UNIT-2

The Data Link Control (DLC) deals with procedures for communication between two adjacent nodes no matter whether the link is dedicated or broadcast. Data link control functions include framing, flow control, and error control.

Framing

The data link layer needs to pack bits into frames, so that each frame is distinguishable from another.

Framing in the data link layer divides a message from one source to a destination by adding a sender address and a destination address. The destination address defines where the packet is to go; the sender address helps the recipient acknowledge the receipt.

Fixed-Size Framing

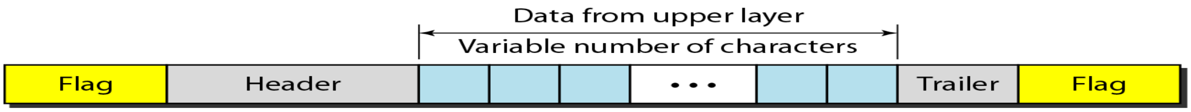
In fixed-size framing, there is no need for defining the boundaries of the frames; the size itself can be used as a delimiter.
An example of this type of framing is the ATM wide-area network, which uses frames of fixed size called cells.

Variable-Size Framing

Variable-size framing, we need a way to define the end of the frame and the beginning of the next. Historically, two approaches were used for this purpose: a character-oriented approach and a bit-oriented approach

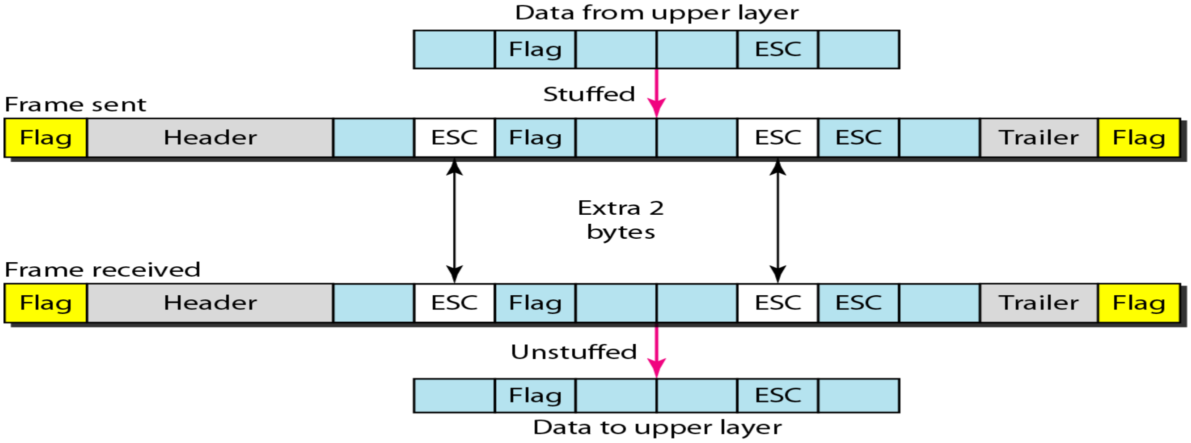
Character Oriented

In a character-oriented protocol, data to be carried are 8-bit characters from a coding system such as ASCII (see Appendix A). The header, which normally carries the source and destination addresses and other control information, and the trailer, which carries error detection or error correction redundant bits, are also multiples of 8 bits. To separate one frame from the next, an 8-bit (I-byte) flag is added at the beginning and the end of a frame. The flag, composed of protocol-dependent special characters, signals the start or end of a frame.



Any pattern used for the flag could also be part of the information. If this happens, the receiver, when it encounters this pattern in the middle of the data, thinks it has reached the end of the frame. To fix this problem, a byte-stuffing strategy was added to character-oriented framing.

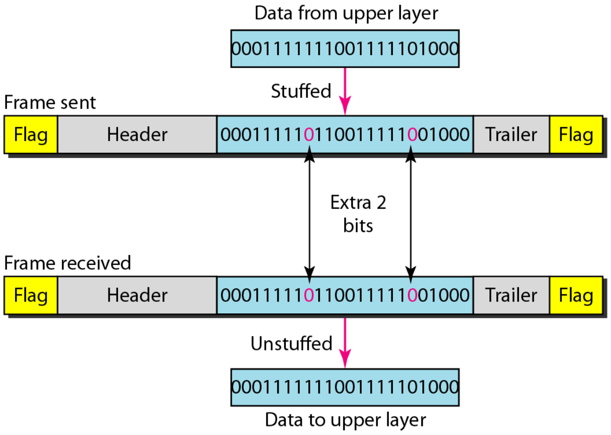
In byte stuffing (or character stuffing), a special byte is added to the data section of the frame when there is a character with the same pattern as the flag. The data section is stuffed with an extra byte. This byte is usually called the escape character (ESC), which has a predefined bit pattern. Whenever the receiver encounters the ESC character, it removes it from the data section and treats the next character as data, not a delimiting flag.



Byte stuffing is the process of adding 1 extra byte wheneverthere is a flag or escape character in the text.

Bit-Oriented Protocols

In a bit-oriented protocol, the data section of a frame is a sequence of bits to be interpreted by the upper layer as text, graphic, audio, video, and so on. However, in addition to headers (and possible trailers), we still need a delimiter to separate one frame from the other. Most protocols use a special 8-bit pattern flag 01111110 as the delimiter to define the beginning and the end of the frame.



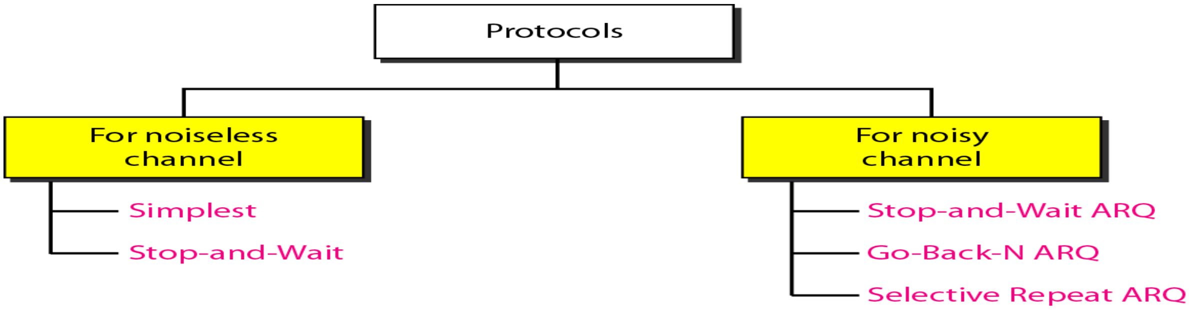
Bit stuffing is the process of adding one extra 0 whenever five consecutive 18 follow a 0 in the data, so that the receiver does not mistake the pattern 0111110 for a flag.

This means that if the flaglike pattern 01111110 appears in the data, it will change to 011111010 (stuffed) and is not mistaken as a flag by the receiver. The real flag 01111110 is not stuffed by the sender and is recognized by the receiver

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 flow of data must not be allowed to overwhelm the receiver. Any receiving device has a limited speed at which it can process incoming data and a limited amount of memory in which to store incoming data. 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.

Flow control refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgment.

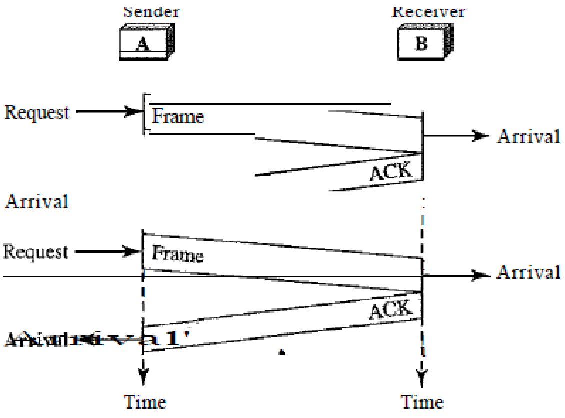


Simplest Protocol

It is a unidirectional protocol in which data frames are traveling in only one direction from the sender to receiver. We assume that the receiver can immediately handle any frame it receives with a processing time that is small enough to be negligible. The data link layer of the receiver immediately removes the header from the frame and hands the data packet to its network layer, which can also accept the packet immediately. In other words, the receiver can never be overwhelmed with incoming frames.

Stop and Wait Protocol

The Sender sends one frame, stops until it receives confirmation from the receiver (okay to go ahead), and then sends the next frame. We still have unidirectional communication for data frames, but auxiliary ACK frames (simple tokens of acknowledgment) travel from the other direction.



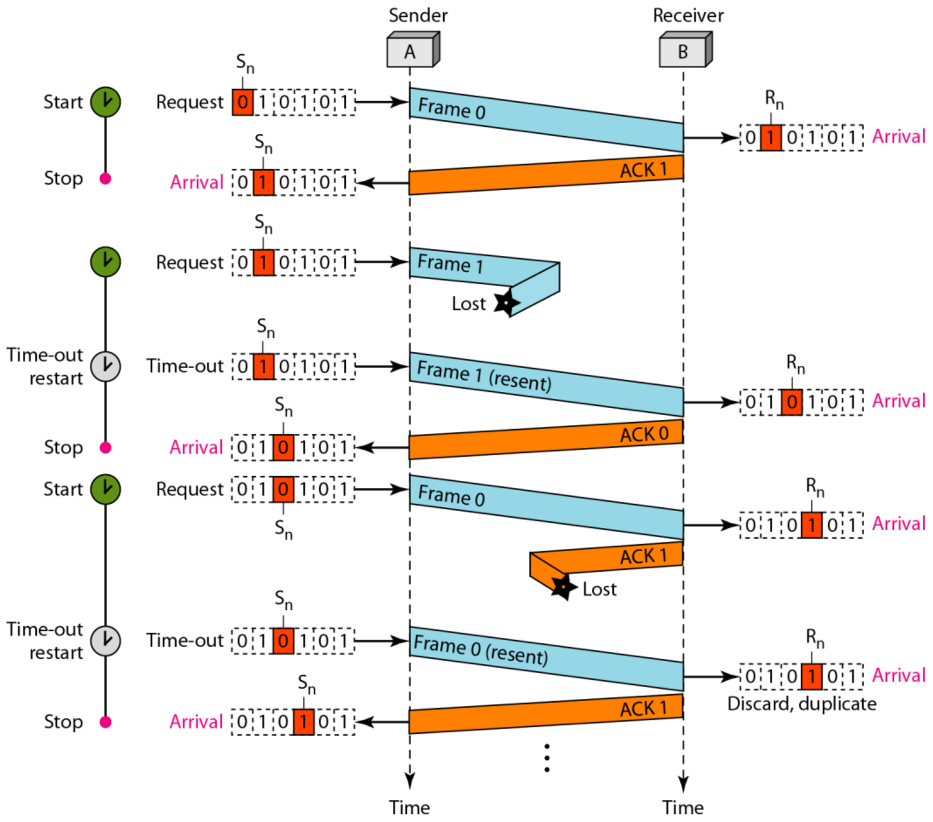
Stop and wait Automatic Repeat Request (ARQ)

Error correction in Stop-and-Wait ARQ is done by keeping a copy of the sent frame and retransmitting of the frame when the timer expires.

In this sequence number is based on modulo 2 arithmetic .Frame having numer in alternate 010101….. In stop and wait ARQ, sequence no. define the frame to be sent and the acknowledgement no. of ACK frame define the next frame to be expected.

Sender control variable (Sn) storing the sequence number of next frame to be sent .
e.g. if Sn store 0 , we send frame 0

Receiver control variable(Rn) storing the sequence number of next frame to be expected .



Stop-and-Wait ARQ. Frame 0 is sent and acknowledged. Frame 1 is lost and resent after the time-out. The resent frame 1 is acknowledged and the timer stops. Frame 0 is sent and acknowledged, but the acknowledgment is lost. The sender has no idea if the frame or the acknowledgment is lost, so after the time-out, it resends frame 0, which is acknowledged.

Go Back – NARQ (Sliding Window protocol)

The first is called Go-Back-N Automatic Repeat Request (the rationale for the name will become clear later). In this protocol we can send several frames before receiving acknowledgments; we keep a copy of these frames until the acknowledgments arrive.

Sequence Numbers

If the header of the frame allows *m* bits for the sequence number, the sequence numbers range from 0 to 2*m* - 1. For example, if *m* is 4, the only sequence numbers are 0 through 15 inclusive. However, we can repeat the sequence. So the sequence numbers are

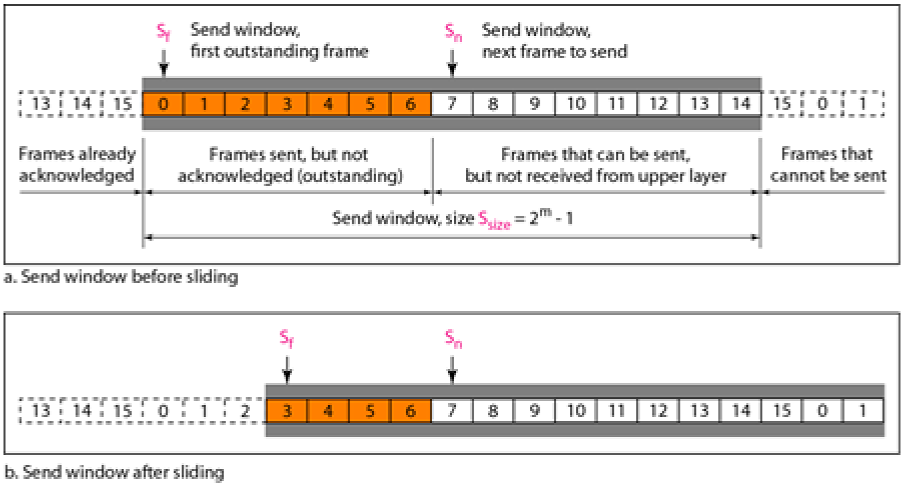
0, 1,2,3,4,5,6, 7,8,9, 10, 11, 12, 13, 14, 15,0, 1,2,3,4,5,6,7,8,9,10, 11, ...

Sliding Window

The sliding window is an abstract concept that defines the range of sequence numbers that is the concern of the sender and receiver.

The maximum size of the window is $\boldsymbol{2^m - 1}$.

Figure 11.12 ***Send window for Go-Back-N ARQ***



S_f
Send window, first outstanding frame
S_n
Send window, next frame to send
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1
Frames already acknowledged
Frames sent, but not acknowledged (outstanding)
Frames that can be sent, but not received from upper layer
Frames that cannot be sent
Send window, size $S_{size} = 2^m - 1$
S_f
S_n
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1

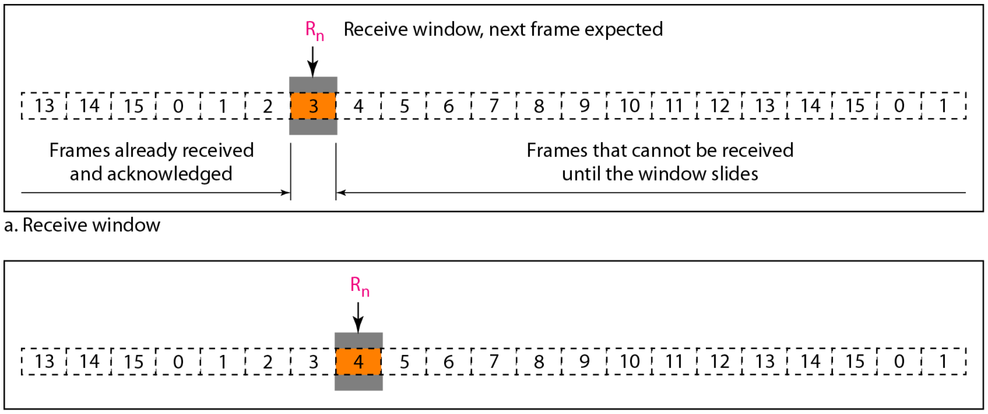
a. Send window before sliding

b. Send window after sliding

11.39

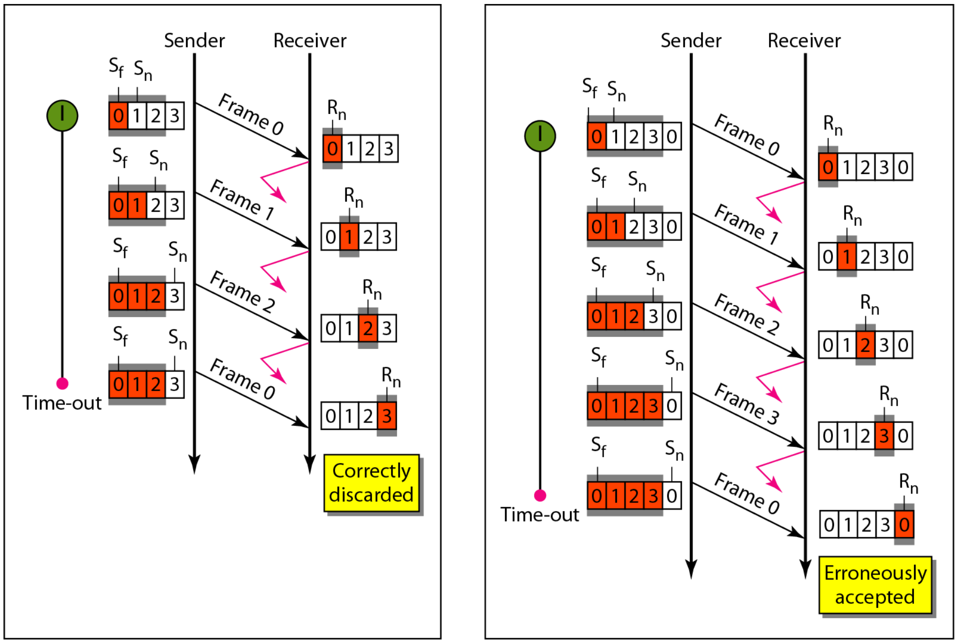
The send window is an abstract concept defining an imaginary box of size $2^m - 1$ with three variables: S_f, S_n, and S_{size}.

The send window can slide oneor more slots when a valid acknowledgment arrives.



a. Receive window

b. Window after sliding



a. Window size < 2^m

b. Window size = 2^m

In Go-Back-N ARQ, the size of the send window must be less than 2^m; the size of the receiver window is always 1.

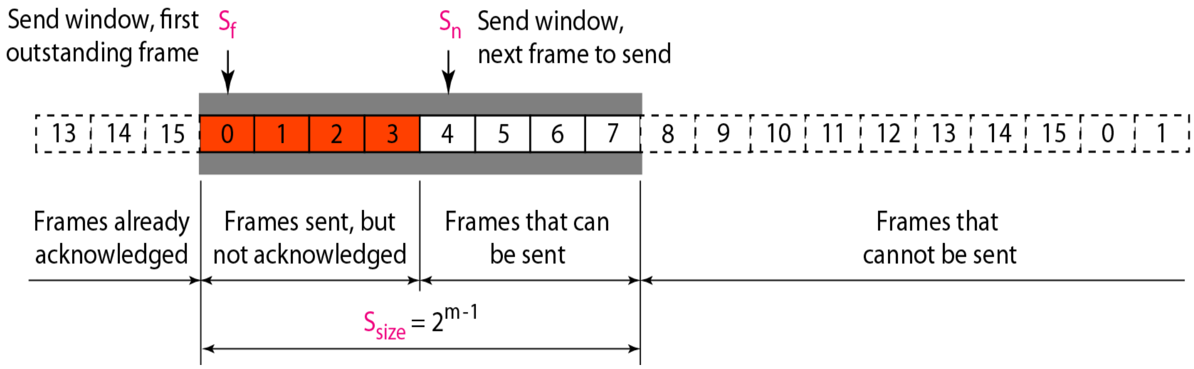
Stop-and-Wait ARQ is a special case of Go-Back-N ARQ in which the size of the send window is 1.

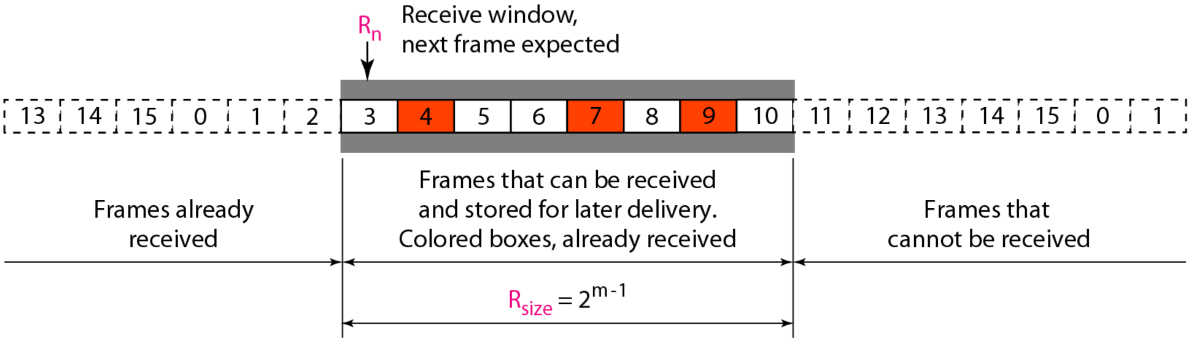
Selective Repeat ARQ

Go-Back-N ARQ simplifies the process at the receiver site. The receiver keeps track of only one variable, and there is no need to buffer out-of-order frames; they are simply discarded. However, this protocol is very inefficient for a noisy link. There is another mechanism that does not resend *N* frames when just one frame is damaged. Only the damaged frame is resent. This mechanism is called Selective Repeat ARQ. It is more efficient for noisy links

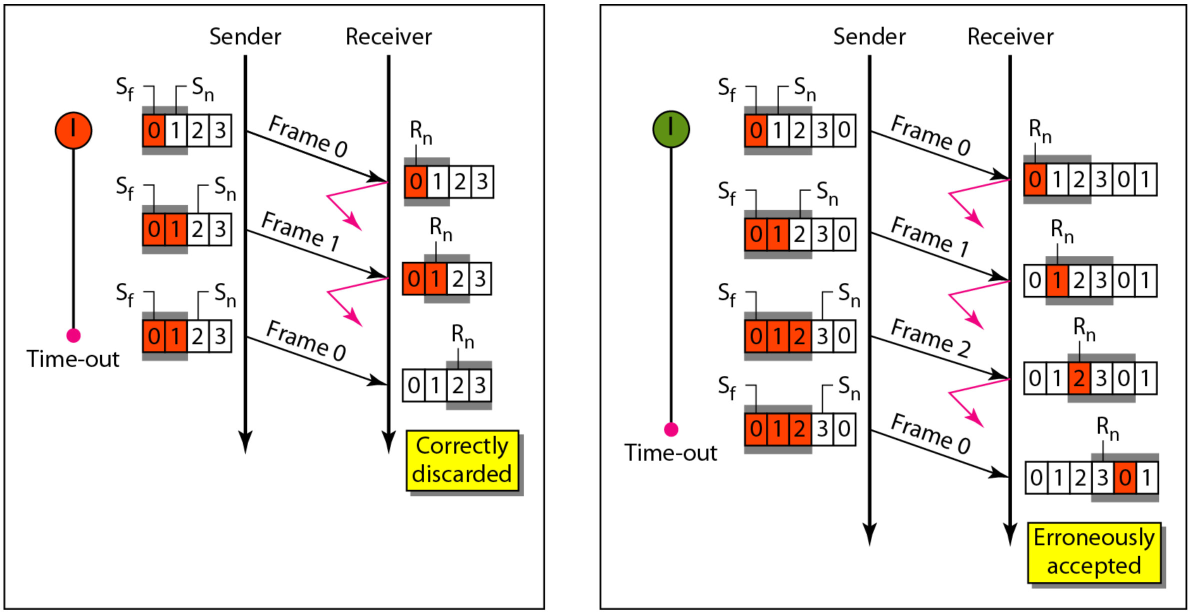
Window Size

In Selective Repeat ARQ, the size of the sender and receiver window must be at most one-half of 2^m


Send window, first outstanding frame
S_f
S_n
Send window, next frame to send
13
14
15
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
0
1
Frames already acknowledged
Frames sent, but not acknowledged
Frames that can be sent
Frames that cannot be sent
$S_{size} = 2^{m-1}$



Rn
Receive window, next frame expected
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1
Frames already received
Frames that can be received and stored for later delivery. Colored boxes, already received
Frames that cannot be received
Rsize = 2m-1



a. Window size = 2^{m-1}

b. Window size > 2^{m-1}

In Selective Repeat ARQ, the size of the sender and receiver window must be at most one-half of 2^m.

Error Control

Error control is both error detection and error correction. 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.

In the data link layer, the term *error control* means error detection and retransmission.

Error Detection and Correction

Data can be corrupted during transmission. Some applications require that errors be detected and corrected. Some applications can tolerate a small level of error. For example, random errors in audio or video transmissions may be tolerable, but when we transfer text, we expect a very high level of accuracy.

Types of errors

If the signal is carrying binary encoded data, such changes can alter the meaning of the data. These errors can be divided into two types: Single-bit error and Burst error.

Single-bit Error

The term single-bit error means that only one bit of given data unit (such as a byte, character, or data unit) is changed from 1 to 0 or from 0 to 1 as shown in Fig. 3.2.1.

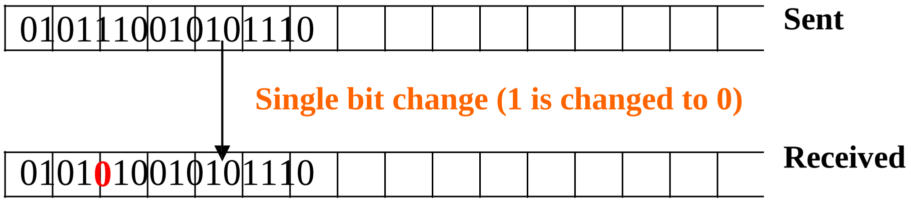
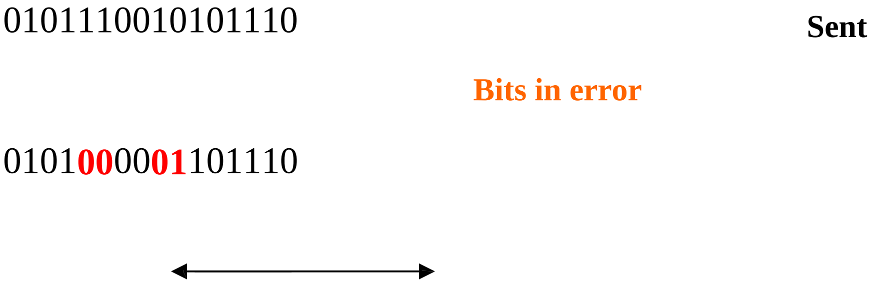


Figure 3.2.1 Single bit error

Burst Error

The term burst error means that two or more bits in the data unit have changed from 0 to 1 or vice-versa. Note that burst error doesn’t necessary means that error occurs in consecutive bits. The length of the burst error is measured from the first corrupted bit to the last corrupted bit. Some bits in between may not be corrupted.



Detection Versus Correction

The correction of errors is more difficult than the detection. 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.
In error correction, we need to know the exact number of bits that are corrupted and more importantly, their location in the message

Hamming Distance

The Hamming distance between two words (of the same size) is the number of differences between the corresponding bits. We show the Hamming distance between two words *x* and *y* as *d(x, y).* The Hamming distance can easily be found if we apply the XOR operation (ffi) on the two words and count the number of Is in the result.

Let us find the Hamming distance between two pairs of words.
1. The Hamming distance *d(000,* 011) is 2 because 000 ffi 011 is 011 (two 1s).
2. The Hamming distance *d(10101,* 11110) is 3 because 10101 ffi11110 is 01011 (three 1s).

3.2.3 Error Detecting Codes

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

- Simple Parity check
- Two-dimensional Parity check
- Checksum
- Cyclic redundancy check

Redundancy

To detect or correct errors, we need to send extra (redundant) bits with data .

Redundant bits are extra binary bits that are generated and added to the information-carrying bits of data transfer to ensure that no bits were lost during the data transfer. A parity bit is a bit appended to a data of binary bits to ensure that the total number of 1's in the data are even or odd.

Simple Parity Checking or One-dimension Parity Check

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

Blocks of data from the source are subjected to a check bit or *Parity bit* generator form, where a parity of 1 is added to the block if it contains an odd number of 1's (ON bits) and 0 is added if it contains an even number of 1's. At the receiving end the parity bit is computed from the received data bits and compared with the received parity bit, as shown in Fig. 3.2.3. This scheme makes the total number of 1's even, that is why it is called *even parity checking*.

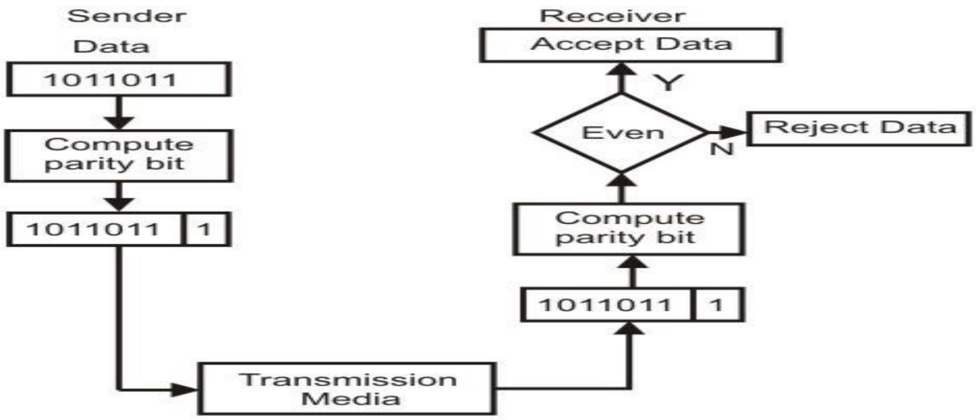


Figure 3.2.3 Even-parity checking scheme

Two dimension Parity Check

Performance can be improved by using two-dimensional parity check, which organizes the block of bits in the form of a table. Parity check bits are calculated for each row, which is equivalent to a simple parity check bit. Parity check bits are also calculated for all columns then both are sent along with the data. At the receiving end these are compared with the parity bits calculated on the received data.

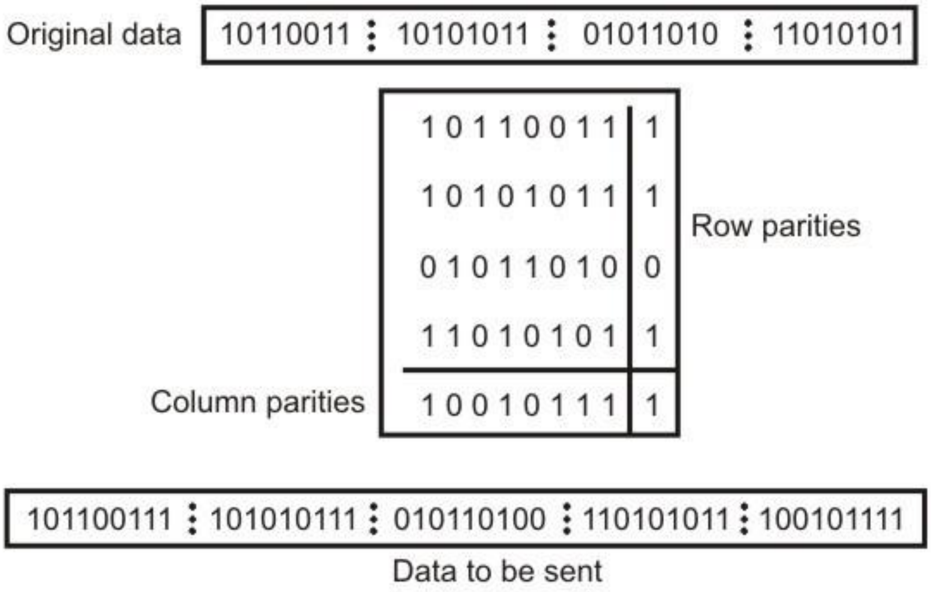


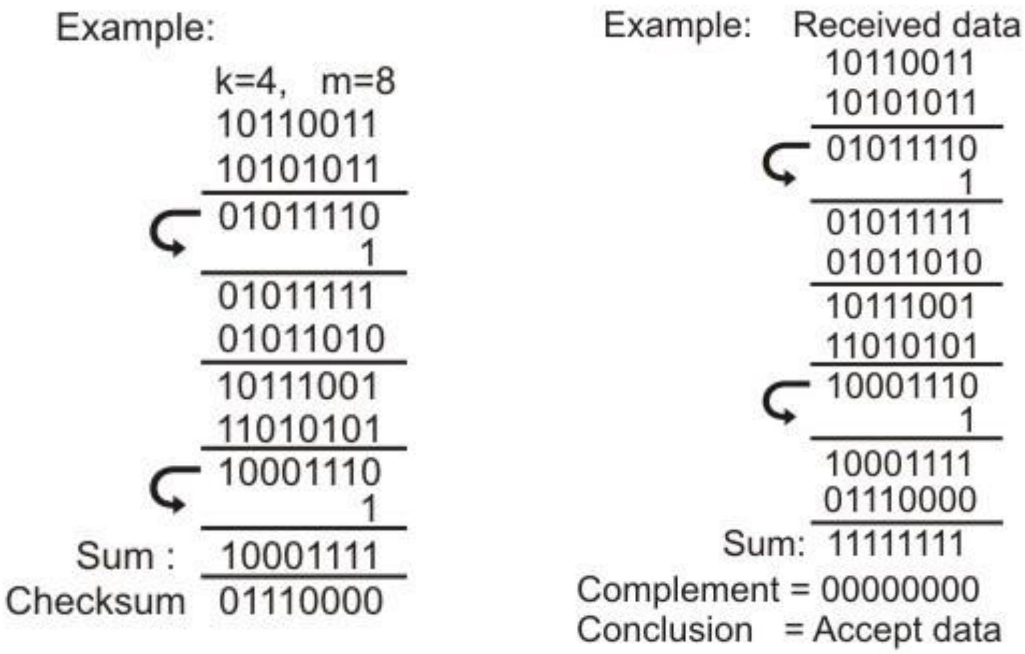
Figure 3.2.4 Two-dimension Parity Checking

Checksum

In checksum error detection scheme, the data is divided into k segments each of m bits. In the sender's end the segments are added using 1's complement arithmetic to get the sum. The sum is complemented to get the checksum. The checksum segment is sent along with the data segments as shown in Fig. 3.2.5 (a). At the receiver's end, all received segments are added using 1's complement arithmetic to get the sum. The sum is complemented. If the result is zero, the received data is accepted; otherwise discarded,

Performance

The checksum detects all errors involving an odd number of bits. It also detects most errors involving even number of bits.



(a) (b)

Figure 3.2.5 (a) Sender's end for the calculation of the checksum, (b) Receiving end for checking the checksum

Cyclic Redundancy Checks(CRC)

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

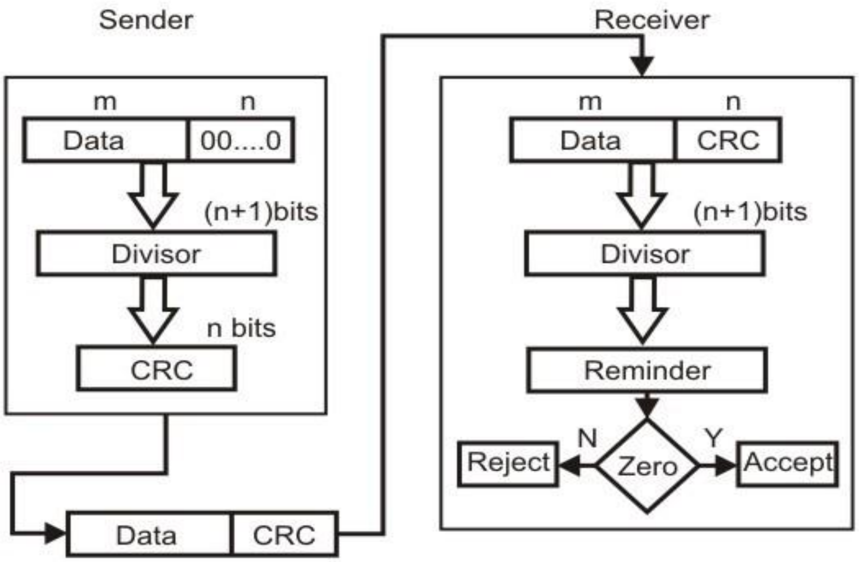
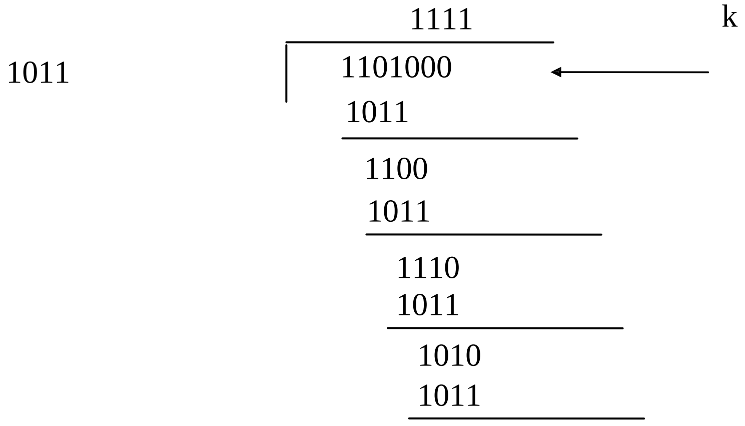


Figure 3.2.6 Basic scheme for Cyclic Redundancy Checking

This mathematical operation performed is illustrated in Fig. 3.2.7 by dividing a sample 4-bit number by the coefficient of the generator polynomial x^3+x+1, which is 1011, using the modulo-2 arithmetic. Modulo-2 arithmetic is a binary addition process without any carry over, which is just the Exclusive-OR operation. Consider the case where k=1101. Hence we have to divide 1101000 (i.e. *k* appended by 3 zeros) by 1011, which produces the remainder r=001, so that the bit frame *(k+r)* =1101001 is actually being transmitted through the communication channel. At the receiving end, if the received number, i.e., 1101001 is divided by the same generator polynomial 1011 to get the remainder as 000, it can be assumed that the data is free of errors.



Error Correcting Codes

The techniques that we have discussed so far can detect errors, but do not correct them. **Error Correction** can be handled in two ways.

- One is when an error is discovered; the receiver can have the sender retransmit the entire data unit. This is known as **backward error correction**.
- In the other, receiver can use an error-correcting code, which automatically corrects certain errors. This is known as **forward error correction**.

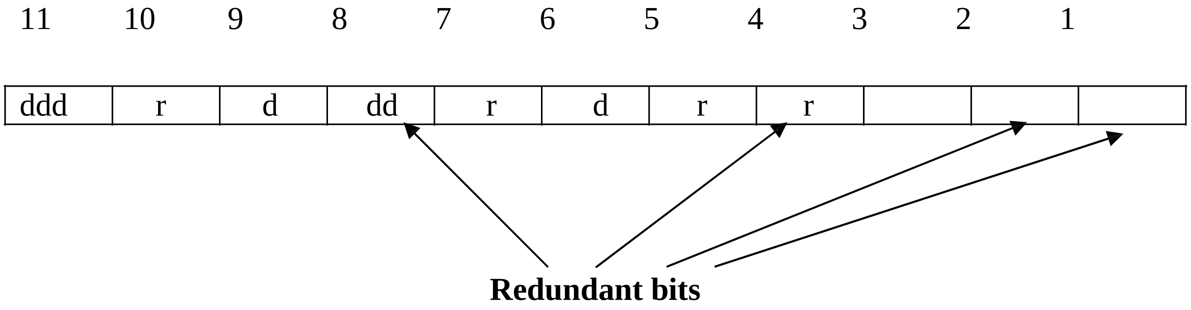
Single-bit error correction (Forward Error Correction)

To calculate the numbers of redundant bits (r) required to correct d data bits, let us find out the relationship between the two. So we have (d+r) as the total number of bits, which are to be transmitted; then r must be able to indicate at least d+r+1 different values. Of these, one value means no error, and remaining d+r values indicate error location of error in each of d+r locations. So, d+r+1 states must be distinguishable by r bits, and r bits can indicates 2^r states. Hence, 2^r must be greater than d+r+1.

$$\mathbf{2^r >= d+r+1}$$

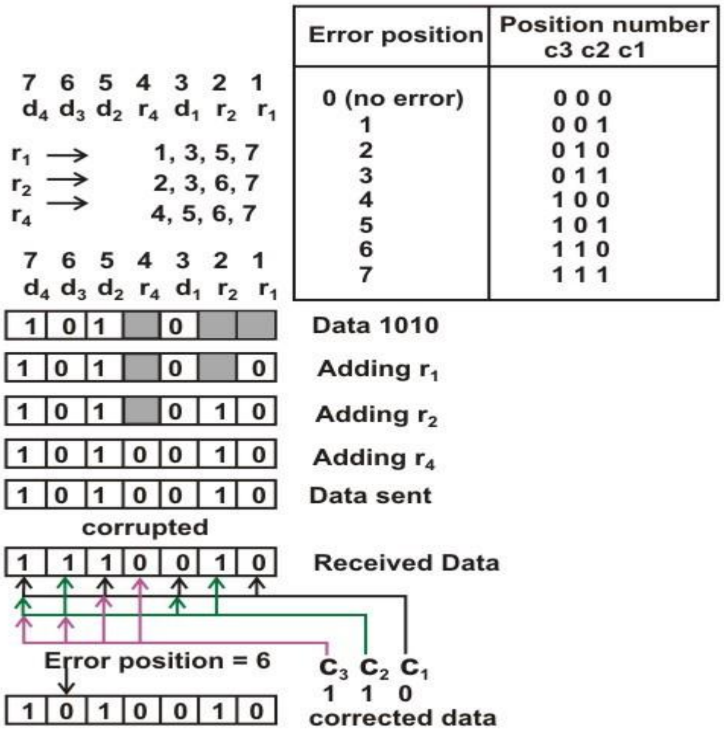
The value of r must be determined by putting in the value of d in the relation. For example, if d is 7, then the smallest value of r that satisfies the above relation is 4. So the total bits, which are to be transmitted is 11 bits (d + r = 7 + 4 =11).

The solution or coding scheme he developed is commonly known as Hamming Code. Hamming code can be applied to data units of any length and uses the relationship between the data bits and redundant bits as discussed.



11	10	9	8	7	6	5	4	3	2	1
ddd	r	d	dd	r	d	r	r			

Figure 3.2.8 Positions of redundancy bits in hamming code



Error position	Position number c3 c2 c1
0 (no error)	0 0 0
1	0 0 1
2	0 1 0
3	0 1 1
4	1 0 0
5	1 0 1
6	1 1 0
7	1 1 1

Figure 3.2.9 Use of Hamming code for error correction for a 4-bit data

Figure 3.2.9 shows how hamming code is used for correction for 4-bit numbers ($d_4d_3d_2d_1$) with the help of three redundant bits ($r_3r_2r_1$). For the example data 1010, first r_1(0) is calculated considering the parity of the bit positions, 1, 3, 5 and 7. Then the parity bits r_2 is calculated considering bit positions 2, 3, 6 and 7. Finally, the parity bits $r_4$is calculated considering bit positions 4, 5, 6 and 7 as shown. If any corruption occurs in any of the transmitted code 1010010, the bit position in error can be found out by calculating $r_3r_2r_1$ at the receiving end. For example, if the received code word is 1110010, the recalculated value of $r_3r_2r_1$is 110, which indicates that bit position in error is 6, the decimal value of 110.

Data Link Control

The two main functions of the data link layer are data link control and media access control. The first, data link control, deals with the design and procedures for communication between two adjacent nodes: node-to-node communication

.

Data link control functions include framing, flow and error control, and software implemented protocols that provide smooth and reliable transmission of frames between nodes.

HDLC

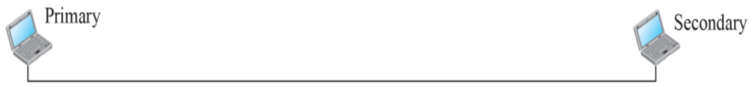
High-level Data Link Control (HDLC) is a bit-oriented protocol for communication over point-to-point and multipoint links. It implements the ARQ mechanisms .

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; asecondary station can only respond. The NRM is used for both point-to-point and multiple-point link

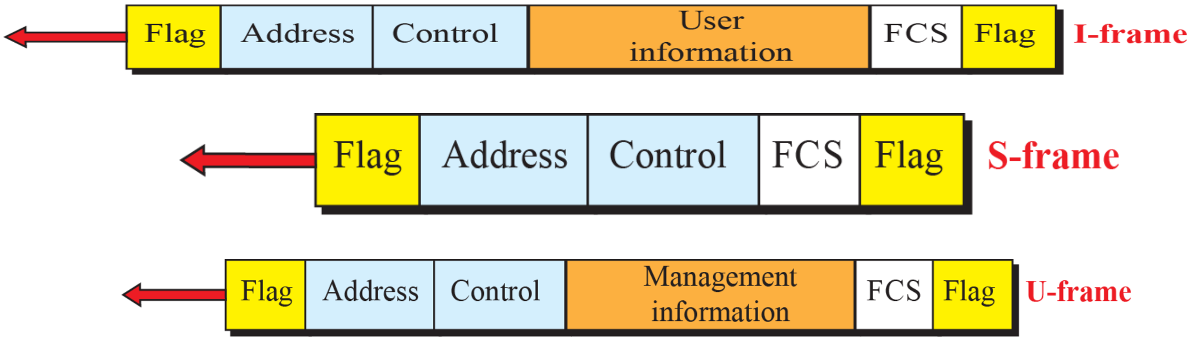


a. Point-to-point

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),

HDLC defines three types of frames: information frames (I-frames), supervisory frames (S-frames), and unnumbered frames (U-frames).



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.

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.

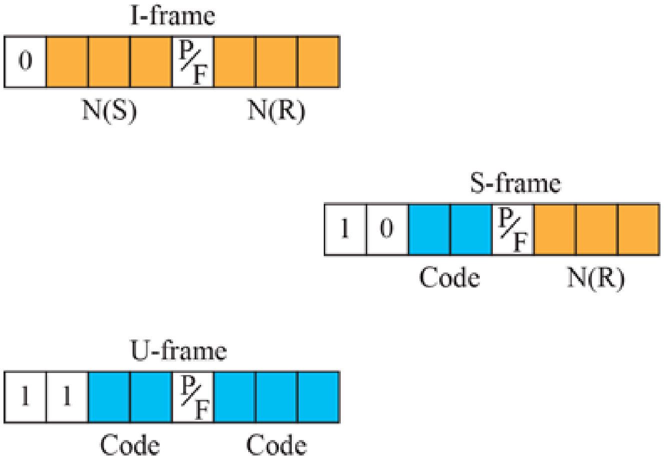
Control field. The control field is a 1- or 2-byte segment of the frame used for flow and error control.

Information field. The information field contains the user's data from the network layer or management information. 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

Figure 111.17: Control field format for the different frame types



11.36

Control fields for I Frame

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 sequence number between and 7; but in the extension format, in which the control field is 2 bytes, this field is larger. 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 *PIF* bit. The *PIP* field is a single bit with a dual purpose. It has meaning only when it is set (bit = 1) and can mean poll or final. It means *poll* when the frame is sent by a primary station to a secondary (when the address field contains the address of the receiver).

Control Fields for S Frame

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.

RR Receive Ready The code is 00
RNR Receive not Ready the bit sequence is 10.
REJ Reject the bit is 01.
SREJ Selective Reject the bit is 11

Control Fields for U Frame

U-frame codes are divided into two sections: a 2-bit prefix before the PtF bit and a 3-bit suffix after the PtF bit. Together, these two segments (5 bits) can be used to create up to 32 different types of U-frames

MAC Sublayer

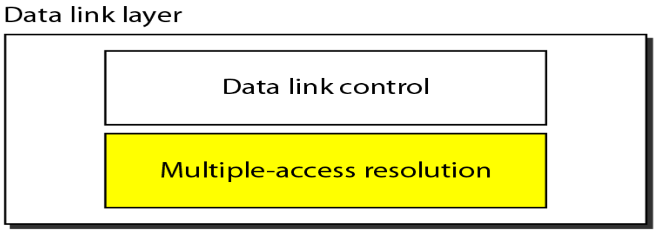
Network can be divided into two categories point to point n/w and broadcast n/w.
In the broadcast n/w the channels called as Multi Access Channels or Random Access Channels

There are 2 different schemes used for channel allocation

1. Static Channel Allocation
2. Dynamic Channel Allocation

MAC (Multiple Access Control)

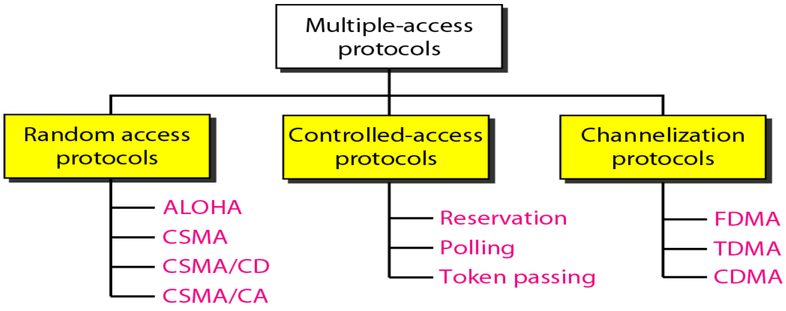
The upper sublayer that is responsible for flow and error control is called the logical link control (LLC) layer; the lower sublayer that is mostly responsible for multipleaccess resolution is called the media access control (MAC) layer



.

MAC (Multiple Access Control) Protocols

In random access or contention methods, no station is superior to another station and none is assigned the control over another. No station permits, or does not permit, another station to send. At each instance, a station that has data to send uses a procedure defined by the protocol to make a decision on whether or not to send.



Random Access

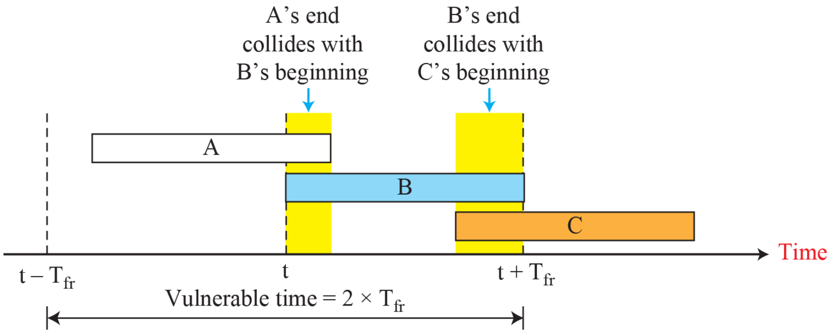
In random access or contention methods, no station is superior to another station and none is assigned the control over another. No station permits, or does not permit, another station to send. At each instance, a station that has data to send uses a procedure defined by the protocol to make a decision on whether or not to send. This decision depends on the state of the medium (idle or busy).

ALOHA /Pure ALOHA

The original ALOHA protocol is called pure ALOHA. This is a simple, but elegant protocol. The idea is that each station sends a frame whenever it has a frame to send. However, since there is only one channel to share, there is the possibility of collision between frames from different stations.
The medium is shared between the stations. When a station sends data, another station may attempt to do so at the same time. The data from the two stations collide and become garbled.

In ALOHA there are 2 types of collision occur partial collision and complete collision.

The **vulnerable time,** in which there is a possibility of collision. Vulnerable period is the time for the frame getting out collide frame each other . In the ALOHA vulnerable period has the length of 2 frame time.


A's end collides with B's beginning
B's end collides with C's beginning
A
B
C
Time
t – T_fr
t
t + T_fr
Vulnerable time = 2 × T_fr

Performance of ALOHA

S – Throughput – Expected number of successfultransmission.If frame time is 1.Maximum throughput is 1.

G – Offer load– Expected number of transmission and retransmission attempts per time unit .

Assumption

1. All frame have a fixed length of one time unit.
2. Infinite user population
3. Offered load is modeled as a poission process with rate G.

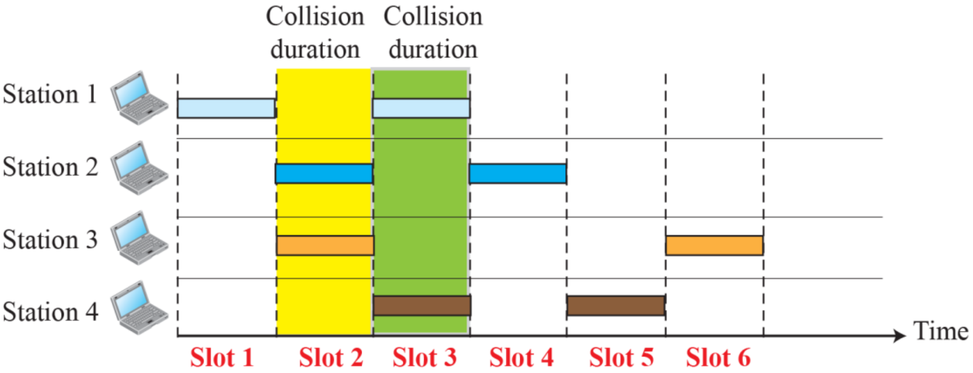
The throughput for pure ALOHA is S =G x *e-2G.*

The maximum throughput Smax=0.184 when G =(1/2).

This is 18.4% capacity used i.e performance of pure ALOHA is low.
Pure ALOHA has a vulnerable time of 2 x *Tfr* .This is so because there is no rule that defines when the station can send. A station may send soon after another station has started or soon before another station has finished. Slotted ALOHA was invented to improve the efficiency of pure ALOHA.

SLOTTED ALOHA

Because a station is allowed to send only at the beginning of the synchronized time slot, if a station misses this moment, it must wait until the beginning of the next time slot. This means that the station which started at the beginning of this slot has already finished sending its frame. Of course, there is still the possibility of collision if two stations try to send at the beginning of the same time slot. However, the vulnerable time is now reduced to one-half, equal to T_{fr}.

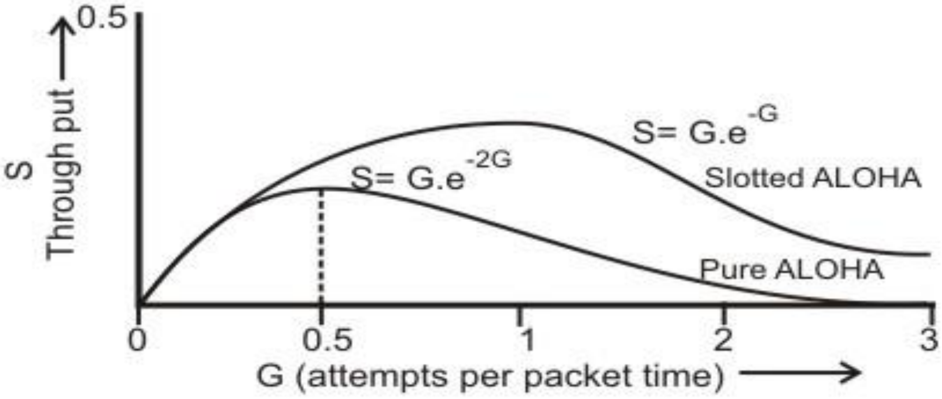


the vulnerable time for slotted ALOHA is one-half that of pure ALOHA.

Slotted ALOHA vulnerable time = T_{fr}

Throughput It can be proved that the average number of successful transmissions for slotted ALOHA is S = G x e^{-G}.
The maximum throughput Smax is 0.368, when G = 1.

In other words, if a frame is generated during one frame transmission time, then 36.8 percent of these frames reach their destination successfully. This result can be expected because the vulnerable time is equal to the frame transmission time.



Throughput versus offered load for ALOHA protocol

Carrier Sense Multiple Access (CSMA)

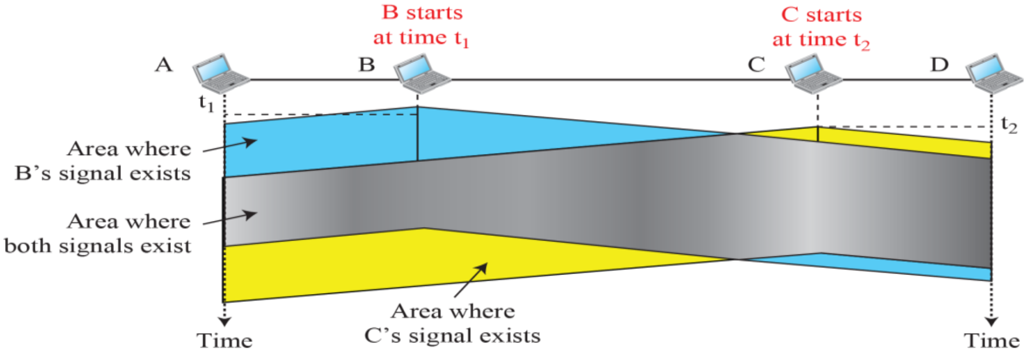
The chance of collision can be reduced if a station senses the medium before trying to use it. Carrier sense multiple access (CSMA) requires that each station first listen to the medium (or check the state of the medium) before sending.

In other words, CSMA is based on the principle "sense before transmit" or "listen before talk." CSMA can reduce the possibility of collision, but it cannot eliminate it.

Stations are connected to a shared channel (usually a dedicated medium).

Vulnerable Time

The vulnerable time for CSMA is the propagation time T_p .This is the time needed for a signal to propagate from one end of the medium to the other. When a station sends a frame, and any other station tries to send a frame during this time, a collision will result. But if the first bit of the frame reaches the end of the medium, every station will already have heard the bit and will refrain from sending.



I-Persistent

The **I-persistent method** is simple and straightforward. **In** this method, after the station finds the line idle, it sends its frame immediately (with probability I). This method has the highest chance of collision because two or more stations may find the line idle and send their frames immediately.

Nonpersistent

In the **nonpersistent method,** a station that has a frame to send senses the line. If the line is idle, it sends immediately. If the line is not idle, it waits a random amount of time and then senses the line again. The nonpersistent approach reduces the chance of collision because it is unlikely that two or more stations will wait the same amount of time and retry to send simultaneously.

p-Persistent

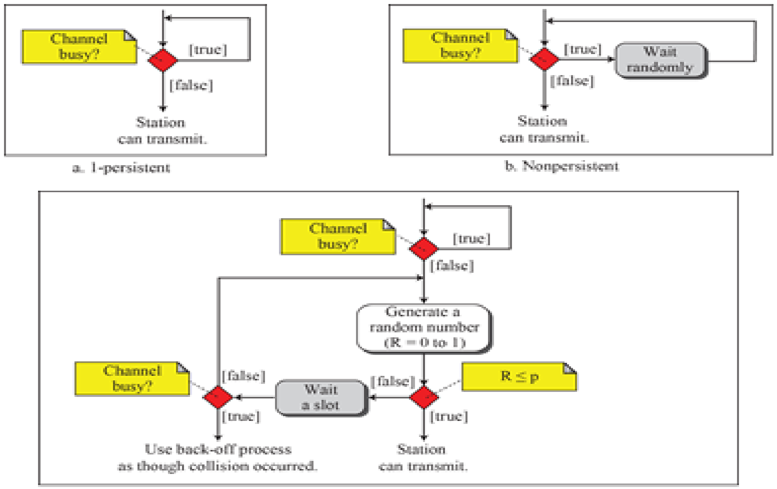
The p-persistent method is used if the channel has time slots with a slot duration equal to or greater than the maximum propagation time. The p-persistent approach combines the advantages of the other two strategies. It reduces the chance of collision and improves efficiency. In this method, after the station finds the line idle it follows these steps:

1. With probability p, the station sends its frame.

2. With probability $q = 1 - p$, the station waits for the beginning of the next time slot and checks the line again.

a. If the line is idle, it goes to step 1.
b. If the line is busy, it acts as though a collision has occurred and uses the backoff procedure.

Figure 12.10: Flow diagram for three persistence methods



a. 1-persistent

b. Nonpersistent

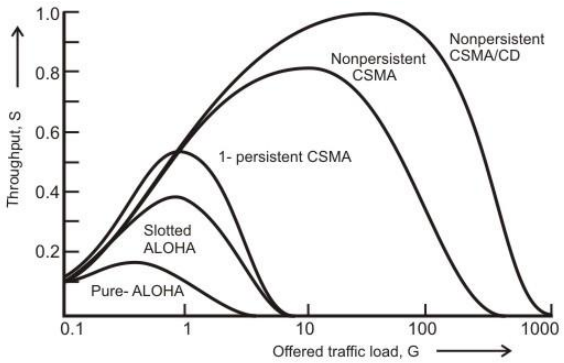
c. *p*-persistent

Carrier Sense Multiple Access with Collision Detection (CSMA/CD) (exponential algorithm)

This algorithm to handle the collision.
In this method, a station monitors the medium after it sends a frame to see if the transmission was successful. If so, the station is finished. If, however, there is a collision, the frame is sent again.

1. This is widely used for topology LANs or IEEE 802.3 Ethernet.
2. It only works if propagation delay is small relative to transmission delay (i.epropogation delay is a small)
3. In CSMA/CD while transmitting the sender is listening the medium for collisions. Sender stops if collision sender stops if collision has occurred.
4. If collisions are detected during transmission stop the transmission and transmit a jam signal to notify other stations of collisions. After sending the jam signal, back-off for a random amount of time then to transmit again.
5. In the CSMA/CD packet should be twice as long as the time to detect a collision.(2 * maximum propagation delay)

Exponential Back-off Algorithm



<u>Collision Free protocols</u>

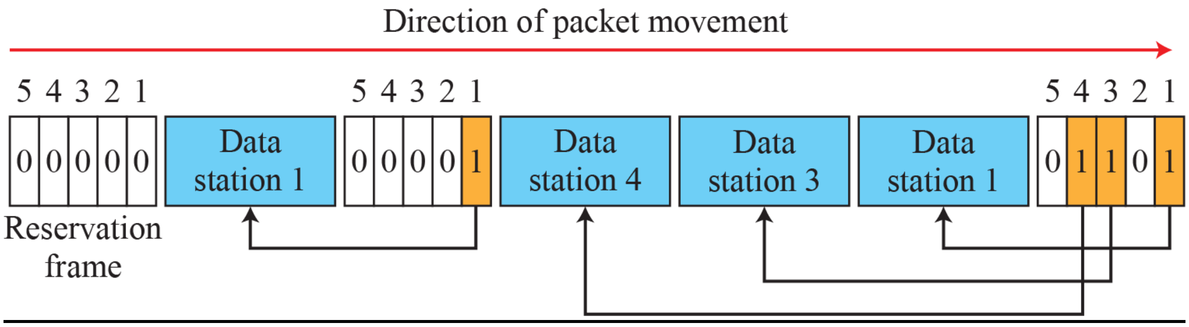
The Medium Access Control (MAC) layer of the OSI model is responsible for handling collision of frames. Collision – free protocols are devised so that collisions do not occur.

collision can still occur during the contention period if more than one stations starts to transmit at the same time. Collision – free protocols resolves collision in the contention period and so the possibilities of collisions are eliminated.

<u>Bit Map protocol (Reservation Protocol)</u>

In the reservation method, a station needs to make a reservation before sending data. Time is divided into intervals. In each interval, a reservation frame precedes the data frames sent in that interval.

If there are *N* stations in the system, there are exactly *N* reservation minislots in the reservation frame. Each minislot belongs to a station. When a station needs to send a data frame, it makes a reservation in its own mini slot. The stations that have made reservations can send their data frames after the reservation frame. a situation with five stations and a five-minislot reservation frame. In the first interval, only stations 1, 3, and 4 have made reservations. In the second interval, only station 1 has made a reservation.



Binary Countdown

This protocol overcomes the overhead of 1 bit per station of the bit – map protocol. Here, binary addresses of equal lengths are assigned to each station. For example, if there are 6 stations, they may be assigned the binary addresses 001, 010, 011, 100, 101 and 110. All stations wanting to communicate broadcast their addresses. The station with higher address gets the higher priority for transmitting.

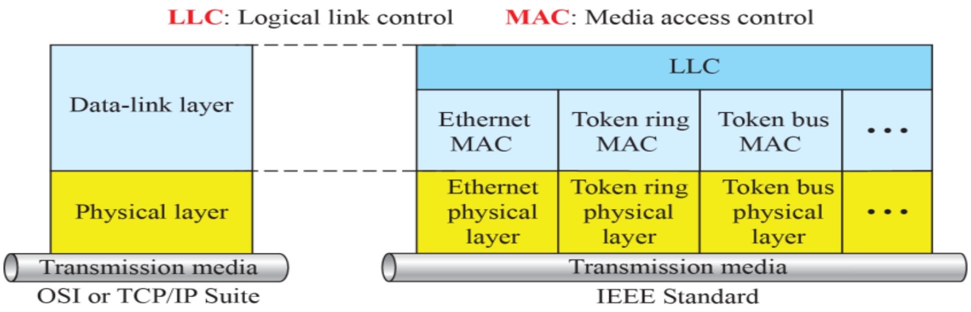
Adaptive Tree walk Protocols

- partition the group of station and limit the contention for each slot.
- Under light load, everyone can try for each slot like aloha
- Under heavy load, only a group can try for each slot
- **How do we do it:**
  1. treat every stations as the leaf of a binary tree
  2. first slot (after successful transmission), all stations can try to get the slot(under the root node).
  3. if no conflict, fine
  4. in case of conflict, only nodes under a subtree get to try for the next one. (depth first search)

<u>IEEE Standard / Wired LANs/ Ethernet</u>

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.

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.



IEEE 802.1 Higher Layer Internetwork

IEEE 802.2

IEEE 802.3 MAC/ Ethernet /CSMA

IEEE 802.4 Token Bus

IEEE 802.5 Token Ring

IEEE 802.11 Wireless LAN

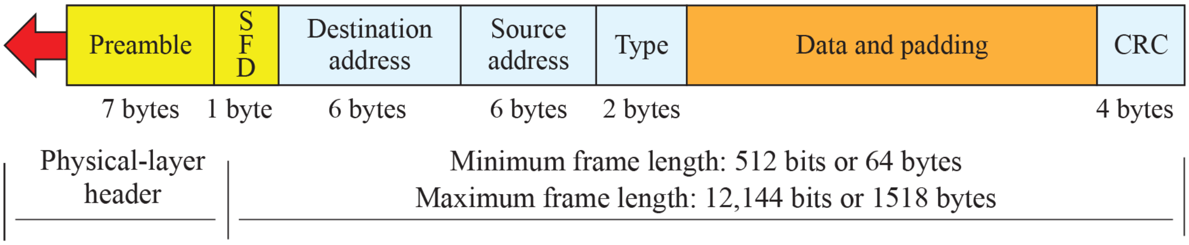
Standard Ethernet/ IEEE 802.3

The Ethernet LAN was developed in the 1970s by Robert Metcalfe and David Boggs. Since then, it has gone through four generations: Standard Ethernet (10 Mbps), Fast Ethernet (100 Mbps), Gigabit Ethernet (1 Gbps), and 10 Gigabit Ethernet (10 Gbps), as shown in Figure 13.2. We briefly discuss all these generations.

MAC Frame format

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 Os and Is 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. The 56-bit pattern allows the stations to miss some bits at the beginning of the frame. The preamble is actually added at the physical layer and is not (formally) part of the frame.

Start frame delimiter (SFD)

The second field (l byte: 10101011) signals thebeginning of the frame. The SFD warns the station or stations that this is the lastchance for synchronization. The last 2 bits is 11 and alerts the receiver that the nextfield 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

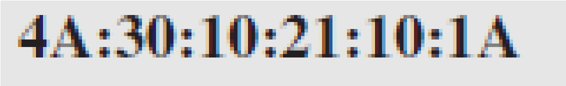
Data.

This field carries data encapsulated from the upper-layer protocols. It is a minimum of 46 and a maximum of 1500 bytes, as we will see later.

CRC. The last field contains error detection information.

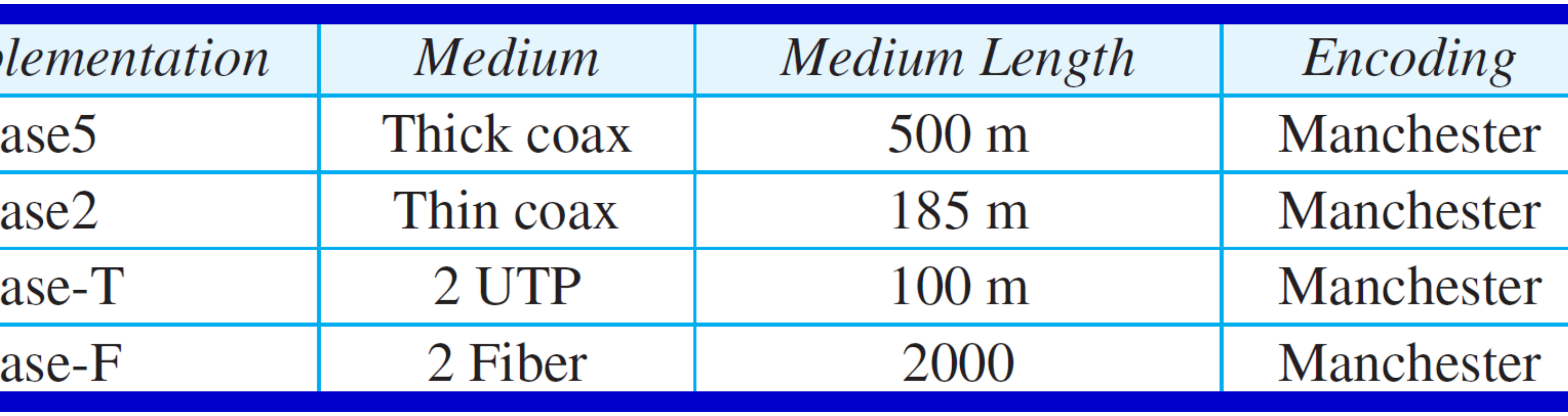
Addressing

Each station on an Ethernet network (such as a PC, workstation, or printer) has its own network interface card (NIC). The NIC fits inside the station and provides the station with a 6-byte physical address. As shown in Figure 13.6, the Ethernet address is 6 bytes (48 bits), nonnally written in hexadecimal notation, with a colon between the bytes.


4A:30:10:21:10:1A

The least significant bit of the first byte defines the type of address.
If the bit is 0, the address is unicast; otherwise, it is multicast

unicast: 0 multicast: 1



lementation	*Medium*	*Medium Length*	*Encoding*
ase5	Thick coax	500 m	Manchester
ase2	Thin coax	185 m	Manchester
ase-T	2 UTP	100 m	Manchester
ase-F	2 Fiber	2000	Manchester

Fast Ethernet

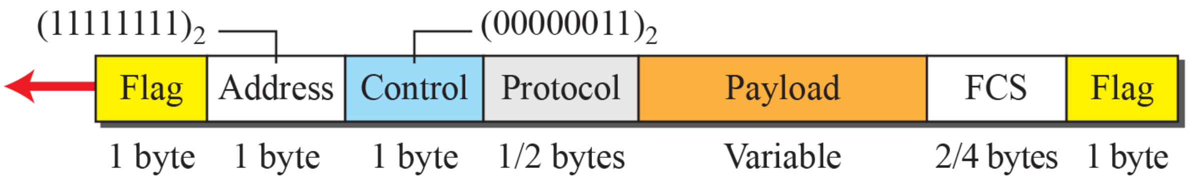
The Fast Ethernet standard (IEEE 802.3u) has been established for Ethernet networks that need higher transmission speeds. This standard raises the Ethernet speed limit from 10 Mbps to 100 Mbps with only minimal changes to the existing cable structure. Fast Ethernet provides faster throughput for video, multimedia, graphics, Internet surfing and stronger error detection and correction.There are three types of Fast Ethernet: 100BASE-TX for use with level 5 UTP cable; 100BASE-FX for use with fiber-optic cable; and 100BASE-T4 which utilizes an extra two wires for use with level 3 UTP cable.

Gigabit Ethernet

Gigabit Ethernet was developed to meet the need for faster communication networks with applications such as multimedia and Voice over IP (VoIP). Also known as “gigabit-Ethernet-over-copper” or 1000Base-T, GigE is a version of Ethernet that runs at speeds 10 times faster than 100Base-T. It is defined in the IEEE 802.3 standard and is currently used as an enterprise backbone. Existing Ethernet LANs with 10 and 100 Mbps cards can feed into a Gigabit Ethernet backbone to interconnect high performance switches, routers and servers. The most important differences between Gigabit Ethernet and Fast Ethernet include the additional support of full duplex operation in the MAC layer and the data rates.

Point to Point Protocol

One of the most common protocols for point-to-point access is the Point-to-Point Protocol (PPP). Today, millions of Internet users who need to connect their home computers to the server of an Internet service provider use PPP. To control and manage the transfer of data, there is a need for a point-to-point protocol at the data-link layer. PPP is by far the most common.PPP is a byte-oriented protocol



PPP provides several services:
1. PPP defines the format of the frame to be exchanged between devices.
2. PPP defines how two devices can negotiate the establishment of the link and the exchange of data.
3. PPP defines how network layer data are encapsulated in the data link frame.
4. PPP defines how two devices can authenticate each other.
5. PPP provides multiple network layer services supporting a variety of network layer protocols.
6. PPP provides connections over multiple links.
7. PPP provides network address configuration. This is particularly useful when a home user needs a temporary network address to connect to the Internet.