MODULE-2- Data Link Layer: Error Detection andCorrection: Introduction Block Coding.

2.1 INTRODUCTION

2.1.1 Types of Errors

- When bits flow from 1 point to another, they are subject to unpredictable-changes '.' of interference.
- The interference can change the shape of the signal.
- Two types of errors: 1) Single-bit error 2) Burst-error.

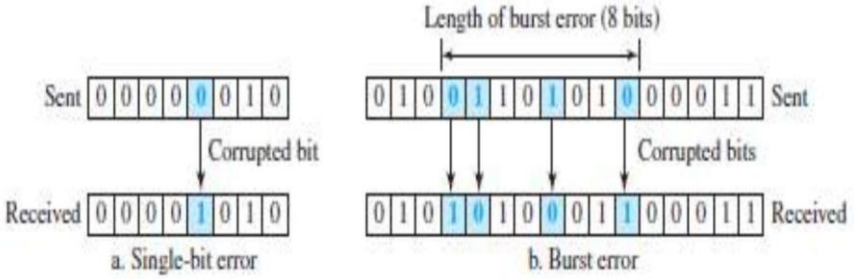


Figure 10.1 *Single-bit and burst error*

1) Single-Bit Error

- Only 1 bit of a given data is changed
  → from 1 to 0 or
  → from 0 to 1 (Figure 10.1a).

2) Burst Error

- Two or more bits in the data have changed

→ from 1 to 0 or

→ from 0 to 1 (Figure 10.1b).

- A burst-error occurs more than a single-bit error. This is because:
  - Normally, the duration of noise is longer than the duration of 1-bit.
  - When noise affects data, the noise also affects the bits.
  - The no. of corrupted-bits depends on data-rate and duration of noise.

2.1.2 Redundancy

• The central concept in detecting/correcting errors is *redundancy*.

• Some extra-bits along with the data have to be sent to detect/correct errors. These extra bits are called redundant-bits.

• The redundant-bits are

→ added by the sender and

→ removed by the receiver.

• The presence of redundant-bits allows the receiver to detect/correct errors.

2.1.3 Error Detection vs. Error Correction

• Error-correction is more difficult than error-detection.

1) Error Detection

- Here, we are checking whether any error has occurred or not.
- The answer is a simple YES or NO.
- We are not interested in the number of corrupted-bits.

2) Error Correction

- Here, we need to know

  → exact number of corrupted-bits and

  → location of bits in the message.
- Two important factors to be considered:

  1) Number of errors and

  2) Message-size.

2.1.4 Coding

- Redundancy is achieved through various coding-schemes.

1) Sender adds redundant-bits to the data-bits. This process creates a relationship between

   → redundant-bits and

   → data-bits.

2) Receiver checks the relationship between redundant-bits & data-bits to detect/correct errors.

- Two important factors to be considered:
  1) Ratio of redundant-bits to the data-bits and
  2) Robustness of the process.
- Two broad categories of coding schemes: 1) Block-coding and 2) Convolution coding.

2.2 Block Coding

- The message is divided into k-bit blocks. These blocks are called data-words.
- Here, r-redundant-bits are added to each block to make the length n=k+r.
- The resulting n-bit blocks are called code-words.
- Since n>k, the number of possible code-words is larger than the number of possible data-words.
- Block-coding process is 1-to-1; the same data-word is always encoded as the same code-word.
- Thus, we have 2n-2k code-words that are not used. These code-words are invalid or illegal.

2.2.1 Error Detection

- If the following 2 conditions are met, the receiver can detect a change in the original code-word:
  1) The receiver has a list of valid code-words.
  2) The original code-word has changed to an invalid code-words.

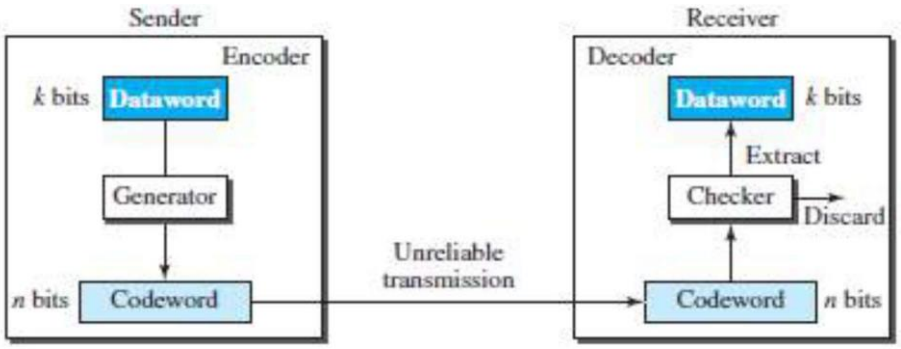


Figure 10.2 *Process of error detection in block coding*

Here is how it works (Figure 10.2):

1) At Sender

- The sender creates code-words out of data-words by using a generator. The generator applies the rules and procedures of encoding.
- During transmission, each code-word sent to the receiver may change.

2) At Receiver

  - If the received code-word is the same as one of the valid code-words, the code-word is accepted and the corresponding data-word is extracted for use. If the received code-word is invalid, the code-word is discarded.
  - However, if the code-word is corrupted but the received code-word still matches a valid codeword, the error remains undetected.

- An error-detecting code can detect only the types of errors for which it is designed; other types of errors may remain undetected.

Example 2.1

Let us assume that $k = 2$ and $n = 3$. Table 10.1 shows the list of datawords and codewords.

Table 10.1 *A code for error detection*

Dataword	*Codeword*	*Dataword*	*Codeword*
00	000	10	101
01	011	11	110

Assume the sender encodes the dataword 01 as 011 and sends it to the receiver. Consider the following cases:

1. The receiver receives 011. It is a valid codeword. The receiver extracts the dataword 01 from it..
2. The codeword is corrupted during transmission, and 111 is received (the leftmost bit is corrupted). This is not a valid codeword and is discarded.
3. The codeword is corrupted during transmission, and 000 is received (the right two bits are corrupted). This is a valid codeword. The receiver incorrectly extracts the dataword 00. Two corrupted bits have made the error undetectable.

2.2.1.1 Hamming Distance

- The main concept for error-control: Hamming distance.
- The Hamming distance b/w 2 words is the number of differences between the corresponding bits.
- Let d(x,y) = Hamming distance b/w 2 words x and y.
- Hamming distance can be found by
  → applying the XOR operation on the 2 words and
  → counting the number of 1s in the result.
- For example:

1) The Hamming distance d(000, 011) is 2 because 000⊕011= 011 (two 1s).
2) The Hamming distance d(10101, 11110) is 3 because 10101⊕11110 = 01011 (three 1s).

Hamming Distance and Error

- Hamming distance between the received word and the sent code-word is the number of bits that are corrupted during transmission.
  For example: Let Sent code-word = 00000
  Received word = 01101
  Hamming distance = d(00000, 01101) =3. Thus, 3 bits are in error.

2.2.1.1.1 Minimum Hamming Distance for Error Detection

- Minimum Hamming distance is the smallest Hamming distance b/w all possible pairs of code-words.
- Let dmin = minimum Hamming distance.
- To find dmin value, we find the Hamming distances between all words and select the smallest one.

Minimum-distance for Error-detection

- If ‘s’ errors occur during transmission, the Hamming distance b/w the sent code-word and received code-word is ‘s’ (Figure 10.3).

- If code has to detect upto ‘s’ errors, the minimum-distance b/w the valid codes must be ‘s+1’ i.e. dmin=s+1.
- We use a geometric approach to define dmin=s+1.

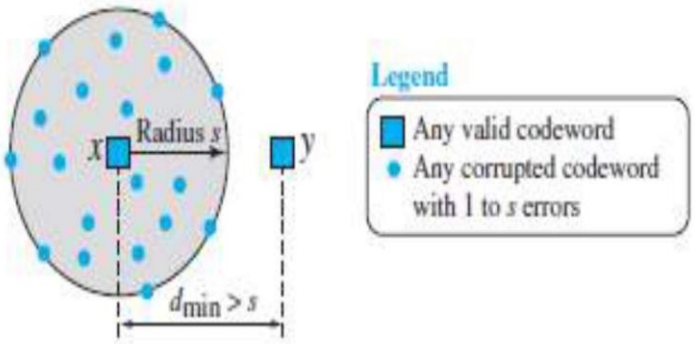


Figure 10.3 *Geometric concept explaining d_{min} in error detection*

  - Let us assume that the sent code-word x is at the center of a circle with radius s.
  - All received code-words that are created by 0 to s errors are points inside the circle or on the perimeter of the circle.
  - All other valid code-words must be outside the circle.
- For example: A code scheme has a Hamming distance dmin = 4. This code guarantees the detection of upto 3 errors (d = s + 1 or s = 3).

2.2.1.2 Linear Block Codes

- Almost all block codes belong to a subset of block codes called linear block codes.
- A linear block code is a code in which the XOR of 2 valid code-words creates another valid code-word. (XOR -> Addition modulo-2).
- The code in below Table 10.1 is a linear block code because the result of XORing any code-word with any other code-word is a valid code-word.
- For example, the XORing of the 2nd and 3rd code-words creates the 4th one.

Table 10.1 *A code for error detection*

Dataword	Codeword	Dataword	Codeword
00	000	10	101
01	011	11	110

2.2.1.2.1 Minimum Distance for Linear Block Codes

- Minimum Hamming distance is no. of 1s in the nonzero valid code-word with the smallest no. of 1s.
- In above Table 10.1, The numbers of 1s in the nonzero code-words are 2, 2, and 2. So the minimum Hamming distance is dmin = 2.

2.2.1.3 Parity Check Code

- This code is a linear block code. This code can detect an odd number of errors.
- A k-bit data-word is changed to an n-bit code-word where n=k+1.
- One extra bit is called the parity-bit.
- The parity-bit is selected to make the total number of 1s in the code-word even.
- Minimum hamming distance dmin = 2. This means the code is a single-bit error-detecting code.

Table 10.2 *Simple parity-check code C(5, 4)*

Dataword	Codeword	Dataword	Codeword
0000	00000	1000	10001
0001	00011	1001	10010
0010	00101	1010	10100
0011	00110	1011	10111
0100	01001	1100	11000
0101	01010	1101	11011
0110	01100	1110	11101
0111	01111	1111	11110

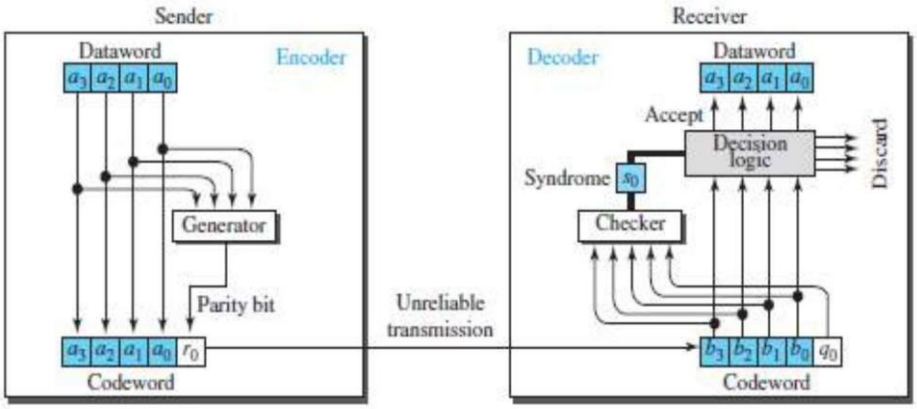


Figure 10.4 *Encoder and decoder for simple parity-check code*

1) At Sender

- The encoder uses a generator that takes a copy of a 4-bit data-word (a0, a1, a2, and a3) and generates a parity-bit r0.
- The encoder

  → accepts a copy of a 4-bit data-word (a0, a1, a2, and a3) and

  → generates a parity-bit r0 using a generator

  → generates a 5-bit code-word
- The parity-bit & 4-bit data-word are added to make the number of 1s in the code-word even.
- The addition is done by using the following:

$$r_0 = a_3 + a_2 + a_1 + a_0 \quad \text{(modulo-2)}$$

- The result of addition is the parity-bit.
  1) If the no. of 1s in data-word = even, result = 0. (r0=0)
  2) If the no. of 1s in data-word = odd, result = 1. (r0=1)
  3) In both cases, the total number of 1s in the code-word is even.
- The sender sends the code-word, which may be corrupted during transmission.

2) At Receiver

- The receiver receives a 5-bit word.
- The checker performs the same operation as the generator with one exception: The addition is done over all 5 bits.

$$s_0 = b_3 + b_2 + b_1 + b_0 + q_0 \quad \text{(modulo-2)}$$

- The result is called the syndrome bit (so).
- Syndrome bit = 0 when the no. of 1s in the received code-word is even; otherwise, it is 1.
- The syndrome is passed to the decision logic analyzer.
  1) If s0=0, there is no error in the received code-word. The data portion of the received code-word is accepted as the data-word.

2) If s0=1, there is error in the received code-word. The data portion of the received code-word is discarded. The data-word is not created.

Example 2.2

Let us look at some transmission scenarios. Assume the sender sends the dataword 1011. The codeword created from this dataword is 10111, which is sent to the receiver. We examine five cases:

1. No error occurs; the received codeword is 10111. The syndrome is 0. The dataword 1011 is created.
2. One single-bit error changes a_1. The received codeword is 10011. The syndrome is 1. No dataword is created.
3. One single-bit error changes r_0. The received codeword is 10110. The syndrome is 1. No dataword is created. Note that although none of the dataword bits are corrupted, no dataword is created because the code is not sophisticated enough to show the position of the corrupted bit.
4. An error changes r_0 and a second error changes a_3. The received codeword is 00110. The syndrome is 0. The dataword 0011 is created at the receiver. Note that here the dataword is wrongly created due to the syndrome value. The simple parity-check decoder cannot detect an even number of errors. The errors cancel each other out and give the syndrome a value of 0.
5. Three bits—a_3, a_2, and a_1—are changed by errors. The received codeword is 01011. The syndrome is 1. The dataword is not created. This shows that the simple parity check, guaranteed to detect one single error, can also find any odd number of errors.

Review Questions

1. What are the types of errors?
2. Compare error detection vs. error correction?
3. What are error detecting block coding techniques?

2.3 Cyclic Codes

- Cyclic codes are special linear block codes with one extra property:
  - If a code-word is cyclically shifted (rotated), the result is another code-word.
  - For ex: if code-word = 1011000 and we cyclically left-shift, then another code-word = 0110001.
  - Let First-word = a0 to a6 and Second-word = b0 to b6, we can shift the bits by using the following:

$$b_1 = a_0 \quad b_2 = a_1 \quad b_3 = a_2 \quad b_4 = a_3 \quad b_5 = a_4 \quad b_6 = a_5 \quad b_0 = a_6$$

2.3.1 Cyclic Redundancy Check (CRC)

- CRC is a cyclic code that is used in networks such as LANs and WANs.

Table 10.3 *A CRC code with C(7, 4)*

Dataword	*Codeword*	*Dataword*	*Codeword*
0000	0000000	1000	1000101
0001	0001011	1001	1001110
0010	0010110	1010	1010011
0011	0011101	1011	1011000
0100	0100111	1100	1100010
0101	0101100	1101	1101001
0110	0110001	1110	1110100
0111	0111010	1111	1111111

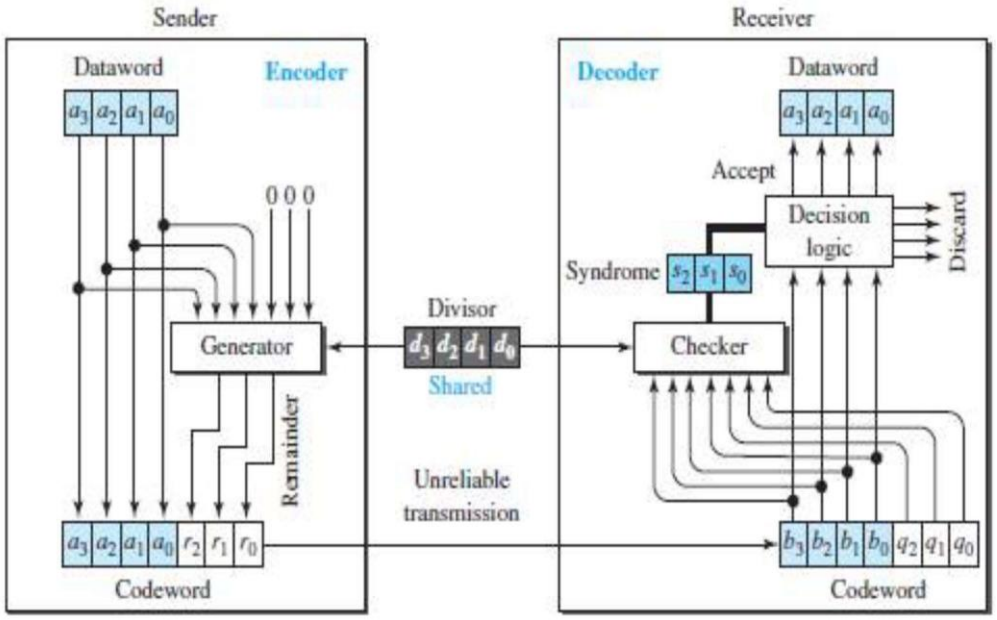


Figure 10.5 *CRC encoder and decoder*

Let Size of data-word = k bits (here k=4).

Size of code-word = n bits (here n=7).

Size of divisor = n-k+1 bits (here n-k+1=4). (Augmented -> increased)

1) At Sender

- n-k 0s is appended to the data-word to create augmented data-word. (here n- =3).
- The augmented data-word is fed into the generator (Figure 10.6).
- The generator divides the augmented data-word by the divisor.
- The remainder is called check-bits ($r_2r_1r_0$).
- The check-bits ($r_2r_1r_0$) are appended to the data-word to create the code-word.

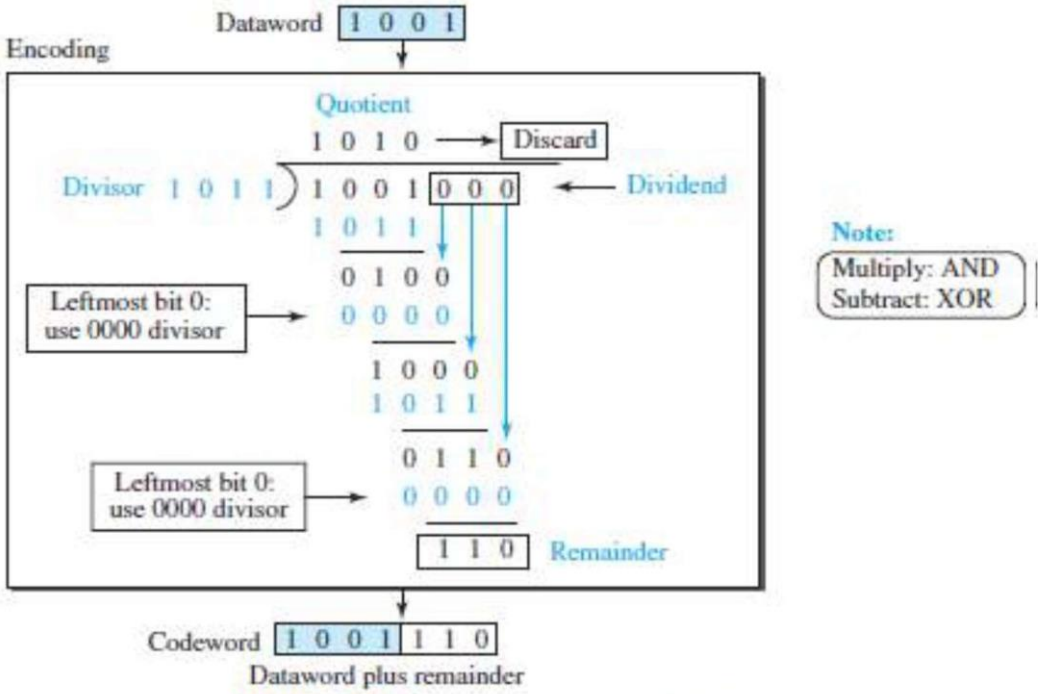


Figure 10.6 *Division in CRC encoder*

2) At Receiver

- The possibly corrupted code-word is fed into the checker.
- The checker is a replica of the generator.
- The checker divides the code-word by the divisor.
- The remainder is called syndrome bits ($r_2r_1r_0$).
- The syndrome bits are fed to the decision-logic-analyzer.
- The decision-logic-analyzer performs following functions:

  i) For No Error

  - If all syndrome-bits are 0s, the received code-word is accepted.
  - Data-word is extracted from received code-word (Figure 10.7a).

  ii) For Error

  - If all syndrome-bits are not 0s, the received code-word is discarded (Figure 10.7b).

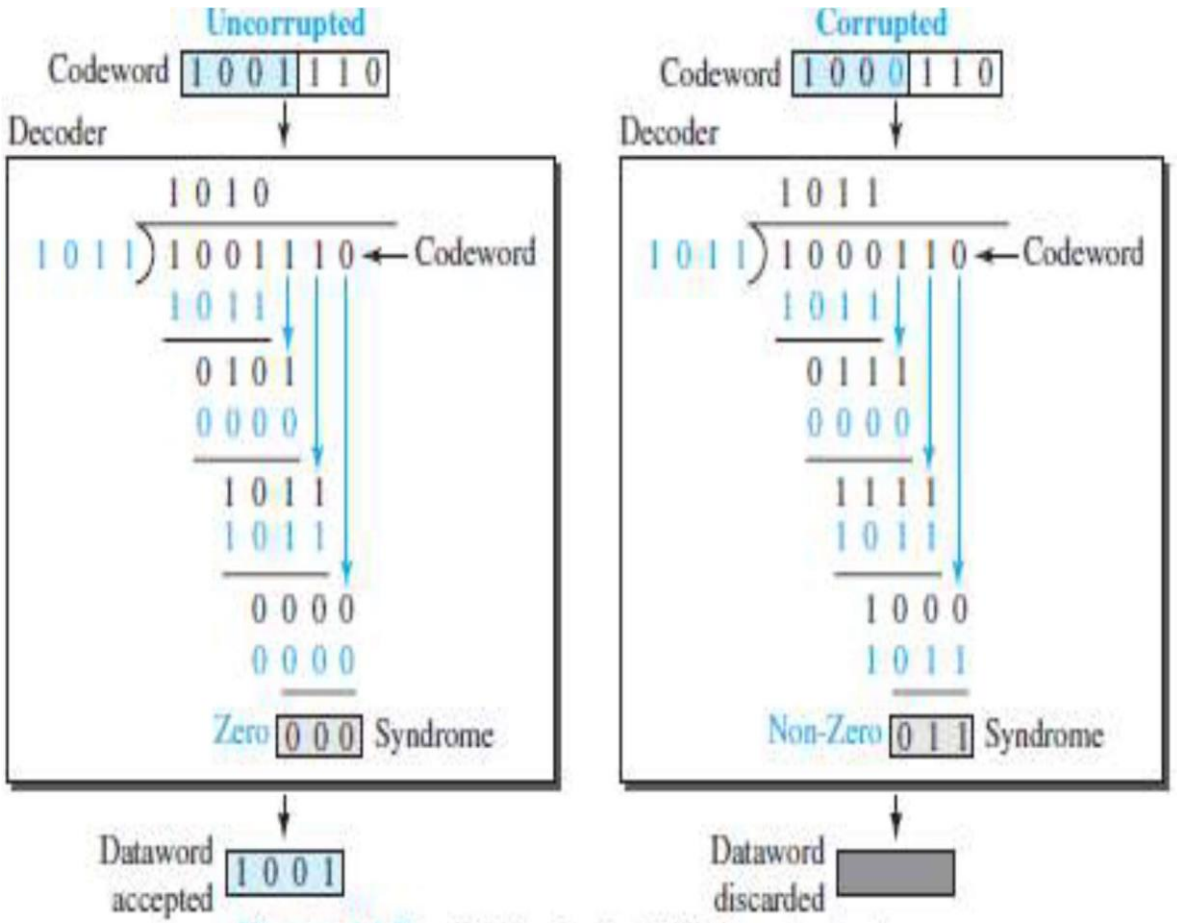


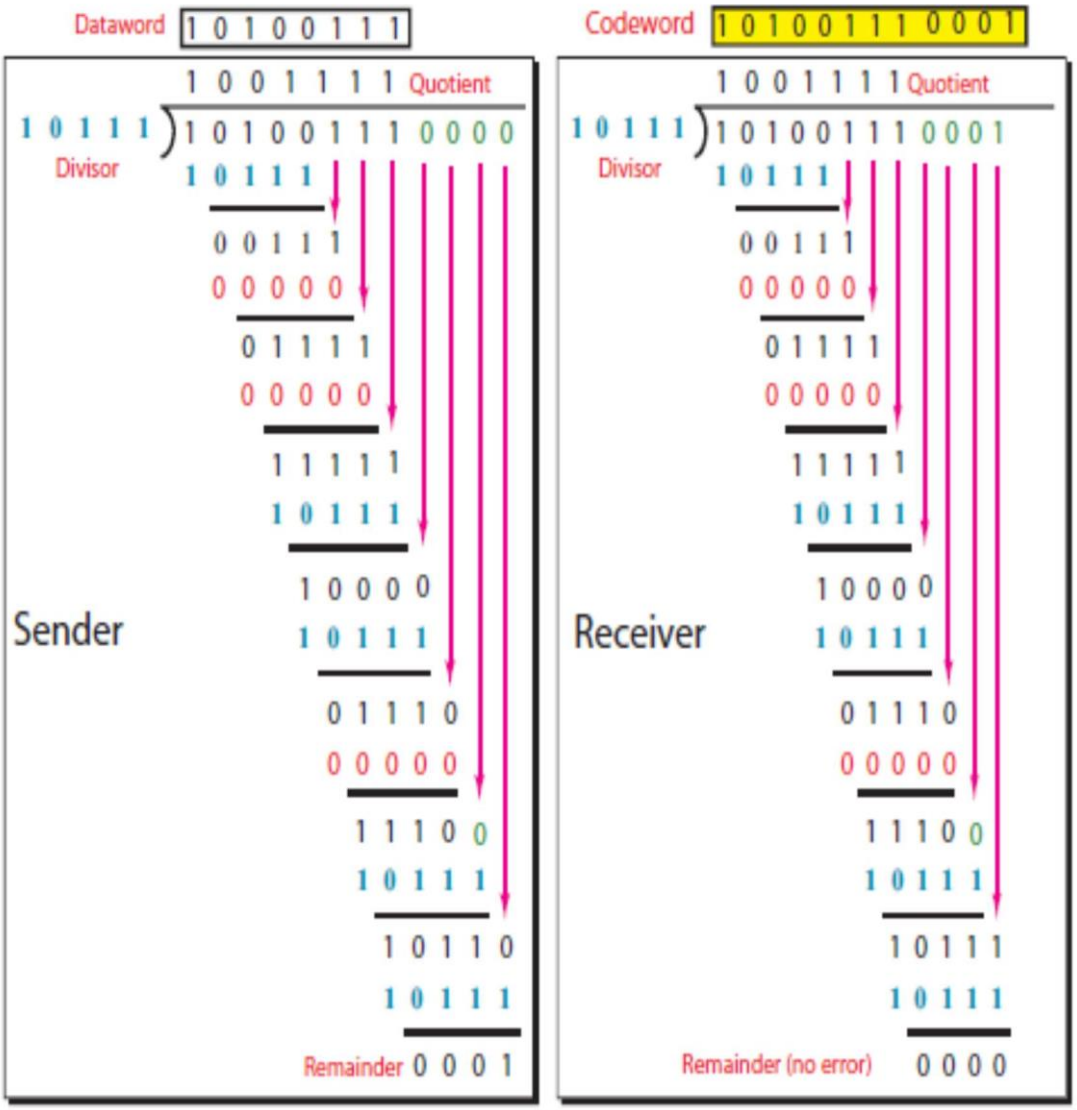
Figure 10.7 *Division in the CRC decoder for two cases*

Example 2.3

Given **Dataword 10100111**

Divisor 10111

Show the generation of codeword at the sender using binary division.



2.3.2 Polynomials

- A pattern of 0s and 1s can be represented as a polynomial with coefficients of 0 and 1 (Figure 10.8).
- The power of each term shows the position of the bit; the coefficient shows the value of the bit.

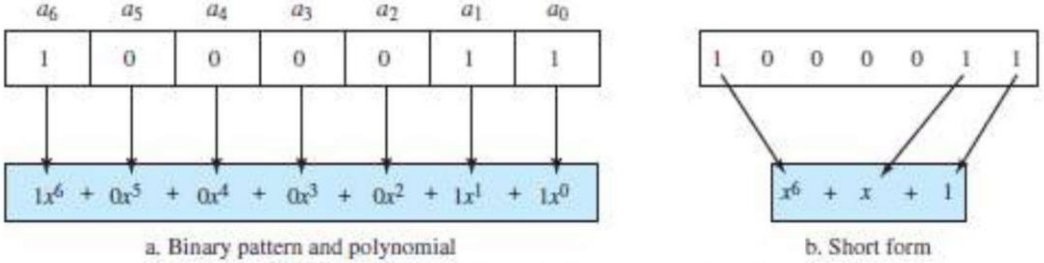


Figure 10.8 *A polynomial to represent a binary word*

2.3.3 Cyclic Code Encoder Using Polynomials

- Let Data-word = 1001 = x3+1.
- Divisor = 1011 = x3+x+1.
- In polynomial representation, the divisor is referred to as generator polynomial t(x) (Figure 10.9).

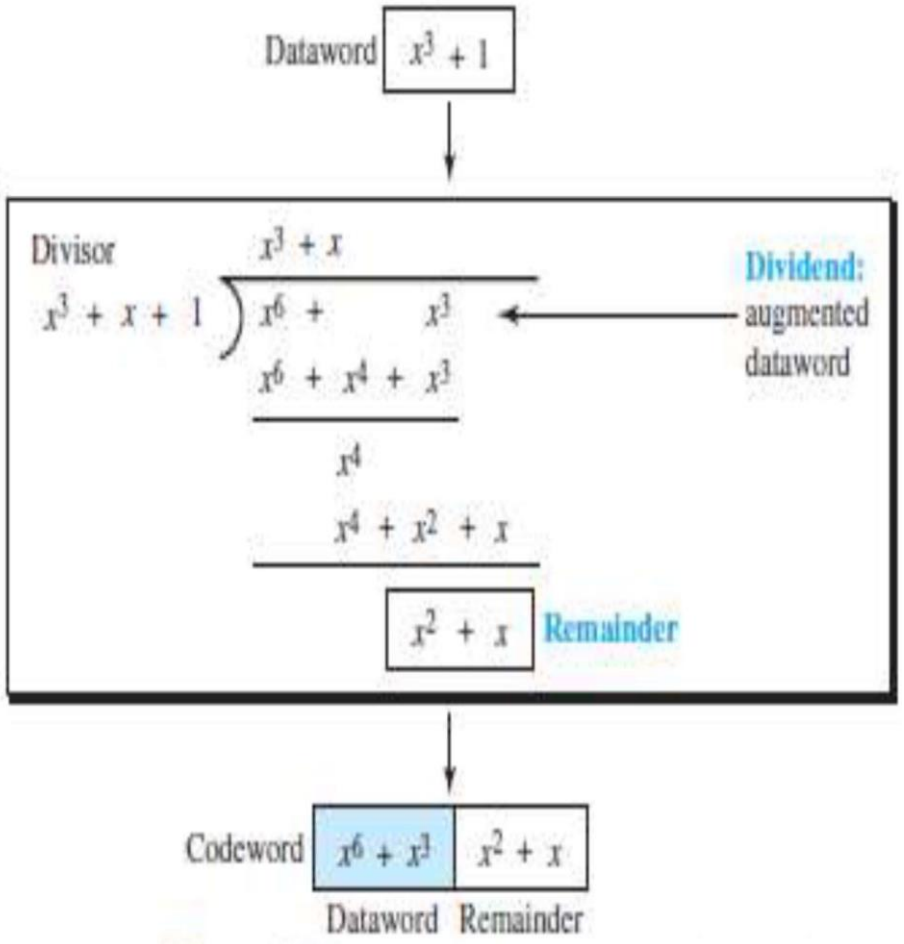


Figure 10.9 *CRC division using polynomials*

2.3.4 Cyclic Code Analysis

We define the following, where f(x) is a polynomial with binary coefficients:

Dataword: $d(x)$ Codeword: $c(x)$ Generator: $g(x)$ Syndrome: $s(x)$ Error: $e(x)$

In a cyclic code,

1. If $s(x) \neq 0$, one or more bits is corrupted.
2. If $s(x) = 0$, either
   a. No bit is corrupted, or
   b. Some bits are corrupted, but the decoder failed to detect them.

Single Bit Error

- If the generator has more than one term and the coefficient of x0 is 1, all single-bit errors can be caught.

Two Isolated Single-Bit Errors

- If a generator cannot divide xi+1 (t between 0 & n-1), then all isolated double errors can be detected (Figure 10.10).

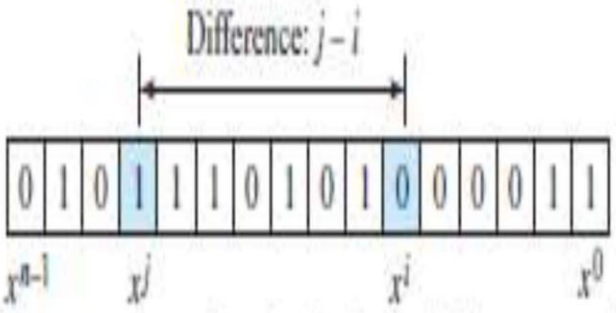


Figure 10.10 Representation of two isolated single-bit errors using polynomials

Odd Numbers of Errors

- A generator that contains a factor of x+1 can detect all odd-numbered errors.

A good polynomial generator needs to have the following characteristics:

1. It should have at least two terms.
2. The coefficient of the term x^0 should be 1.
3. It should not divide $x^t + 1$, for t between 2 and $n - 1$.
4. It should have the factor $x + 1$.

Standard Polynomials

Table 10.4 *Standard polynomials*

Name	*Polynomial*	*Used in*
CRC-8	x^8+x^2+x+1 **100000111**	ATM header
CRC-10	$x^{10}+x^9+x^5+x^4+x^2+1$ **11000110101**	ATM AAL
CRC-16	$x^{16}+x^{12}+x^5+1$ **10001000000100001**	HDLC
CRC-32	$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$ **10000010011000001000111011011 0111**	LANs

2.3.5 Advantages of Cyclic Codes

- The cyclic codes have a very good performance in detecting
  → single-bit errors
  → double errors
  → odd number of errors and
  → burst-errors.
- They can easily be implemented in hardware and software. They are fast when implemented in hardware.

Lecture 11: Checksum

2.4 Checksum

- Checksum is an error-detecting technique.
- In the Internet,

  → The checksum is mostly used at the network and transport layer.

  → The checksum is not used in the data link layer.

Like linear and cyclic codes, the checksum is based on the concept of redundancy.

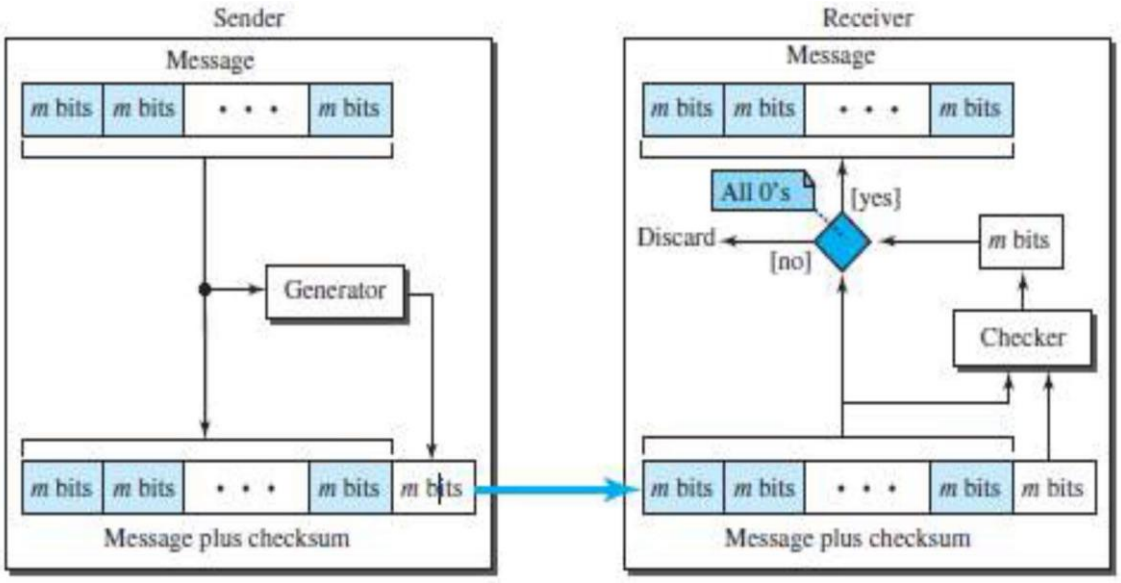


Figure 10.15 *Checksum*

1) At Source

- Firstly the message is divided into m-bit units.
- Then, the generator creates an extra m-bit unit called the checksum.
- The checksum is sent with the message.

2) At Destination

- The checker creates a new checksum from the combination of the message and sent checksum.

  i) If the new checksum is all 0s, the message is accepted.

  ii) If the new checksum is not all 0s, the message is discarded.

2.4.1 Concept of Checksum

Consider the following example:

Example 2.4

- Our data is a list of five 4-bit numbers that we want to send to a destination.
- In addition to sending these numbers, we send the sum of the numbers.
- For example: Let set of numbers = (7, 11, 12, 0, 6).
  - We send (7, 11, 12, 0, 6, 36), where 36 is the sum of the original numbers.
  - The receiver adds the five numbers and compares the result with the sum.
  - If the result & the sum are the same,
  - The receiver assumes no error, accepts the five numbers, and discards the sum.
  - Otherwise, there is an error somewhere and the data are not accepted.

Example 2.5

- To make the job of the receiver easy if we send the negative (complement) of the sum, called the checksum.
  - In this case, we send (7, 11, 12, 0, 6, -36).
  - The receiver can add all the numbers received (including the checksum).
  - If the result is 0, it assumes no error; otherwise, there is an error.

2.4.1.1 One's Complement

- The previous example has one major drawback.

  All of our data can be written as a 4-bit word (they are less than 15) except for the checksum.
- Solution: Use one's complement arithmetic.
  - We can represent unsigned numbers between 0 and 2^n-1 using only n bits.

- If the number has more than n bits, the extra leftmost bits need to be added to the n rightmost bits (wrapping).
- A negative number can be represented by inverting all bits (changing 0 to 1 and 1 to 0).
- This is the same as subtracting the number from 2^n-1.

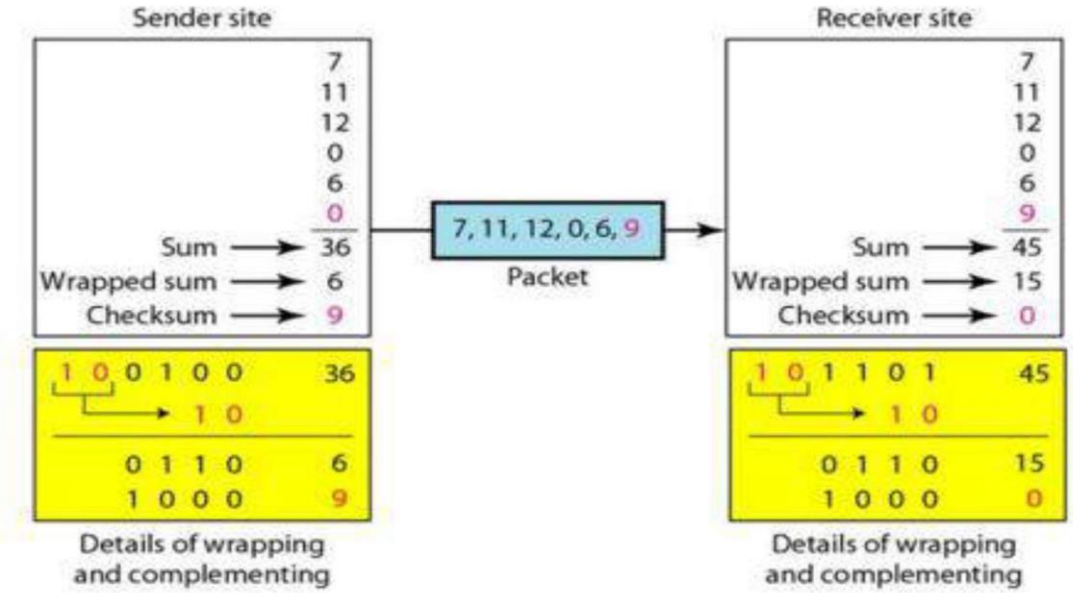


Figure 10.16

1) At Sender

- The sender initializes the checksum to 0 and adds all data items and the checksum. The result is 36.
- However, 36 cannot be expressed in 4 bits. The extra two bits are wrapped and added with the sum to create the wrapped sum value 6. The sum is then complemented, resulting in the checksum value 9 (15 - 6 = 9).
- The sender now sends six data items to the receiver including the checksum 9.

2) At Receiver

- The receiver follows the same procedure as the sender.
- It adds all data items (including the checksum); the result is 45.
- The sum is wrapped and becomes 15. The wrapped sum is complemented and becomes 0.

- Since the value of the checksum is 0, this means that the data is not corrupted. The receiver drops the checksum and keeps the other data items.
- If the checksum is not zero, the entire packet is dropped.

2.4.1.2 Internet Checksum

- Traditionally, the Internet has been using a 16-bit checksum.
- The sender or the receiver uses five steps.

Table 10.5 *Procedure to calculate the traditional checksum*

Sender	Receiver
1. The message is divided into 16-bit words.	1. The message and the checksum are received.
2. The value of the checksum word is initially set to zero.	2. The message is divided into 16-bit words.
3. All words including the checksum are added using one's complement addition.	3. All words are added using one's complement addition.
4. The sum is complemented and becomes the checksum.	4. The sum is complemented and becomes the new checksum.
5. The checksum is sent with the data.	5. If the value of the checksum is 0, the message is accepted; otherwise, it is rejected.

2.4.1.3 Algorithm

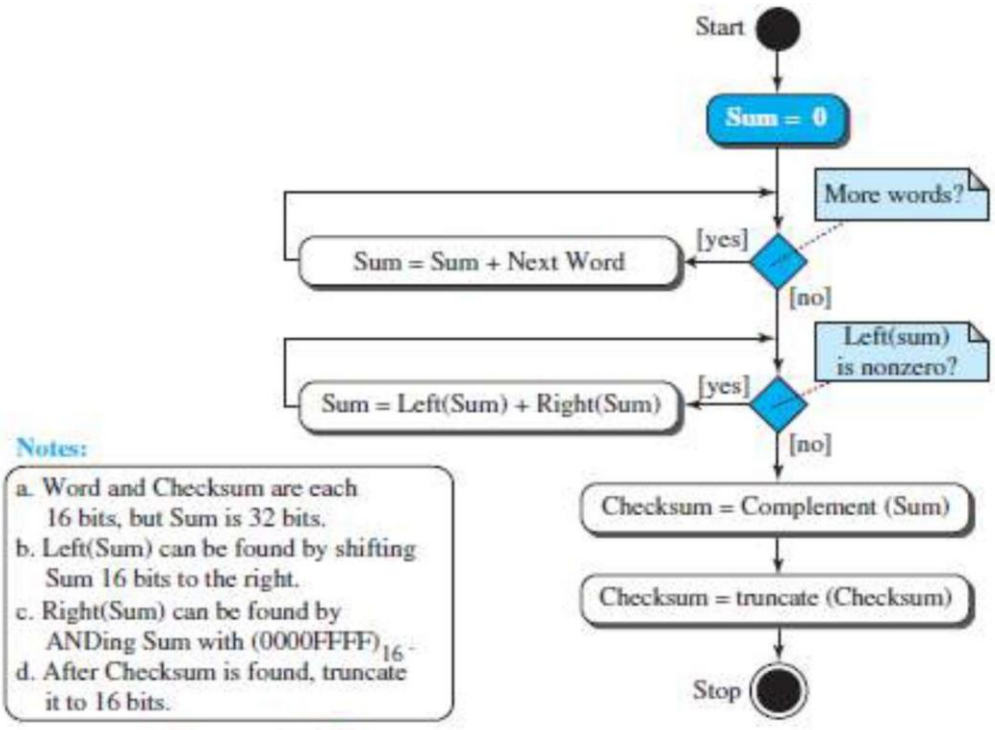


Figure 10.17 *Algorithm to calculate a traditional checksum*

2.4.2 Other Approaches to the Checksum

- If two 16-bit items are transposed in transmission, the checksum cannot catch this error.
- The reason is that the traditional checksum is not weighted: it treats each data item equally.
- In other words, the order of data items is immaterial to the calculation.
- Two approaches have been used to prevent this problem: 1)Fletcher and 2)Adler

2.4.2.1 Fletcher Checksum

- The Fletcher checksum was devised to weight each data item according to its position.
- Fletcher has proposed two algorithms: 8-bit and 16-bit (Figure 10.18).

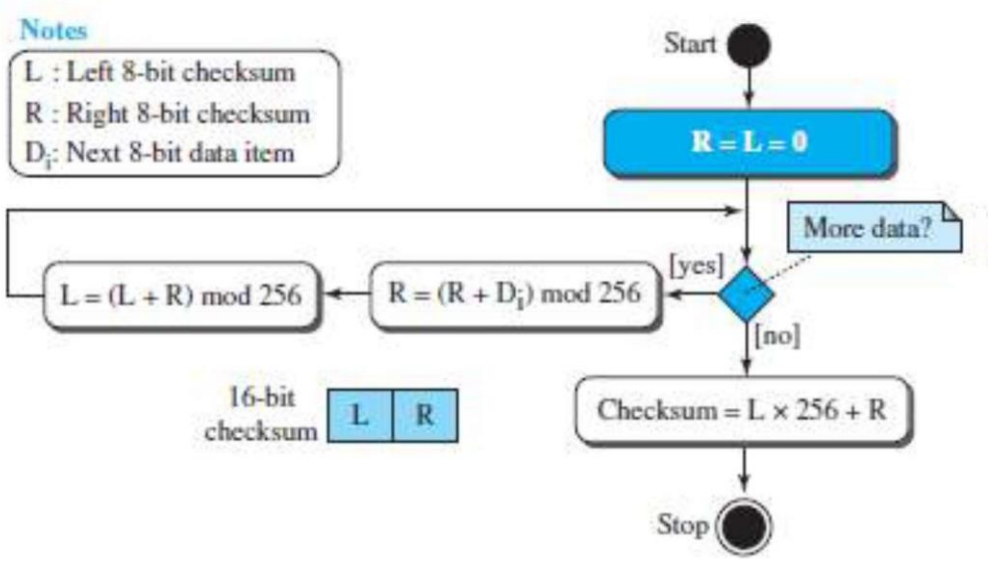


Figure 10.18 *Algorithm to calculate an 8-bit Fletcher checksum*

- The first, 8-bit Fletcher, calculates on 8-bit data items and creates a 16-bit checksum.
- The second, 16-bit Fletcher, calculates on 16-bit data items and creates a 32-bit checksum.
- The 8-bit Fletcher is calculated over data octets (bytes) and creates a 16-bit checksum.

- The calculation is done modulo 256 (28), which means the intermediate results are divided by 256 and the remainder is kept.
- The algorithm uses two accumulators, L and R.
- The first simply adds data items together. The second adds a weight to the calculation.

2.4.2.2 Adler Checksum

- The Adler checksum is a 32-bit checksum.
- It is similar to the 16-bit Fletcher with three differences (Figure 10.19).

  1) Calculation is done on single bytes instead of 2 bytes at a time.

  2) The modulus is a prime number (65,521) instead of 65,536.

  3) L is initialized to 1 instead of 0.
- A prime modulo has a better detecting capability in some combinations of data.

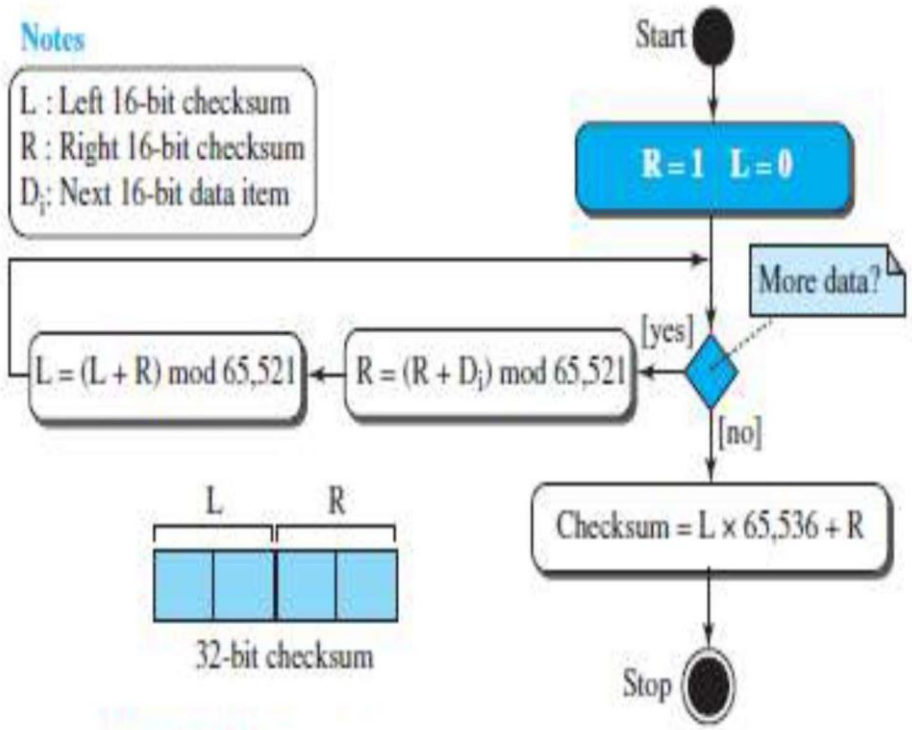


Figure 10.19 *Algorithm to calculate an Adler checksum*

Review Questions

1. Why hamming distance is used?
2. What is parity-check code?
3. How Secure is CRC code?
4. What polynomial codes?
5. Why internet checksum is Needed?
6. Compare i) Fletcher checksum and ii) Adler checksum.

Data link control: DLC Services: Framing, Flow Control,Error Control, Connectionless and Connection Oriented

2.5 DLC SERVICES

- The data link control (DLC) deals with procedures for communication between two adjacent nodes.
- Data link control functions include 1) Framing and 2) Flow control and 3) Error control.

2.5.1 Framing

- A frame is a group of bits. Framing means organizing the bits into a frame that are carried by the physical layer.
- The data-link-layer needs to form frames, so that each frame is distinguishable from another.
- Framing separates a message from other messages by adding sender-address & destination-address.
- The destination-address defines where the packet is to go.
- The sender-address helps the recipient acknowledge the receipt.
- Q: Why the whole message is not packed in one frame?
- Ans: Large frame makes flow and error-control very inefficient. Even a single-bit error requires the re-transmission of the whole message. When a message is divided into smaller frames, a single-bit error affects only that small frame.

2.5.1.1 Frame Size

Types of frames

1) Fixed-Size Framing, There is no need for defining boundaries of frames; the size itself can be used as a delimiter. For example: ATM WAN uses frames of fixed size called cells.

2) Variable-Size Framing, We need to define the end of the frame and the beginning of the next frame. Two approaches are used: i) Character-oriented approach ii) Bit-oriented approach.

2.5.1.2 Character-Oriented Framing

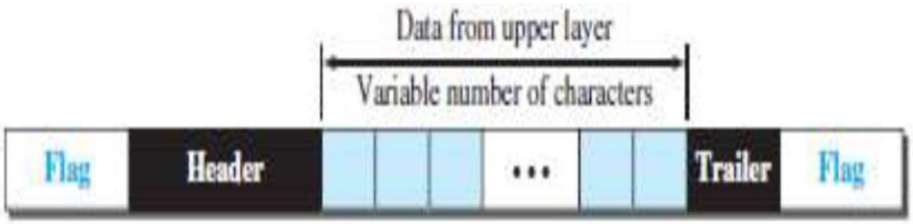


Figure 11.1 *A frame in a character-oriented protocol*

- Data to be carried are 8-bit characters from a coding system such as ASCII (Figure 11.1).
- The header and the trailer are also multiples of 8 bits.
- Header carries the source and destination-addresses and other control information. Trailer carries error-detection or error-correction redundant bits.
- To separate one frame from the next frame, an 8- bit (I-byte) flag is added at the beginning and the end of a frame.
- The flag is composed of protocol-dependent special characters and it signals the start or end of a frame.

Problem:

- Character-oriented framing is suitable when only text is exchanged by the data-link-layers.
- However, if we send other type of information (say audio/video), then any pattern used for the flag can also be part of the information.
- If the flag-pattern appears in the data-section, the receiver might think that it has reached the end of the frame.

Solution: A **byte-stuffing** is used. (Byte stuffing -> character stuffing)

In byte 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 called the escape character (ESC), which has a predefined bit pattern.
- When a receiver encounters the ESC character, the receiver removes ESC character from the data-section and treats the next character as data, not a delimiting flag.

Problem:

- What happens if the text contains one or more escape characters followed by a flag?
- The receiver removes the escape character, but keeps the flag, which is incorrectly interpreted as the end of the frame.

Solution:

- Escape characters part of the text must also be marked by another escape character (Fig 11.2).

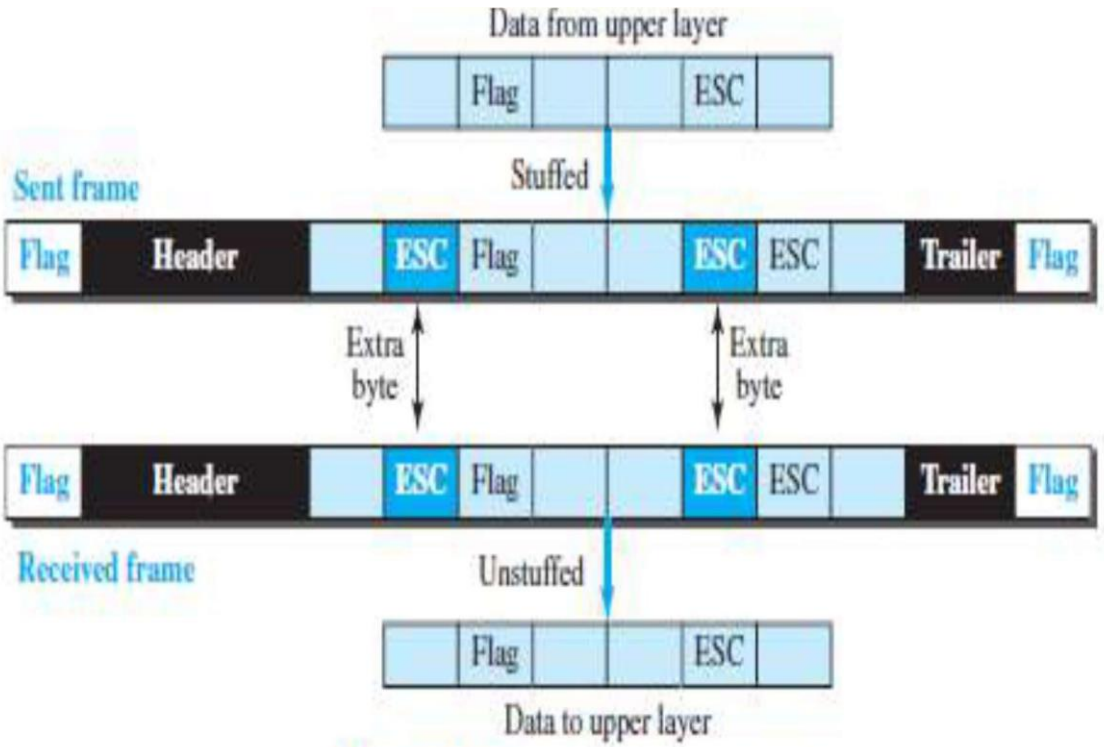


Figure 11.2 *Byte stuffing and unstuffing*

Note: byte stuffing is the process of adding one extra byte whenever there is a flag or escape character in the text.

2.5.1.3 Bit-Oriented Framing

- The data-section of a frame is a sequence of bits to be interpreted by the upper layer as text, audio, video, and so on.
- However, in addition to headers and trailers, we 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 (Figure 11.3).

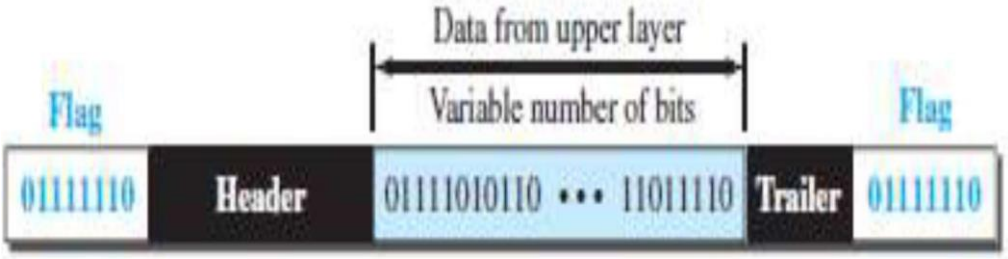


Figure 11.3 *A frame in a bit-oriented protocol*

Problem:

- If the flag-pattern appears in the data-section, the receiver might think that it has reached the end of the frame.

Solution: **A bit-stuffing** is used.

- In bit stuffing, if a 0 and five consecutive 1 bits are encountered, an extra 0 is added. This extra stuffed bit is eventually removed from the data by the receiver. (Figure 11.4).
- This guarantees that the flag field sequence does not inadvertently appear in the frame.

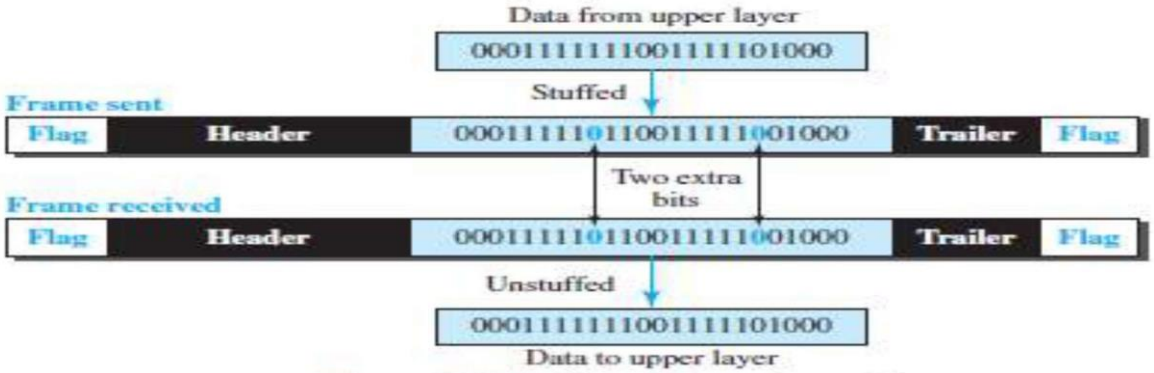


Figure 11.4 *Bit stuffing and unstuffing*

2.5.2 Flow Control and Error Control

One of the responsibilities of the DLC sublayer is flow and error control at the data-link layer.

2.5.2.1 Flow Control

- Whenever an entity produces items and another entity consumes them, there should be a balance between production and consumption rates.
- If the items are produced faster than they can be consumed, the consumer can be overwhelmed and may need to discard some items.
- We need to prevent losing the data items at the consumer site.
- At the sending node, the data-link layer tries to push frames toward the data-link layer at the receiving node (Figure 11.5).

- If the receiving node cannot process and deliver the packet to its network at the same rate that the frames arrive, it becomes overwhelmed with frames.
- Here, flow control can be feedback from the receiving node to the sending node to stop or slow down pushing frames.

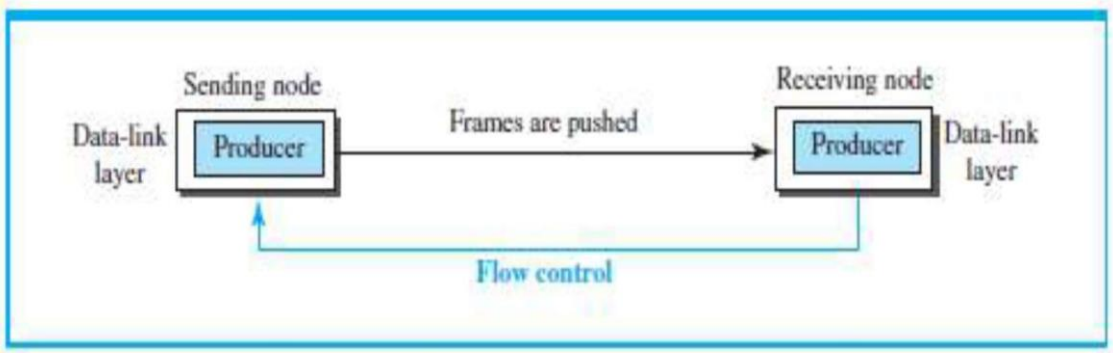


Figure 11.5 *Flow control at the data-link layer*

2.5.2.1.1 Buffers

- Flow control can be implemented by using buffer.
- A buffer is a set of memory locations that can hold packets at the sender and receiver.
- Normally, two buffers can be used.1) First buffer at the sender. 2) Second buffer at the receiver.
- The flow control communication can occur by sending signals from the consumer to the producer.
- When the buffer of the receiver is full, it informs the sender to stop pushing frames.

2.5.2.2 Error Control

- Error-control includes both error-detection and error-correction.
- Error-control allows the receiver to inform the sender of any frames lost/damaged in transmission.
- A CRC is added to the frame header by the sender and same is checked by the receiver.

- At the data-link layer, error control is normally implemented using one of the following two methods.
  - First method: If the frame is corrupted, it is discarded. If the frame is not corrupted, the packet is delivered to the network layer. This method is used mostly in wired LANs such as Ethernet.
  - Second method: If the frame is corrupted, it is discarded. If the frame is not corrupted, an acknowledgment is sent to the sender. Acknowledgment is used for the purpose of both flow and error control.

2.5.2.2.1 Combination of Flow and Error Control

- Flow and error control can be combined.
- The acknowledgment that is sent for flow control can also be used for error control to tell the sender the packet has arrived uncorrupted.
- The lack of acknowledgment means that there is a problem in the sent frame.
- A frame that carries an acknowledgment is normally called an ACK to distinguish it from the data frame.

2.5.3 Connectionless and Connection-Oriented

- A DLC protocol can be either connectionless or connection-oriented.

Connectionless Protocol

- Frames are sent from one node to the next without any relationship between the frames. Each frame is independent.
- The term connectionless does not mean that there is no physical connection (transmission medium) between the nodes; it means that there is no connection between frames.
- The frames are not numbered and there is no sense of ordering.
- Most of the data-link protocols for LANs are connectionless protocols.

Connection-Oriented Protocol

- A logical connection should first be established between the two nodes (setup phase).
- After all frames that are somehow related to each other are transmitted (transfer phase), the logical connection is terminated (teardown phase).
- The frames are numbered and sent in order.
- If the frames are not received in order, the receiver needs to wait until all frames belonging to the same set are received and then deliver them in order to the network layer.
- Connection oriented protocols are rare in wired LANs, but we can see them in some point-to-point protocols, some wireless LANs, and some WANs.

Review Questions

1. What are the types of frames.
2. Why character oriented protocols are used?
3. What is byte stuffing and unstuffing?
4. Explain bit oriented protocol?
5. Differentiate between character oriented and bit oriented format for Framing?

Data Link Layer Protocols

2.6 DATA LINK LAYER PROTOCOLS

- Traditionally 2 protocols have been defined for the data-link layer to deal with flow and error control: 1) Simple Protocol and 2) Stop-and-Wait Protocol.
- The behaviour of a data-link-layer protocol can be better shown as a finite state machine (FSM).
- An FSM is a machine with a finite number of states (Figure 11.6).
- The machine is always in one of the states until an event occurs.
- Each event is associated with 2 reactions:

  1) Defining the list (possibly empty) of actions to be performed.

  2) Determining the next state (which can be the same as the current state).
- One of the states must be defined as the initial state, the state in which the machine starts when it turns on.

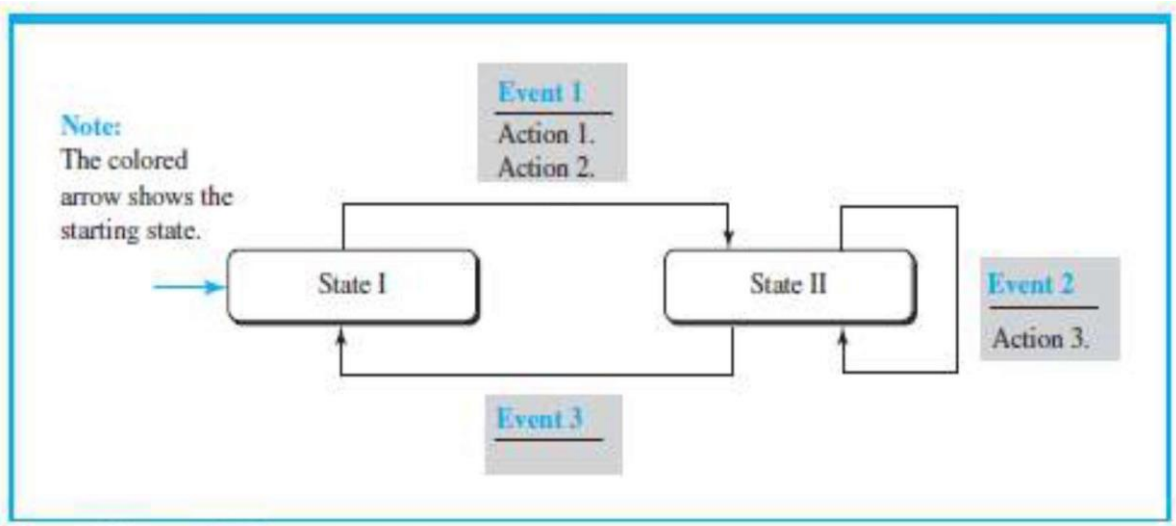


Figure 11.6 *Connectionless and connection-oriented service represented as FSMs*

2.6.1 Simplest Protocol

Assumptions:

- The protocol has no flow-control or error-control.
- The protocol is a unidirectional protocol (in which frames are traveling in only one direction).

- The receiver can immediately handle any frame it receives.

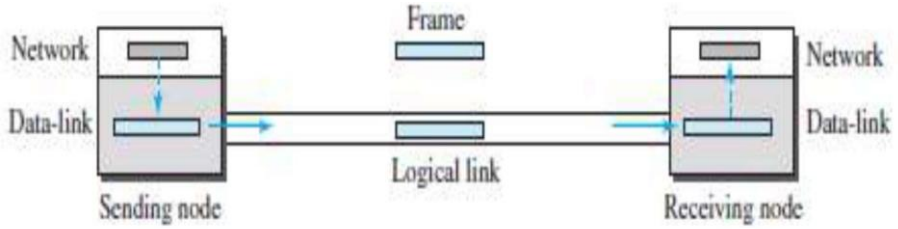


Figure 11.7 *Simple protocol*

2.6.1.1 Design

1) At Sender

- The data-link-layer
  → gets data from its network-layer
  → makes a frame out of the data and
  → sends the frame.

2) At Receiver

- The data-link-layer
  → receives a frame from its physical layer
  → extracts data from the frame and
  → delivers the data to its network-layer.
- Data-link-layers of sender & receiver provide transmission services for their network-layers.
- Data-link-layers use the services provided by their physical layers for the physical transmission of bits.

2.6.1.2 FSMs

- Two main requirements:
  1. The sender-site cannot send a frame until its network-layer has a data packet to send.
  2. The receiver-site cannot deliver a data packet to its network-layer until a frame arrives.

- The above two requirements are shown using two FSMs. Each FSM has only one state, the ready state.

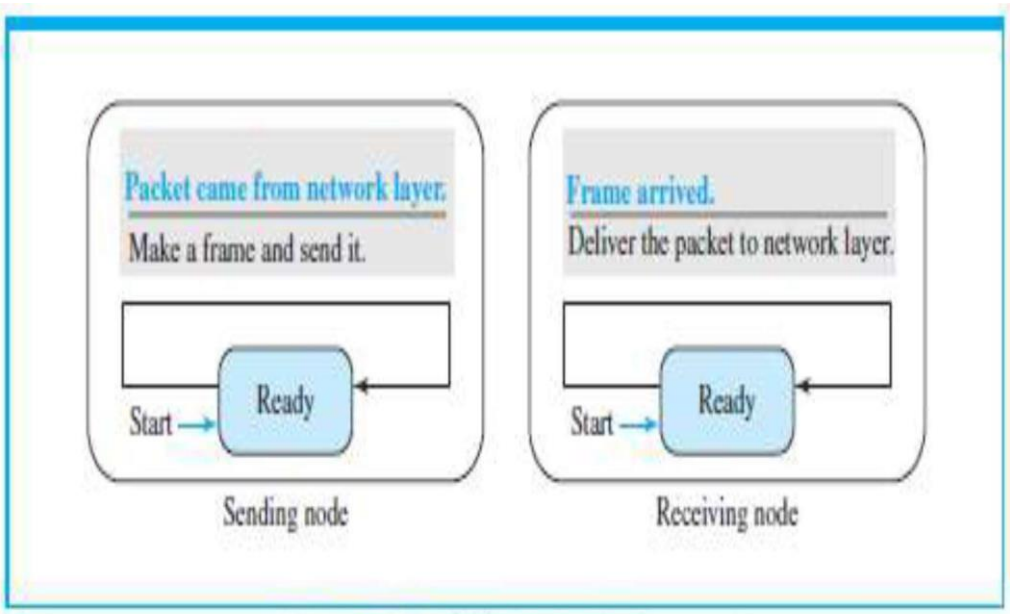


Figure 11.8 *FSMs for the simple protocol*

1) At Sending Machine

- The sending machine remains in the ready state until a request comes from the process in the network layer.
- When this event occurs, the sending machine encapsulates the message in a frame and sends it to the receiving machine.

2) At Receiving Machine

- The receiving machine remains in the ready state until a frame arrives from the sending machine.
- When this event occurs, the receiving machine decapsulates the message out of the frame and delivers it to the process at the network layer.

Example 2.6

Figure 11.9 shows an example of communication using this protocol. It is very simple. The sender sends frames one after another without even thinking about the receiver.

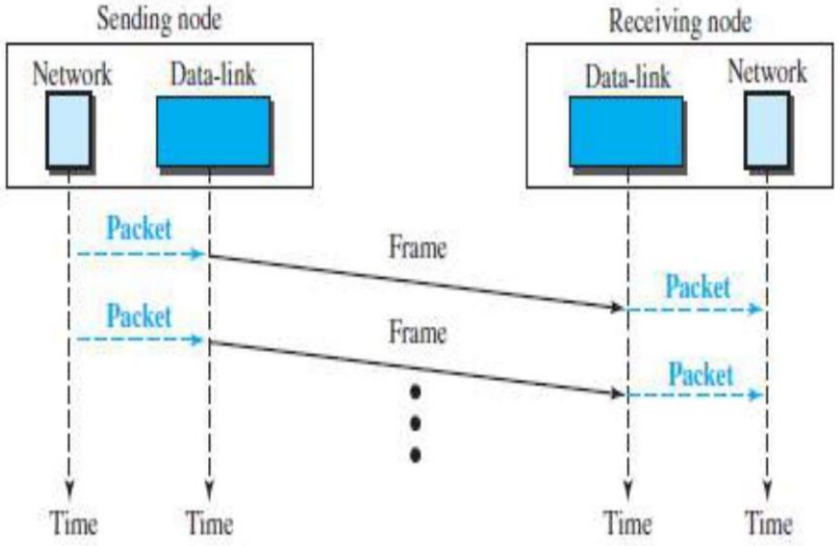
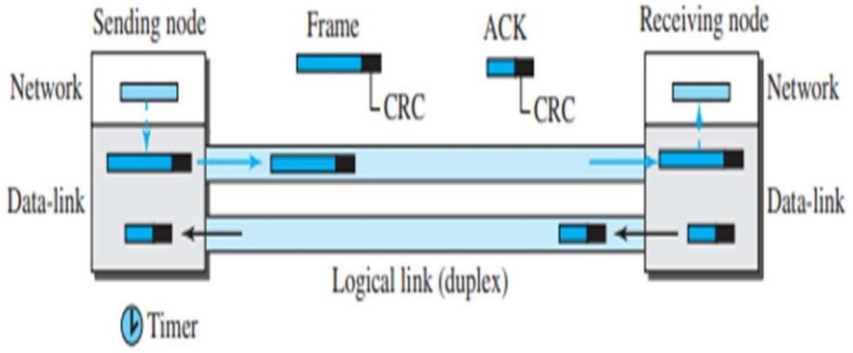


Figure 11.9 *Flow diagram*

2.6.2 Stop & Wait Protocol

- This uses both flow and error control. Normally, the receiver has limited storage-space.
- If the receiver is receiving data from many sources, the receiver may be overloaded with frames & discard the frames.
- To prevent the receiver from being overloaded with frames, we need to tell the sender to slow down.

Figure 11.10 *Stop-and-Wait protocol*



2.6.2.1 Design

1) At Sender

- The sender
  - sends one frame & starts a timer
  - keeps a copy of the sent-frame and
  - waits for ACK-frame from the receiver (okay to go ahead).
  - If an ACK-frame arrives before the timer expires, the timer is stopped and the sender sends the next frame. Also, the sender discards the copy of the previous frame.
  - If the timer expires before ACK-frame arrives, the sender resends the previous frame and restarts the timer

At Receiver

- To detect corrupted frames, a CRC is added to each data frame.
- When a frame arrives at the receiver-site, the frame is checked.
- If frame's CRC is incorrect, the frame is corrupted and discarded.
- The silence of the receiver is a signal for the sender that a frame was either corrupted or lost.

2.6.2.2 FSMs

Sender States

- Sender is initially in the ready state, but it can move between the ready and blocking state.
- **Ready State**: When the sender is in this state, it is only waiting for a packet from the network layer.
  - If a packet comes from the network layer, the sender creates a frame, saves a copy of the frame, starts the only timer and sends the frame. The sender then moves to the blocking state.

- **Blocking State**: When the sender is in this state, three events can occur:

  a) If a time-out occurs, the sender resends the saved copy of the frame and restarts the timer.

  b) If a corrupted ACK arrives, it is discarded.

  c) If an error-free ACK arrives, the sender stops the timer and discards the saved copy of the frame. It then moves to the ready state.

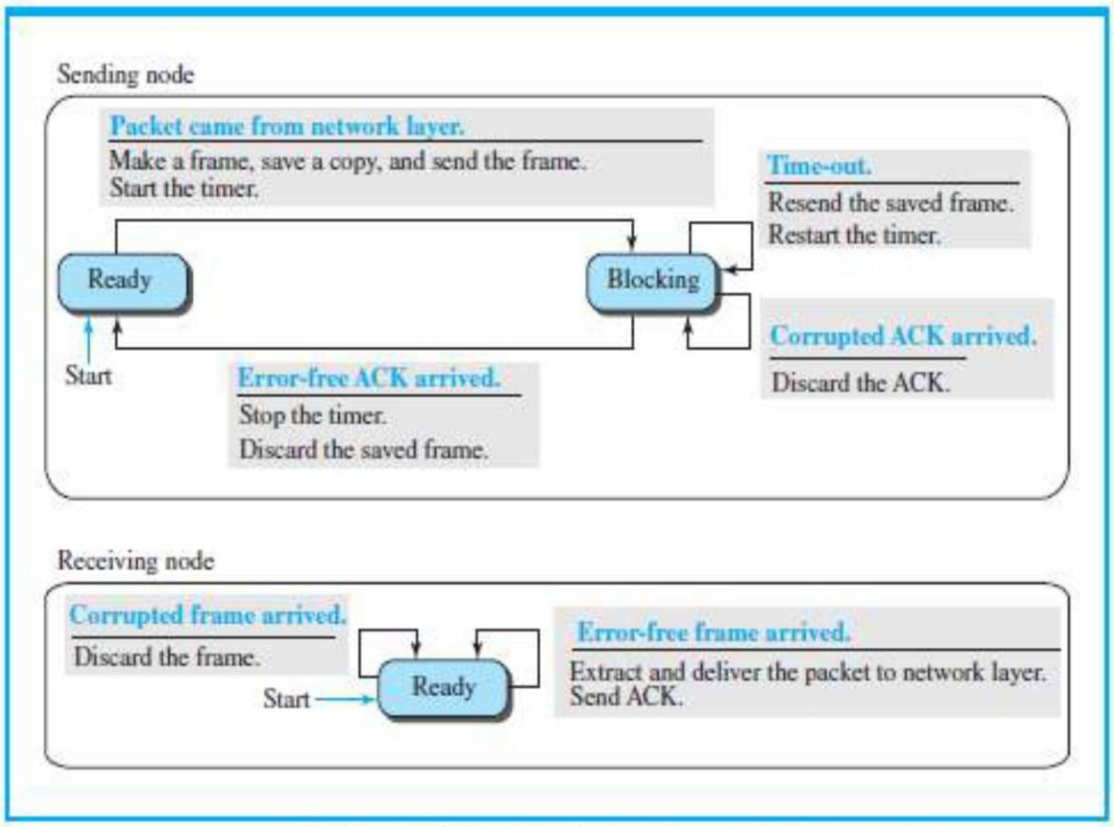


Figure 11.11 *FSM for the Stop-and-Wait protocol*

Receiver

- The receiver is always in the ready state. Two events may occur:

  a) If an error-free frame arrives, the message in the frame is delivered to the network layer and an ACK is sent.

  b) If a corrupted frame arrives, the frame is discarded.

Example:2.7

Figure 11.12 shows an example. The first frame is sent and acknowledged. The second frame is sent, but lost. After time-out, it is resent. The third frame is sent and acknowledged, but the acknowledgment is lost. The frame is resent. However, there is a problem with this scheme. The network layer at the receiver site receives two copies of the third packet, which is not right. In the next section, we will see how we can correct this problem using sequence numbers and acknowledgment numbers.

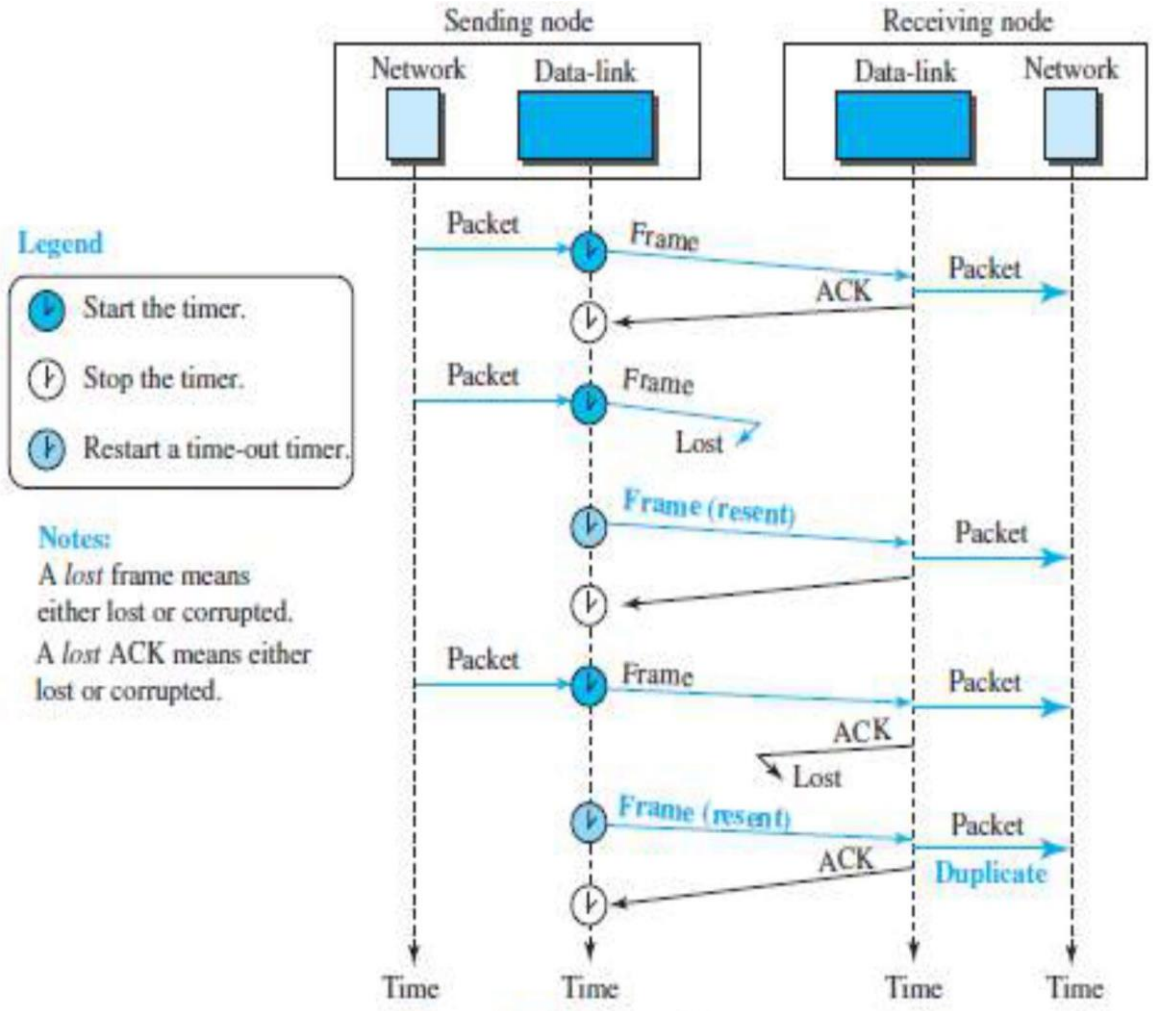


Figure 11.12 *Flow diagram*

2.6.2.3 Sequence and Acknowledgment Numbers

- If the corrupted-frame arrives at the receiver-site, then the frame is simply discarded.
- If the receiver receives out-of-order data-frame, then it means that frames were lost. The lost-frames need to be resent.

Problem in Stop and Wait protocols

- There is no way to identify a frame.
- The received-frame could be the correct one, or a duplicate, or a frame out of order.

Solution: 1) Use sequence-number for each data frame.

2) Use Acknowledgment-number for each ACK frame.

Sequence Numbers

- Frames need to be numbered. This is done by using sequence-numbers.
- A sequence-number field is added to the data-frame.
- Sequence numbers are 0, 1, 0, 1, 0, 1, . .

Acknowledgment Numbers

- An acknowledgment-number field is added to the ACK-frame.
- The acknowledgment numbers can also be 1, 0, 1, 0, 1, 0, …
- The acknowledgment-numbers always announce the sequence-number of the next frame expected by the receiver.
- For example, If frame-0 has arrived safely, the receiver sends an ACK-frame with acknowledgment-1 (meaning frame-1 is expected next).

Example 2.8

Figure 11.13 shows how adding sequence numbers and acknowledgment numbers can prevent duplicates. The first frame is sent and acknowledged. The second frame is sent, but lost. After time-out, it is resent. The third frame is sent and acknowledged, but the acknowledgment is lost. The frame is resent.

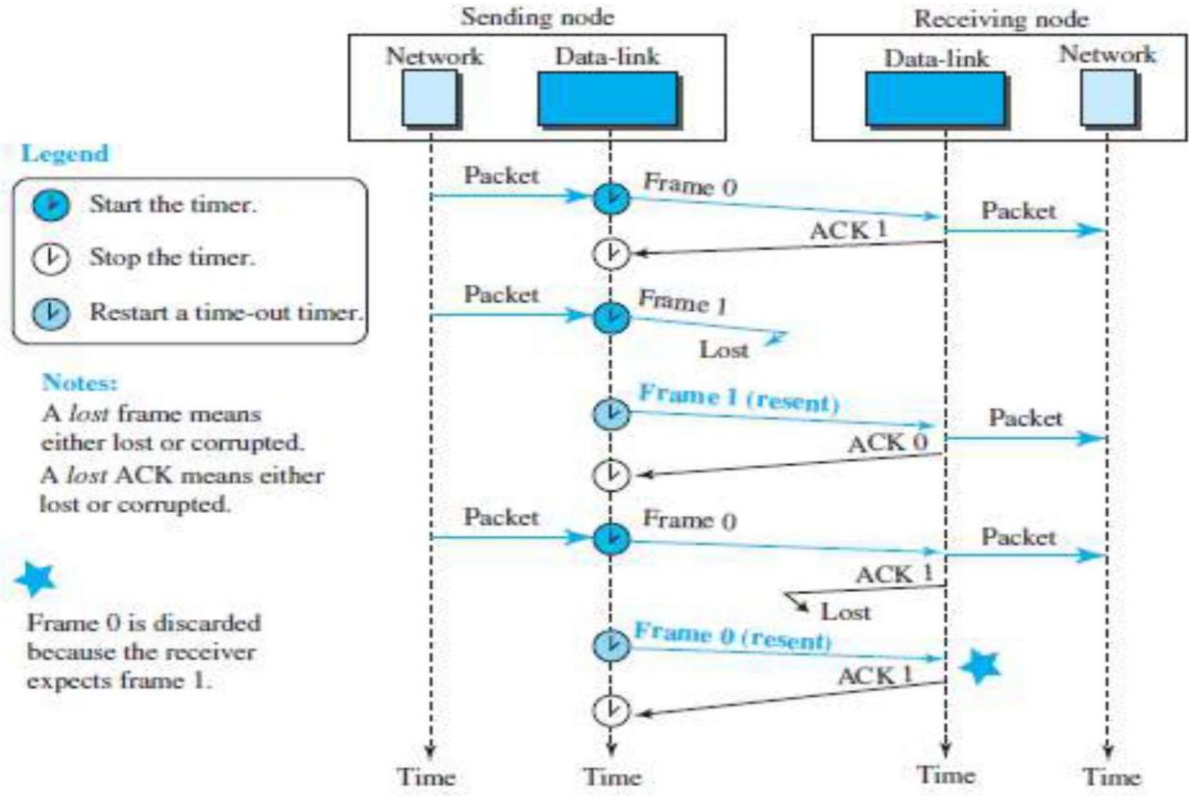


Figure 11.13 *Flow diagram*

2.6.3 Piggybacking

- A technique called piggybacking is used to improve the efficiency of the bidirectional protocols.
- The data in one direction is piggybacked with the acknowledgment in the other direction.
- In other words, when node A is sending data to node B, Node A also acknowledges the data received from node B.

Review Questions

1. What is a simplest protocol?
2. What are the functions at sender and receiver site for the simplest protocol?
3. What are the benefits of Stop-and-Wait protocol?
4. Why Piggybacking is needed?

High Level Data Link Control and Point to Point Protocol

2.7 High-Level Data Link Control (HDLC)

HDLC is a bit-oriented protocol for communication over point-to-point and multipoint links.

HDLC implements the ARQ mechanisms.

2.7.1 Configurations and Transfer Modes

- HDLC provides 2 common transfer modes that can be used in different configurations: 1) Normal response mode (NRM) 2) Asynchronous balanced mode (ABM).

NRM

- The station configuration is unbalanced (Figure 11.14).
- 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.

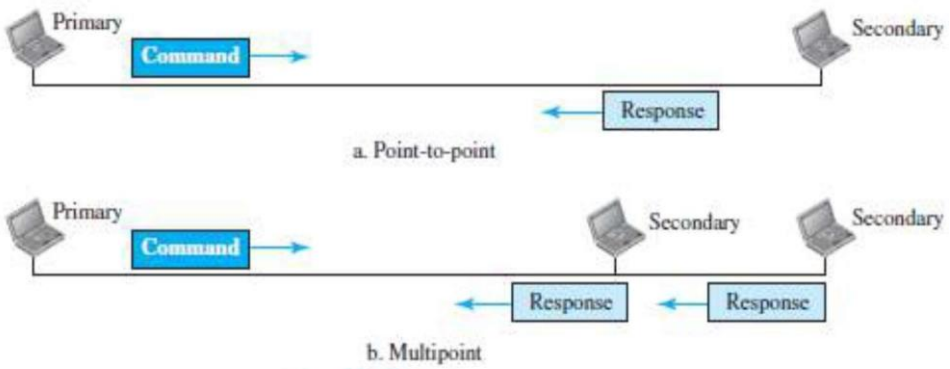


Figure 11.14 *Normal response mode*

ABM

- The configuration is balanced (Figure 11.15).
- Link is point-to-point, and each station can function as a primary and a secondary (acting as peers). This is the common mode today.

Figure 11.15 *Asynchronous balanced mode*

2.7.2 Framing

- To provide the flexibility necessary to support all the options possible in the modes and configurations, **HDLC defines three types of frames**:
  - **Information frames** (I-frames): are used to transport user data and control information relating to user data (piggybacking).
  - **Supervisory frames** (S-frames): are used only to transport control information.
  - **Unnumbered frames** (U-frames): are reserved for system management.
  - Information carried by U-frames is intended for managing the link itself.
- Each type of frame serves as an envelope for the transmission of a different type of message.

2.7.2.1 Frame Format

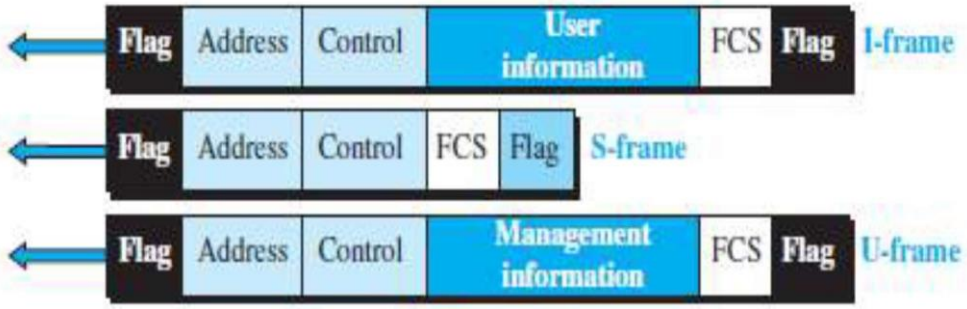


Figure 11.16 *HDLC frames*

Various fields of HDLC frame

1) Flag Field

- This field has a synchronization pattern 01111110.
- This field identifies both the beginning and the end of a frame.

2) Address Field

- This field 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.
- This field can be 1 byte or several bytes long, depending on the needs of the network.

3) Control Field

- This field is one or two bytes used for flow and error control.

4) Information Field

- This field contains the user's data from the network-layer or management information. Its length can vary from one network to another.

5) FCS Field

- This field is the error-detection field. (FCS Frame Check Sequence)
- This field can contain either a 2- or 4-byte standard CRC.

2.7.2.1.1 Control Fields of HDLC Frames

- The control field determines the type of frame and defines its functionality (Figure 11.17).

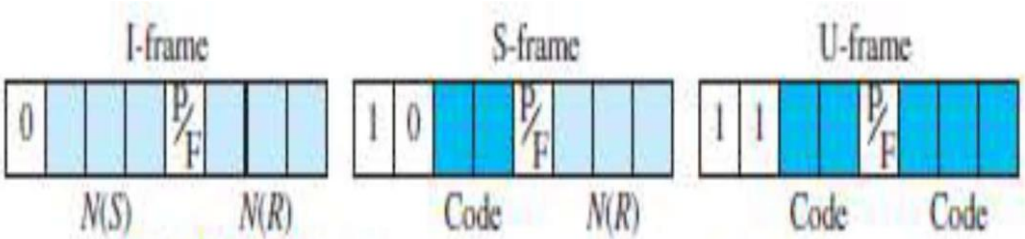


Figure 11.17 *Control field format for the different frame types*

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 subfields in the control field are:

1. The first bit defines the type. If the first bit of the control field is 0, this means the frame is an I-frame.
2. The next 3 bits N(S) define the sequence-number of the frame. With 3 bits, we can define a sequence-number between 0 and 7
3. The last 3 bits N(R) correspond to the acknowledgment-number when piggybacking is used.
4. The single bit between N(S) and N(R) is called the P/F bit.
    - The P/F field is a single bit with a dual purpose. It can mean poll or final.
        - It means poll when the frame is sent by a primary station to a secondary (when the address field contains the address of the receiver).
        - It means final when the frame is sent by a secondary to a primary (when the address field contains the address of the sender).

Control Field for S-Frames

- Supervisory frames are used for flow and error-control whenever piggybacking is either impossible or inappropriate (e.g., when the station either has no data of its own to send or needs to send a command or response other than an acknowledgment).
- S-frames do not have information fields.
- The subfields in the control field are:
    1. If the first 2 bits of the control field is 10, this means the frame is an S-frame.
    2. The last 3 bits N(R) corresponds to the acknowledgment-number (ACK) or negative acknowledgment-number (NAK).
    3. 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:

- **Receive ready (RR) = 00**
  - This acknowledges the receipt of frame or group of frames.
  - The value of N(R) is the acknowledgment-number.
- **Receive not ready (RNR) = 10**
  - This is an RR frame with 1 additional function to announces that the receiver is busy and cannot receive more frames.
  - It acts as congestion control mechanism by asking the sender to slow down.
  - The value of N(R) is the acknowledgment-number.
- **Reject (REJ) = 01**
  - It is a NAK frame used in Go-Back-N ARQ to improve the efficiency of the process.
  - It informs the sender, before the sender time expires, that the last frame is lost or damaged.
  - The value of N(R) is the negative acknowledgment-number.
- **Selective reject (SREJ) = 11**
  - This is a NAK frame used in Selective Repeat ARQ.
  - 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.
- U-frames contain an information field used for system management information, but not user data.
- Much of the information carried by U-frames is contained in codes included in the control field.

- U-frame codes are divided into 2 sections:

  i) A 2-bit prefix before the P/F bit

  ii) 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.

Example 2.9

Figure 11.18 shows how U-frames can be used for connection establishment and connection release. Node A asks for a connection with a set asynchronous balanced mode (SABM) frame; node B gives a positive response with an unnumbered acknowledgment (UA) frame. After these two exchanges, data can be transferred between the two nodes (not shown in the figure). After data transfer, node A sends a DISC (disconnect) frame to release the connection; it is confirmed by node B responding with a UA (unnumbered acknowledgment).

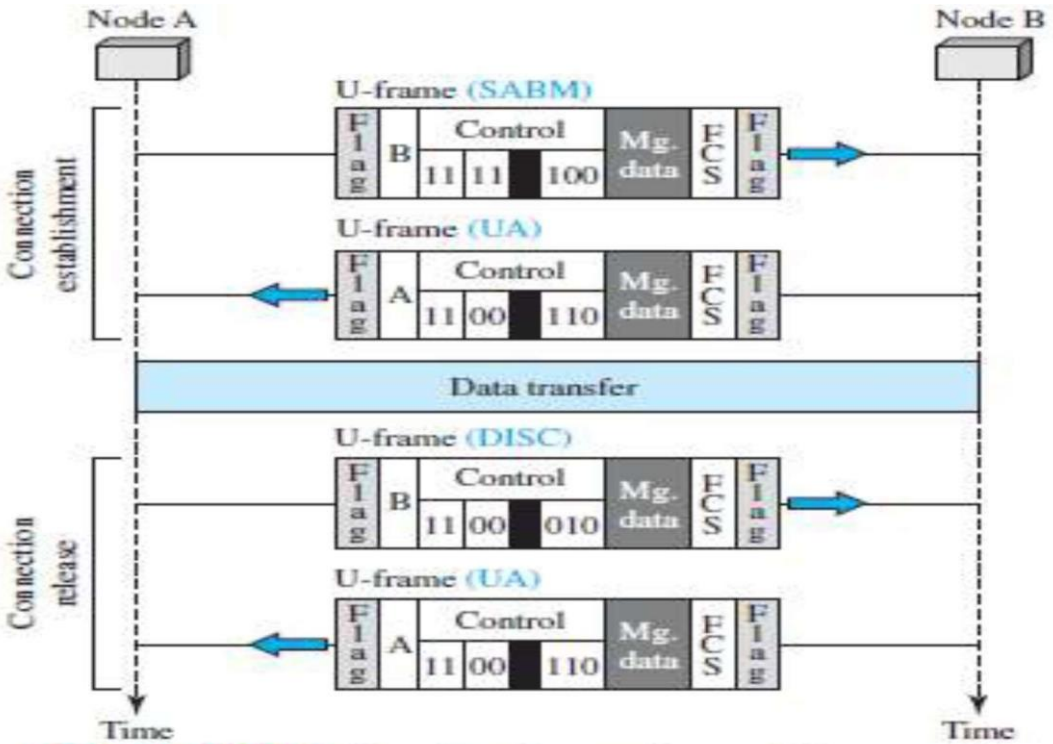


Figure 11.18 *Example of connection and disconnection*

2.8 POINT-TO-POINT PROTOCOL (PPP)

- PPP is one of the most common protocols for point-to-point access.
- Today, millions of Internet users who connect their home computers to the server of an ISP use PPP.

2.8.1 Framing

- PPP uses a character-oriented (or byte-oriented) frame (Figure 11.20).

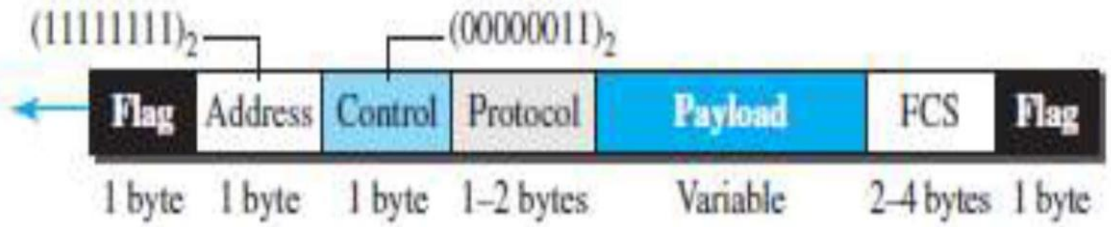


Figure 11.20 *PPP frame format*

Various fields of PPP frame

1) Flag

- This field has a synchronization pattern 01111110.
- This field identifies both the beginning and the end of a frame.

2) Address

- This field is set to the constant value 11111111 (broadcast address).

3) Control

- This field is set to the constant value 00000011 (imitating unnumbered frames in HDLC).
- PPP does not provide any flow control and Error control is also limited to error detection.

4) Protocol

- This field defines what is being carried in the payload field.
- Payload field carries either i) user data or ii) other control information.
- By default, size of this field = 2 bytes.

5) Payload field

- This field carries either i) user data or ii) other control information.
- By default, maximum size of this field = 1500 bytes.
- This field is byte-stuffed if the flag-byte pattern appears in this field.
- Padding is needed if the payload-size is less than the maximum size.

6) FCS

- This field is the PPP error-detection field.
- This field can contain either a 2- or 4-byte standard CRC.

2.8.1.1 Byte Stuffing

- Since PPP is a byte-oriented protocol, the flag in PPP is a byte that needs to be escaped whenever it appears in the data section of the frame.
- The escape byte is 01111101, which means that every time the flag like pattern appears in the data, this extra byte is stuffed to tell the receiver that the next byte is not a flag.
- Obviously, the escape byte itself should be stuffed with another escape byte.

2.8.2 Transition Phases

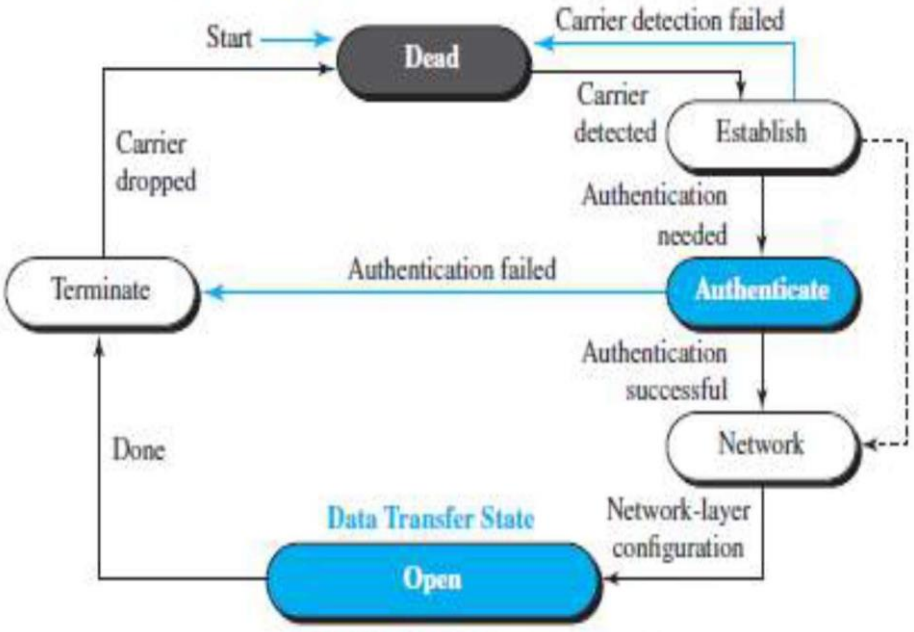


Figure 11.21 *Transition phases*

The transition diagram starts with the dead state (Figure 11.21).

1) Dead State

- In dead state, there is no active carrier and the line is quiet.

2) Establish State

- When 1 of the 2 nodes starts communication, the connection goes into the establish state.
- In establish state, options are negotiated between the two parties.

3) Authenticate State

- If the 2 parties agree that they need authentication, then the system needs to do authentication. Otherwise, the parties can simply start communication.

4) Open State

- Data transfer takes place in the open state.

5) Terminate State

- When 1 of the endpoints wants to terminate connection, the system goes to terminate state.

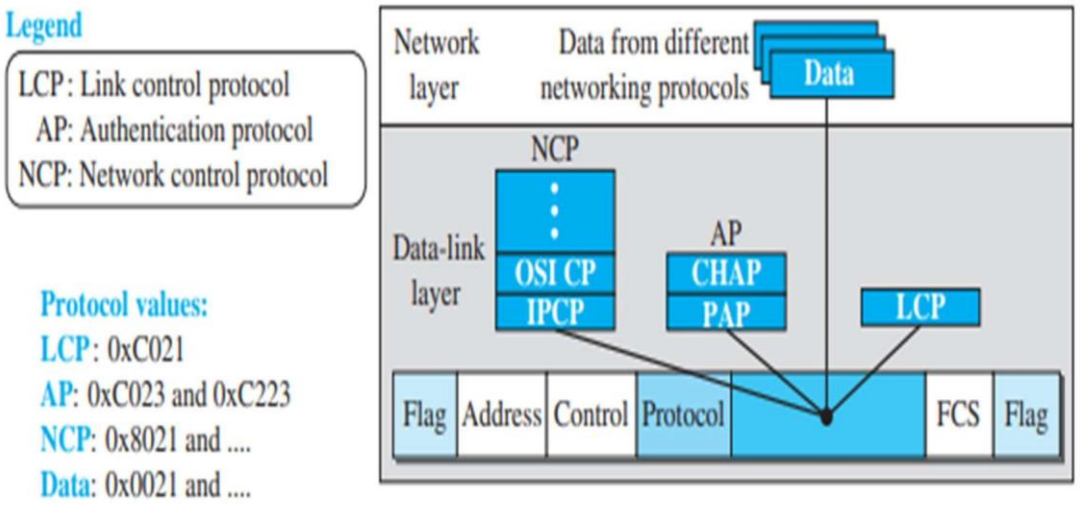
2.8.3 Multiplexing

- Although PPP is a link-layer protocol, it uses another set of protocols to establish the link, authenticate the parties involved, and carry the network-layer data.
- Three sets of protocols are defined to make PPP powerful: the Link Control Protocol (LCP), two Authentication Protocols (APs), and several Network Control Protocols (NCPs).

Link Control Protocol

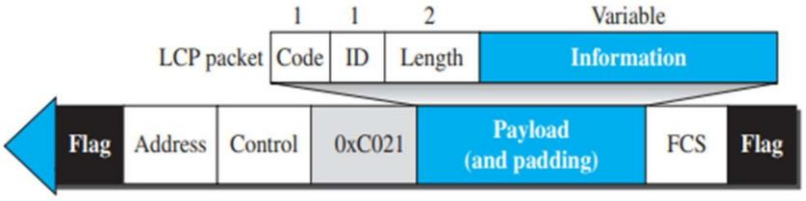
- The Link Control Protocol (LCP) is responsible for establishing, maintaining, configuring, and terminating links.
- It also provides negotiation mechanisms to set options between the two endpoints. Both endpoints of the link must reach an agreement about the options before the link can be established.

Figure 11.22 *Multiplexing in PPP*



- All LCP packets are carried in the payload field of the PPP frame with the protocol field set to C021 in hexadecimal (see Figure 11.23).

Figure 11.23 *LCP packet encapsulated in a frame*



- The code field defines the type of LCP packet. There are 11 types of packets, as shown in Table 11.1.

Table 11.1 *LCP packets*

Code	Packet Type	Description
0x01	Configure-request	Contains the list of proposed options and their values
0x02	Configure-ack	Accepts all options proposed
0x03	Configure-nak	Announces that some options are not acceptable
0x04	Configure-reject	Announces that some options are not recognized
0x05	Terminate-request	Request to shut down the line
0x06	Terminate-ack	Accept the shutdown request
0x07	Code-reject	Announces an unknown code
0x08	Protocol-reject	Announces an unknown protocol
0x09	Echo-request	A type of hello message to check if the other end is alive

Authentication Protocols

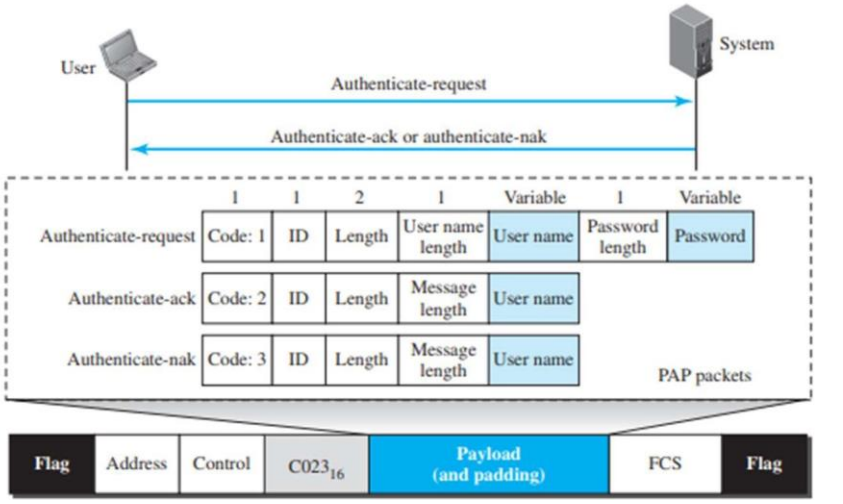
The Password Authentication Protocol (PAP)

- It is a simple authentication procedure with a two-step process:

  a. The user who wants to access a system sends an authentication identification and a password.

  b. The system checks the validity of the identification and password and either accepts or denies connection.

Figure 11.24 *PAP packets encapsulated in a PPP frame*



CHAP

- The Challenge Handshake Authentication Protocol (CHAP) is a three-way handshaking authentication protocol that provides greater security than PAP.

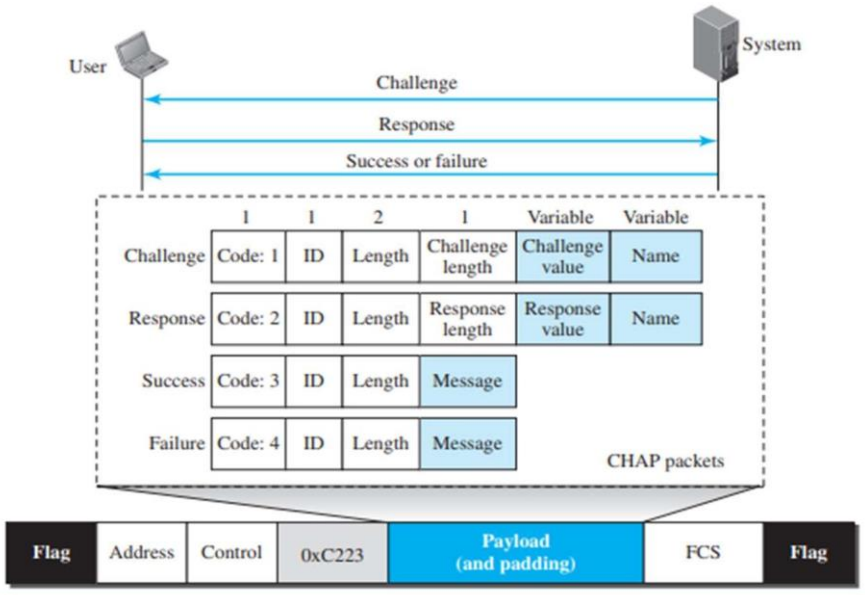
  a. The system sends the user a challenge packet containing a challenge value, usually a few bytes.

  b. The user applies a predefined function that takes the challenge value and the user's own password and creates a result. The user sends the result in the response packet to the system.

  c. The system does the same. It applies the same function to the password of the user (known to the system) and the challenge value to create a result.

If the result created is the same as the result sent in the response packet, access is granted; otherwise, it is denied.

Figure 11.25 *CHAP packets encapsulated in a PPP frame*

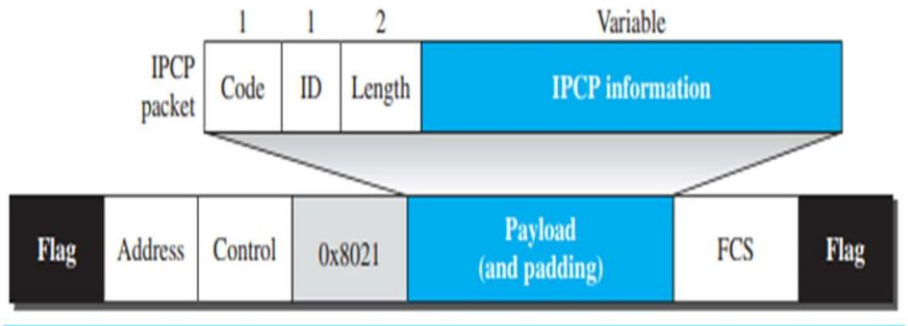


Network Control Protocols

IPCP

- One NCP protocol is the Internet Protocol Control Protocol (IPCP). This protocol configures the link used to carry IP packets in the Internet. The format of an IPCP packet is shown in Figure 11.26.

Figure 11.26 *IPCP packet encapsulated in PPP frame*



- IPCP defines seven packets, distinguished by their code values, as shown in Table 11.3.

Table 11.3 *Code value for IPCP packets*

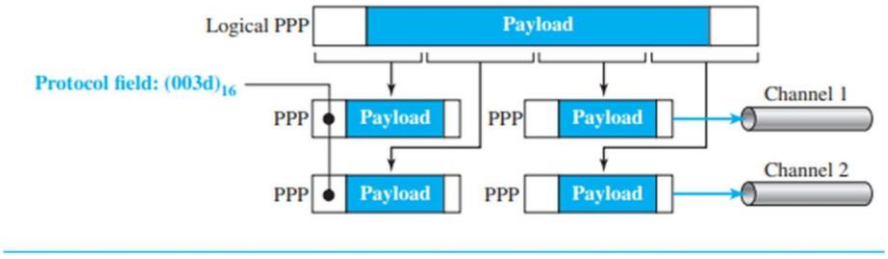
Code	*IPCP Packet*
0x01	Configure-request
0x02	Configure-ack
0x03	Configure-nak
0x04	Configure-reject
0x05	Terminate-request
0x06	Terminate-ack
0x07	Code-reject

Data from the Network Layer

Multilink PPP

- In Multilink, a logical PPP frame is divided into several actual PPP frames. A segment of the logical frame is carried in the payload of an actual PPP frame, as shown in Figure 11.28.

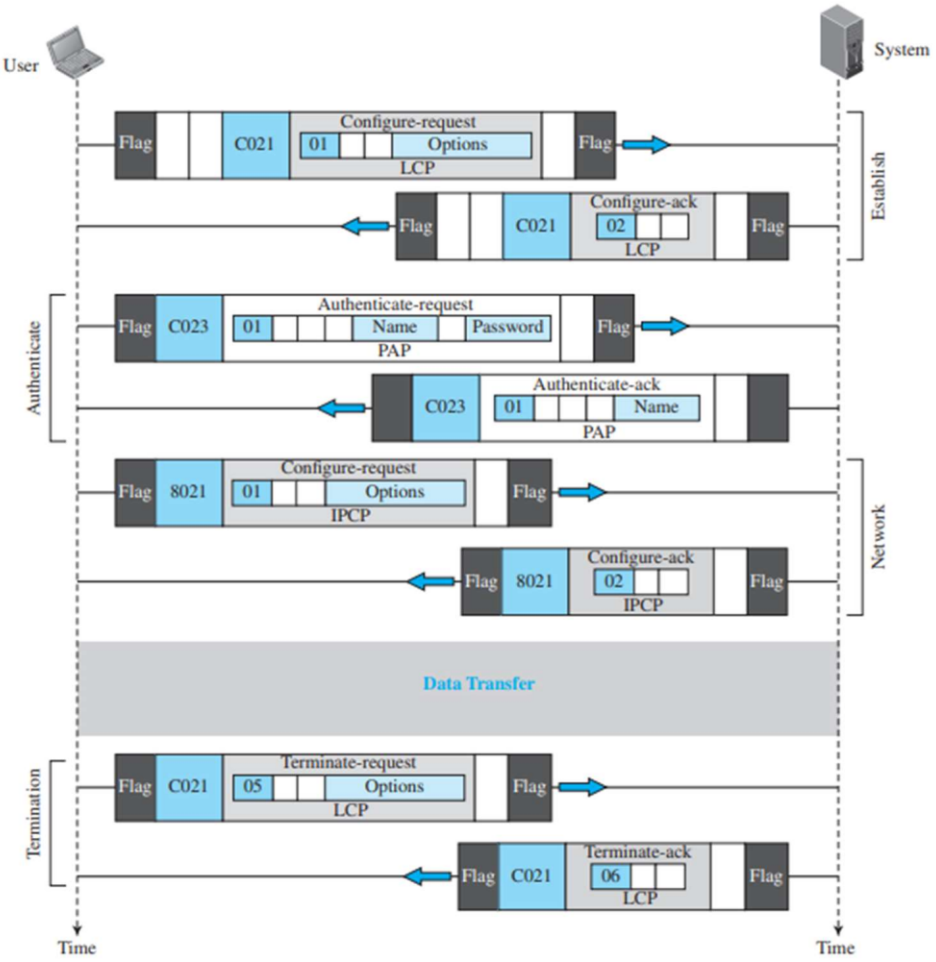
Figure 11.28 *Multilink PPP*



Example 11.7

Let us go through the phases followed by a network layer packet as it is transmitted through a PPP connection. Figure 11.29 shows the steps. For simplicity, we assume unidirectional movement of data from the user site to the system site (such as sending an e-mail through an ISP).

Figure 11.29 *An example*



Review Questions

1. What is the significance of HDLC frame format?
2. What types of frame are used in HDLC?
3. What is the frame structure of PPP protocol?
4. What are functions in transition phase of Point-to-Point Protocol?

Media Access Control: Random Access

- When nodes use shared-medium, we need multiple-access protocol to coordinate access to medium.
- Many protocols have been designed to handle access to a shared-link (Figure 12.1).

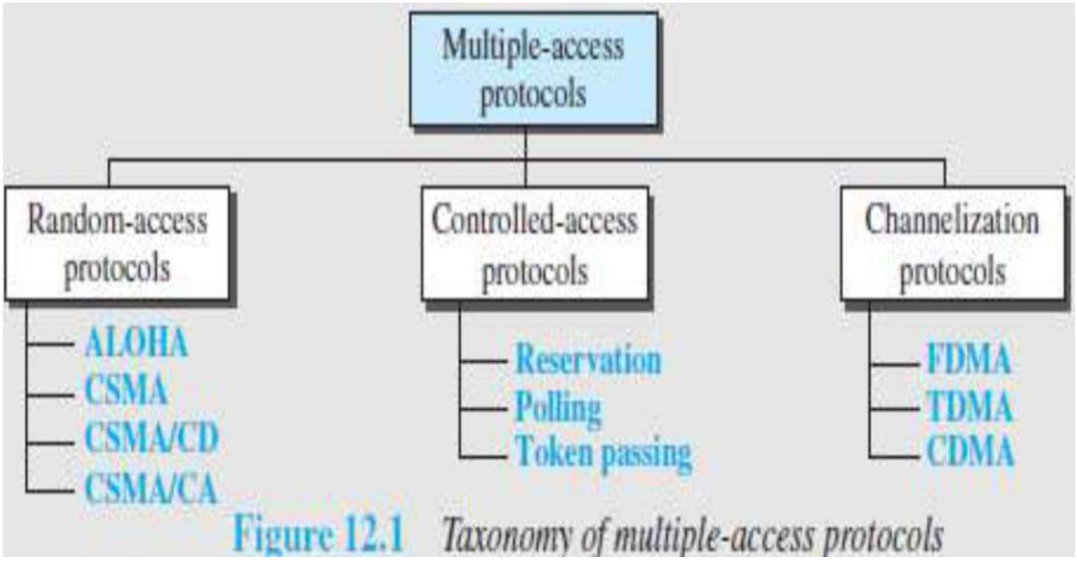


Figure 12.1 *Taxonomy of multiple-access protocols*

- These protocols belong to a sublayer in the data-link layer called Media Access Control (MAC).
- Four random-access protocols (or Contention Methods)
  i) ALOHA ii) CSMA iii) CSMA/CD iv) CSMA/CA
- These protocols are mostly used in LANs and WANs.
- Three controlled-access protocols:
  i) Reservation ii) Polling iii) Token-passing

2.9 RANDOM ACCESS PROTOCOL

- No station is superior to another station and No station is assigned control over other station.
- To send the data, a station uses a procedure to make a decision on whether or not to send. This decision depends on the state of the medium: idle or busy.

- This is called Random Access because, Transmission is random among the stations and there is no scheduled-time for a station to transmit.
- This is called Contention Method because, Stations compete with one another to access the medium.
- If more than one station tries to send, there is an access-conflict (i.e. collision) and the frames will be destroyed.
- Each station follows a procedure that answers the following questions:
  1) When can the station access the medium?
  2) What can the station do if the medium is busy?
  3) How can the station determine the success or failure of the transmission?
  4) What can the station do if there is a collision?
- Four random-access protocols (or Contention methods):
  1) ALOHA
  2) CSMA (Carrier Sense Multiple Access)
  3) CSMA/CD (Carrier Sense Multiple Access with Collision-detection)
  4) CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance)

2.9.1 ALOHA

- ALOHA was designed for a wireless LAN, but it can be used on any shared medium.
- Since the medium is shared between the stations, there is possibility of collisions.
- When 2 or more stations send the data simultaneously, there is possibility of collision & data loss.

2.9.1.1 Pure ALOHA

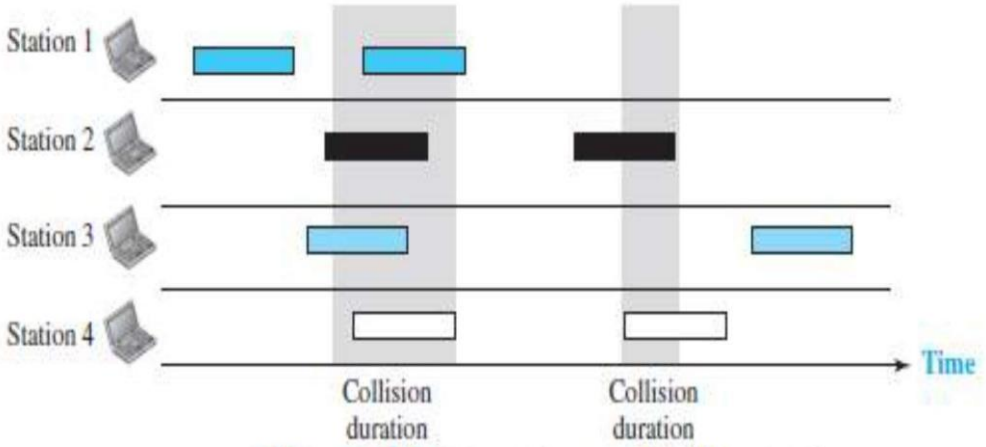


Figure 12.2 *Frames in a pure ALOHA network*

1) The sender sends a frame & starts the timer.
2) The receiver receives the frame and responds with an acknowledgment.
3) If the acknowledgment does not arrive after a time-out period, the sender resends the frame. The sender assumes that the frame (or the acknowledgment) has been destroyed.
4) Since the medium is shared between the stations, there is possibility of collisions.
5) If two stations try to resend the frames after the time-out, the frames will collide again.

Two methods to deal with collision:

1. Randomness

- When the time-out period passes, each station waits a random amount of time before resending the frame. This time is called back-off time TB.
- The randomness will help avoid more collisions.

2. Limit Maximum Retransmission

- This method prevents congestion by reducing the number of retransmitted frames.
- After a maximum number of retransmission-attempts Kmax, a station must give up and try later (Figure 12.3).

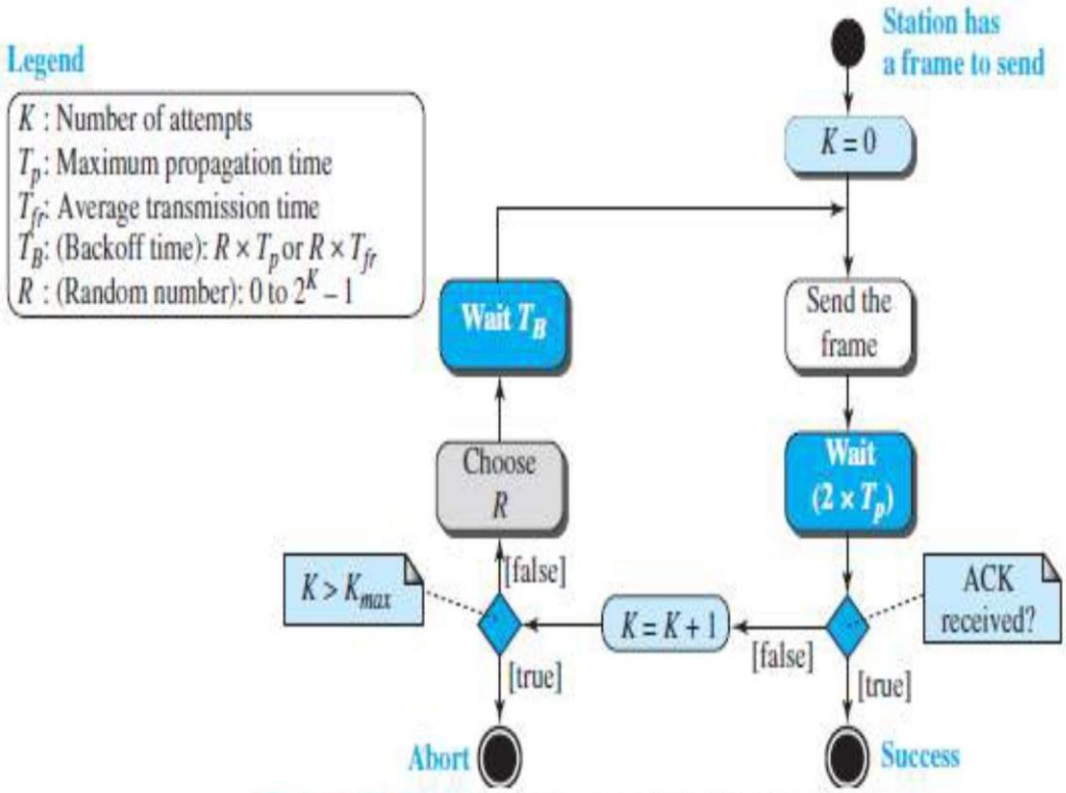


Figure 12.3 *Procedure for pure ALOHA protocol*

2.9.1.1.1 Vulnerable time

The vulnerable-time is defined as a time during which there is a possibility of collision.

Pure ALOHA vulnerable time = $2 \times T_{fr}$

where T_{fr} = Frame transmission time.

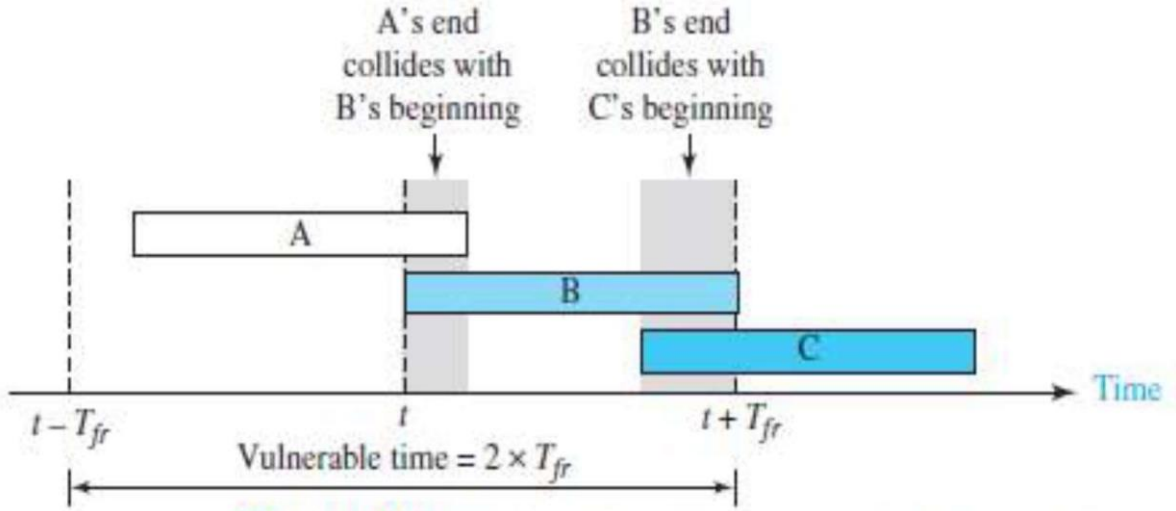


Figure 12.4 *Vulnerable time for pure ALOHA protocol*

- If station B sends a frame between t-T_{fr} and t, this leads to a collision between the frames from station A and station B.
- If station C sends a frame between t and t+T_{fr}, this leads to a collision between the frames from station A and station C.

Example 2.10

A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the requirement to make this frame collision-free?

Solution

Average frame transmission time T_{fr} is 200 bits/200 kbps or 1 ms. The vulnerable time is 2×1 ms = 2 ms. This means no station should send later than 1 ms before this station starts transmission and no station should start sending during the period (1 ms) that this station is sending.

2.9.1.1.2 Throughput

- The average number of successful transmissions is given by

$$S = G \times e^{-2G}$$

  where G = average no. of frames in one frame transmission time (T_{fr})
- For G = 1, the maximum throughput Smax = 0.184.
- In other words, out of 100 frames, 18 frames reach their destination successfully.

Example 2.11

A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the throughput if the system (all stations together) produces

a. 1000 frames per second? b. 500 frames per second? c. 250 frames per second?

Solution

The frame transmission time is 200/200 kbps or 1 ms.

a. If the system creates 1000 frames per second, or 1 frame per millisecond, then $G = 1$. In this case $S = G \times e^{-2G} = 0.135$ (13.5 percent). This means that the throughput is $1000 \times 0.135 = 135$ frames. Only 135 frames out of 1000 will probably survive.

b. If the system creates 500 frames per second, or 1/2 frames per millisecond, then $G = 1/2$. In this case $S = G \times e^{-2G} = 0.184$ (18.4 percent). This means that the throughput is $500 \times 0.184 = 92$ and that only 92 frames out of 500 will probably survive. Note that this is the *maximum* throughput case, percentagewise.

c. If the system creates 250 frames per second, or 1/4 frames per millisecond, then $G = 1/4$. In this case $S = G \times e^{-2G} = 0.152$ (15.2 percent). This means that the throughput is $250 \times 0.152 = 38$. Only 38 frames out of 250 will probably survive.

2.9.1.2 Slotted ALOHA

- Slotted ALOHA was invented to improve the efficiency of pure ALOHA.
- The time is divided into time-slots of T_{fr} seconds (Figure 12.5).
- The stations are allowed to send only at the beginning of the time-slot.

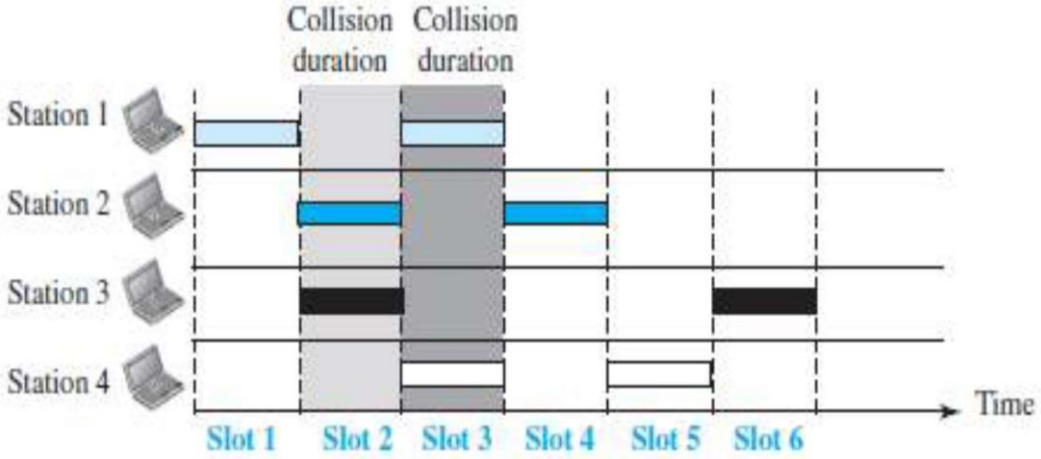


Figure 12.5 *Frames in a slotted ALOHA network*

- If a station misses the time-slot, the station must wait until the beginning of the next time-slot.
- If 2 stations try to resend at beginning of the same time-slot, the frames will collide again (Fig 12.6).
- The vulnerable time is given by: **vulnerable time= T_{fr}**

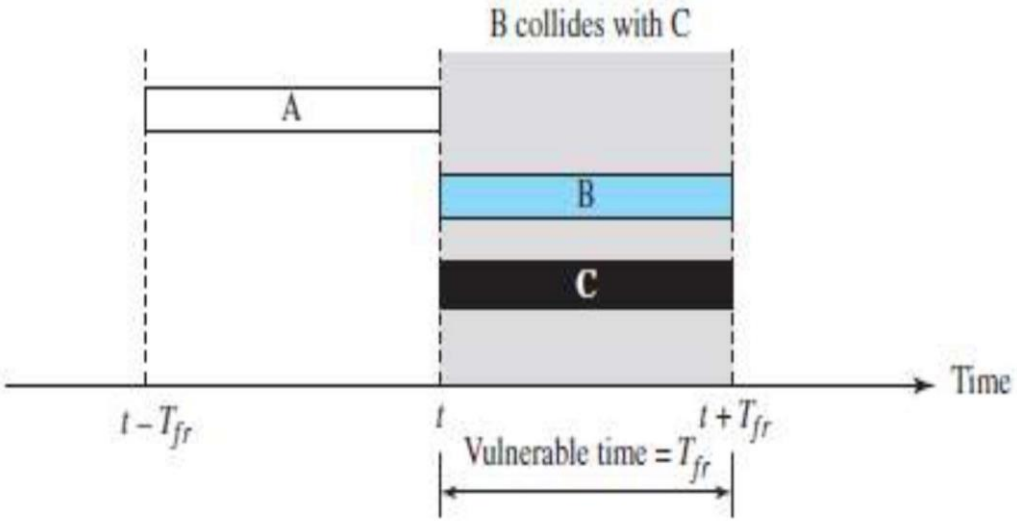


Figure 12.6 *Vulnerable time for slotted ALOHA protocol*

2.9.1.2.1 Throughput

- The average number of successful transmissions is given by

$$S = G \times e^{-G}$$

- For G = 1, the maximum throughput Smax = 0.368.
- In other words, out of 100 frames, 36 frames reach their destination successfully.

Example 2.12

A slotted ALOHA network transmits 200-bit frames using a shared channel with a 200-kbps bandwidth. Find the throughput if the system (all stations together) produces

a. 1000 frames per second.
b. 500 frames per second.
c. 250 frames per second.

Solution

This situation is similar to the previous exercise except that the network is using slotted ALOHA instead of pure ALOHA. The frame transmission time is 200/200 kbps or 1 ms.

a. In this case G is 1. So $S = G \times e^{-G} = 0.368$ (36.8 percent). This means that the throughput is $1000 \times 0.0368 = 368$ frames. Only 368 out of 1000 frames will probably survive. Note that this is the maximum throughput case, percentagewise.
b. Here G is 1/2. In this case $S = G \times e^{-G} = 0.303$ (30.3 percent). This means that the throughput is $500 \times 0.0303 = 151$. Only 151 frames out of 500 will probably survive.
c. Now G is 1/4. In this case $S = G \times e^{-G} = 0.195$ (19.5 percent). This means that the throughput is $250 \times 0.195 = 49$. Only 49 frames out of 250 will probably survive.

2.9.2 CSMA

- CSMA was developed to minimize the chance of collision and, therefore, increase the performance.
- CSMA is based on the principle “sense before transmit” or “listen before talk.”

1) Each station checks the state of the medium: idle or busy.

2) i) If the medium is idle, the station sends the data.

ii) If the medium is busy, the station defers sending.

- CSMA can reduce the possibility of collision, but it cannot eliminate it.

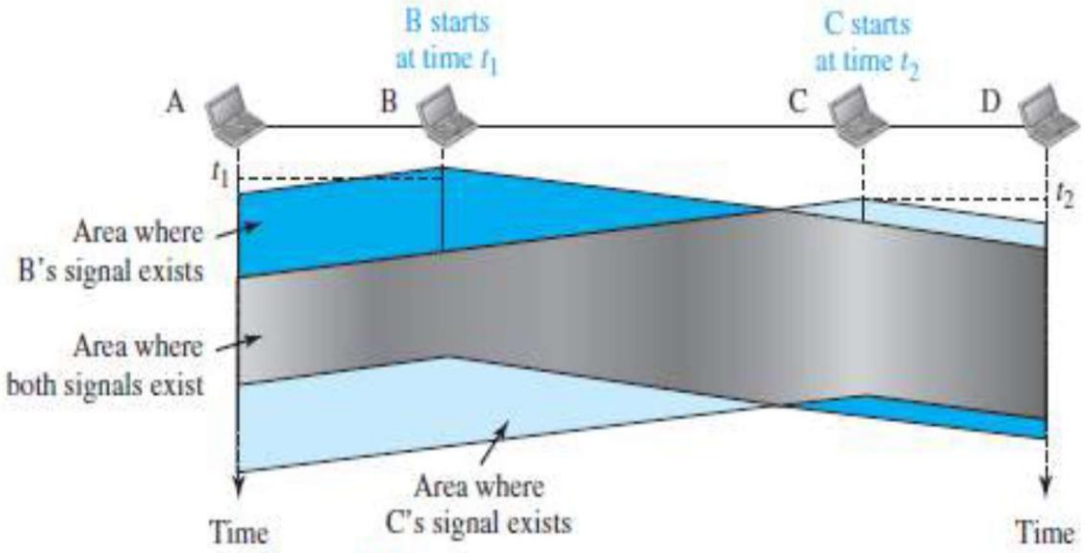


Figure 12.7 *Space/time model of a collision in CSMA*

- The possibility of collision still exists.
  - For example: When a station sends a frame, it still takes time for the first bit to reach every station and for every station to sense it.
  - For example: In Figure 12.7,
    - At time t1, station B senses & finds the medium idle, so sends a frame.
    - At time t2, station C senses & finds the medium idle, so sends a frame.
    - The 2 signals from both stations B & C collide and both frames are destroyed.

2.9.2.1 Vulnerable Time

- The vulnerable time is the propagation time Tp (Figure 12.8).
- The propagation time is the time needed for a signal to propagate from one end of the medium to the other.

- Collision occurs when a station sends a frame, and other station also sends a frame during propagation time.
- If the first bit of the frame reaches the end of the medium, every station will refrain from sending.

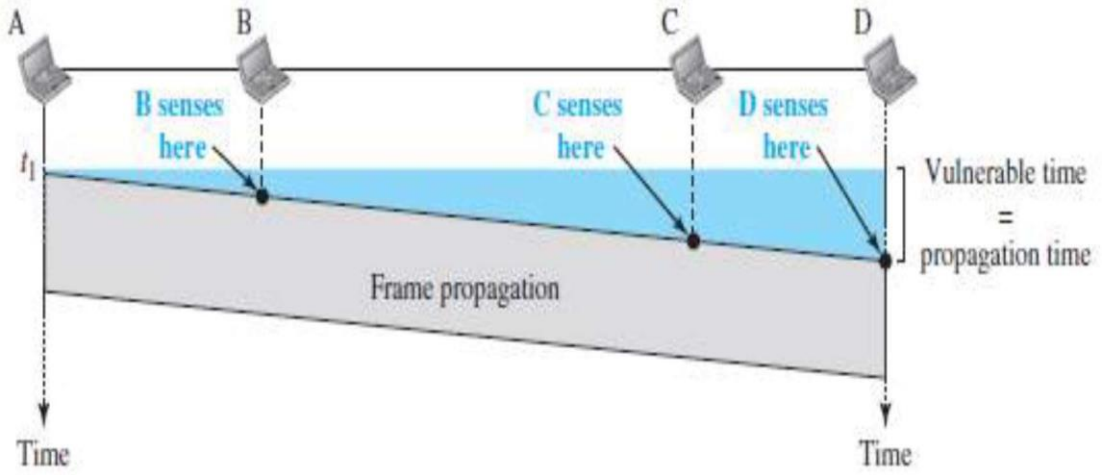


Figure 12.8 *Vulnerable time in CSMA*

2.9.2.2 Persistence Methods

Q: What should a station do if the channel is busy or idle?

Three methods can be used to answer this question:

1) 1-persistent method.

2) Non-persistent method.

3) p-persistent method.

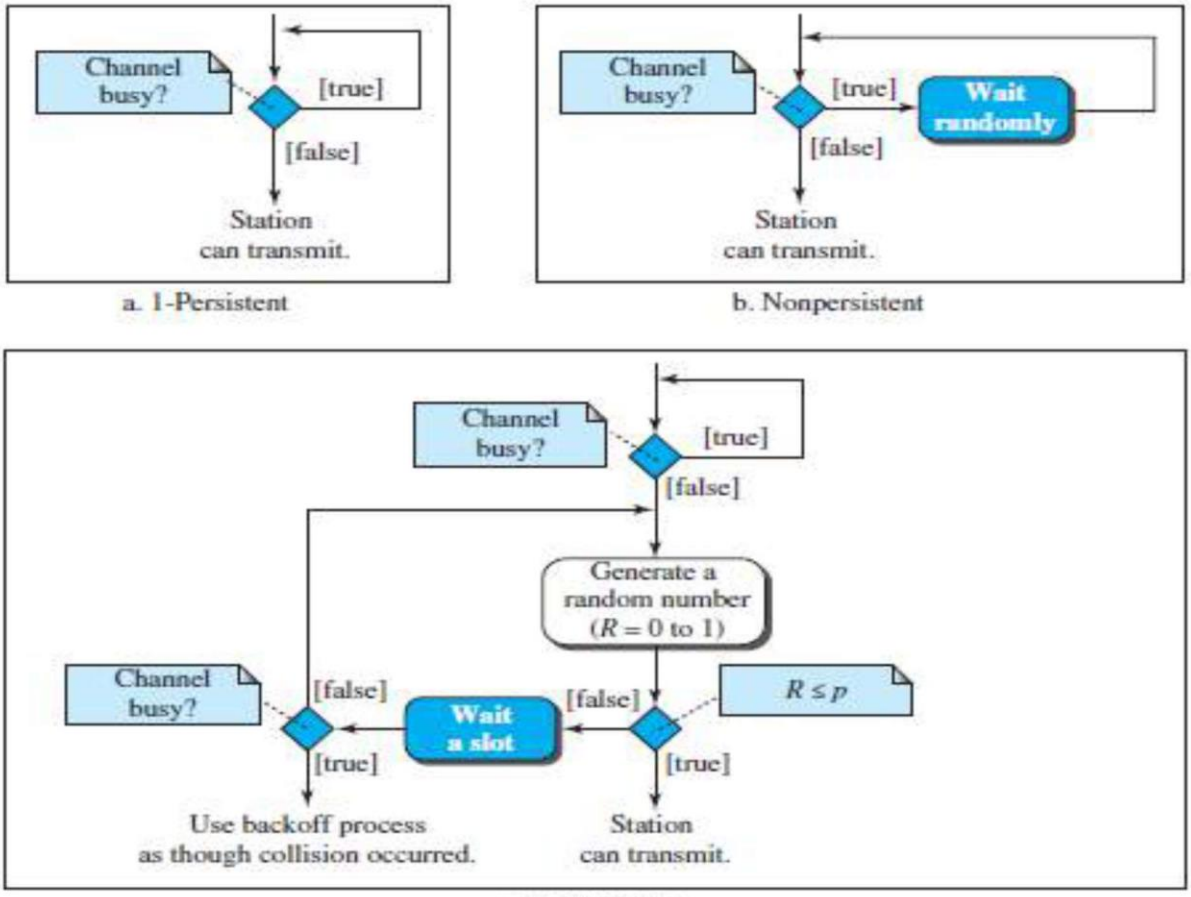


Figure 12.10 *Flow diagram for three persistence methods*

1-Persistent

- Before sending a frame, a station senses the line (Figure 12.10a).

  i) If the line is idle, the station sends immediately (with probability = 1).

  ii) If the line is busy, the station continues sensing the line.

- This method has the highest chance of collision because 2 or more stations:

  → may find the line idle and

  → send the frames immediately.

Non-persistent

- Before sending a frame, a station senses the line (Figure 12.10b).

  i) If the line is idle, the station sends immediately.

  ii) If the line is busy, the station waits a random amount of time and then senses the line again.

- This method reduces the chance of collision because 2 or more stations:
  → will not wait for the same amount of time and
  → will not retry to send simultaneously.

P-Persistent

- This method is used if the channel has time-slots with a slot-duration equal to or greater than the maximum propagation time (Figure 12.10c).
- Advantages:
  i) It combines the advantages of the other 2 methods.
  ii) It reduces the chance of collision and improves efficiency.
- After the station finds the line idle, it follows these steps:
  1) With probability p, the station sends the frame.
  2) With probability q=1-p, the station waits for the beginning of the next time-slot and checks the line again.
  i) If line is idle, it goes to step 1.
  ii) If line is busy, it assumes that collision has occurred and uses the back off procedure.

2.9.3 CSMA/CD

- Disadvantage of CSMA: CSMA does not specify the procedure after a collision has occurred.
- Solution: CSMA/CD enhances the CSMA to handle the collision.

1) A station sends the frame & then monitors the medium to see if the transmission was successful or not.
2) If the transmission was unsuccessful (i.e. there is a collision), the frame is sent again.

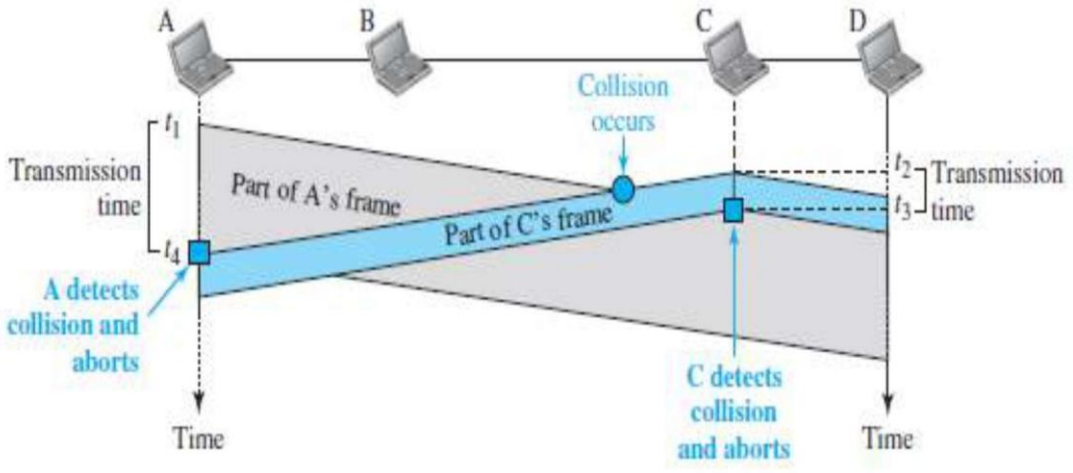


Figure 12.12 *Collision and abortion in CSMA/CD*

- At time t_1, station A has executed its procedure and starts sending the bits of its frame.
- At time t_2, station C has executed its procedure and starts sending the bits of its frame.
- The collision occurs sometime after time t_2.
- Station C detects a collision at time t_3 when it receives the first bit of A's frame. Station C immediately aborts transmission.
- Station A detects collision at time t_4 when it receives the first bit of C's frame. Station A also immediately aborts transmission.
- Station A transmits for the duration $t_4 – t_1$.
- Station C transmits for the duration $t_3 - t_2$.
- For the protocol to work: The length of any frame divided by the bit rate must be more than either of these durations.

2.9.3.1 Minimum Frame Size

- For CSMA/CD to work, we need to restrict the frame-size.
- Before sending the last bit of the frame, the sender must detect a collision and abort the transmission.
- This is so because the sender does not keep a copy of the frame and does not monitor the line for collision-detection.

- Frame transmission time T_{fr} is given by

$T_{fr}=2T_p$ where T_p=maximum propagation time

Example 2.13

A network using CSMA/CD has a bandwidth of 10 Mbps. If the maximum propagation time (including the delays in the devices and ignoring the time needed to send a jamming signal, as we see later) is 25.6 μs, what is the minimum size of the frame?

Solution

The minimum frame transmission time is $T_{fr} = 2 \times T_p = 51.2$ μs. This means, in the worst case, a station needs to transmit for a period of 51.2 μs to detect the collision. The minimum size of the frame is 10 Mbps × 51.2 μs = 512 bits or 64 bytes. This is actually the minimum size of the frame for Standard Ethernet, as we will see later in the chapter.

2.9.3.2 Procedure

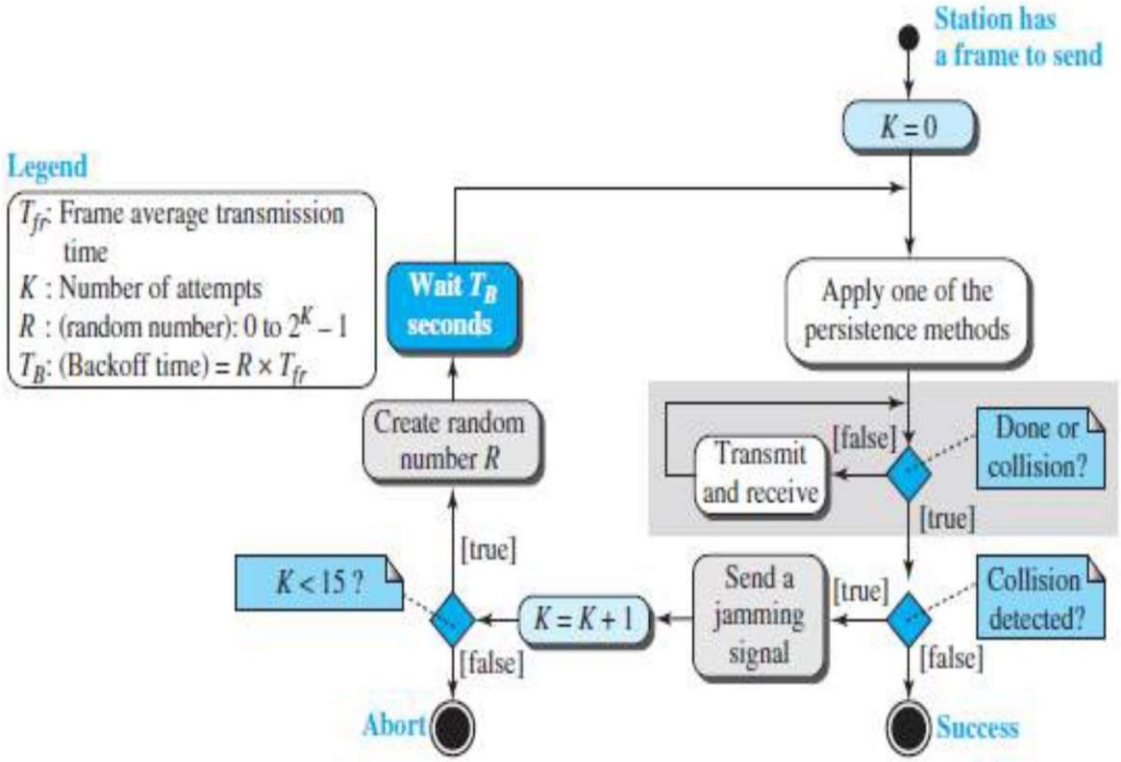


Figure 12.13 *Flow diagram for the CSMA/CD*

- CSMA/CD is similar to ALOHA with 2 differences (Figure 12.13):

1) Addition of the persistence process.
    - We need to sense the channel before sending the frame by using non-persistent, 1- persistent or p-persistent.

2) Frame transmission.
    - In ALOHA, first the entire frame is transmitted and then acknowledgment is waited for.
    - In CSMA/CD, transmission and collision-detection is a continuous process.

2.9.3.3 Energy Level

- In a channel, the energy-level can have 3 values: 1) Zero 2) Normal and 3) Abnormal.
    1) At zero level, the channel is idle (Figure 12.14).
    2) At normal level, a station has successfully captured the channel and is sending its frame.
    3) At abnormal level, there is a collision and the level of the energy is twice the normal level.
- A sender needs to monitor the energy-level to determine if the channel is Idle or Busy or in Collision mode.

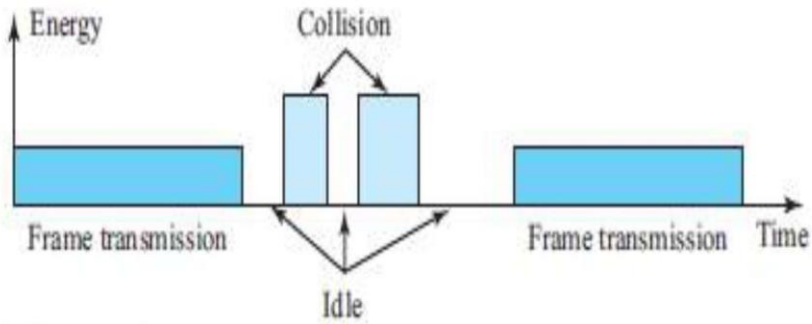
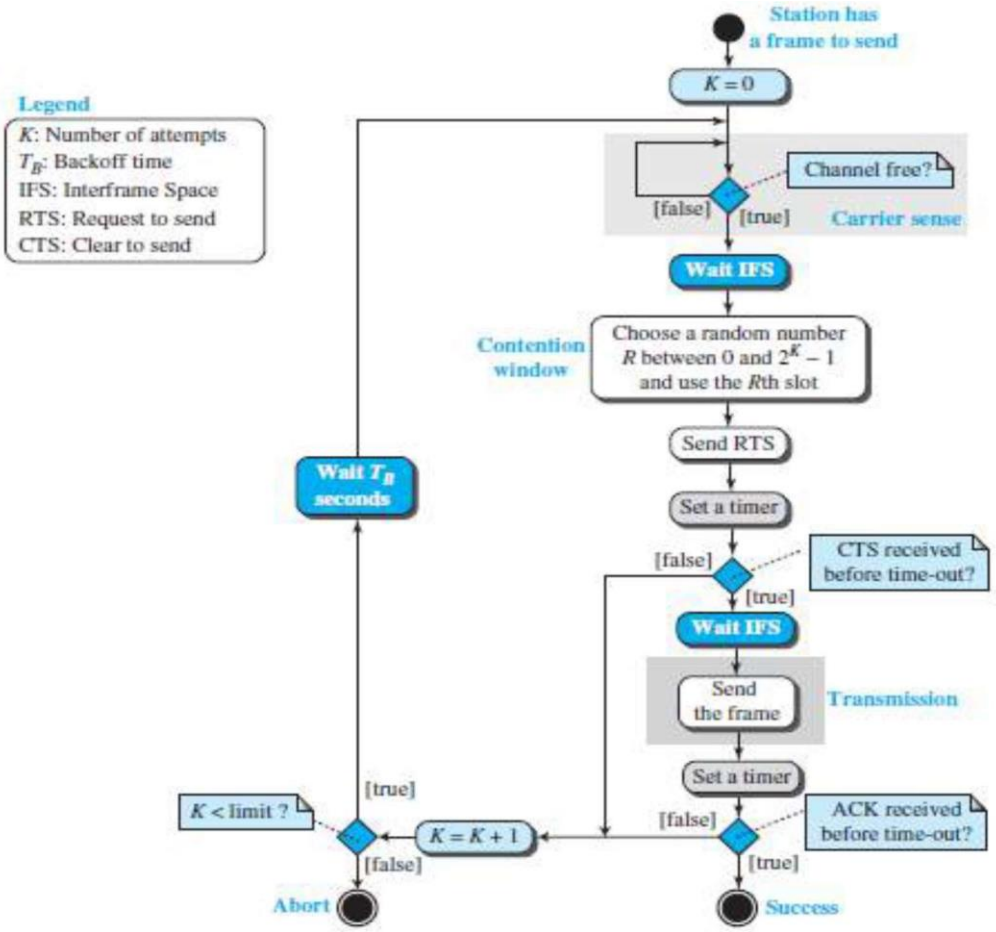


Figure 12.14 *Energy level during transmission, idleness, or collision*

2.9.3.4 Throughput

- The throughput of CSMA/CD is greater than pure or slotted ALOHA.
- The maximum throughput is based on different value of G , persistence method used (non-persistent, 1-persistent, or p-persistent) and 'p" value in the p-persistent method.
- For 1-persistent method, the maximum throughput is 50% when G =1.
- For non-persistent method, the maximum throughput is 90% when G is between 3 and 8.

2.9.4 CSMA/CA



1) A station needs to be able to receive while transmitting to detect a collision.

i) When there is no collision, the station receives one signal: its own signal.

ii) When there is a collision, the station receives 2 signal, Its own signal and Signal transmitted by a second station.

2) To distinguish b/w these 2 cases, the received signals in these 2 cases must be different.

- CSMA/CA was invented to avoid collisions on wireless networks.
- Three methods to avoid collisions (Figure 12.16):

  1) Interframe space

  2) Contention window

  3) Acknowledgments

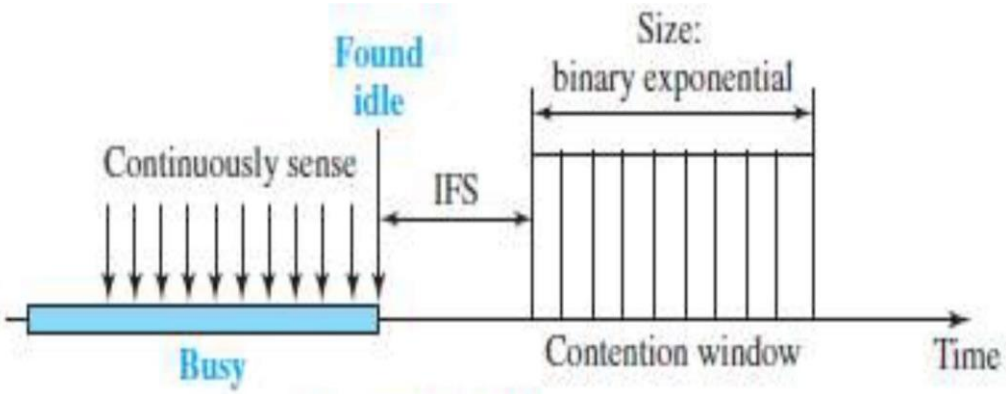


Figure 12.16 *Contention window*

1) Interframe Space (IFS)

- Collisions are avoided by deferring transmission even if the channel is found idle.
- When the channel is idle, the station does not send immediately.
- Rather, the station waits for a period of time called the inter-frame space or IFS. After the IFS time, if the channel is still idle, then, the station waits for the contention-time & finally, the station sends the frame.
- IFS variable can also be used to prioritize stations or frame types.
- For example, a station that is assigned a shorter IFS has a higher priority.

2) Contention Window

- The contention-window is an amount of time divided into time-slots.
- A ready-station chooses a random-number of slots as its wait time.

- In the window, the number of slots changes according to the binary exponential back-off strategy.
- **For example**: At first time, number of slots is set to one slot and Then, number of slots is doubled each time if the station cannot detect an idle channel.

3) Acknowledgment

- There may be a collision resulting in destroyed-data.
- In addition, the data may be corrupted during the transmission.
- To help guarantee that the receiver has received the frame, we can use

  i) Positive acknowledgment.

  ii) Time-out timer.

2.9.4.1 Frame Exchange Time Line

- Two control frames are used:

  1) Request to send (RTS)

  2) Clear to send (CTS)
- The procedure for exchange of data and control frames in time (Figure 12.17):

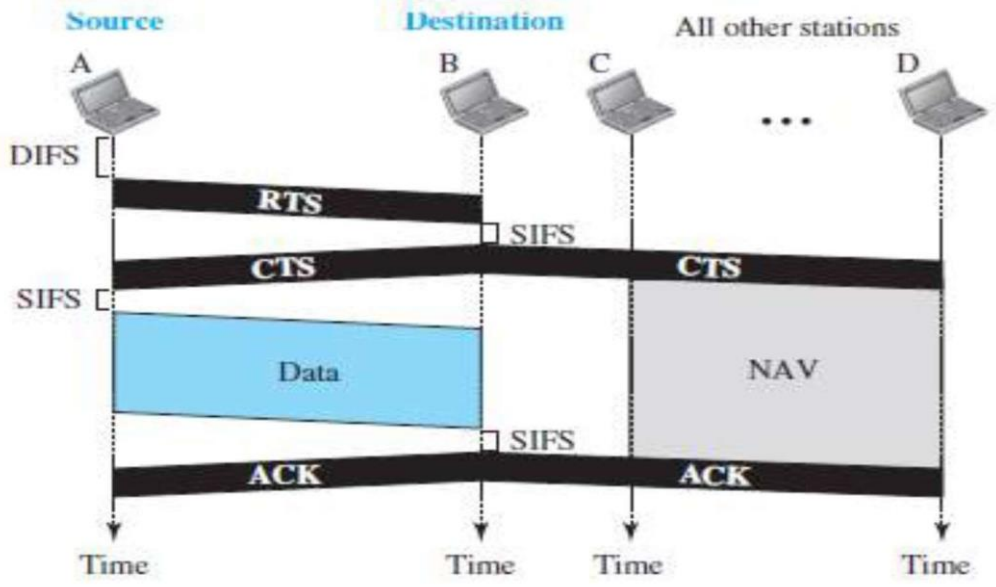


Figure 12.17 *CSMA/CA and NAV*

1. **The source** senses the medium by checking the energy level at the carrier frequency. If the medium is idle, then the source waits for a period of time called the DCF interframe space (DIFS). Finally, the source sends a RTS.
2. **The destination** receives the RTS, waits a period of time called the short interframe space (SIFS) and sends a control frame CTS to the source. CTS indicates that the destination station is ready to receive data.
3. **The source** receives the CTS, waits a period of time SIFS and sends a data to the destination
4. **The destination** receives the data, waits a period of time SIFS and sends a acknowledgment ACK to the source. ACK indicates that the destination has been received the frame.

2.9.4.2 Network Allocation Vector

- When a source-station sends an RTS, it includes the duration of time that it needs to occupy the channel.
- The remaining stations create a timer called a network allocation vector (NAV).
- NAV indicates waiting time to check the channel for idleness. Each time a station accesses the system and sends an RTS frame, other stations start their NAV.

2.9.4.3 Collision During Handshaking

- Two or more stations may try to send RTS at the same time. These RTS may collide.
- The source assumes there has been a collision if it has not received CTS from the destination. The backoff strategy is employed, and the source tries again.

2.9.4.4 Hidden-Station Problem

- Figure 12.17 also shows that the RTS from B reaches A, but not C.

- However, because both B and C are within the range of A, the CTS reaches C.
- Station C knows that some hidden station is using the channel and refrains from transmitting until that duration is over.

2.9.4.5 CSMA/CA and Wireless Networks

- CSMA/CA was mostly intended for use in wireless networks.
- However, it is not sophisticated enough to handle some particular issues related to wireless networks, such as hidden terminals or exposed terminals.

Review Questions

1. Why random access protocols are needed?
2. Compare pure ALOHA and slotted ALOHA?
3. Why CSMA is used?
4. What are different persistence methods used and which one is efficient?

Controlled Access Protocols

2.10 CONTROLLED ACCESS PROTOCOLS

- Here, the stations consult one another to find which station has the right to send.
- A station cannot send unless it has been authorized by other stations.
- Three popular controlled-access methods are:

  1) Reservation 2) Polling 3) Token Passing

2.10.1 Reservation

- Before sending data, each station needs to make a reservation of the medium.
- Time is divided into intervals. In each interval, a reservation-frame precedes the data-frames.
- If no. of stations = N, then there are N reservation mini-slots in the reservation-frame. Each mini-slot belongs to a station.
- When a station wants to send a data-frame, it makes a reservation in its own minislot. The stations that have made reservations can send their data-frames.

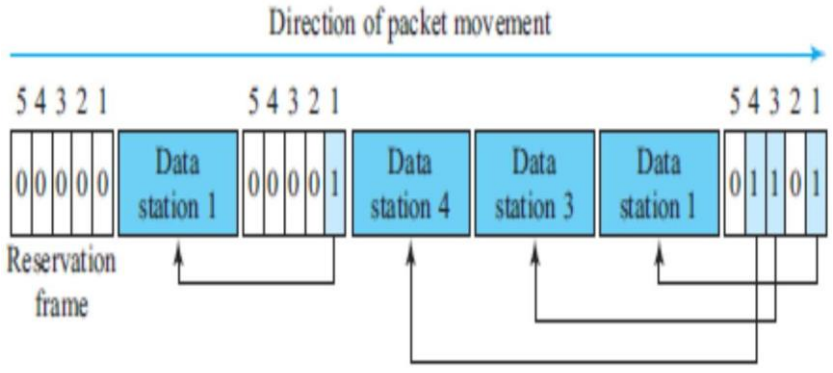


Figure 12.18 *Reservation access method*

- For example (Figure 12.18):
  - 5 stations have a 5-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.

2.10.2 Polling

- In a network, One device is designated as a primary station and Other devices are designated as secondary stations.
- Functions of primary-device:
  - The primary-device controls the link.
  - The primary-device is always the initiator of a session.
  - The primary-device is determines which device is allowed to use the
  - channel at a given time.
  - All data exchanges must be made through the primary-device.
- The secondary devices follow instructions of primary-device.
- Disadvantage: If the primary station fails, the system goes down.

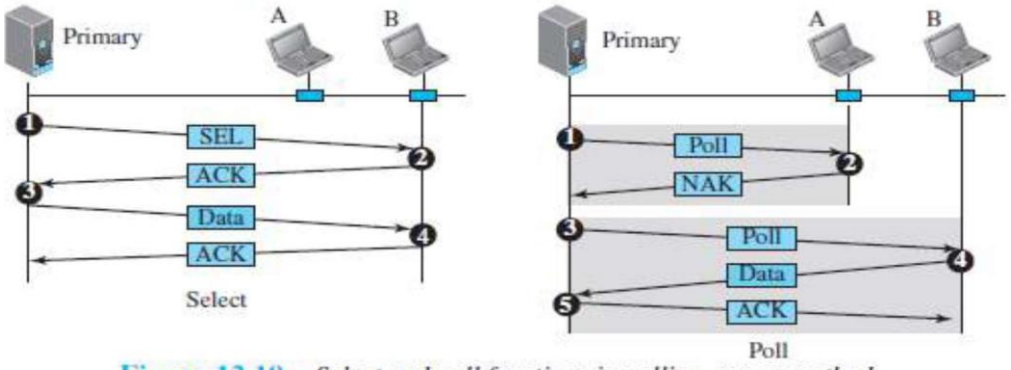


Figure 12.19 *Select and poll functions in polling-access method*

- **Poll and select** functions are used to prevent collisions (Figure 12.19).

Select

If the primary wants to send data, it tells the secondary to get ready to receive; this is called select function.

The primary, alerts the secondary about upcoming transmission by sending select frame (SEL), then waits for an acknowledgment (ACK) from secondary, then sends the data frame and finally waits for an acknowledgment (ACK) from the secondary.

Poll

- If the primary wants to receive data, it asks the secondaries if they have anything to send, this is **called poll function**.
- When the first secondary is approached, it responds either with a NAK frame if it has no data to send or with data-frame if it has data to send.
- If the response is negative (NAK frame), then the primary polls the next secondary in the same manner.
- When the response is positive (a data-frame), the primary reads the frame and returns an acknowledgment (ACK frame).

2.10.3 Token Passing

- In a network, the stations are organized in a ring fashion i.e. for each station; there is a predecessor and a successor.
  1) The predecessor is the station which is logically before the station in the ring.
  2) The successor is the station which is after the station in the ring.
- The current station is the one that is accessing the channel now.
- A token is a special packet that circulates through the ring.
- A station can send the data only if it has the token.
- When a station wants to send the data, it waits until it receives the token from its predecessor. Then, the station holds the token and sends its data.
- When the station finishes sending the data, the station releases the token and passes the token to the successor.
- Main functions of token management:
  1. Stations must be limited in the time they can hold the token.
  2. The token must be monitored to ensure it has not been lost or destroyed.
  3. Assign priorities to the stations and the types of data being transmitted.
  4. Make low-priority stations release the token to high priority stations.

2.10.3.1 Logical Ring

- In a token-passing network, stations do not have to be physically connected in a ring. the ring can be a logical one.
- Four physical topologies to create a logical ring (Figure 12.20):
  1) Physical ring
  2) Dual ring
  3) Bus ring
  4) Star ring

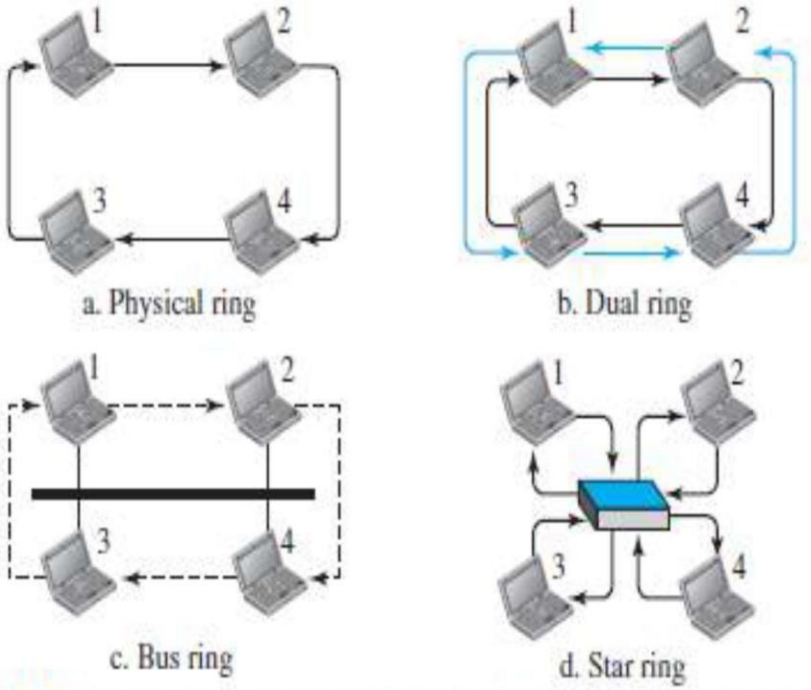


Figure 12.20 *Logical ring and physical topology in token-passing access method*

Physical Ring Topology

- When a station sends token to its successor, token cannot be seen by other stations. (Figure 12.20a). This means that the token does not have the address of the next successor.
- Disadvantage: If one of the links fails, the whole system fails.

Dual Ring Topology

- A second (auxiliary) ring is used along with the main ring (Figure 12.20b) which operates in the reverse direction compared with the main ring and is used for emergencies only (such as a spare tire for a car).
- If the main ring fails, the system automatically combines the 2 rings to form a temporary ring. After the failed link is restored, the second ring becomes idle again.
- Each station needs to have 2 transmitter-ports and 2 receiver-ports.
- This topology is used in FDDI (Fiber Distributed Data Interface) and CDDI (Copper Distributed Data Interface).

Bus Ring Topology

- The stations are connected to a single cable called a bus (Figure 12.20c).
- This makes a logical ring, because each station knows the address of its successor and predecessor.
- When a station has finished sending its data, the station releases the token and inserts the address of its successor in the token. Only that station gets the token to access the shared media.
- This topology is used in the Token Bus LAN.

Star Ring Topology

- The physical topology is a star (Figure 12.20d).
- There is a hub that acts as the connector.
- The wiring inside the hub makes the ring i.e. the stations are connected to the ring through the 2 wire connections.
- This topology is used in the Token Ring LAN.
    - **Advantages:** This topology is **less prone to failure** because If a link goes down, then the link will be bypassed by the hub and the rest of the stations can operate. **Also adding and removing** stations from the ring is easier.

.Question Bank

1) Explain two types of errors (4)

2) Compare error detection vs. error correction (2)

3) Explain error detection using block coding technique. (10)

4) Explain hamming distance for error detection (6)

5) Explain parity-check code with block diagram. (6)

6) Explain CRC with block diagram & an example. (10)

7) Write short notes on polynomial codes. (5)

8) Explain internet checksum algorithm along with an example. (6)

9) Explain the following:

i) Fletcher checksum and ii) Adler checksum (8)

10) Explain various types of checksum. (6)

11) Explain two types of frames. (6)

12) Explain character oriented protocol. (6)

13) Explain the concept of byte stuffing and unsnuffing with example. (6)

14) Explain bit oriented protocol. (6)

15) Differentiate between character oriented and bit oriented format for framing.(6)

16) Compare flow control and error control. (4)

17) With a neat diagram, explain the design of the simplest protocol with no flow control. (6)

18) Write algorithm for sender site and receiver site for the simplest \ protocol. (6)

19) Explain Stop-and-Wait protocol (8)

20) Explain the concept of Piggybacking (4)

21) Explain in detail HDLC frame format. (8)

22) Explain 3 type of frame used in HDLC (8)

23) With a neat schematic, explain the frame structure of PPP protocol. (8)

24) Explain framing and transition phases in Point-to-Point Protocol. (8)

25) Explain random access protocol. (4)

26) Explain pure ALOHA. (6)

27) Explain slotted ALOHA. (4)

28) Explain CSMA. (6)

29) Explain different persistence methods of CSMA. (6)

30) Explain CSMA/CA. (6)

31) Explain CSMA/CD. (10)

32) List & explain different controlled access protocols. (10)

33) Explain reservation access method. (4)

34) Explain polling access method. (6)

35) Explain token passing access method. (6)