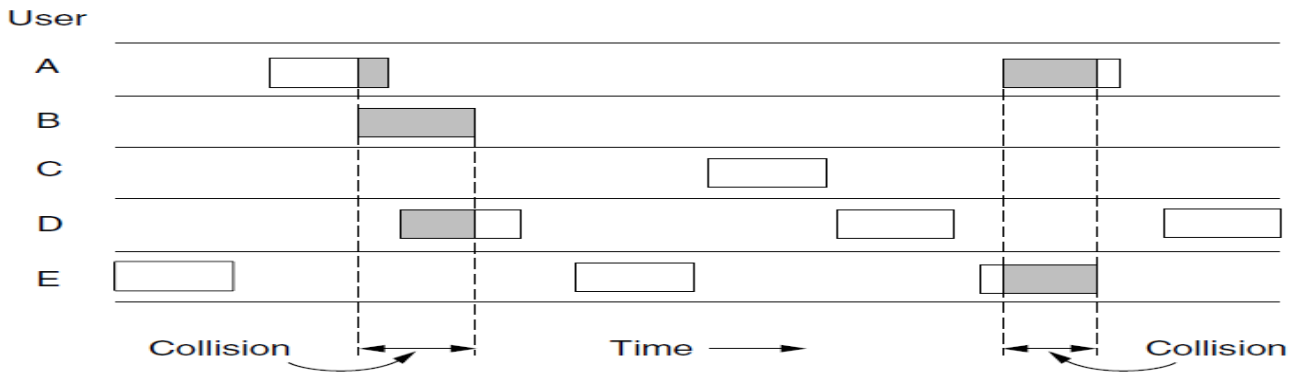
- **THE CHANNEL ALLOCATION PROBLEM**
- *Static Channel Allocation*
- The traditional way of allocating a single channel, are FDM and TDM.

- In FDM, If there are *N* users, the bandwidth is divided into *N* equal-sized portions, with each user being assigned one portion.
- When there is only a small and constant number of users, each of which has a steady stream or a heavy load of traffic, this division is a simple and efficient allocation mechanism.
- However, when the number of senders is large and varying or the traffic is bursty, FDM presents problems.
- Precisely the same arguments that apply to FDM also apply to TDM.

- **Dynamic Channel Allocation**
- *Assumptions for Dynamic Channel Allocation*
- **Independent Traffic**. The model consists of *N* independent **stations** (e.g., computers, telephones), each with a program or user that generates frames for transmission. The expected number of frames generated in an interval of length Δt is $\lambda \Delta t$, where λ is a constant (the arrival rate of new frames). Once a frame has been generated, the station is blocked and does nothing until the frame has been successfully transmitted.

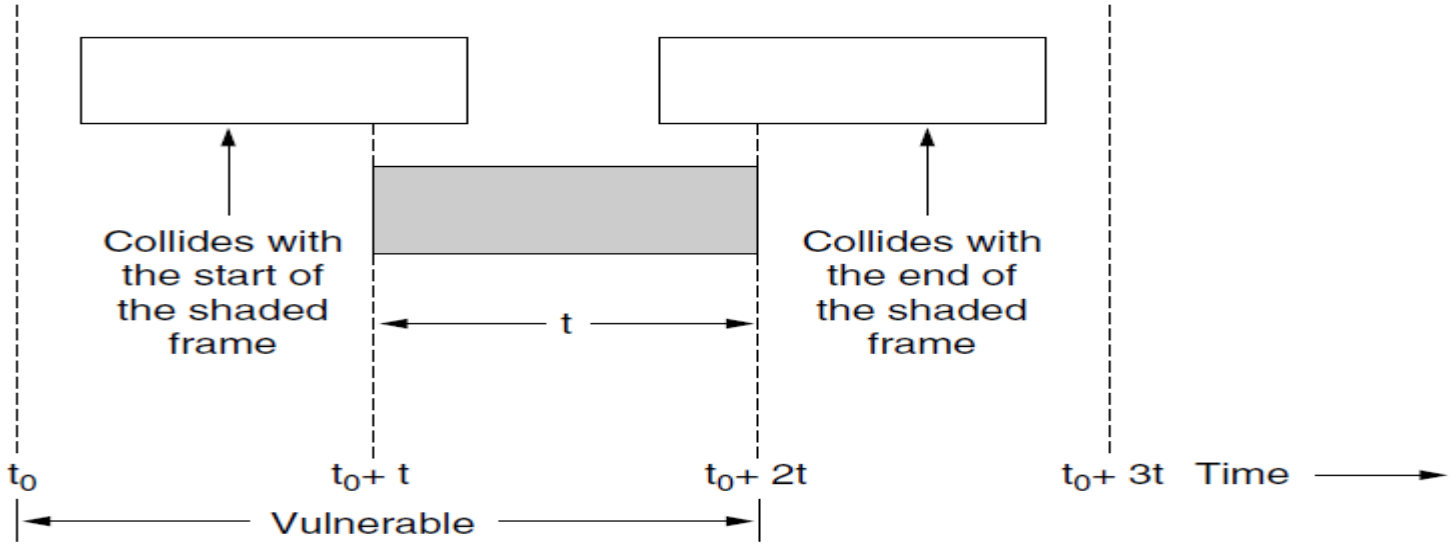
- **Single Channel**. A single channel is available for all communication. All stations can transmit on it and all can receive from it. The stations are assumed to be equally capable, though protocols may assign them different roles (e.g., priorities).
- **Observable Collisions**. If two frames are transmitted simultaneously, they overlap in time and the resulting signal is garbled. This event is called a **collision**. All stations can detect that a collision has occurred. A collided frame must be transmitted again later. No errors other than those generated by collisions occur.
- **Continuous or Slotted Time**. Time may be assumed continuous, in which case frame transmission can begin at any instant. Alternatively, time may be slotted or divided into discrete intervals (called slots). Frame transmissions must then begin at the start of a slot. A slot may contain 0, 1, or more frames, corresponding to an idle slot, a successful transmission, or a collision, respectively.
- **Carrier Sense or No Carrier Sense**. With the carrier sense assumption, stations can tell if the channel is in use before trying to use it. No station will attempt to use the channel while it is sensed as busy. If there is no carrier sense, stations cannot sense the channel before trying to use it. They just go ahead and transmit.

- **MULTIPLE ACCESS PROTOCOLS**
- **ALOHA**
- Two categories of ALOHA are defined: Pure and Slotted.
- **Pure ALOHA**
- Stations follow continuous time and no carrier sense assumptions.
- Users transmit whenever they have data to be sent. There will be collisions, and the colliding frames will be damaged.
- The sender might be able to listen for collisions after transmitting.
- If the frame was destroyed, the sender just waits a random amount of tim
- Sys way that car



- If the first bit of a new frame overlaps with just the last bit of a frame that has almost finished, both frames will be totally destroyed and both will have to be retransmitted later.
- what is the efficiency of an ALOHA channel?
- Let us first consider an infinite collection of users typing at their terminals (stations).
- Let the ‘‘frame time’’ denote the amount of time needed to transmit the standard, fixed-length frame.
- Let the new frames generated be a mean of N frames per frame time.
- If $N > 1$, the user community is generating frames at a higher rate than the channel can handle, and nearly every frame will suffer a collision.
- For reasonable throughput, we would expect $0 < N < 1$.
- In addition to the new frames, the stations also generate retransmissions of frames that previously suffered collisions.
- Let us further assume that the old and new frames combined are well modeled by a Poisson distribution, with mean of G frames per frame time.
- Clearly, $G \geq N$.

- At low load (i.e., $N \approx 0$), there will be few collisions, hence few retransmissions, so $G \approx N$.
- At high load, there will be many collisions, so $G > N$.
- Under all loads, the throughput, S, is just the offered load, G, times the probability, P_0, of a transmission succeeding—that is, $S = GP_0$, where P_0 is the probability that a frame does not suffer a collision.
- A frame will not suffer a collision if no other frames are sent within one frame time of its start.



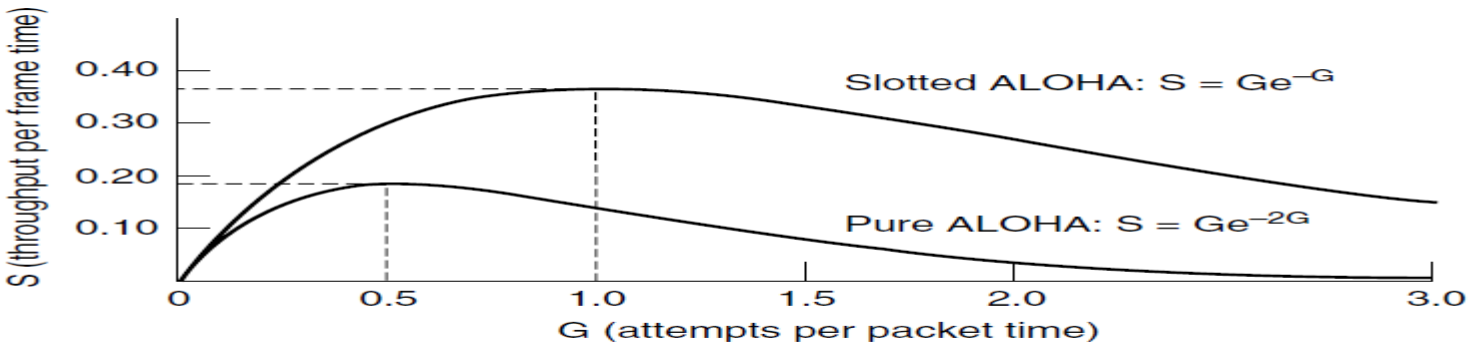
- Let t be the time required to send one frame. If any other user has generated a frame between time t_0 and $t_0 + t$, the end of that frame will collide with the beginning of the shaded one.

- Similarly, any other frame started between t_0 + t and t_0 + 2t will bump into the end of the shaded frame.
- The vulnerable period is two frame times.
- The probability that *k* frames are generated during a given frame time, in which *G* frames are expected, is given by the Poisson distribution

$$\Pr[k] = \frac{G^k e^{-G}}{k!}$$

- So the probability of zero frames is just e^{-G}.
- In an interval two frame times long, the mean number of frames generated is 2*G.*
- The probability of no frames being initiated during the entire vulnerable period is thus given by $P_0 = e^{-2G}$.
- Using $S = GP_0$, we get $S = Ge^{-2G}$.
- The maximum throughput occurs at $G = 0.5$, with $S = 1/2e$, which is about 0.184. In other words, the best we can hope for is a channel utilization of 18%.

- **Slotted ALOHA**
- Time is divided into discrete intervals called **slots**, each interval corresponding to one frame.
- This approach requires the users to agree on slot boundaries.
- A station is not permitted to send whenever the user types a line. Instead, it is required to wait for the beginning of the next slot.
- Thus, the continuous time ALOHA is turned into a discrete time one. This halves the vulnerable period.
- The probability of no other traffic during the same slot as our test frame is then e^{-G}, which leads to $S = Ge^{-G}$.
- Slotted ALOHA peaks at $G = 1$, with a throughput of $S = 1/e$ or about 0.368, twice that of pure ALOHA.
- The best we can hope for using slotted ALOHA is 37% success



- **Carrier Sense Multiple Access Protocols**
- In LANs, it is often possible for stations to detect what other stations are doing, and thus adapt their behavior accordingly. These networks can achieve a much better utilization than 1*/e.*
- Protocols in which stations listen for a carrier (i.e., a transmission) and act accordingly are called **carrier sense protocols**.

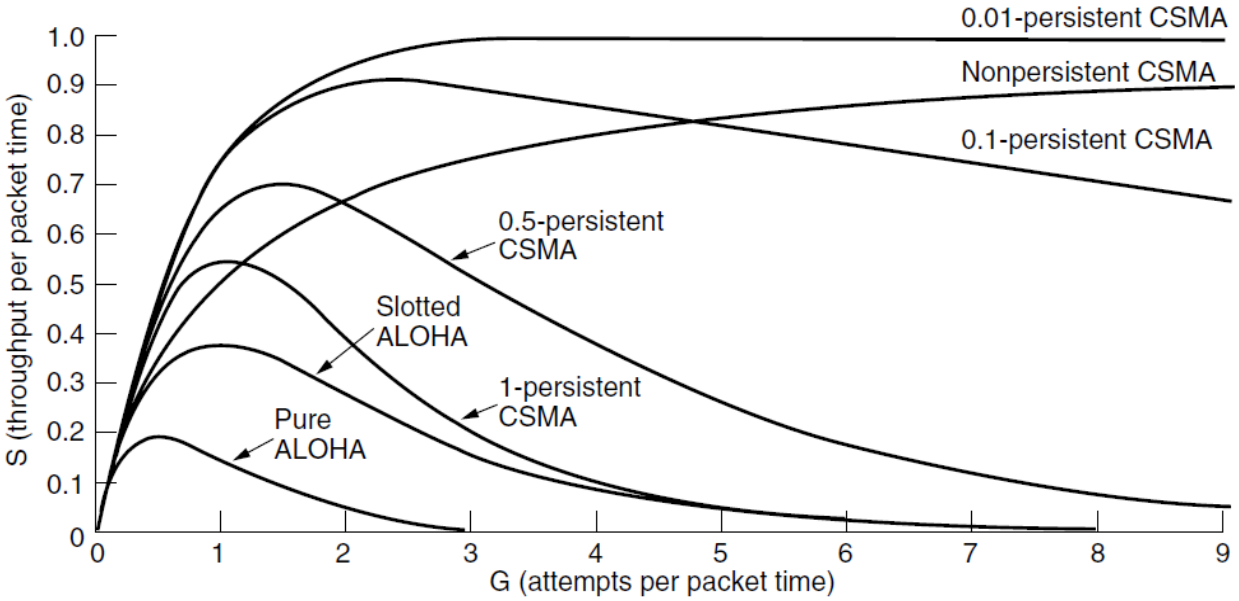
- **1-persistent CSMA**
- When a station has data to send, it first listens to the channel to see if anyone else is transmitting at that moment. If the channel is idle, the stations sends its data. Otherwise, if the channel is busy, the station just waits until it becomes idle. Then the station transmits a frame.
- If a collision occurs, the station waits a random amount of time and starts all over again.
- The protocol is called 1-persistent because the station transmits with a probability of 1 when it finds the channel idle.
- If two stations become ready in the middle of a third station's transmission, both will wait politely until the transmission ends, and then both will begin transmitting exactly simultaneously, resulting in a collision.

- The propagation delay has an important effect on collisions.

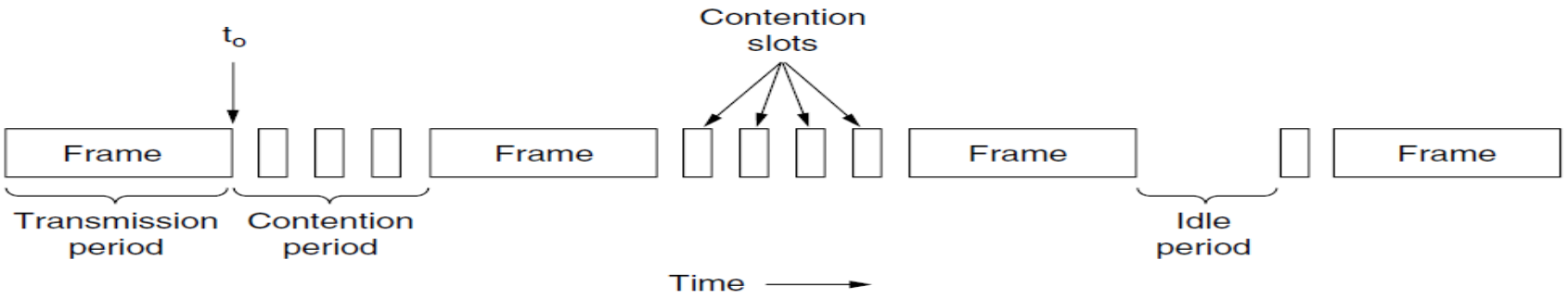
- **nonpersistent CSMA**
- In this protocol, a conscious attempt is made to be less greedy than in the previous one.
- As before, a station senses the channel when it wants to send a frame, and if no one else is sending, the station begins doing so itself.
- However, if the channel is already in use, the station does not continually sense it for the purpose of seizing it immediately upon detecting the end of the previous transmission.
- Instead, it waits a random period of time and then repeats the algorithm.
- Consequently, this algorithm leads to better channel utilization but longer delays than 1-persistent CSMA.

- **p-persistent CSMA**
- It applies to slotted channels.

- When a station becomes ready to send, it senses the channel. If it is idle, it transmits with a probability p. With a probability $q = 1 - p$, it defers until the next slot. If that slot is also idle, it either transmits or defers again, with probabilities p and q.
- This process is repeated until either the frame has been transmitted or another station has begun transmitting. In the latter case, the unlucky station acts as if there had been a collision (i.e., it waits a random time
- until the next
-



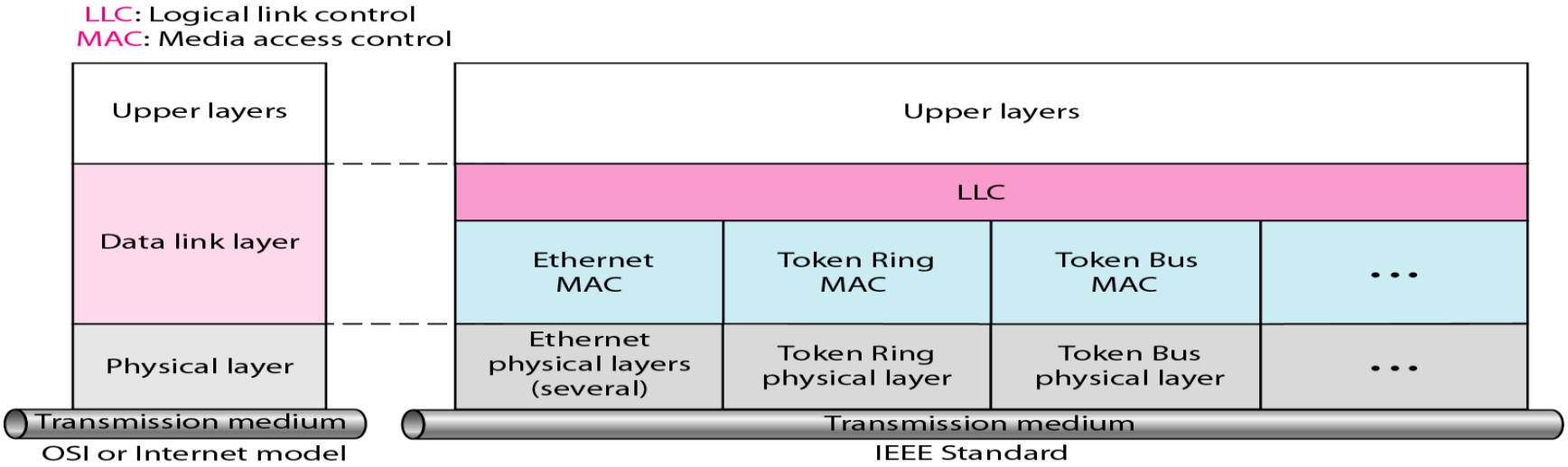
- **CSMA with Collision Detection**
- If two stations sense the channel to be idle and begin transmitting simultaneously, their signals will still collide.
- An improvement in CSMA/CD is for the stations to quickly detect the collision and abruptly stop transmitting, (rather than finishing them) since they are irretrievably garbled anyway. This strategy saves time and bandwidth.
- If the signal that is read back is different from the signal that is put out, it known a collision.
- CSMA/CD uses the below conceptual model.



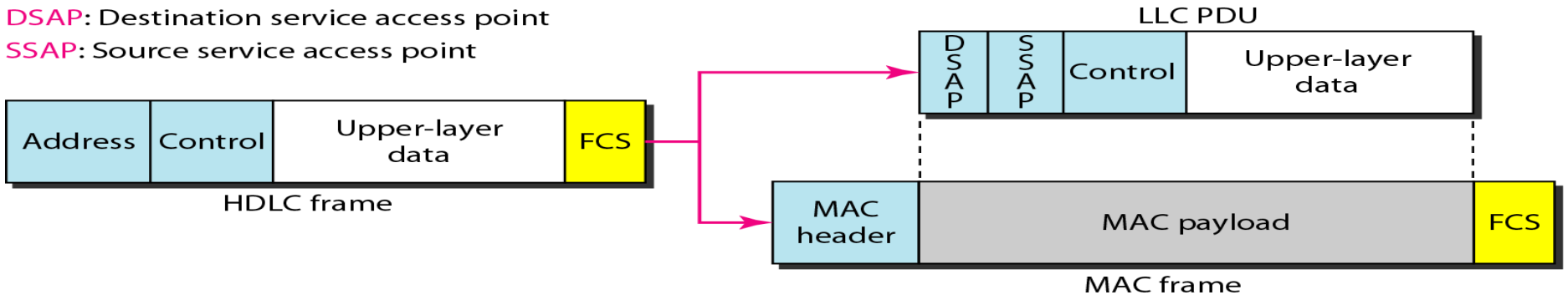
- The CSMA/CD will consist of alternating contention and transmission periods, with idle periods occurring when all stations are quite.

- The main issue is the length of the contention period, which determines the efficiency.
- Considering worst case scenario.
- Let the time for a signal to propagate between the two farthest stations be τ.
- At t_0, one station begins transmitting. At t_0 + τ − ε, an instant before the signal arrives at the most distant station, that station also begins transmitting.
- Of course, it detects the collision almost instantly and stops, but the little noise burst caused by the collision does not get back to the original station until time 2τ − ε.
- In the worst case a station cannot be sure that it has seized the channel until it has transmitted for 2τ without hearing a collision.
- we can think of CSMA/CD contention as a slotted ALOHA system with a slot width of 2τ.

- **IEEE STANDARDS 802 FOR LANS**
- In 1985, the Computer Society of the IEEE started a project, called Project 802, to set standards to enable intercommunication among equipment from a variety of manufacturers.
- Project 802 is a way of specifying functions of the physical layer and the data link layer of major LAN protocols.
- The IEEE has subdivided the data link layer into two sublayers: logical link control (LLC) and media access control (MAC). IEEE has also created several physical layer standards for different LAN protocols.



- *Logical Link Control (LLC)*
- In IEEE Project 802, flow control, error control, and part of the framing duties are collected into one sublayer called the logical link control. Framing is handled in both the LLC sublayer and the MAC sublayer.
- The LLC provides one single data link control protocol for all IEEE LANs. In this way, the LLC is different from the media access control sublayer, which provides different protocols for different LANs.
- LLC defines a protocol data unit (PDU) that is somewhat similar to that of HDLC.

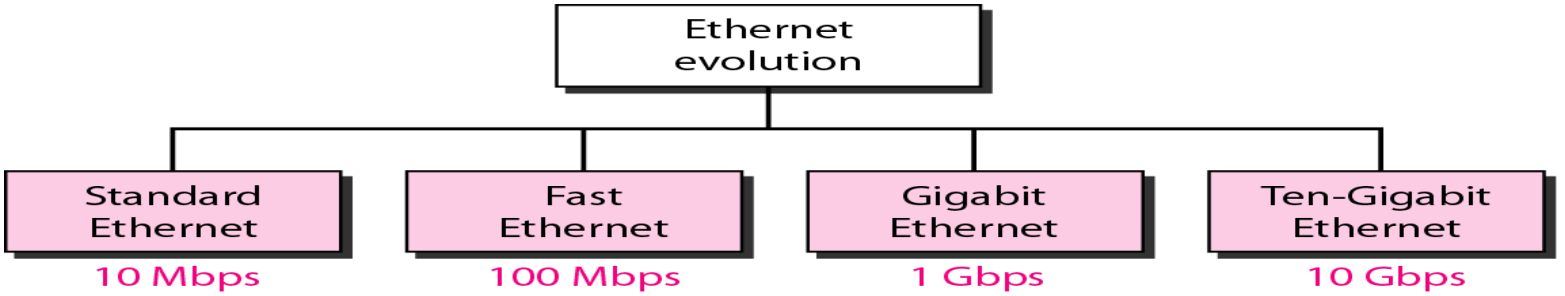


- The header contains a control field like the one in HDLC; this field is used for flow and error control.

- The two other header fields define the upper-layer protocol at the source and destination that uses LLC. These fields are called the destination service access point (DSAP) and the source service access point (SSAP).
- A frame defined in HDLC is divided into a PDU at the LLC sublayer and a frame at the MAC sublayer.

- *Media Access Control (MAC)*
- IEEE Project 802 has created a sublayer called media access control that defines the specific access method for each LAN.
- For example, it defines *CSMA/CD* as the media access method for Ethernet LANs and the token passing method for Token Ring and Token Bus LANs. Part of the framing function is also handled by the MAC layer.
- In contrast to the LLC sublayer, the MAC sublayer contains a number of distinct modules; each defines the access method and the framing format specific to the corresponding LAN protocol.
- *Physical Layer*
- The physical layer is dependent on the implementation and type of physical media used. IEEE defines detailed specifications for each LAN implementation.

- **STANDARD ETHERNET**
- The original Ethernet was created in 1976 at Xerox's Palo Alto Research Center (PARC). Since then, it has gone through four generations: Standard Ethernet (10 Mbps), Fast Ethernet (100 Mbps), Gigabit Ethernet (1 Gbps), and Ten-Gigabit Ethernet (10 Gbps).

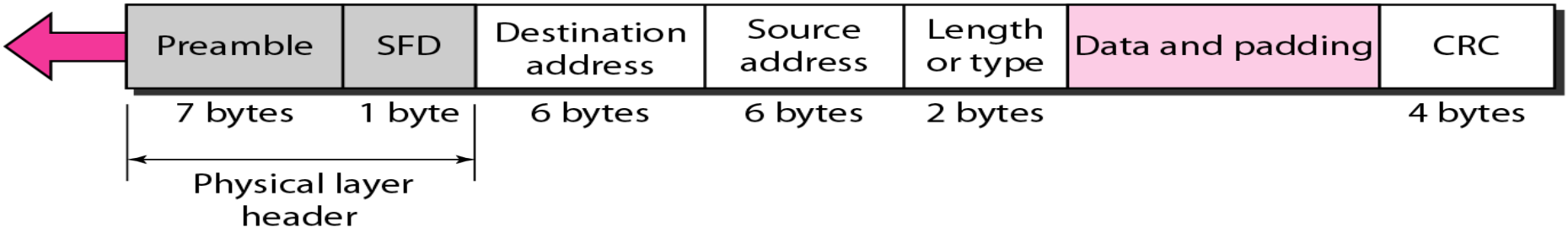


- *MAC Sublayer*
- In Standard Ethernet, the MAC sublayer governs the operation of the access method. It also frames data received from the upper layer and passes them to the physical layer.
- *Frame Format*
- The Ethernet frame contains seven fields: preamble, SFD, DA, SA, length or type of protocol data unit (PDU), upper-layer data, and the CRC.

- *802.3 MAC frame*

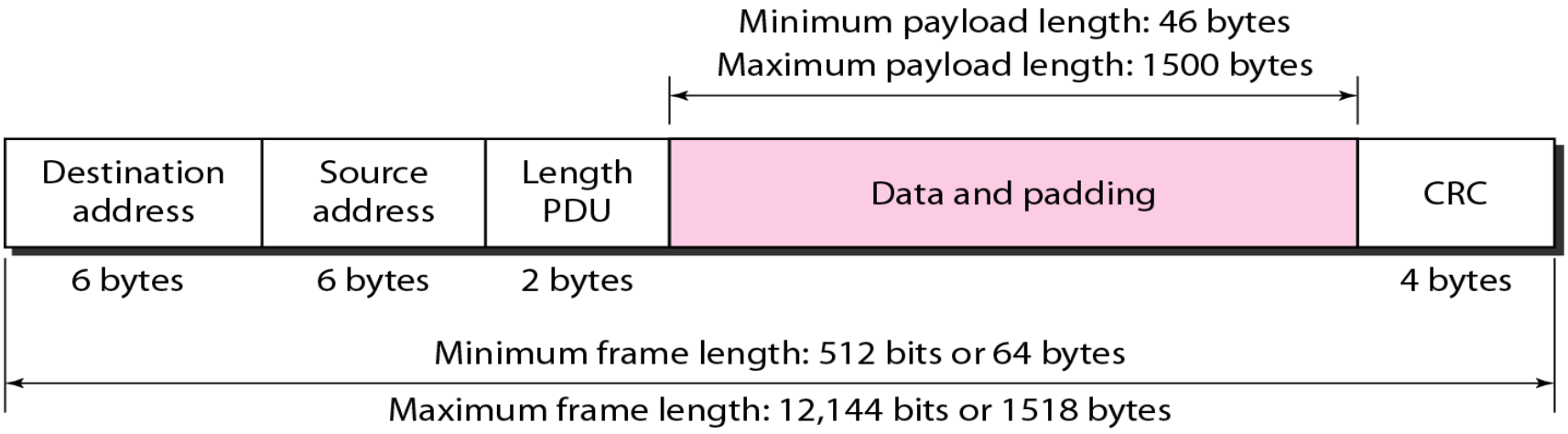
Preamble: 56 bits of alternating 1s and 0s.

SFD: Start frame delimiter, flag (10101011)



- **Preamble.** The first field of the 802.3 frame contains 7 bytes (56 bits) of alternating 0s and 1s that alerts the receiving system to the coming frame and enables it to synchronize its input timing. The pattern provides only an alert and a timing pulse.
- **Start frame delimiter (SFD).** The second field (1 byte: 10101011) signals the beginning of the frame. The SFD warns the station or stations that this is the last chance for synchronization. The last 2 bits is 11 and alerts the receiver that the next field is the destination address.
- **Destination address (DA).** The DA field is 6 bytes and contains the physical address of the destination station or stations to receive the packet.

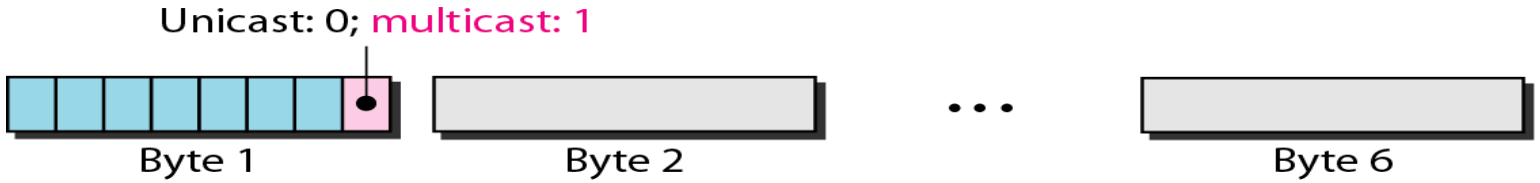
- **Source address (SA).** The SA field is also 6 bytes and contains the physical address of the sender of the packet.
- **Length or type.** This field is defined as a type field or length field. The original Ethernet used this field as the type field to define the upper-layer protocol using the MAC frame. The IEEE standard used it as the length field to define the number of bytes in the data field. Both uses are common today.
- **Data.** This field carries data encapsulated from the upper-layer protocols. It is a minimum of 46 and a maximum of 1500 bytes.
- **CRC.** The last field contains error detection information, in this case a CRC-32.
- *Frame Length*



- An Ethernet frame needs to have a minimum length of 512 bits or 64 bytes.
- If the upper-layer packet is less than 46 bytes, padding is added to make up the difference.
- The minimum length restriction is required for the correct operation of *CSMA/CD.*
- The standard defines the maximum length of a frame (without preamble and SFD field) as 1518 bytes.
- The maximum length restriction has two historical reasons.
- First, memory was very expensive when Ethernet was designed: a maximum length restriction helped to reduce the size of the buffer.
- Second, the maximum length restriction prevents one station from monopolizing the shared medium, blocking other stations that have data to send.

- *Addressing*
- The Ethernet address is 6 bytes (48 bits), normally written in hexadecimal notation, with a colon between the bytes.

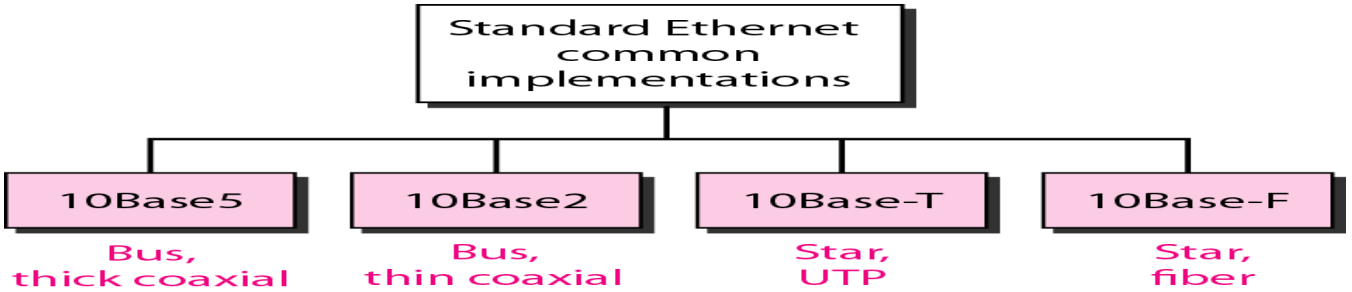
- *Unicast, Multicast, and Broadcast Addresses.* A source address is always a unicast address-the frame comes from only one station. The destination address, however, can be unicast, multicast, or broadcast.
- If the least significant bit of the first byte in a destination address is 0, the address is unicast; otherwise, it is multicast.



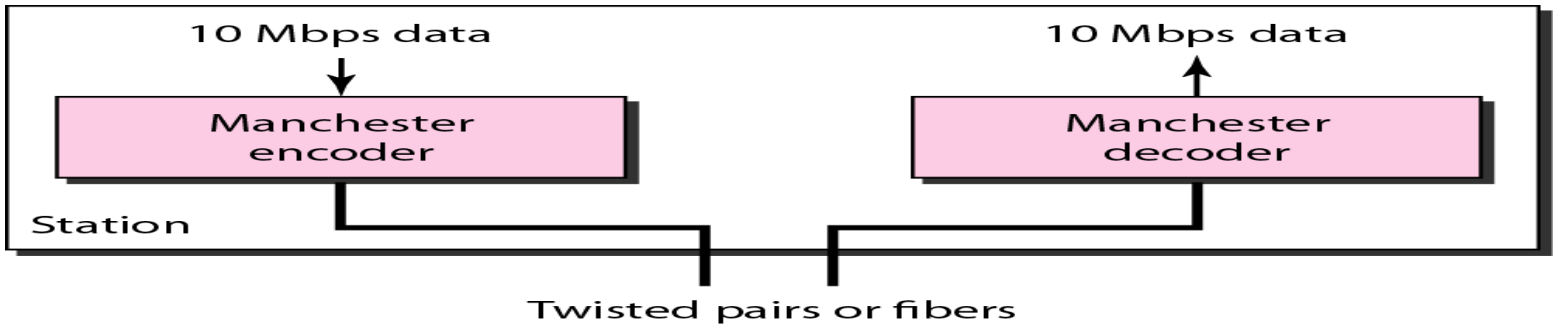
- The broadcast address is a special case of the multicast address; the recipients are all the stations on the LAN. A broadcast destination address is forty-eight 1s.

- *4A:30:10:21:10:1A*
- *47:20:1B:2E:08:EE*
- *FF:FF:FF:FF:FF:FF*

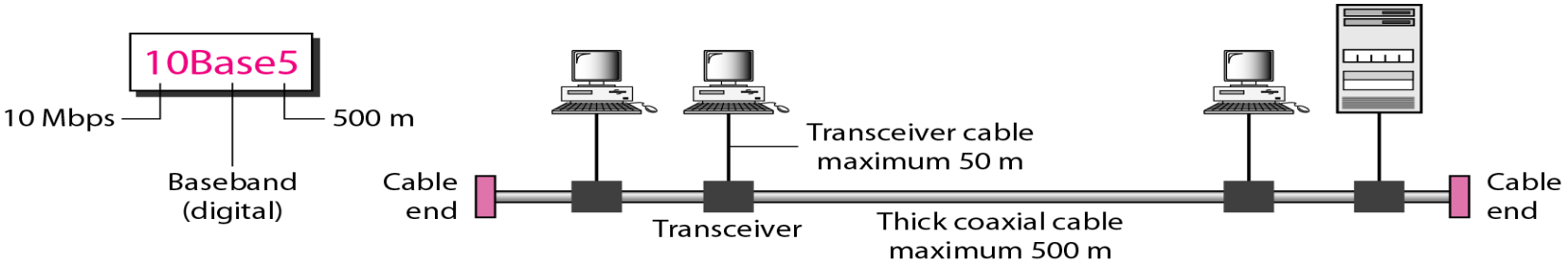
- ***Physical Layer.***
- The Standard Ethernet defines several physical layer implementations.



- *Encoding and Decoding*
- All standard implementations use digital signaling (baseband) at 10 Mbps.



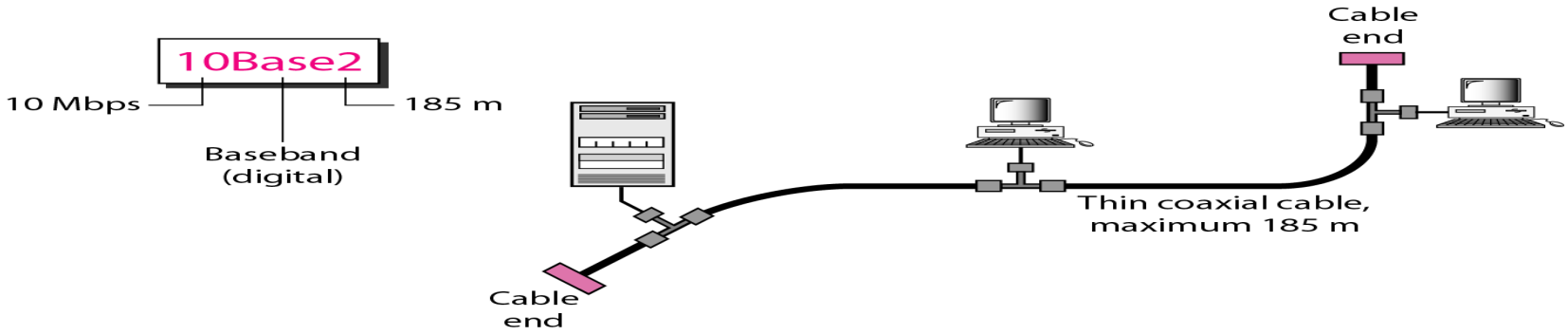
- At the sender, data are converted to a digital signal using the Manchester scheme; at the receiver, the received signal is interpreted as Manchester and decoded into data.
- *10Base5: Thick Ethernet*
- The first implementation is called **10Base5, thick Ethernet, or Thicknet.**
- 10Base5 was the first Ethernet specification to use a bus topology with an external **transceiver** (transmitter/receiver) connected via a tap to a thick coaxial cable.



- The transceiver is responsible for transmitting, receiving, and detecting collisions.
- The transceiver is connected to the station via a transceiver cable that provides separate paths for sending and receiving. This means that collision can only happen in the coaxial cable.

- If a length of more than *500* m is needed, up to five segments, each a maximum of 500-meter, can be connected using repeaters.

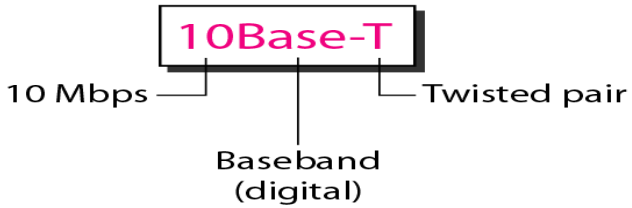
- *10Base2: Thin Ethernet*
- The second implementation is called 10Base2, **thin** Ethernet, or cheapernet.
- 10Base2 also uses a bus topology, but the cable is much thinner and more flexible. The cable can be bent to pass very close to the stations.
- The transceiver is normally part of the network interface card (NIC), which is installed inside the station.

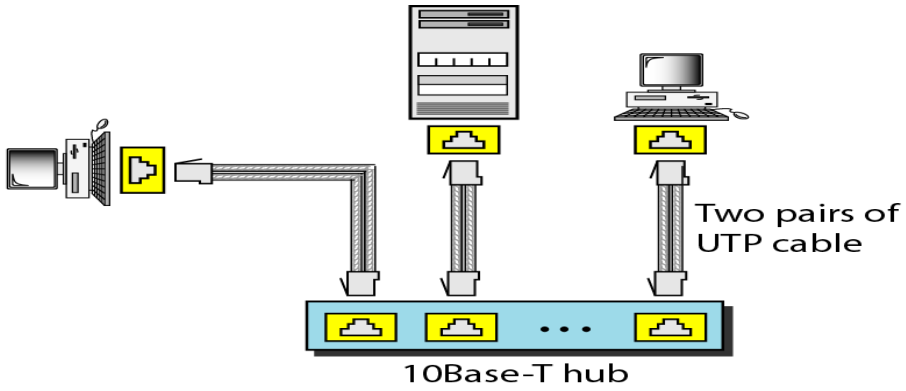


- The collision here occurs in the thin coaxial cable. This implementation is more cost effective than 10Base5 because thin coaxial cable is less expensive than thick coaxial and the tee connections are much cheaper than taps.

- Installation is simpler because the thin coaxial cable is very flexible.
- However, the length of each segment cannot exceed *185* m (close to 200 m) due to the high level of attenuation in thin coaxial cable.

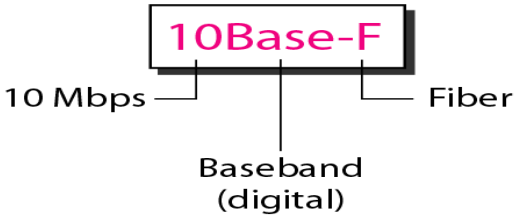
- *10Base-T: Twisted-Pair Ethernet*
- 10Base-T uses a physical star topology. The stations are connected to a hub via two pairs of twisted cable.

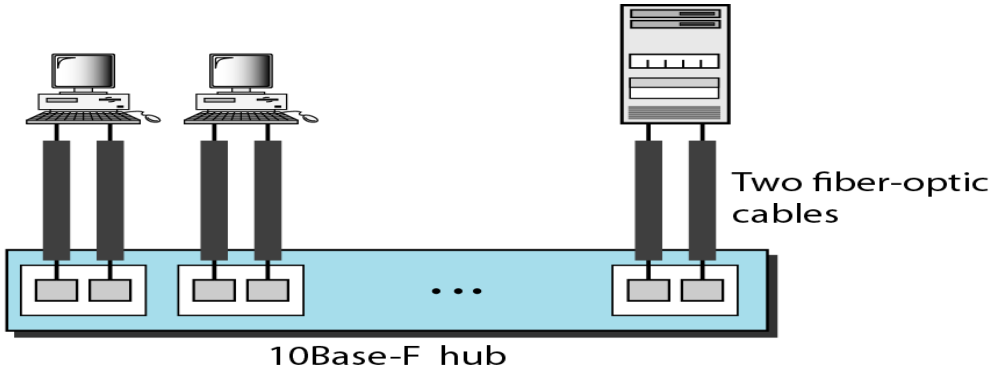




- Any collision here happens in the hub.
- Compared to 10Base5 or 10Base2, the hub actually replaces the coaxial cable as far as a collision is concerned.
- The maximum length of the twisted cable here is defined as 100 m, to minimize the effect of attenuation in the twisted cable.

- *10Base-F: Fibre Ethernet*
- 10Base-F uses a star topology to connect stations to a hub. The stations are connected to the hub using two fiber-optic cables.



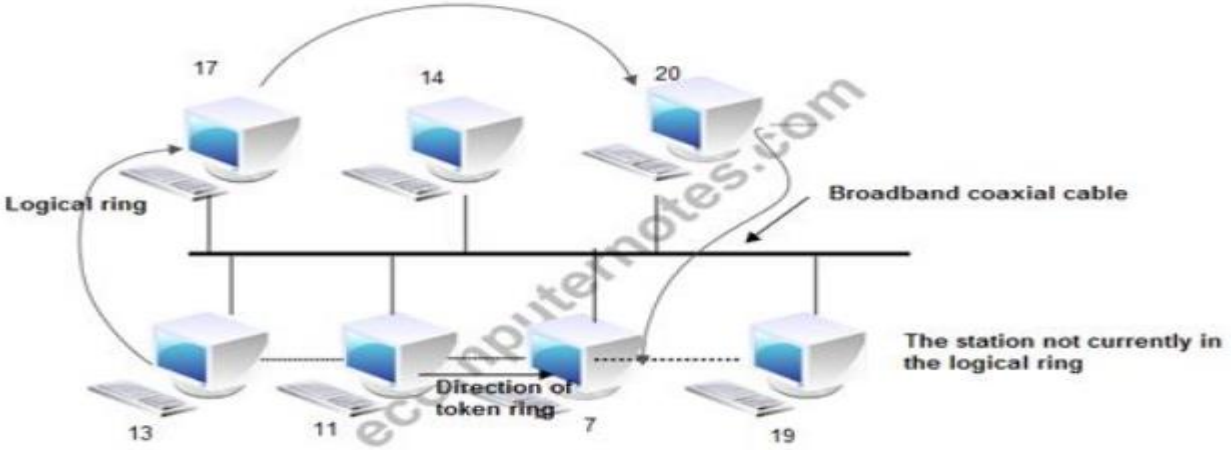


- *Summary of Standard Ethernet implementations*

Characteristics	*10Base5*	*10Base2*	*10Base-T*	*10Base-F*
Media	Thick coaxial cable	Thin coaxial cable	2 UTP	2 Fiber
Maximum length	500 m	185 m	100 m	2000 m
Line encoding	Manchester	Manchester	Manchester	Manchester

- **IEEE 802.4 Token Bus**
- The IEEE 802.4 Committee has defined **token bus** standards as broadband computer networks, as opposed to Ethernet's baseband transmission technique.
- In token bus, computer network station must have possession of a token before it can transmit on the computer network. Logically they follow token ring format.

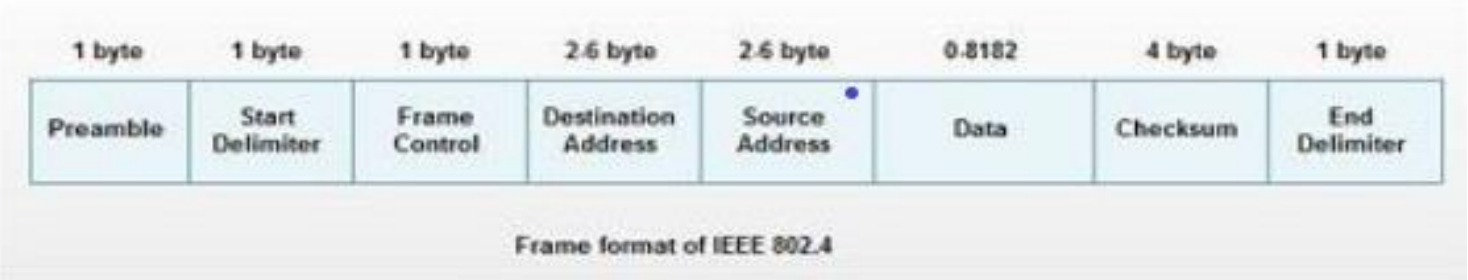
- *Physical Layer*
- Physically, the token bus is a linear or tree-shape cable to which the stations are attached.



A Token Bus

- The topology of the computer network can include groups of workstations connected by long trunk cables. These devices can branch from hubs in a star configuration, so the network has both a bus and star topology.
- The conventional 75 ohm coaxial cable used for the cable TV is used as the physical medium of the token bus.
- The different modulation schemes are used. They are, phase continuous frequency shift keying, phase coherent frequency shift keying, and the multilevel duo binary amplitude-modulated phase shift keying.
- Signal speeds in the range 1 Mbps, 5 Mbps, and 10 Mbps are achievable. The physical layer of the token bus is totally incompatible to the IEEE 802.3 standard.

- *MAC Sublayer*
- **Frame format of Token Bus**

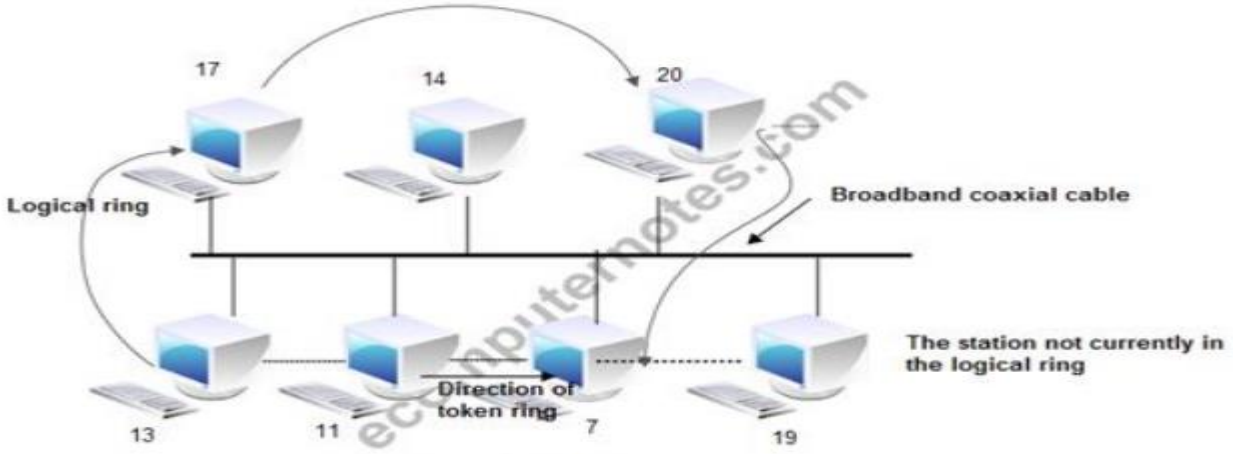


Frame format of IEEE 802.4

- **Preamble**: This. Field is at least 1 byte long. It is used for bit synchronization.
- **Start Delimiter**: This one byte field marks the beginning of frame.
- **Frame Control:** This one byte field specifies the type of frame. It distinguishes data frame from control frames. For data frames it carries frame's priority. For control frames, it specifies the frame type. The control frame types include token passing and various ring maintenance frames, including the mechanism for letting new station enter the ring, the mechanism for allowing stations to leave the ring.
- **Destination address**: It specifies 2 to 6 bytes destination address.
- **Source address**: It specifies 2 to 6 bytes source address.
- **Data**: This field may be upto 8182 bytes long when 2 bytes addresses are used & upto 8174 bytes long when 6 bytes address is used.
- **Checksum**: This 4 byte field detects transmission errors.
- **End Delimiter**: This one byte field marks the end of frame.

- The devices logically follow **token ring** format to control access to the network.

- Logically, the stations are organized into a ring.
- When the ring is initialized, stations are inserted into it in order of station address, from highest to lowest.
- Token passing is done from high to low address.
- Whenever a station acquires the token, it can transmit frames for a specific amount of time.
- If a station has no data, it passes the token immediately upon receiving it.
- The token bus defines four priority classes, 0, 2, 4, and 6 for traffic, with 0 the lowest and 6 the highest. Each station is internally divided into four substations, one at each priority level *i.e.* 0,2,4 and 6.



A Token Bus

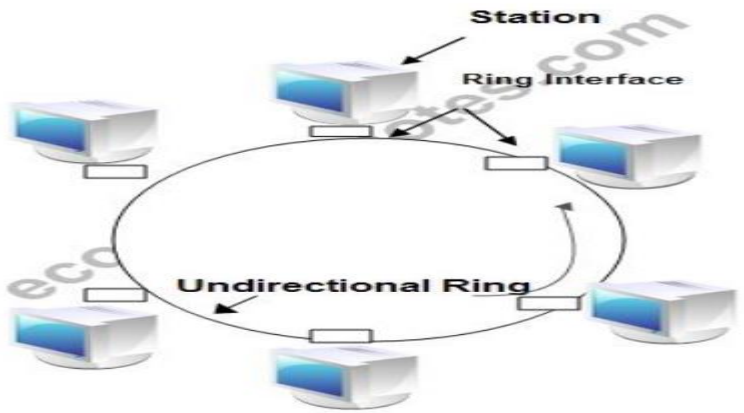
- **Frame control formats and ring maintenance**

Farme Control Field	Name	Meaning
00000000	Claim_token	Claim token during ring initialization
00000001	Solicit successor_1	Allow station to enter the ring
00000010	Solicit successor_2	Allow stations to enter the ring
00000011	Who_follows	Recover from lost token.
00000100	Resolve_contention	Used when multiple stations want to enter.
00001000	Token	Pass the token
00001100	Set successor	Allow station to leave the ring.

- Control frames are frames used for controlling and maintaining the logical ring. There are different bit patterns used in the frame control field to define different types of control frames.
- ***Adding* a *new station.*** To add a new station to the logical ring, the station which currently holds the token broadcasts periodically a special control frame called a Solicit-successor frame. This frame contains the address of the sending station and its successor. This intimates the stations within the specified range to join the ring. Hence, the ring remains sorted.

- ***Leaving the logical ring.*** A station can simply leave the ring by sending its successor address to its predecessor and asks it to set this as its successor. To perform this, it passes a special control frame called set successor. The frame contains the address of predecessor as destination address and the address of the successor as sending address.
- ***Ring initialization*** When the network is powered on, initially, all the stations are off. As soon as the first station is initiated, it checks the channel for the presence of any contenders, by sending a special control frame called claim-token. If there is no response, it generates a new token; thereby, creating a new logical ring with a single station. Periodically it transmits solicit successor tokens, hence, new stations are added frequently, making 'the logical ring grow.
- ***Failure* of *token holder*** If the token holder fails, none of the other stations can get the token. Each station in the ring has a timer internally. They wait until the timer expires and transmits a claim-token frame, and the network is reestablished.

- **IEEE 802.5 Token Ring**
- Token ring is the IEEE 802.5 standard for a token-passing ring in Communication networks.
- A ring consists of a collection of ring interfaces connected by point-to-point lines *i.e.* ring interface of one station is connected to the ring interfaces of its left station as well as right station.
- Internally, signals travel around the Communication network from one station to the next in a ring.

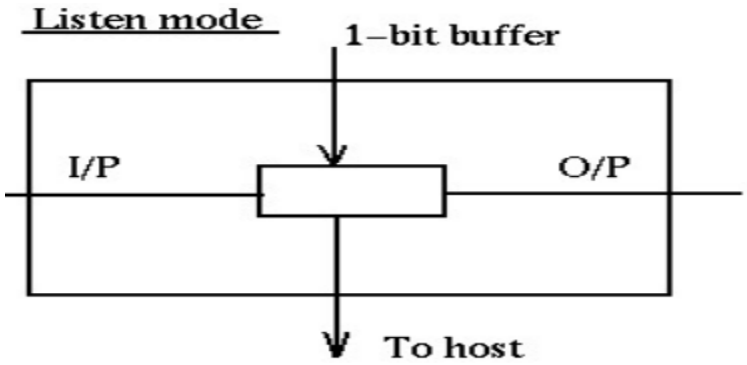


A Ring Network

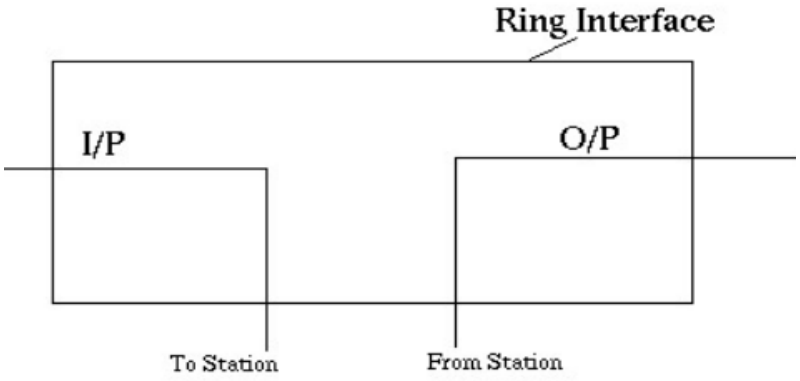
- *Physical layer*
- The cable recommended for a token ring by IEEE 802.5 contains two pairs of twisted-cables covered by a shield.

- Signal speed of this media is 1 Mbps or 4 Mbps. But IBM released a Token ring version that can operate at a speed of 16Mbps.
- Differential ‘Manchester’ encoding scheme is used for encoding the digital data.
- Any single point failure on the cable may cause the ring to disappear instantly.

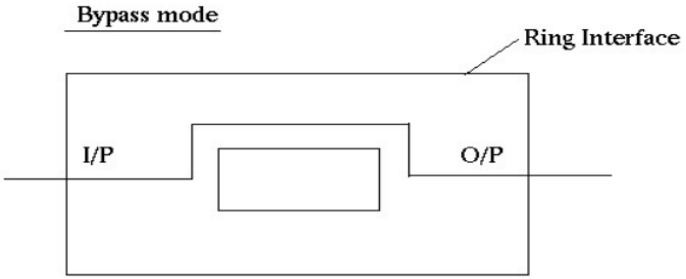
- *Ring interface Modes of Operation*
- **Listen Mode:** In this mode the node listens to the data and transmits the data to the next node. In this mode there is a one-bit delay associated with the transmission.



- **Transmit Mode:** In this mode the node just discards the any data and puts the data onto the network.

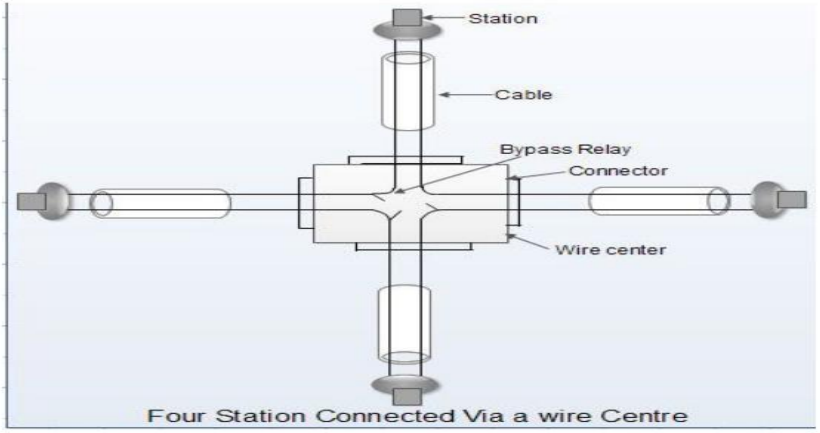


- **By-pass Mode:** In this mode reached when the node is down. Any data is just bypassed. There is no one-bit delay in this mode.



- If the cable breaks, the entire ring network goes down. This can completely stop the propagation of token in the ring.

- This problem can be solved by using wire center as shown in fig.



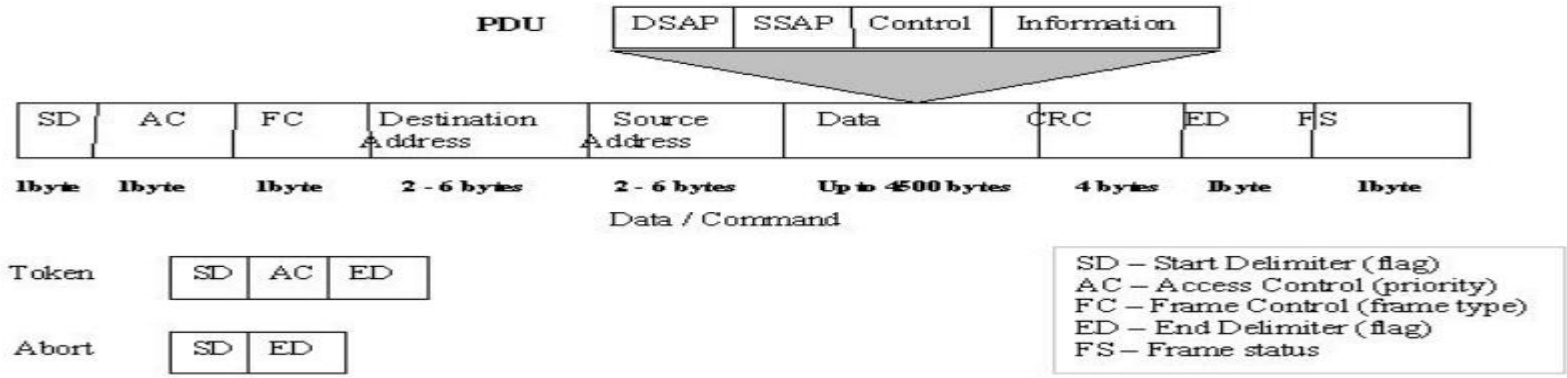
Four Station Connected Via a wire Centre

- Each station is connected to wire center by a cable containing two twisted pairs, one for data to station and one for data from the station.
- Inside the wire center are bypass relays that are energized by the current from the stations.
- If the ring breaks or a station goes down loss of drive current will release the relay and bypass the station.

- *MAC Sublayer*
- Token-passing networks move a small frame, called a token, around the network. Possession of the token grants the right to transmit.

- If a node receiving the token in order to transmit data, it seizes the token, alters 1 bit of the token (which turns the token into a start-of-frame sequence), appends the information that it wants to transmit, and sends this information to the next station on the ring.
- Since only one station can possess the token and transmit data at any given time, there are no collisions.
- Since the token rotates in the ring it is guaranteed that every node gets the token with in some specified time. So there is an upper bound on the time of waiting to grab the token so that starvation is avoided.
- There is also an upper limit of 250 on the number of nodes in the network.
- A station may hold the token for the token-holding time. which is 10 ms unless the installation sets a different value.
- If there is enough time left after the first frame has been transmitted to send more frames, then these frames may be sent as well. After all pending frames have been transmitted or the transmission frame would exceed the token-holding time, the station regenerates the 3-byte token frame and puts it back on the ring.

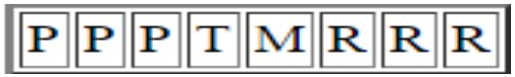
- **Frame format**



- **Start Delimiter.** The first field of the data/command frame, SD, is one byte long and is used to alert the receiving station to the arrival of a frame as well as to allow it to synchronize its retrieval timing.
- **Access control.** The AC field is one byte long and includes four subfields. The first three bits are the priority field. The fourth bit is called the token bit.
- **Frame Control.** The FC field is one byte long and contains two fields. The first is a one-bit field used to indicate the type of information contained in the PDU.
- **Destination Address**. The two-to-six byte DA field contains the physical address of the frame's next destination.

- **Source Address.** The SA field is also two to six bytes long and contains the physical address of the sending station.
- **Data.** The sixth field, data, is allotted 4500 bytes and contains the PDU.
- **CRC.** The CRC field is four bytes long and contains a CRC-32 error detection sequence.
- **End delimiter.** The ED is a second flag field of one byte and indicates the end of the sender's data and control information.
- **Frame status.** The last byte of the frame is the FS field. It can be set by the receiver to indicate that the frame has been read, or by the monitor to indicate that the frame has already been around the ring.
- **Token frame.** Token frame is a reservation frame, it has three fields: SD, AC, and ED. The SD indicates that a frame is coming. The AC indicates that the frame is a token and includes the priority and reservation fields. The ED indicates the end of the frame.
- **Abort frame.** An abort frame carries no information. It has only two fields: SD, and ED. It can be generated by the sender to stop its own transmission.

- Access Control Format:



- **T=Token**
- T = “0” for Token,
  T = “1” for Frame.
- When a station with a Frame to transmit detects a token which has a priority equal to or less than the Frame to be transmitted, it may change the token to a start-of-frame sequence and transmit the Frame.
- **Priority (P)**
  Priority Bits indicate tokens priority, and therefore, which stations are allowed to use it. Station can transmit if its priority as at least as high as that of the token.
- **M = Monitor**
  The monitor bit is used to prevent a token whose priority is greater than 0 or any frame from continuously circulating on the ring. If an active monitor detects a frame or a high priority token with the monitor bit equal to 1, the frame or token is aborted. This bit shall be transmitted as 0 in all frame and tokens. The active monitor inspects and modifies this bit. All other stations shall repeat this bit as received.

- **R = Reserved bits.** The reserved bits allow station with high priority Frames to request that the next token be issued at the requested priority.
- **Frame Status:**
- It contains the A and C bits.
- A bit set to 1: destination recognized the packet.
  C bit set to 1: destination accepted the packet.
- This arrangement provides an automatic acknowledgement for each frame. The A and C bits are present twice in the Frame Status to increase reliability in as much as they are not covered by the checksum.

- **Token Ring Maintenance**
- **Monitor stations** Each ring has a monitor station that performs monitoring the ring. It also takes appropriate action when the ring is broken. It also clears the ring when there are garbled frames on the ring. If the monitor station fails, some other station is chosen by means of a special contention protocol.

- **Ring Initialization** When the network is powered up, initially there is no monitor. The first station on the network transmits a claim-token frame. If there is nothing else, a claim-token frame is setup the first station becomes the token owner as well as the monitor.
- **Lost tokens** The monitor station has an internal timer that is set for the longest possible time interval without a token. If the token is not produced within the period, the monitor station clears the ring and issues a new token.
- **Orphan frames** Stations crashed after transmitting a short frame form orphan frames. The transmitted frame simply circulates around the ring. The monitor station sets the monitor bit in the access control byte whenever a frame passes through it. If a frame already contains this bit set that means the frame is passing through the monitor for the second time. The monitor immediately removes it from the ring.