DATA – LINK CONTROL

UNIT - 4

SYLLABUS

Unit IV

Data-Link Control: Framing: Character-Oriented Framing, Bit-Oriented Framing. Error Control: Types of Errors, Block Coding: Error Detection, Hamming Distance, Linear Block Codes: Parity-Check Code. Cyclic Codes: Cyclic Redundancy Check, Point-to-Point Protocol, Media Access Control – Random Access- CSMA, CSMA/CD, CSMA/CA, Controlled access.

Data Link Control

- Data Link Control(DLC) deals with procedures for communication between two adjacent nodes – node-to-node communication – no matter whether the link is dedicated or broadcast.
- Its function include Framing and Error Control.

Framing

- Data transmission in the physical layer means moving bits in the form of a signal from the source to the destination.
- The data-link layer, needs to pack bits into frames, so that each frame is distinguishable from another.
- Postal system practices a type of framing.
  - Inserting a letter into an envelope separates one piece of information from another; the envelope serves as the delimiter.

- Framing in the data-link layer separates a message from one source to a destination by adding a sender address and a destination address.
- Although the whole message could be packed in one frame, that is not normally done.

Frame Size:

- Frames can be of fixed or variable size.
- In fixed-size framing, there is no need to define the boundaries .

  Example : ATM WAN → which uses fixed size called cells.
- In variable-size framing, we need a way to define the end of one frame and the beginning of the next.

  Example : Local Area Networks.
- Two approaches have been used for this purpose:
  - Character-oriented approach and
  - Bit-oriented approach.

Character-Oriented Framing

- In character-oriented (or byte-oriented) framing, data to be carried are 8-bit characters from a coding system such as ASCII.
- The header, which normally carries the source and destination addresses and other control information, and the trailer, which carries error detection redundant bits, are also multiples of 8 bits.
- To separate one frame from the next, an 8-bit(1-byte) flag is added at the beginning and the end of a frame.

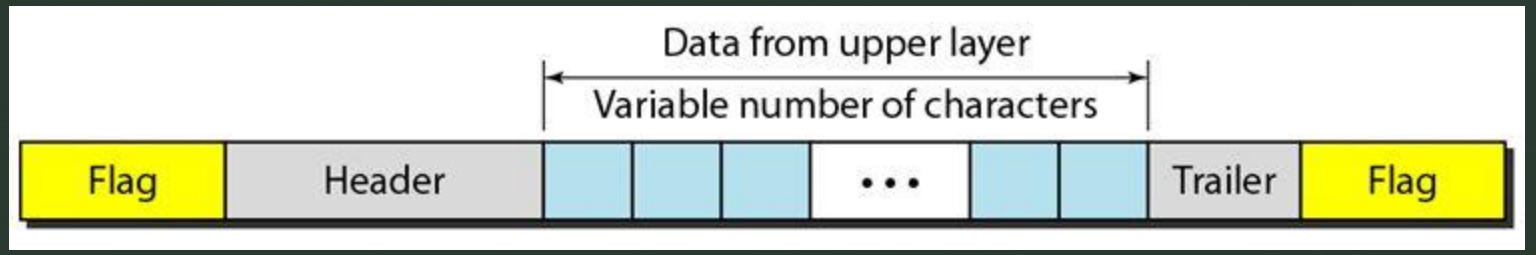


Figure : A frame in a character-oriented protocol

- Character-oriented framing was popular when only text was exchanged by the data-link layers.
- Suppose if we send other types of information such as graphs, audio, and video; any pattern used for the flag could also be part of the information.
- If this happens, the receiver, when it encounters this pattern in the middle of the data, thinks it has reached the end of the frame.
- To fix this problem, a byte-stuffing strategy was added to character-oriented framing.
- In byte stuffing (or character stuffing), a special byte is added to the data section of the frame when there is a character with the same pattern as the flag.
- The data section is stuffed with an extra byte. This byte is usually called the escape character (ESC) and has a predefined bit pattern.

- The receiver removes the escape character, but keeps the next byte, which is incorrectly interpreted as the end of the frame. To solve this problem, the escape characters that are part of the text must also be marked by another escape character.

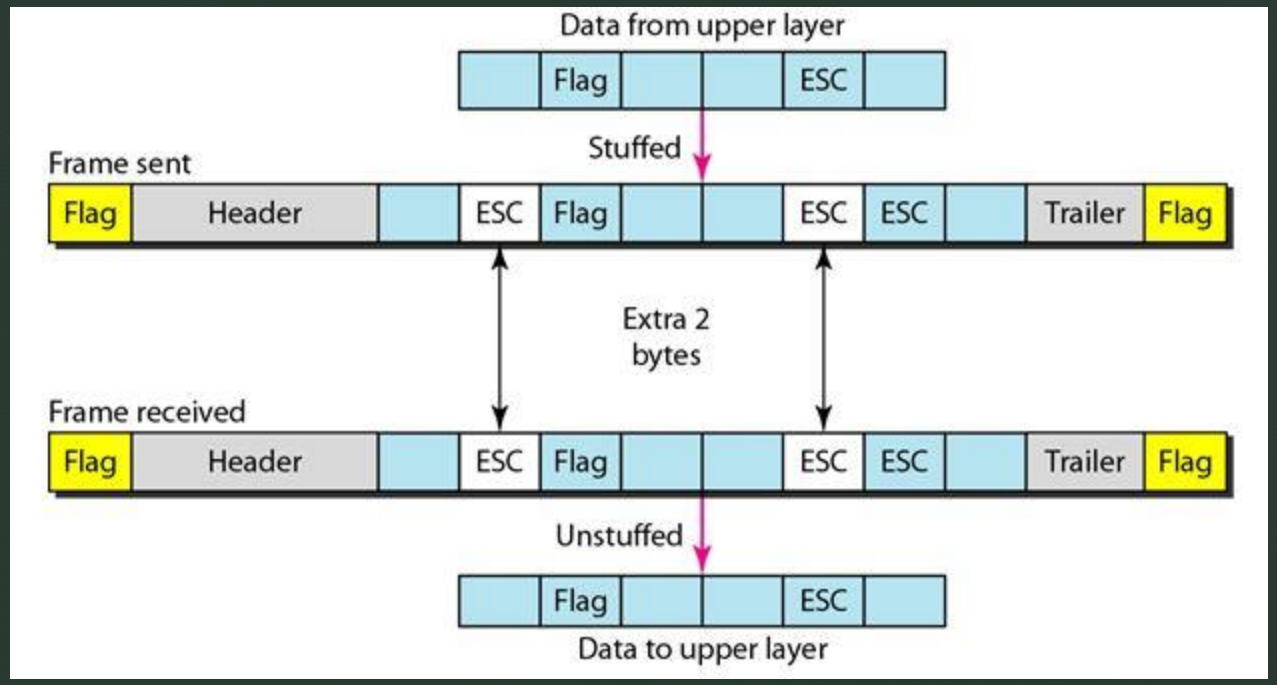


Figure : Byte stuffing and unstuffing

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

- Character-oriented protocols present another problem in data communications. The universal coding systems in use today, such as Unicode, have 16-bit and 32-bit characters that conflict with 8-bit characters.

Bit-Oriented Framing

- In bit-oriented framing, the data section of a frame is a sequence of bits to be interpreted by the upper layer as text, graphic, audio, video, and so on.
- Most protocols use a special 8-bit pattern flag, 01111110, as the delimiter to define the beginning and end of the frame.

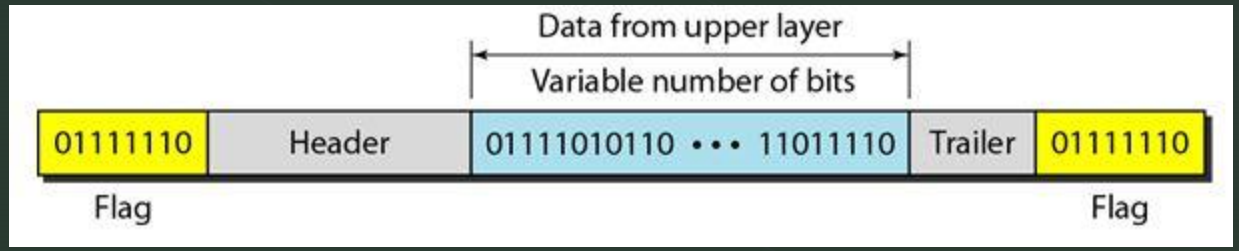


Figure : A frame in a bit-oriented protocol

- Bit stuffing : Stuffing one single bit (instead of 1 byte) to prevent the pattern from looking like a flag.
- In bit stuffing, if a 0 and five consecutive 1 bits are encountered, an extra 0 is added.

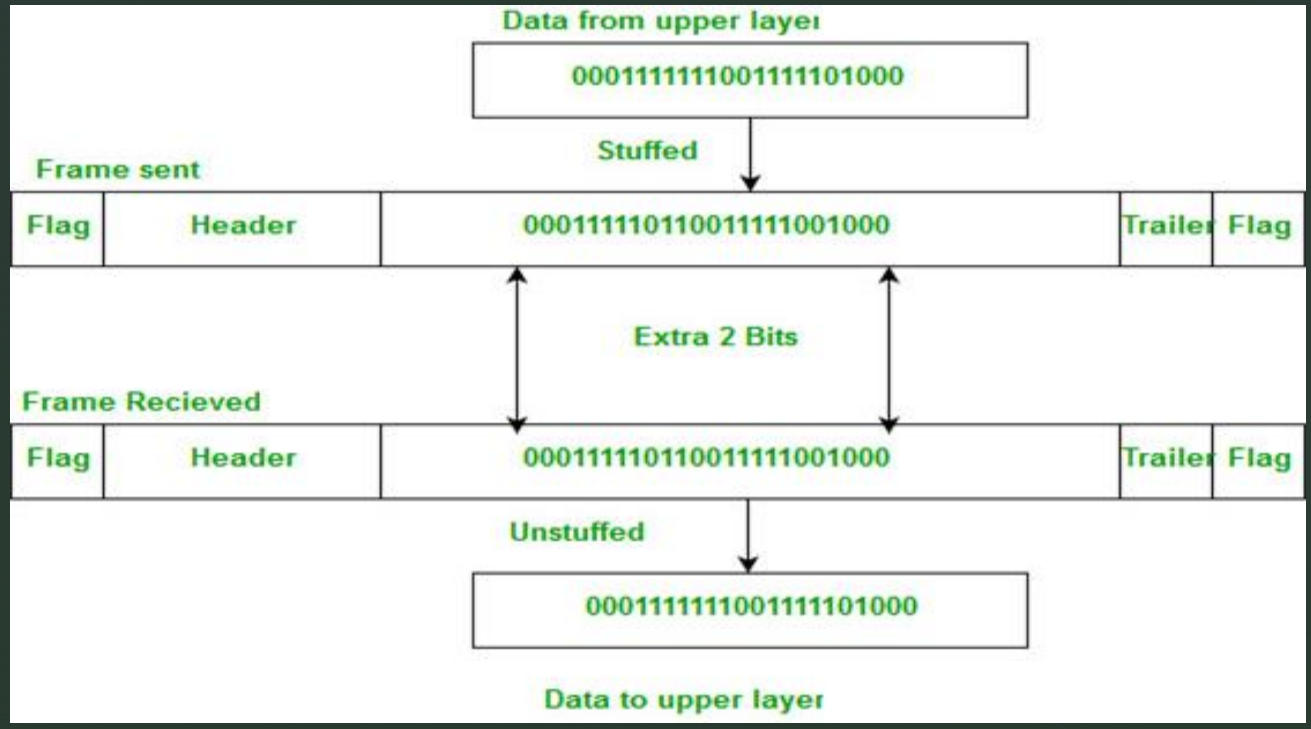


Figure : Bit stuffing and unstuffing

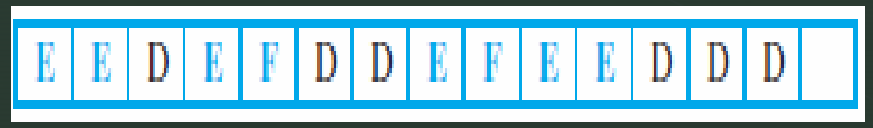
Example 1

- Byte-stuff the following frame payload in which
  - E is the escape byte
  - F is the flag byte
  - D is a data byte
  - other than an escape or a flag character.

F D E E D D E F D D E E E E D E F D F

Example 2

- Unstuff the following frame payload in which
  - E is the escape byte
  - F is the flag byte
  - D is a data byte
  - other than an escape or a flag character.



E D F D D F E D D D

Example 3

- Bit-stuff the following frame payload:

  0001111111001111101000111111111110000111

  00011111101100111111001000111111011111010000111

Example 4

- Unstuff the following frame payload:

  000111110000011111011101001110111110000001111

  00011111**0**0000111111**0**1110100111011111**0**00001111

  000111111000011111111101001110111111000011111

Programming Examples

- Write and test a program that simulates the byte stuffing and byte unstuffing
- Write and test a program that simulates the bit stuffing and bit unstuffing

Error Control

- Error control is both error detection and error correction.
- It allows the receiver to inform the sender of any frames lost or damaged in transmission and coordinates the retransmission of those frames by the sender.
- In the data-link layer, the term error control refers primarily to methods of error detection and retransmission (error correction is done using retransmission of the corrupted frame).

Types of Errors

- Whenever bits flow from one point to another, they are subject to unpredictable changes because of interference.
- This interference can change the shape of the signal.
- The term single-bit error means that only 1 bit of a given data unit (such as a byte, character, or packet) is changed from 1 to 0 or from 0 to 1.
- The term burst error means that two or more bits in the data unit have changed from 1 to 0 or from 0 to 1.

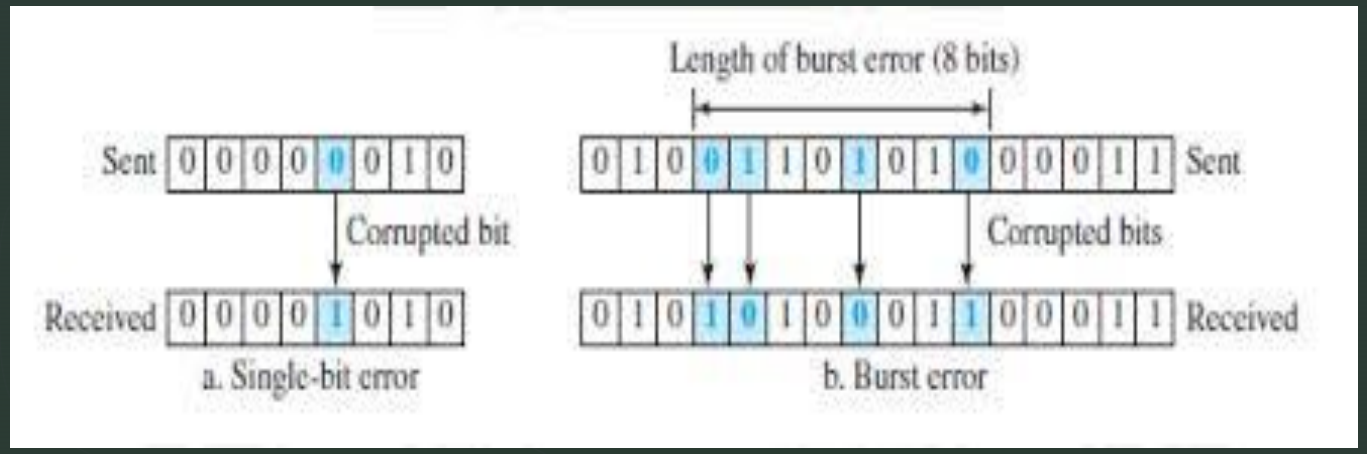


Figure : Single-bit error and burst error

- A burst error is more likely to occur than a single-bit error because the duration of the noise signal is normally longer than the duration of one bit, which means that when noise affects data, it affects a set of bits.
- The number of bits affected depends on the data rate and duration of noise.

Redundancy :

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

Detection versus Correction :

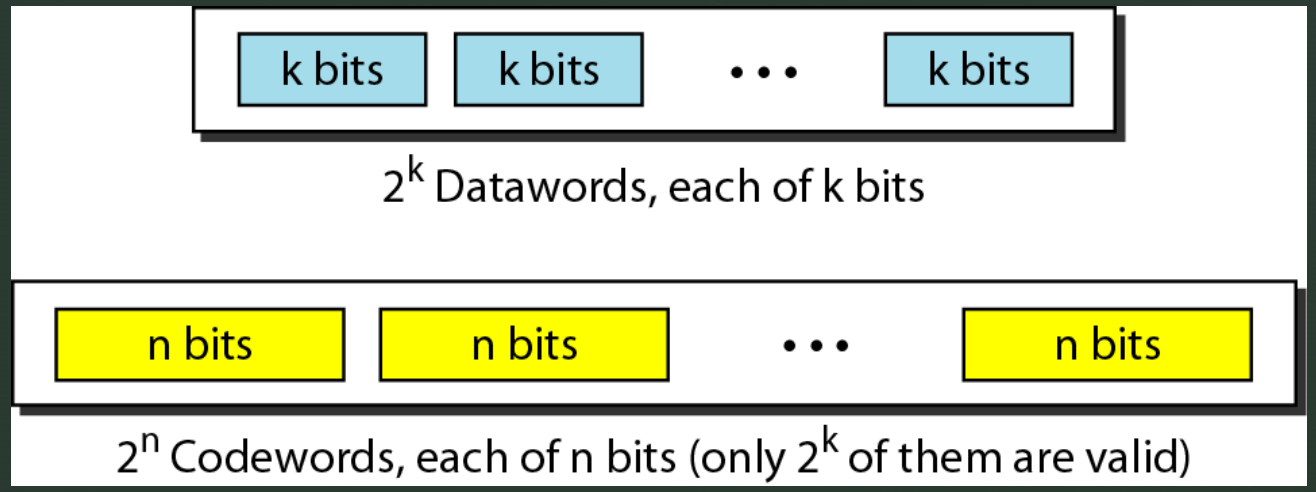
- The correction of errors is more difficult than the detection.
- In error detection, we are looking only to see if any error has occurred.
- A single-bit error is the same for us as a burst error.
- In error correction, we need to know the exact number of bits that are corrupted and, more importantly, their location in the message.
- If we need to correct a single error in an 8-bit data unit, we need to consider eight possible error locations; if we need to correct two errors in a data unit of the same size, we need to consider 28 (permutation of 8 by 2) possibilities.

Coding

- Redundancy is achieved through various coding schemes.
- The sender adds redundant bits through a process that creates a relationship between the redundant bits and the actual data bits.
- The receiver checks the relationships between the two sets of bits to detect errors.
- The ratio of redundant bits to the data bits and the robustness of the process are important factors in any coding scheme.
- We can divide coding schemes into two broad categories:
  - Block coding and
  - Convolution coding.

Block Coding

- In block coding, we divide our message into blocks, each consisting of k bits, called Datawords.
- We add r redundant bits to each block to make the length n = k + r. The resulting n-bit blocks are called Codewords.



- With k bits, we can create a combination of 2^k datawords; with n bits, we can create a combination of 2^n codewords.
- The block-coding process is one-to-one;
  - The same dataword is always encoded as the same codeword.
- This means that we have $2^n - 2^k$ codewords that are not used. We call these codewords invalid or illegal.
- The trick in error detection is the existence of these invalid codes. If the receiver receives an invalid codeword, this indicates that the data were corrupted during transmission.

Error Detection

If the following two conditions are met, the receiver can detect a change in the original codeword.

1. The receiver has (or can find) a list of valid codewords.

2. The original codeword has changed to an invalid one.

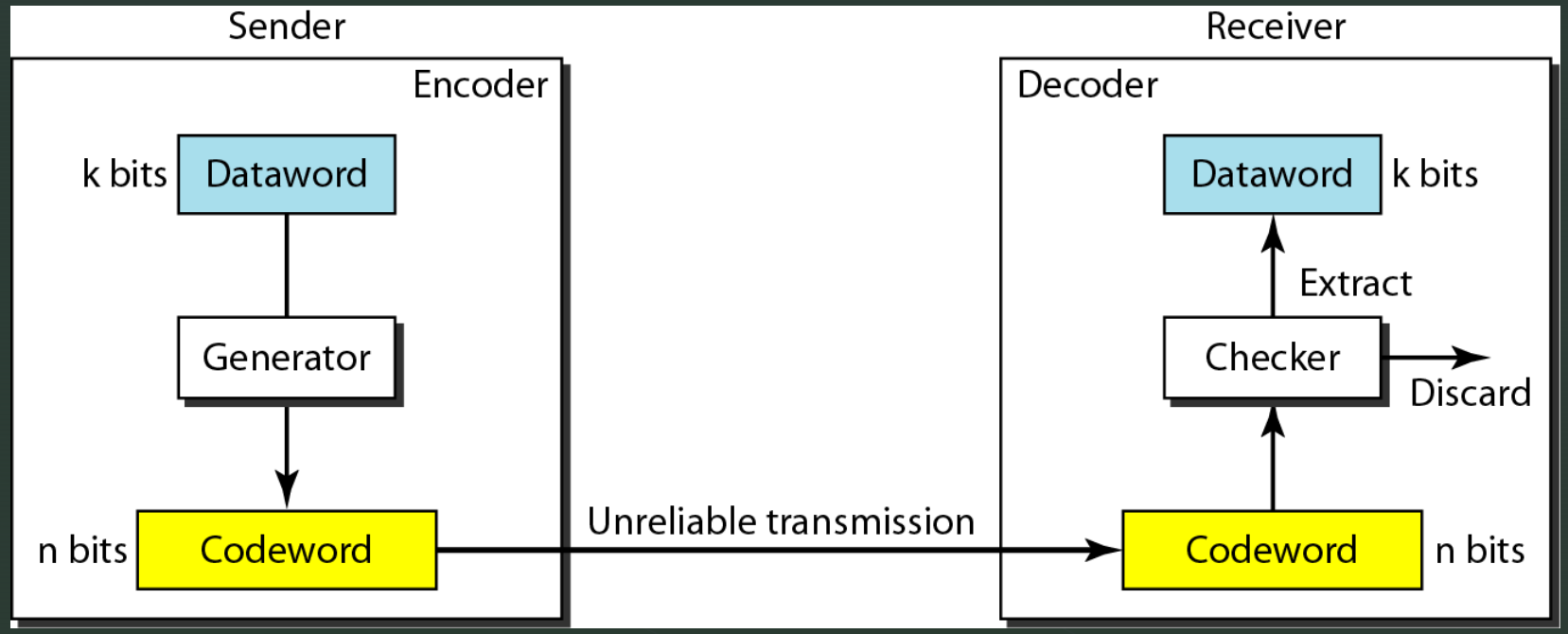


Figure : Process of error detection in block coding

- If the codeword is corrupted during transmission but the received word still matches a valid codeword, the error remains undetected.

Error-detecting Code Example :

- Let us assume that *k = 2 and n = 3*
- *The list of datawords and codewords*

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

- Sender encodes the dataword 01 as 011 and sends it to the receiver
  - Receiver receives 011→ a valid codeword → extracts 01 dataword
  - Receiver receives 111 →not a valid codeword → discarded
  - Receiver receives 000→ a valid codeword → incorrectly extracts 00 dataword → Two corrupted bits have made the error undetectable

Hamming Distance

- The Hamming distance between two words (of the same size) is the number of differences between the corresponding bits.
- The Hamming distance between two words x and y as d(x, y).
- Why the Hamming distance is important for error detection.
  - The reason is that the Hamming distance between the received codeword and the sent codeword is the number of bits that are corrupted during transmission.

Hamming Distance between two words is the number of differences between corresponding bits.

- Example :
  - if the codeword 00000 is sent and 01101 is received, 3 bits are in error and the Hamming distance between the two is d(00000, 01101) = 3.
- In other words, if the Hamming distance between the sent and the received codeword is not zero, the codeword has been corrupted during transmission.
- The Hamming distance can easily be found if we apply the XOR operation ($\oplus$) on the two words and count the number of 1s in the result.
- Note that the Hamming distance is a value greater than or equal to zero.

Example 2 :

- Let us find the Hamming distance between two pairs of words.
  - The Hamming distance d (000, 011).
  - The Hamming distance d (10101, 11110).

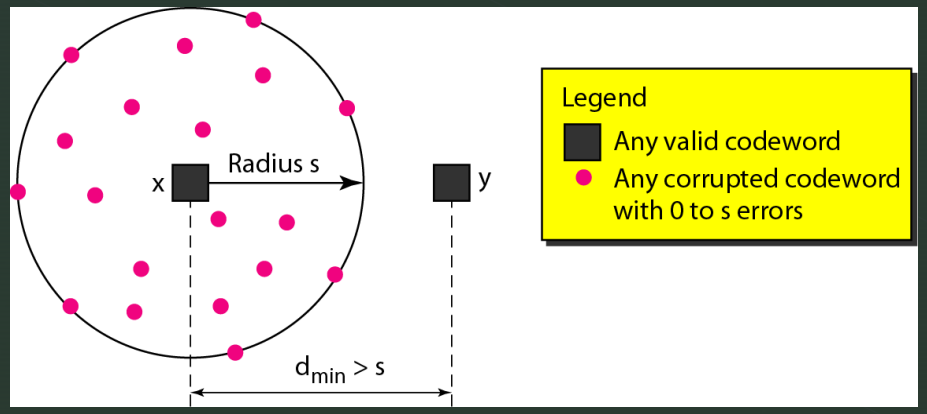
Minimum Hamming Distance for Error Detection:

- In a set of codewords, the minimum Hamming distance is the smallest Hamming distance between all possible pairs of codewords.
- If s errors occur during transmission, the Hamming distance between the sent codeword and received codeword is s.
- If our system is to detect up to s errors, the minimum distance between the valid codes must be (s + 1).
- Although a code with $d_{min} = s + 1$ may be able to detect more than s errors in some special cases, only s or fewer errors are guaranteed to be detected.

Geometric concept explaining d_{min} in error detection

- To guarantee the ***detection*** of up to **s**-bit errors, the minimum Hamming distance in a block code must be

$$d_{min} = s + 1$$



A code scheme with Hamming distance $d_{min} = 4$ *guarantees the detection of up to* three errors (*d = s + 1 or s = 3*)

Example:

- The minimum hamming distance for the code in table is 2. this code guarantees detection of only a single error.
- If the third codeword(101) is sent and one error occurs, the received codeword does not match any valid codeword. If two errors occur, however, the received codeword may match a valid codeword and the errors are not detected.

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

Linear Block Codes

- Almost all block codes used today belong to a subset of block codes called linear block codes.
- The use of nonlinear block codes for error detection and correction is not as widespread because their structure makes theoretical analysis and implementation difficult.
- The formal definition of linear block codes requires the knowledge of abstract algebra.
- We therefore give an informal definition. → a linear block code is a code in which the exclusive OR (addition modulo-2) of two valid codewords creates another valid codeword.

Example :

- The code in Table is a linear block code because the result of XORing any codeword with any other codeword is a valid codeword.
- For example, the XORing of the second and third codewords creates the fourth one.

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

First codeword	**000**	**000**	**000**	**011**	**011**	**101**
Second codeword	011	101	110	101	110	110
result	011	101	110	110	101	011
Valid/invalid	V	V	V	V	V	V

Minimum Distance for Linear Block Codes

- The minimum Hamming distance for a linear block code is simple to find. It is the number of 1s in the nonzero valid codeword with the smallest number of 1s.

Example :

- In our first code (Table), the numbers of 1s in the nonzero codewords are 2, 2, and 2. So the minimum Hamming distance is $d_{min} = 2$.

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

First codeword	**000**	**000**	**000**	**011**	**011**	**101**
Second codeword	011	101	110	101	110	110
result	011	101	110	110	101	011
Hamming distance	2	2	2	2	2	2

Parity-Check Code

- The most familiar error-detecting code is the parity-check code. This code is a linear block code.
- A k-bit dataword is changed to an n-bit codeword where n = k + 1. The extra bit, called the parity bit, is selected to make the total number of 1s in the codeword even.
- The minimum Hamming distance for this category is $d_{min} = 2$, which means that the code is a single-bit error-detecting code.

Parity Check Examples

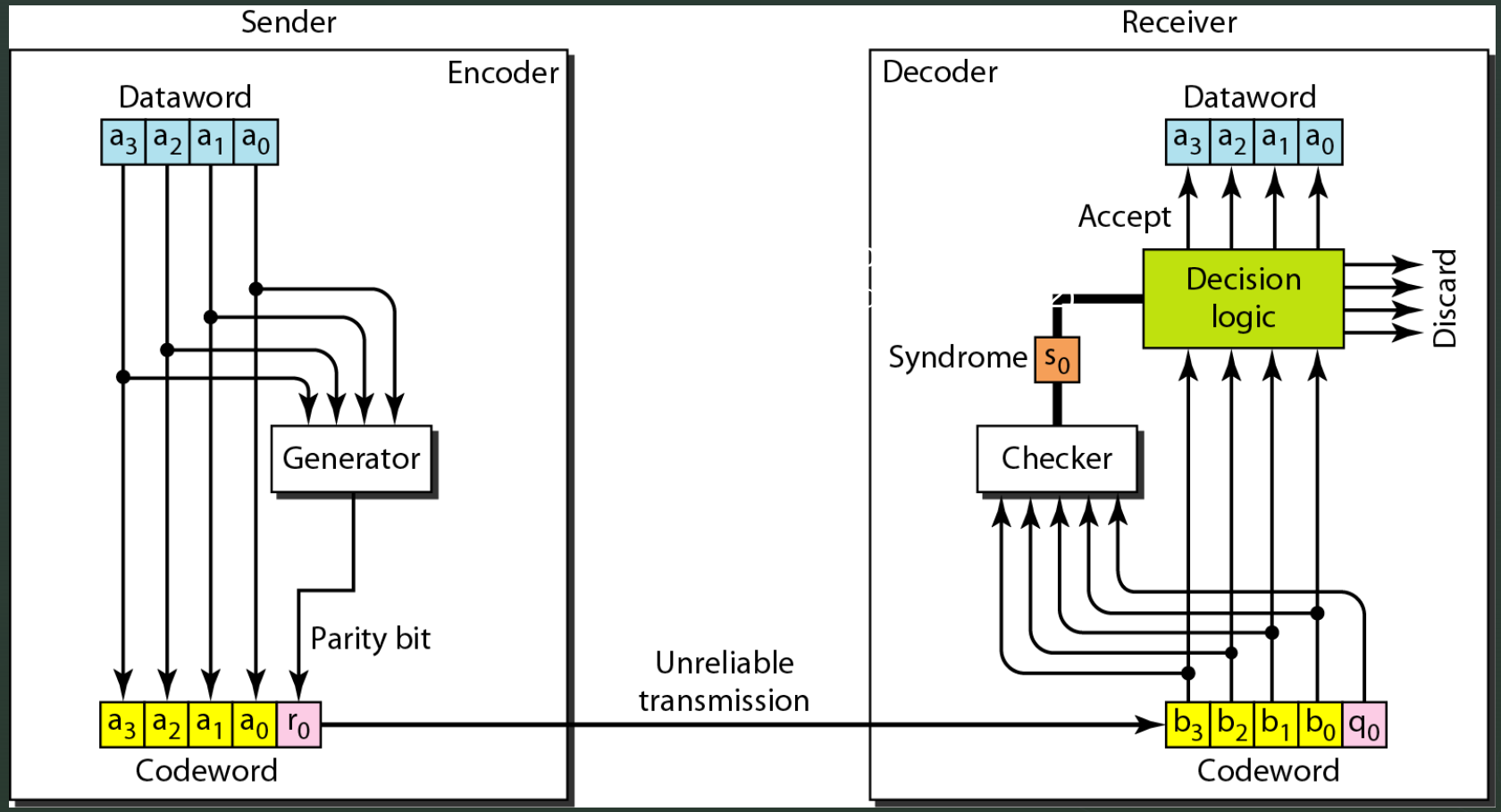
- A parity-check code *k = 2 and n = 3*

Dataword	codeword
00	00**0**
01	01**1**
10	10**1**
11	11**0**

- A parity-check code with *k = 4 and n = 5*

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

Parity-Check: Encoding/Decoding



Rule to make total number of 1s in the codeword is even: If the number of 1s is even, r0=0 else r0=1

Rule of the decision logic analyzer : If s0=0, no detectable error in the received codeword (accept) else detectable error (discard)

- If the number of 1s is even, the result is 0; if the number of 1s is odd, the result is 1. In both cases, the total number of 1s in the codeword is even.
- The result, which is called the syndrome, is just 1 bit. The syndrome is 0 when the number of 1s in the received codeword is even; otherwise, it is 1.

$$s0 = b3 + b2 + b1 + b0 + q0$$

- If the syndrome is 0, there is no detectable error in the received codeword; the data portion of the received codeword is accepted as the dataword.
- If the syndrome is 1, the data portion of the received codeword is discarded. The dataword is not created.

Parity-Check: transmission scenarios

- Scenario: Assume the sender dataword 1011 with codeword is 10111, which is sent to the receiver.

Receiver Scenario	received codeword	syndrome	data word	Note
No error occurs	10111	0	1011	-
One single-bit error - changes a1	10011	1	Not created	-
One single-bit error changes r0	10110	1	Not created	none of the dataword bits are corrupted still no dataword is created because the code is not sophisticated enough to show the position of the corrupted bit.
An error changes r0 and a second error changes a3	00110	0	0011	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.
3-bits—a3, a2, and changed by errors	01011	1	Not created	shows that the simple parity check, guaranteed to detect one single error, can also find any odd number of errors.

A parity-check code can detect an odd number of errors.

- Two schemes:
  - Even parity – Maintain even number of 1s
    - *Extra bit used to make the total* number of 1s in the codeword even
    - E.g., 1011 → 1011**1**
  - Odd parity – Maintain odd number of 1s
    - *Extra bit used to make the total* number of 1s in the codeword odd
    - E.g., 1011 → 1011**0**

Example: *Parity Check*

Suppose the sender wants to send the word *world*. In ASCII the five characters are coded (with ***even parity***) as

1110111 1101111 1110010 1101100 1100100

The following shows the actual bits sent

11101110 11011110 11100100 11011000 11001001

Receiver receives this sequence of words:

11111110 11011110 11101100 11011000 11001001

Which blocks are accepted? Which are rejected?

11111110 11011110 11101100 11011000 11001001

R A R A A

Cyclic Codes

- Cyclic codes are special linear block codes with one extra property.
- In a cyclic code, if a codeword is cyclically shifted (rotated), the result is another codeword.
- For example, if 1011000 is a codeword and we cyclically left-shift, then 0110001 is also a codeword.
- b1 = a0 b2 = a1 b3 = a2 b4 = a3 b5 = a4 b6 = a5 b0 = a6
- In the rightmost equation, the last bit of the first word is wrapped around and becomes the first bit of the second word.

Cyclic Redundancy Check (CRC)

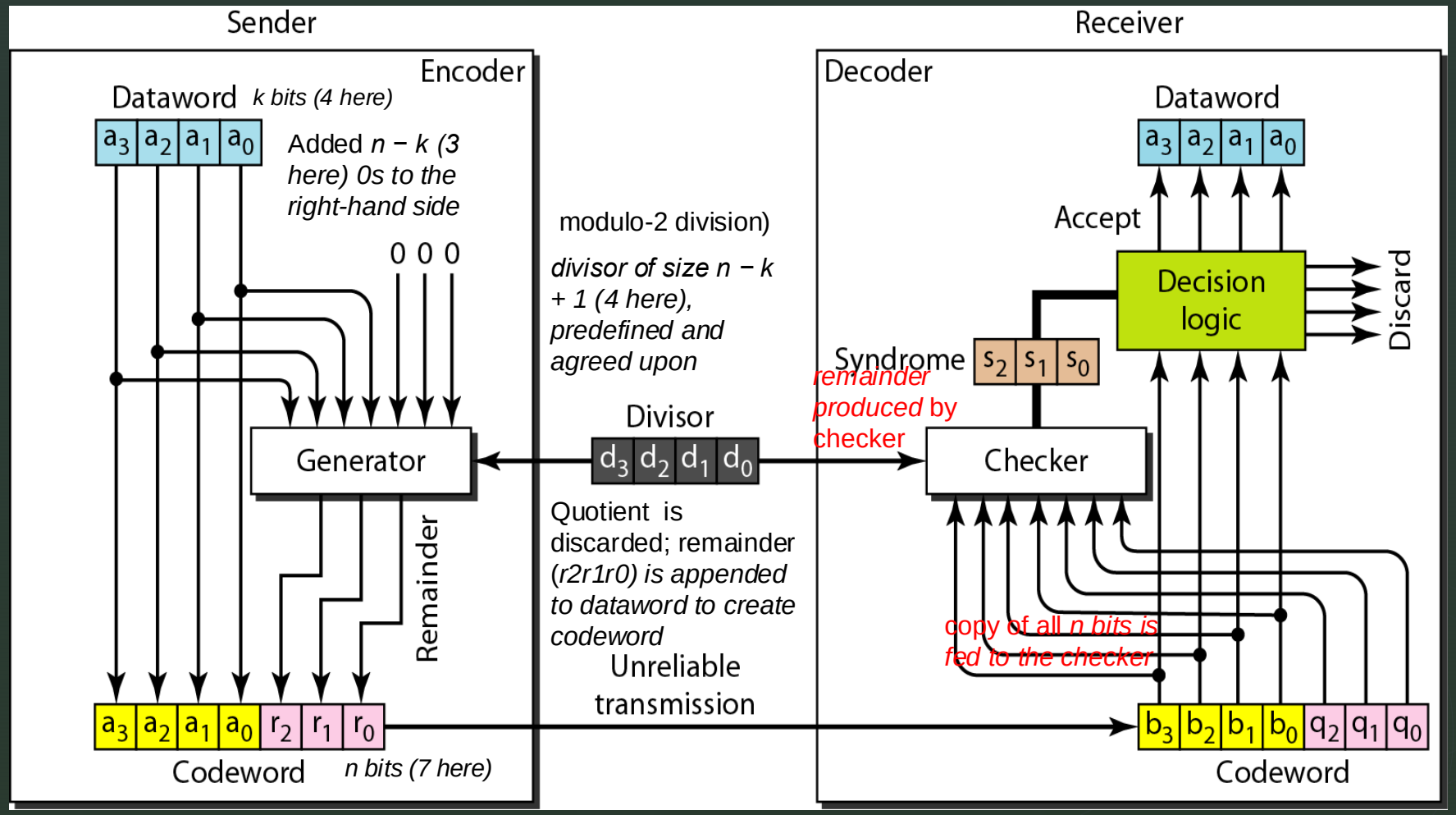
- A subset of cyclic codes.
- Has both the linear and cyclic properties
- Used in networks such as LANs and WANs
- *A CRC code with C(7, 4)*

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

CRC Encoder/Decoder

One possible design:

If the syndrome bits are all 0s, the 4 leftmost bits of the codeword are accepted as the dataword (interpreted as no error); otherwise, the 4 bits are discarded (error).



Encoder :

- The encoder takes a dataword and augments it with n – k number of 0s. It then divides the augmented dataword by the divisor, as shown in Figure
- Divisor : degree of polynomial.

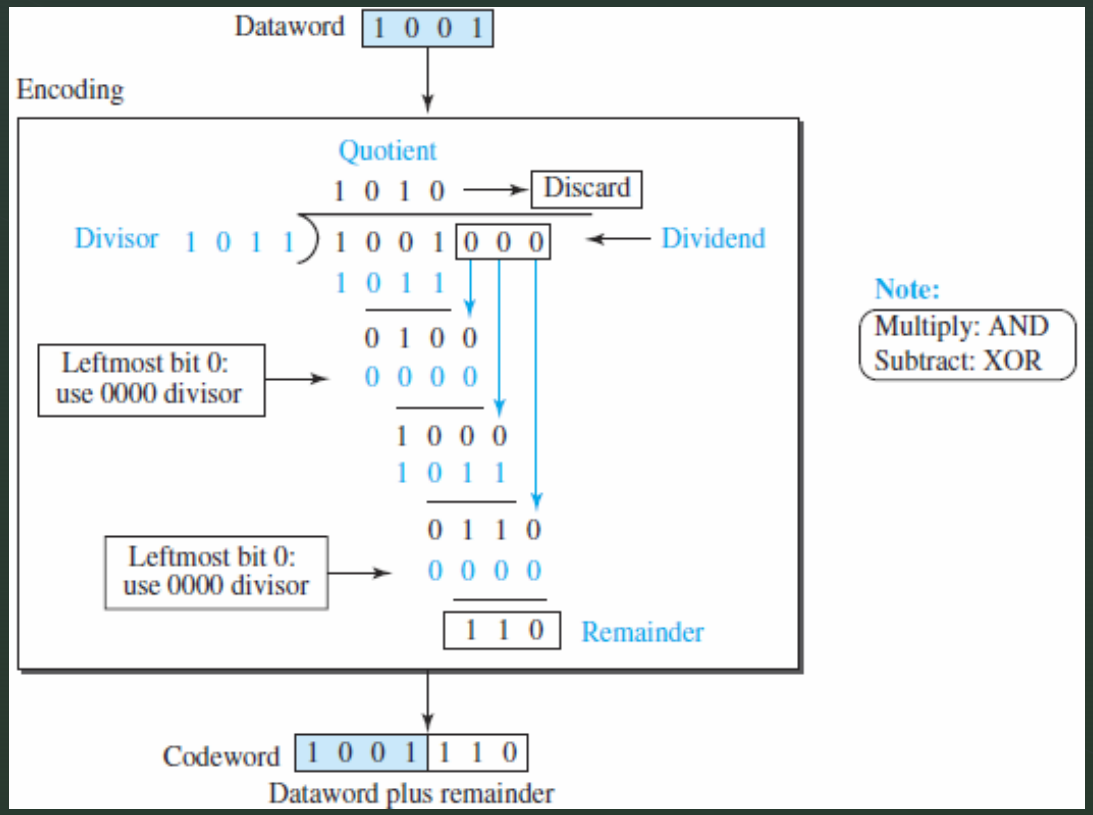


Figure : Division in CRC encoder

Decoder

- The decoder does the same division process as the encoder.
- The remainder of the division is the syndrome. If the syndrome is all 0s, there is no error.
- the dataword is separated from the received codeword and accepted. Otherwise, everything is discarded.

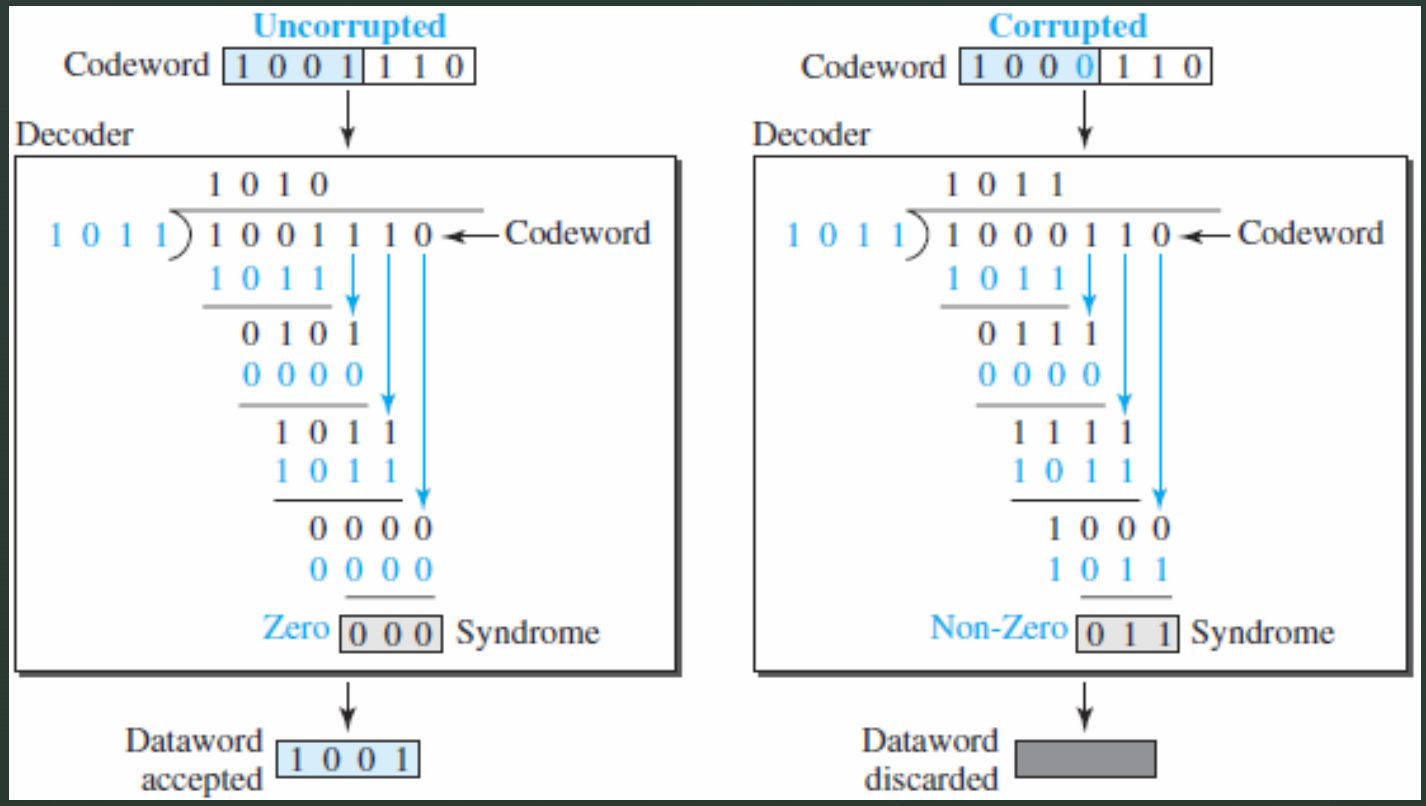


Figure : Division in the CRC decoder for two cases

Requirement:

A bit pattern needs to have at least two properties to be considered a generator (divisor):

1. The pattern should have at least 2 bits.

2. The rightmost and leftmost bits should both be 1s.

Performance

The following shows the performance of CRC.

- Single errors : detect any single-bit error.
- Odd number of errors: Only some odd number errors are detected.
- Burst errors.

Advantages of Cyclic Codes

- Cyclic codes can easily be implemented in hardware and software.

Checksum

- Checksum is an error-detecting technique that can be applied to a message of any length.
- In the Internet, the checksum technique is mostly used at the network and transport layer rather than the data-link layer.

Two DLC Protocol

- Two DLC protocol:
  - First, High Level Data Link Control, is the base of many protocols that have been designed for LANs.
  - Second, Point-to-point Protocol is derived from HDLC

Point-to-Point Protocol

- One of the most common protocols for point-to-point access is the Point-to-Point Protocol (PPP).
- The majority of these users have a traditional modem; they are connected to the Internet through a telephone line, which provides the services of the physical layer.
- But to control and manage the transfer of data, there is a need for point-to-point access at the data-link layer.
- PPP is a byte-oriented protocol.

PPP Services

Services Provided by PPP

- Format of the frame to be exchanged between devices
- How two devices can negotiate the establishment of the link and the exchange of data
- Accept payloads from several network layers (not only IP)
- Optional authentication
- The new version of PPP, called *Multilink PPP, provides connections over multiple links*
- Network address configuration
- useful when a home user needs a temporary network address to connect to the Internet

PPP Services

Services Not Provided by PPP

- No flow control - A sender can send several frames one after another with no concern about overwhelming the receiver
- A very simple mechanism for error control - A CRC field is used to detect errors
  - If the frame is corrupted, it is silently discarded; the upper-layer protocol needs to take care of the problem
    - Lack of error control and sequence numbering may cause a packet to be received out of order
- No a sophisticated addressing mechanism to handle frames in a multipoint configuration

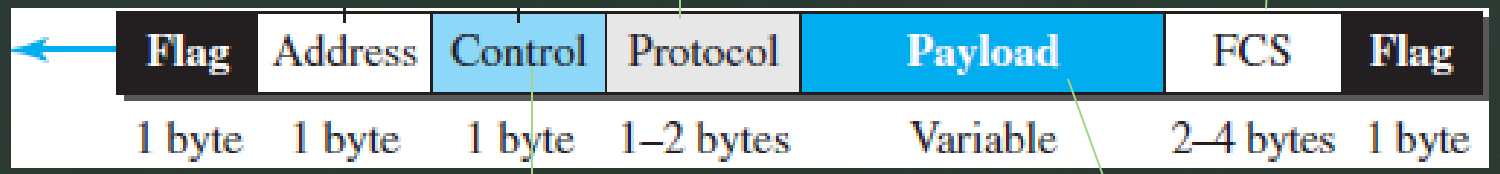
PPP frame format

- a character-oriented (or byte-oriented) frame
- starts and ends with a 1-byte flag with the bit pattern 01111110

a constant value and set to 11111111 (broadcast address)

defines what is being carried in the data field: either user data or other information

frame check sequence (FCS) is simply a 2-byte or 4-byte standard CRC



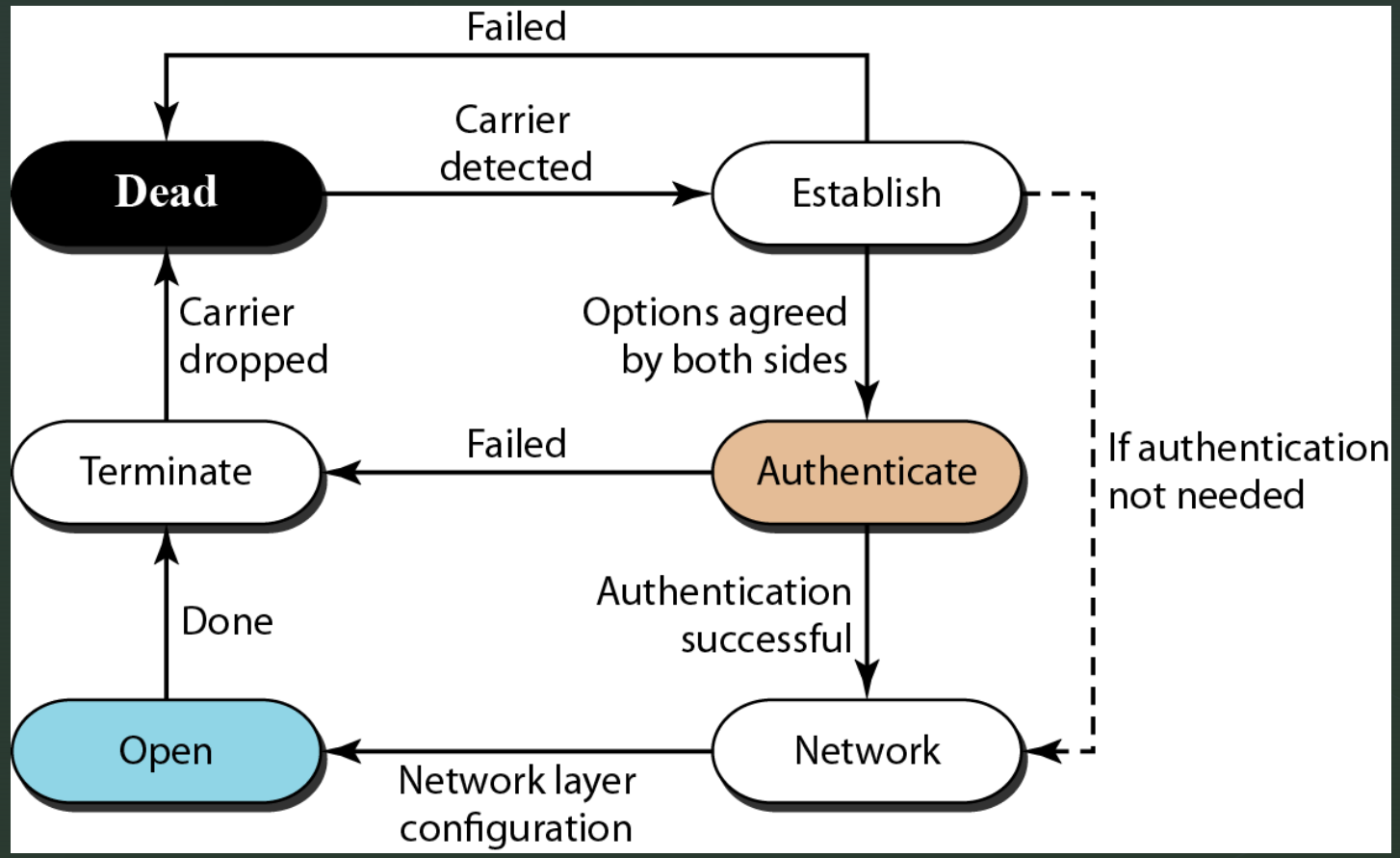
constant value 00000011 (imitating unnumbered frames in HDLC)

carries either the user data or other information (byte-stuffed with escape byte as 01111101) - maximum of 1500 bytes; but this can be changed during negotiation

Byte Stuffing :

- 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.
- The escape byte itself should be stuffed with another escape byte.

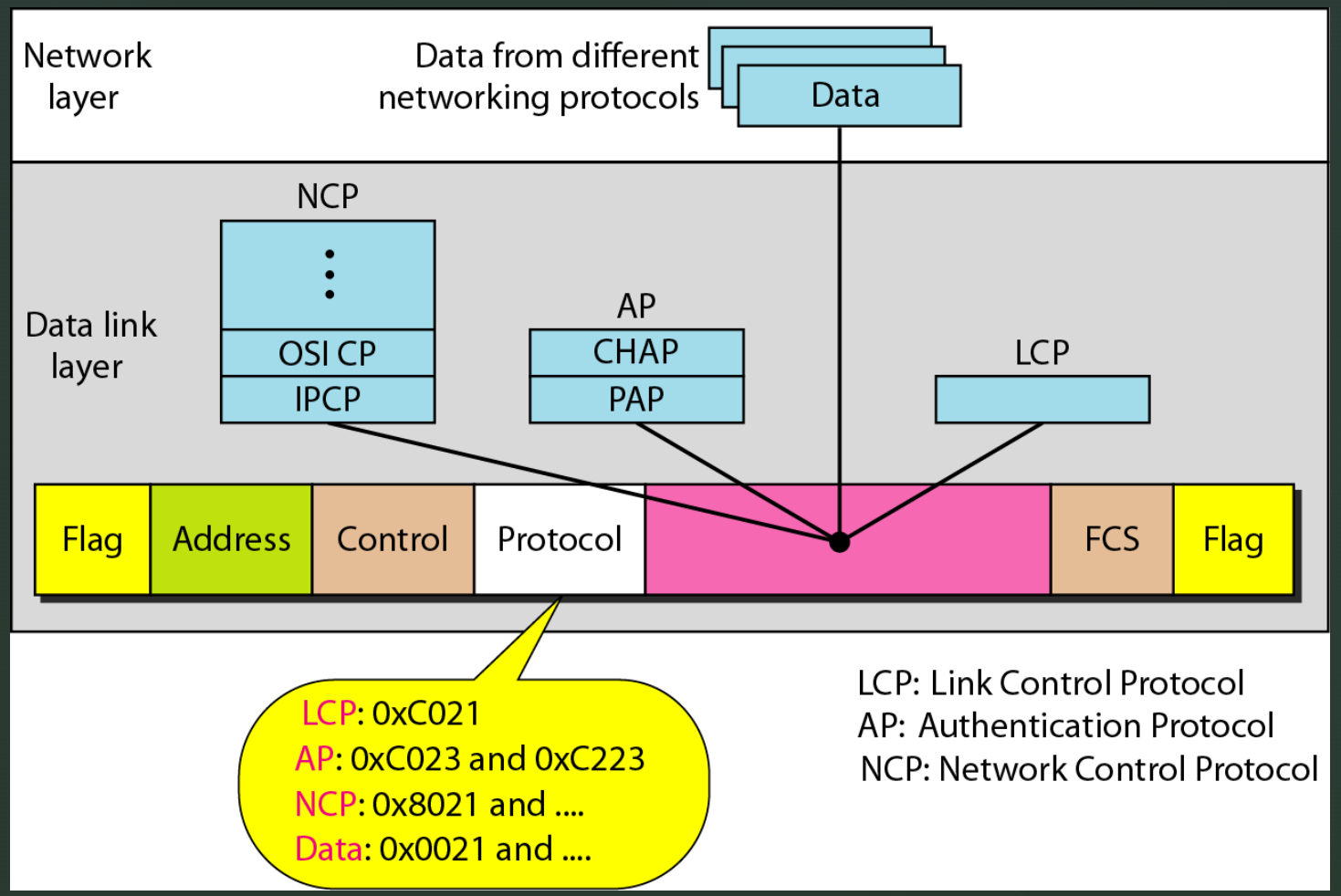
Transition Phases : A PPP connection goes through phases that can be shown in a transition phase diagram.



- The transition diagram starts with the dead state. In this state, there is no active carrier (at the physical layer) and the line is quiet.
- When one of the two nodes starts the communication, the connection goes into the establish state. In this state, options are negotiated between the two parties.
- If the two ends agree with authentication, the system goes to the authenticate state; otherwise, the system goes to the network state.
- The Link Control Protocol packets, are used for this purpose. Several packets may be exchanged here.
- Data transfer takes place in the open state. When a connection reaches this state, the exchange of data packets can be started.
- The connection remains in this state until one of the endpoints wants to terminate the connection. In this case, the system goes to the terminate state.
- The system remains in this state until the carrier (physical-layer signal) is dropped, which moves the system to the dead state again.

Multiplexing :

- Three sets of protocols are defined to make PPP powerful:
  - Link Control Protocol (LCP),
  - Authentication Protocols (APs), and
  - Network Control Protocols (NCPs).
- At any moment, a PPP packet can carry data from one of these protocols in its data field.
- Data may also come from several different network layers.


Network layer
Data from different networking protocols
Data
NCP
Data link layer
OSI CP
IPCP
AP
CHAP
PAP
LCP
Flag
Address
Control
Protocol
FCS
Flag
LCP: 0xC021
AP: 0xC023 and 0xC223
NCP: 0x8021 and ....
Data: 0x0021 and ....
LCP: Link Control Protocol
AP: Authentication Protocol
NCP: Network Control Protocol

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.

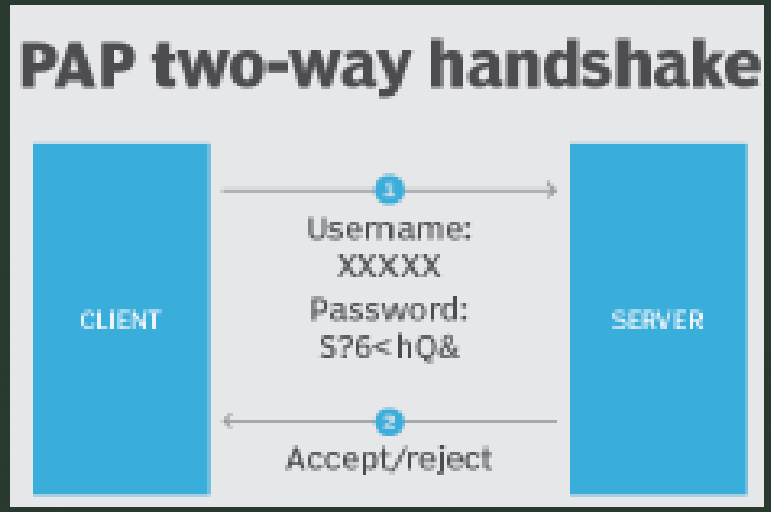
Authentication Protocols:

- PPP is designed for use over dial-up links where verification of user identity is necessary.
- Authentication means validating the identity of a user who needs to access a set of resources.
- PPP has created two protocols for authentication:
  - Password Authentication Protocol and
  - Challenge Handshake Authentication Protocol.

Note: These protocols are used during the authentication phase.

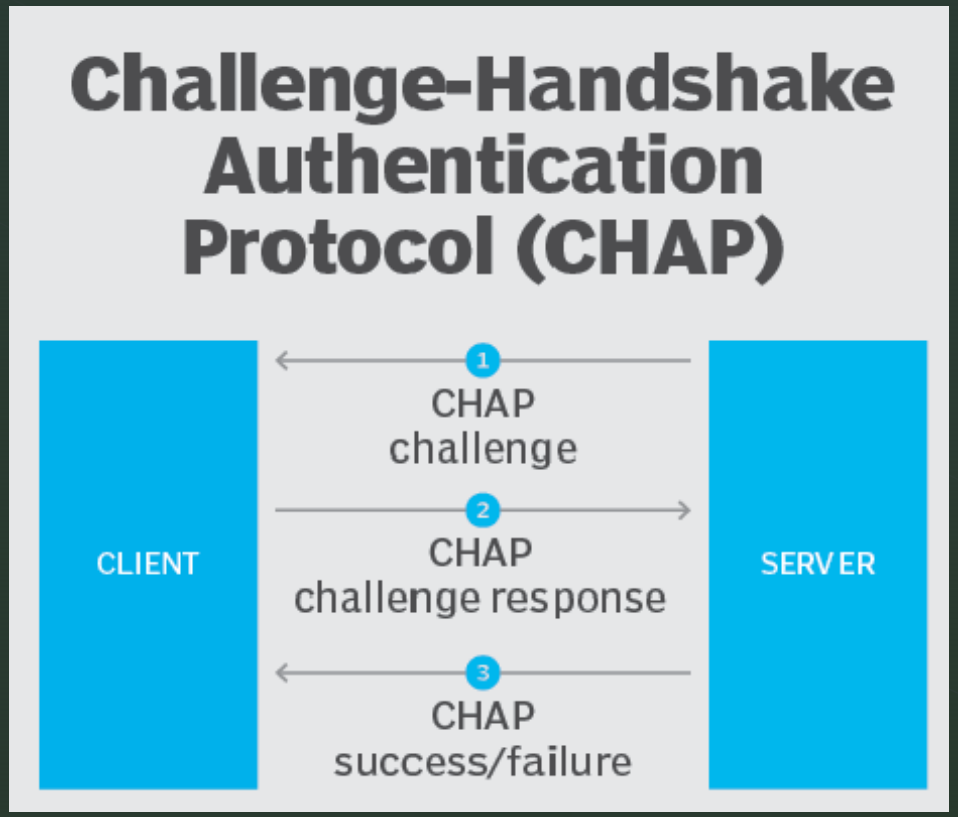
PAP.

- The Password Authentication Protocol (PAP) is a simple authentication procedure with a two-step process:
  - The user who wants to access a system sends an authentication identification (usually the user name) and a password.
  - The system checks the validity of the identification and password and either accepts or denies connection.



CHAP.

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



- In this method, the password is kept secret; it is never sent online.
  - The system sends the user a challenge packet containing a challenge value, usually a few bytes.
  - 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.
  - 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.

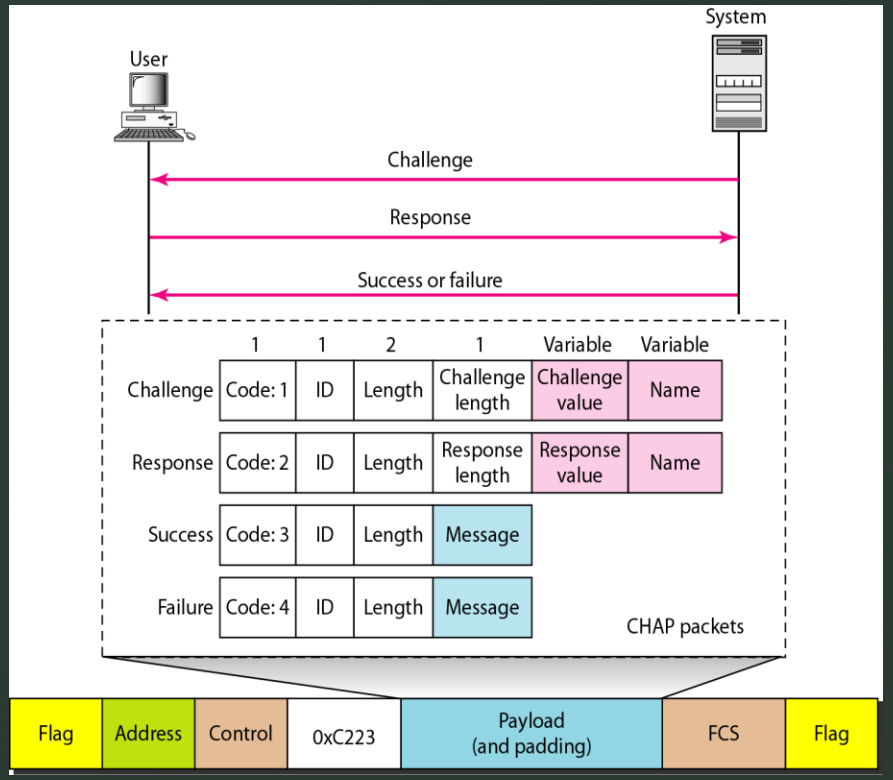


Figure : CHAP packets encapsulated in a PPP frame

- There are four CHAP packets:
  - Challenge - used by the system to send the challenge value
  - Response - used by the user to return the result of the calculation
  - Success - used by the system to allow access to the system
  - Failure - used by the system to deny access to the system

Network Control Protocols

- PPP is a multiple-network-layer protocol.
- It can carry a network-layer data packet from protocols defined by the Internet, OSI, Xerox, DECnet, AppleTalk, Novel, and so on.
- Note that none of the NCP packets carry network-layer data; they just configure the link at the network layer for the incoming data.
- One NCP protocol is the Internet Protocol Control Protocol (IPCP).
  - This protocol configures the link used to carry IP data packets in the Internet.

Data from the Network Layer

- After the network-layer configuration is completed by one of the NCP protocols, users can exchange data packets from the network layer.
- Here again, there are different protocol fields for different network layers.
- For example, if PPP is carrying data from the IP network layer, the field value is $(0021)_{16}$.
- If PPP is carrying data from the OSI network layer, the protocol field value is $(0023)_{16}$, and so on.

Multilink PPP

- The availability of multiple channels in a single point-to-point link motivated the development of Multilink PPP
- 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.

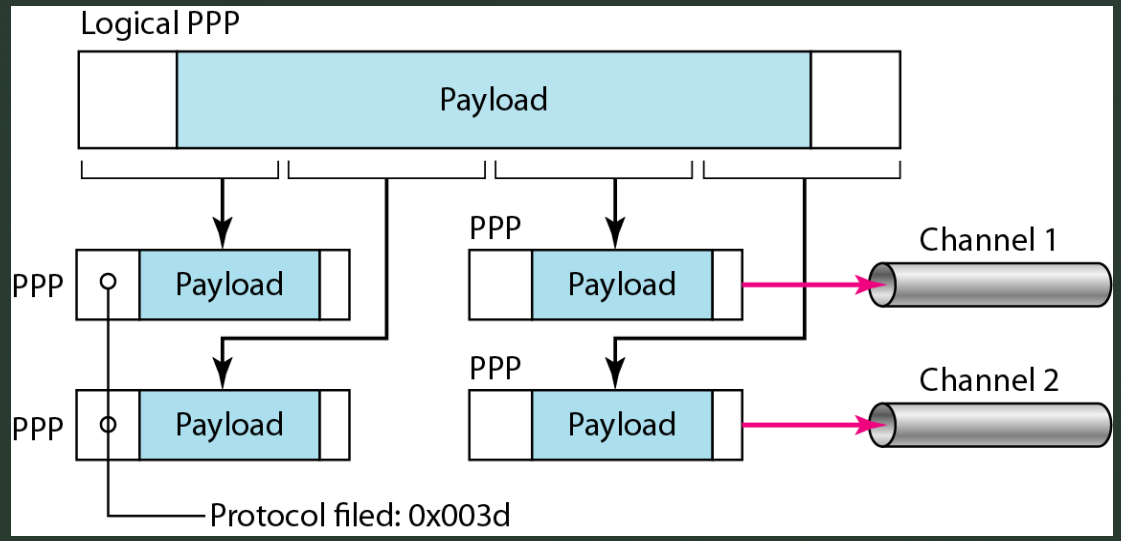


Figure: Multilink PPP

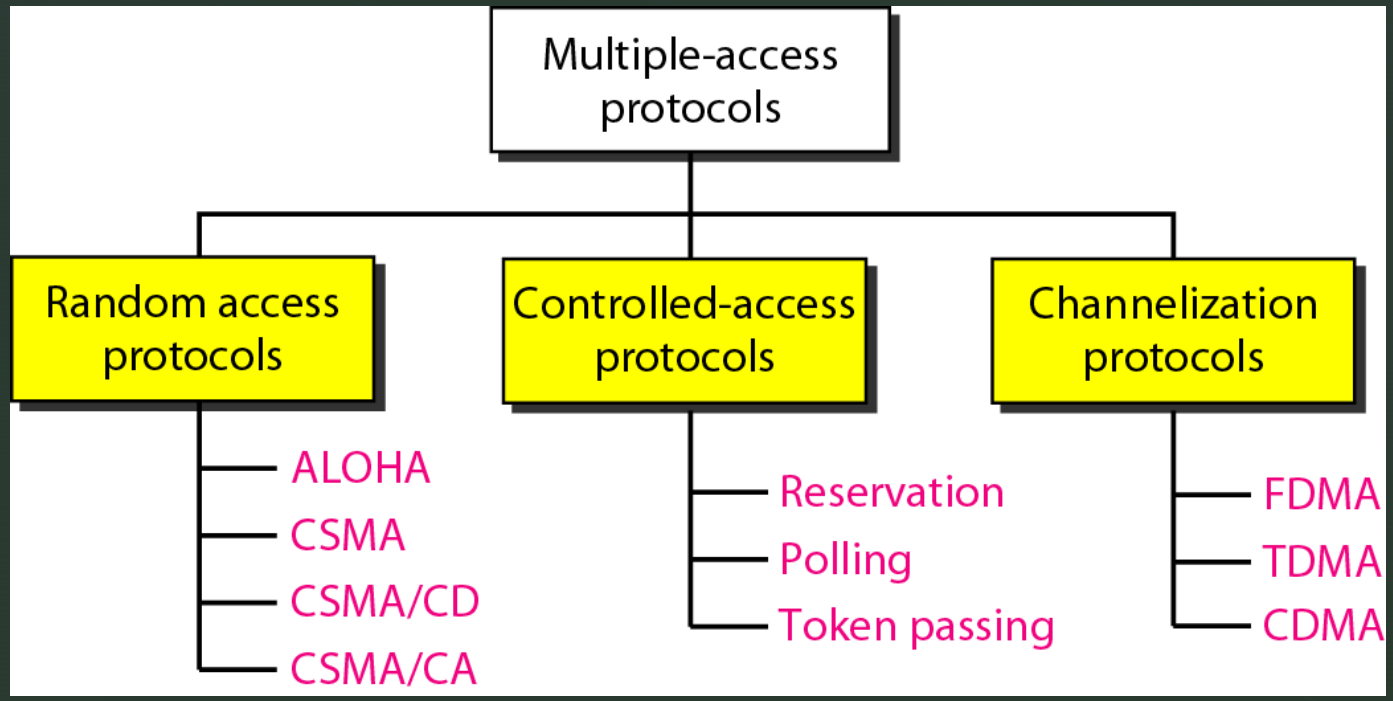
- To show that the actual PPP frame is carrying a fragment of a logical PPP frame, the protocol field is set to $(003d)_{16.}$
- This new development adds complexity
  - For example, a sequence number needs to be added to the actual PPP frame to show a fragment's position in the logical frame.

MEDIA ACCESS PROTOCOLS

- Data-link layer is divided into two sublayers:
  - Data-link control (DLC) and
  - Media access control.
- While using a dedicated link, such as a dial-up telephone line, we need only a data-link-control protocol, such as the Point-to-Point Protocol (PPP), that manages the data transfer between the two ends.
- On the other hand, while sharing the media, wire or air, with other users, we need to have a protocol to first manage the sharing process and then to do the data transfer.

- When nodes or stations are connected and use a common link, called a multipoint or broadcast link, we need a multiple-access protocol to coordinate access to the link.
- The problem of controlling access to the medium is similar to the rules of speaking in an assembly.
- The procedures guarantee that the right to speak is upheld and ensure that two people do not speak at the same time, do not interrupt each other, do not monopolize the discussion, and so on.
- The situation is similar for multipoint networks. We need to be sure that each node gets access to the link.
- The first goal is to prevent any collision between nodes. If somehow a collision does occur, the second goal is to handle the collision.

Many protocols have been devised to handle access to a shared link. They are categorize into three groups. Protocols belonging to each group are shown in Figure



Random Access

- In random-access or contention methods, no station is superior to another station and none is assigned control over another.
- At each instance, a station that has data to send uses a procedure defined by the protocol to make a decision on whether or not to send.
- Decision depends on the state of the medium (idle or busy).
- It follows the predefined procedure, including the testing of the state of the medium.

Two features give this method its name.

- First, there is no scheduled time for a station to transmit. Transmission is random among the stations. That is why these methods are called random access.
- Second, no rules specify which station should send next. Stations compete with one another to access the medium. That is why these methods are also called contention methods.

- If more than one station tries to send, there is an access conflict—collision—and the frames will be either destroyed or modified.
- To avoid access conflict or to resolve it when it happens, each station follows a procedure that answers the following questions:
  - When can the station access the medium?
  - What can the station do if the medium is busy?
  - How can the station determine the success or failure of the transmission?
  - What can the station do if there is an access conflict?

- The random-access methods has evolved from a very interesting protocol known as ALOHA, which uses a very simple procedure called Multiple Access (MA).
- The method was improved with the addition of a procedure that forces the station to sense the medium before transmitting. This was called Carrier Sense Multiple Access (CSMA).
- This method later evolved into two parallel methods:
  - Carrier Sense Multiple Access with Collision Detection (CSMA/CD), which tells the station what to do when a collision is detected, and
  - Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), which tries to avoid the collision.

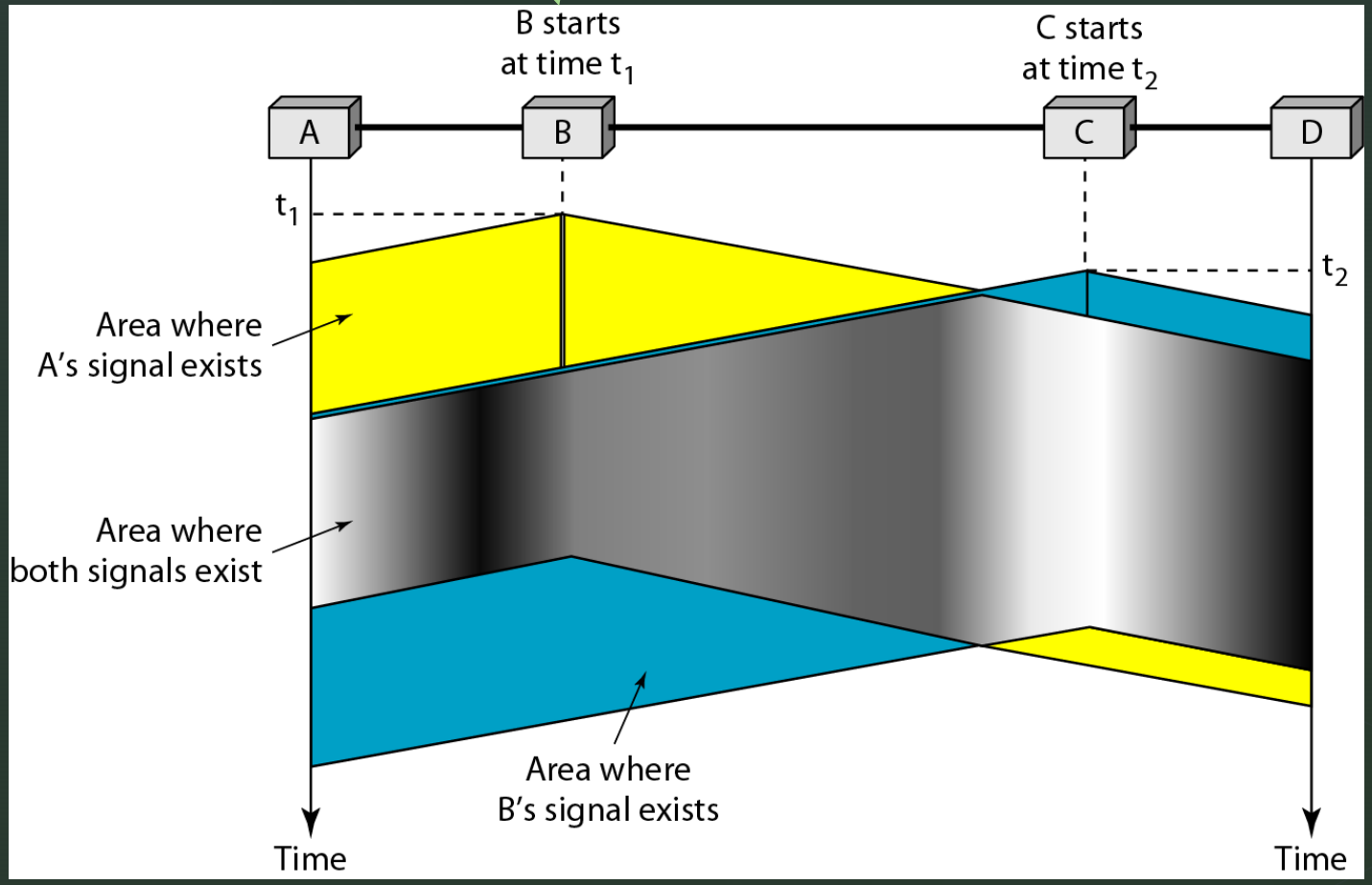
Carrier Sense Multiple Access (CSMA)

- Minimize the chance of collision, therefore, increase the performance
- How to minimize chance of collision?
  - A station should sense the medium before trying to use it (space and time model)
    - Each station first listen to the medium (or check the state of the medium) before sending
      - Principle “sense before transmit” or “listen before talk”

CSMA - space and time model

station B senses the medium and finds it idle, so it sends a frame

station C senses the medium and finds it idle as the first bits from station B have not reached station C*, so it sends a frame*



two signals collide and both frames are destroyed

CSMA - space and time model:

- Stations are connected to a shared channel
- The possibility of collision still exists because of propagation delay.
  - When a station sends a frame, it still takes time (although very short) for the first bit to reach every station and for every station to sense it
    - A station may sense the medium and find it idle, only because the first bit sent by another station has not yet been received

CSMA - vulnerable time

- *Vulnerable time =* ***propagation time T_p***
  - T_p is time needed for a signal to propagate from one end of the medium to the other.
  - When a station sends a frame and any other station tries to send a frame during this time, a collision will result.
    - If the first bit of the frame reaches the end of the medium, every station will already have heard the bit and will refrain from sending

- Worst Case:
  - Leftmost station, A, sends a frame at time t1, which reaches the rightmost station, D, at time t1 + $T_{p.}$
- The blue area shows the vulnerable area in time and space.

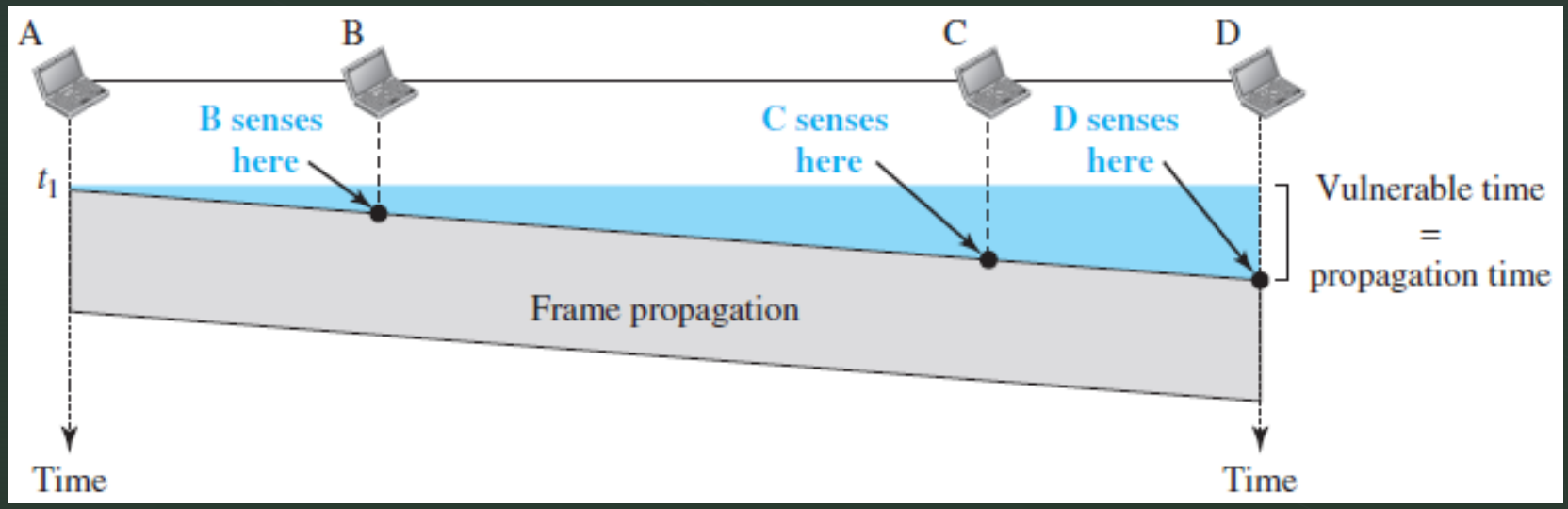


Figure: Vulnerable time in CSMA

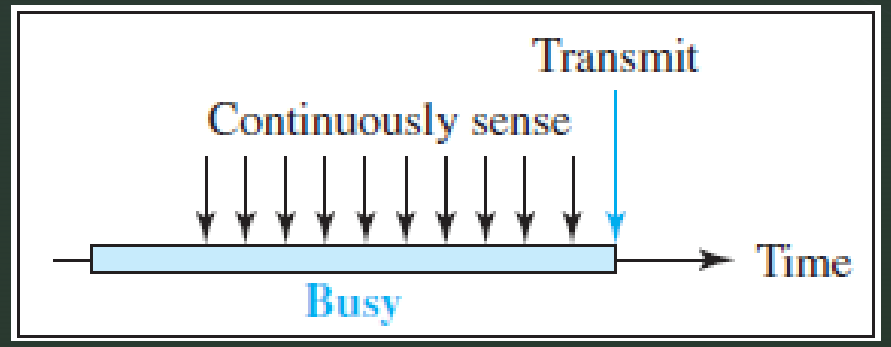
Persistence Methods

1. What should a station do if the channel is busy?
2. What should a station do if the channel is idle?

- Three methods have been devised to answer these questions:
  - 1- persistent method,
  - Nonpersistent method, and
  - p-persistent method.

1-Persistent:

- The 1-persistent method is simple and straightforward.
- After the station finds the line idle, it sends its frame immediately (with probability 1).
- This method has the highest chance of collision because two or more stations may find the line idle and send their frames immediately.
- Ethernet uses this method.



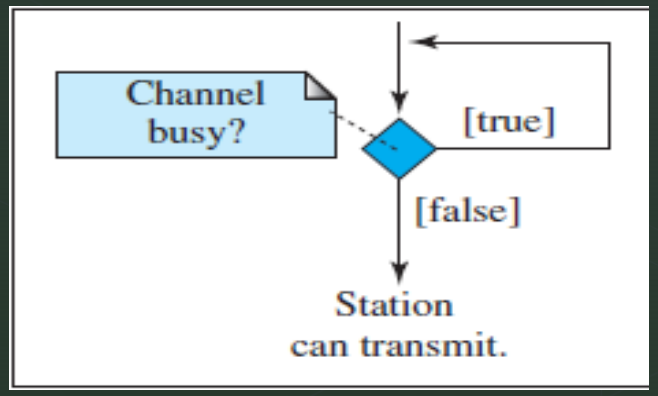


Figure: 1-persistent

Nonpersistent :

- In the nonpersistent method, a station that has a frame to send senses the line.
- If the line is idle, it sends immediately. If the line is not idle, it waits a random amount of time and then senses the line again.
- The nonpersistent approach reduces the chance of collision because it is unlikely that two or more stations will wait the same amount of time and retry to send simultaneously.
- However, this method reduces the efficiency of the network because the medium remains idle when there may be stations with frames to send.

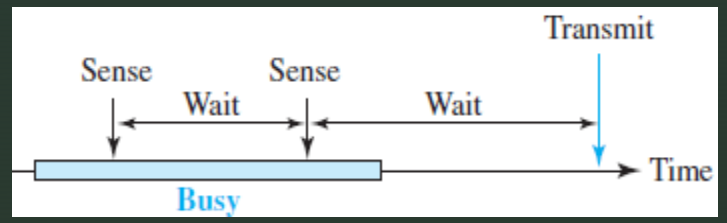
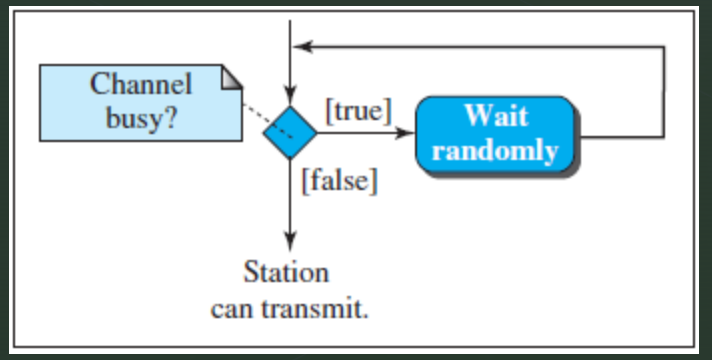


Figure: Nonpersistent



p-Persistent:

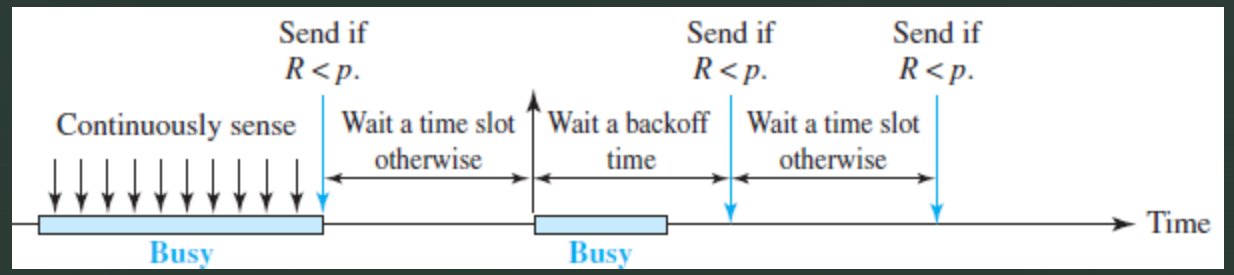
- The p-persistent method is used if the channel has time slots with a slot duration equal to or greater than the maximum propagation time.
- The p-persistent approach combines the advantages of the other two strategies.
- It reduces the chance of collision and improves efficiency.
- In this method,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

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

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

    a. If the line is idle, it goes to step 1.

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



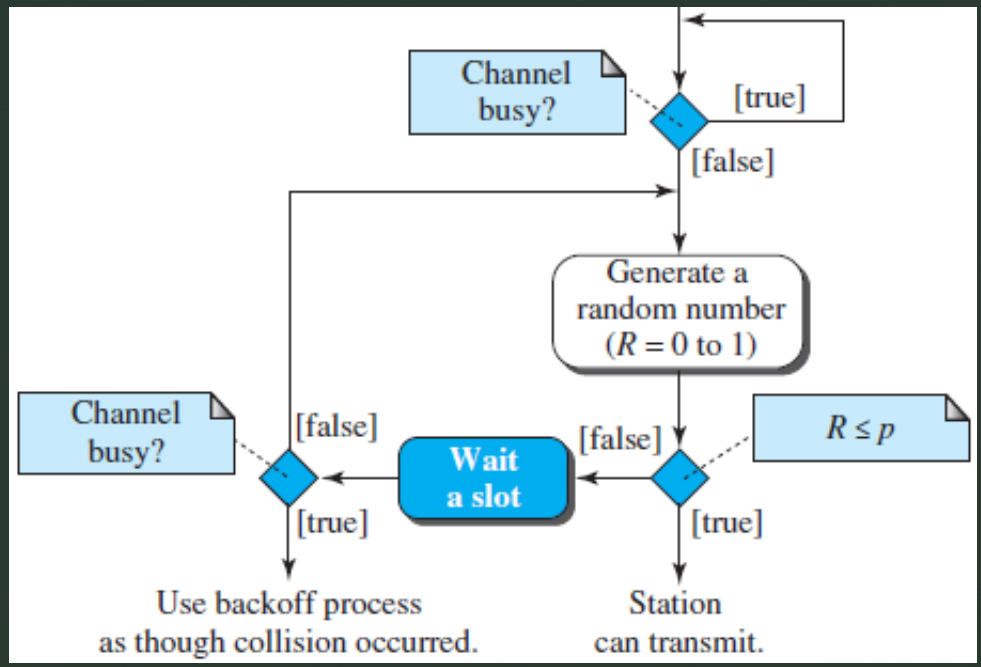


Figure: p-persistent

CSMA/CD

- Carrier sense multiple access with collision detection (CSMA/CD) augments the algorithm to handle the collision.
- In this method, a station monitors the medium after it sends a frame to see if the transmission was successful.
- If so, the station is finished. If, however, there is a collision, the frame is sent again.

Let us look at the first bits transmitted by the two stations involved in the collision. Although each station continues to send bits in the frame until it detects the collision, we show what happens as the first bits collide.

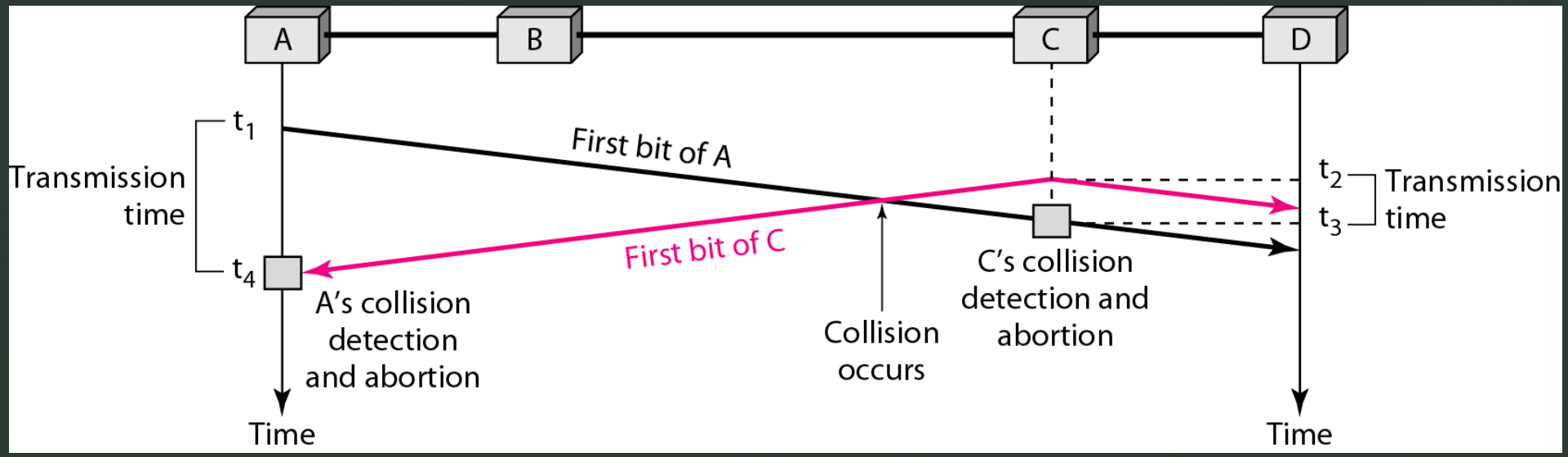


Figure : Collision of the first bits in CSMA/CD

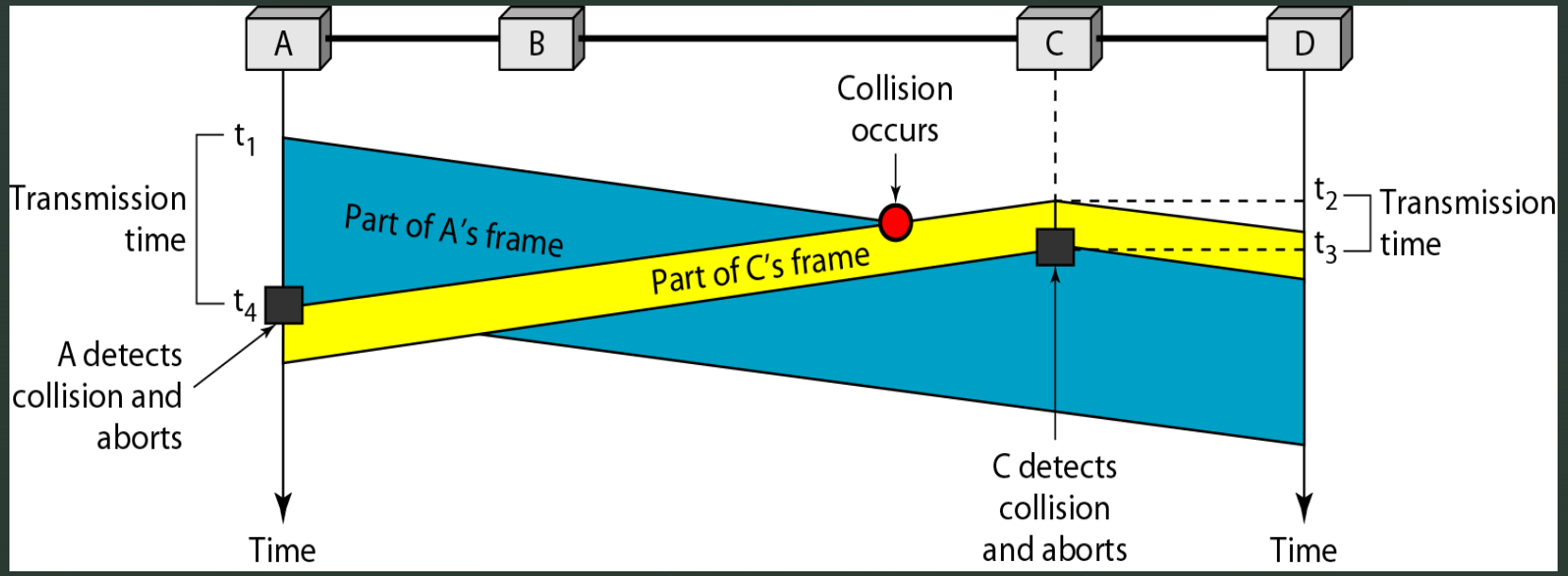


Figure : Collision and abortion in CSMA/CD

Minimum Frame Size

- Need a restriction on the frame size
- Before sending the last bit of the frame, the sending station must detect a collision, if any, and abort the transmission.
  - Because the station, once the entire frame is sent, does not keep a copy of the frame and does not monitor the line for collision detection.
  - The frame transmission time T_{fr} *must be at least two times the maximum propagation time* Tp.
  - Worst-case scenario: If the two stations involved in a collision are the maximum distance apart, the signal from the first takes time T_p *to reach* the second, and the effect of the collision takes another time T_P *to reach the first.*
  - *So the* requirement is that the first station must still be transmitting after $2T_p$.

CSMA/CD Example:

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) is 25.6 μs, what is the minimum size of the frame?

Solution

- The minimum frame transmission time is $T_{fr} = 2 \times T_p = 51.2\ \mu s$.
- *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.

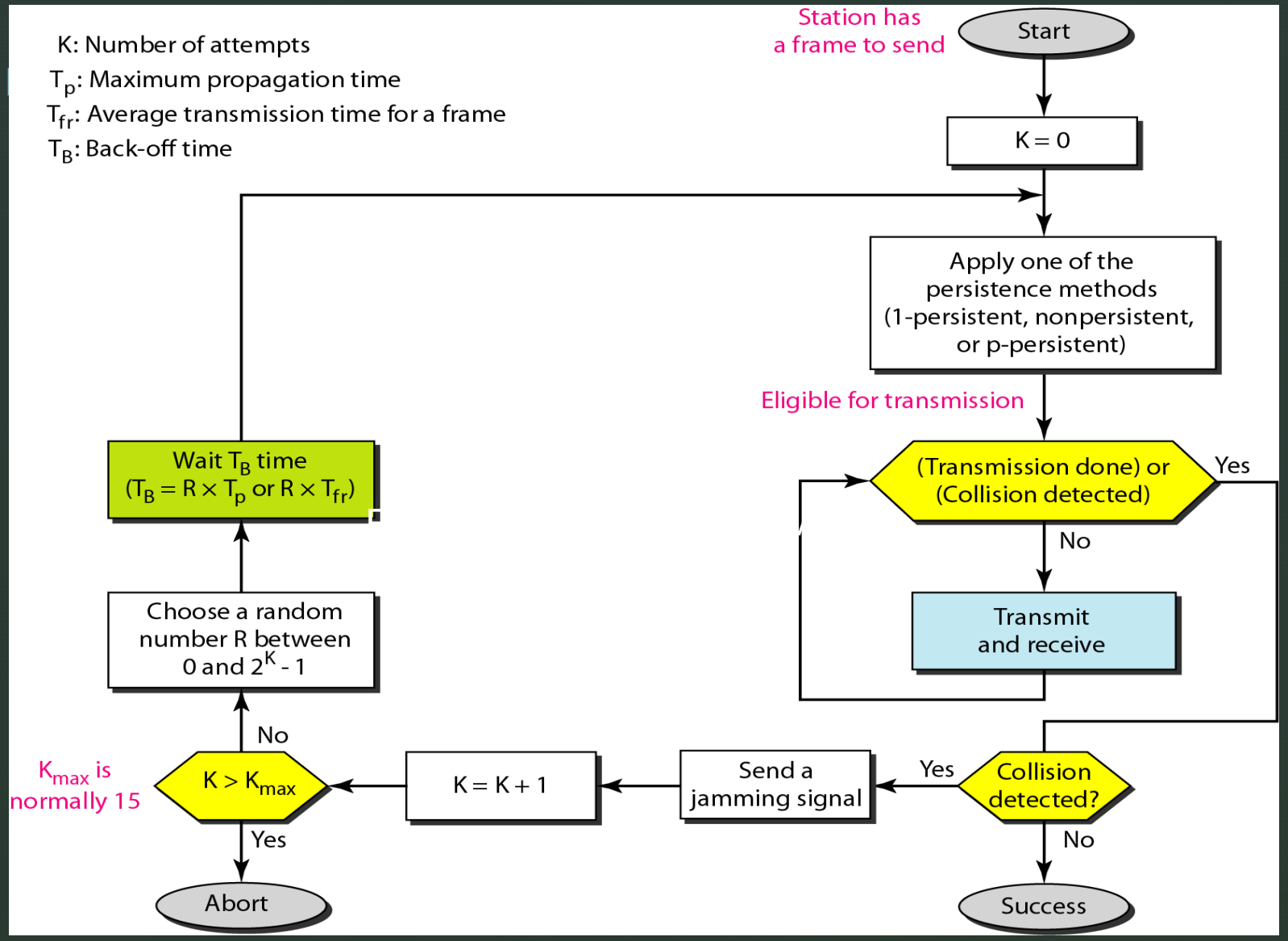


Figure : Flow diagram for the CSMA/CD

- Similar to the one for the ALOHA protocol with 3-differences
  1. Addition of the persistence process - sense the channel before start sending the frame by using one of the persistence processes (Nonpersistent, 1-persistent, or *p-persistent).*
  2. Frame transmission:
     - In ALOHA, first transmit the entire frame and then wait for an acknowledgment
     - In CSMA/CD, transmission and collision detection are continuous processes (shown as a loop)
       - constantly monitor in order to detect one of two conditions to stop transmission
         - either transmission is finished or
         - a collision is detected
       - On loop exit, if a collision has not been detected, it means that transmission is complete; the entire frame is transmitted. Otherwise, a collision has occurred
     - Sending of a short **jamming signal** **to make sure that all** other stations become aware of the collision

Energy Level:

- Level of energy in a channel can have three values: zero, normal, and abnormal.
- At the zero level, the channel is idle.
- At the normal level, a station has successfully captured the channel and is sending its frame.
- At the abnormal level, there is a collision and the level of the energy is twice the normal level.
- A station that has a frame to send or is sending a frame needs to monitor the energy level to determine if the channel is idle, busy, or in collision mode.

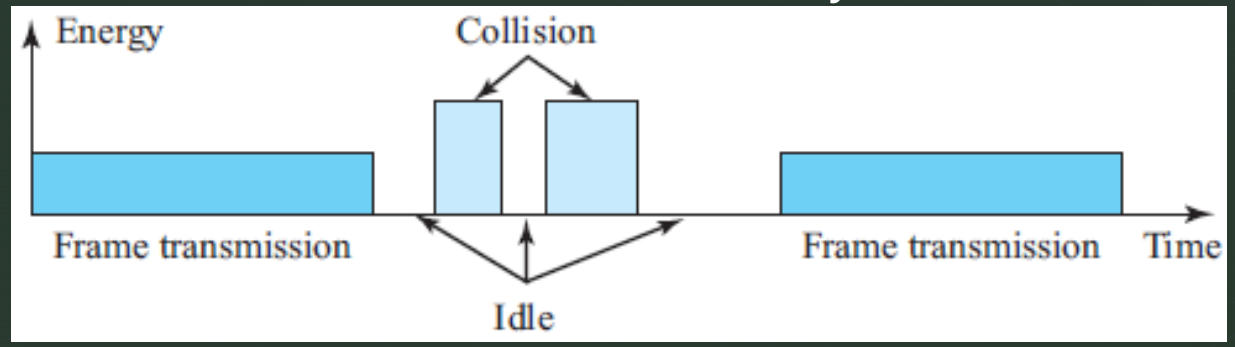


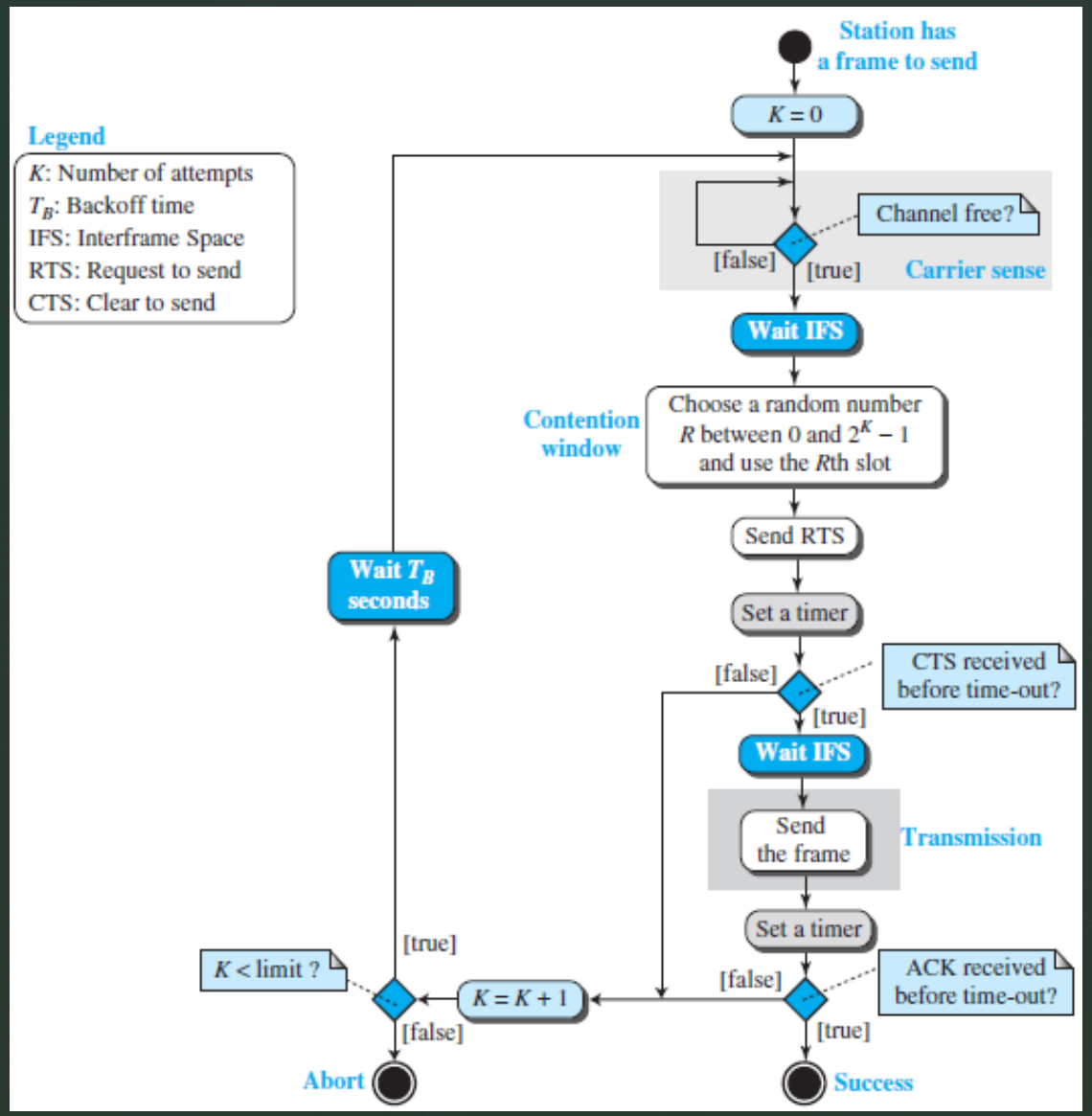
Figure : Energy level during transmission, idleness, or collision

CSMA/CD - *Throughput*

- The maximum throughput occurs at a different value of *G(*G the average number of frames generated by the system during one frame transmission time.) *and is based on the persistence* method and the value of *p in the p-persistent approach*
  - *For the 1-persistent method, the* maximum throughput is around 50 percent when *G = 1*
  - *For the nonpersistent method,* the maximum throughput can go up to 90 percent when *G is between 3 and 8*
- *Traditional Ethernet is a* broadcast LAN that used the 1-persistence method to control access to the common media with the data rate of 10 Mbps.

Carrier sense multiple access with collision avoidance - CSMA/CA

- **Invented** for wireless networks
- Collisions are avoided through the use of CSMA/CA's three strategies:
  - the interframe space
  - the contention window
  - acknowledgments



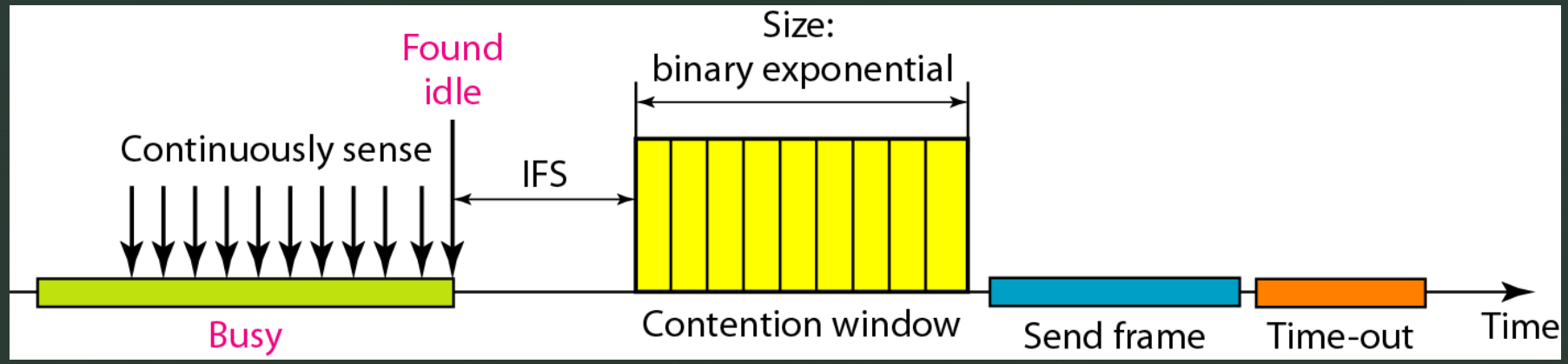
CSMA/CA- *Interframe Space (IFS) :*

- *Avoids collisions by deferring transmission even* if the channel is found idle.
- When an idle channel is found, the station waits for a period of *IFS* time
- IFS time allows the front of the transmitted signal by the distant station to reach this station
- After waiting an IFS time, if the channel is still idle, the station can send, but it still needs to wait a time equal to the contention window
- The 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

CSMA/CA - *Contention Window :*

- ***An amount of time divided into*** slots
- A station that is ready to send chooses a random number of slots as its wait time
- The number of slots in the window changes according to the binary exponential backoff strategy
  - It is set to one slot the first time and then doubles each time the station cannot detect an idle channel after the IFS time
  - Very similar to the *p-persistent method except that a random outcome defines the* number of slots taken by the waiting station
- The station needs to sense the channel after each time slot
- However, if the station finds the channel busy, it does not restart the process; it just stops the timer and restarts it when the channel is sensed as idle
- This gives priority to the station with the longest waiting time

CSMA/CA - Timing



CSMA/CA - *Acknowledgment*

- With all these precautions, there still may be a collision resulting in destroyed data
- In addition, the data may be corrupted during the transmission
- The positive acknowledgment and the time-out timer can help guarantee that the receiver has received the frame

- CSMA/CA -*Frame Exchange Time Line :*

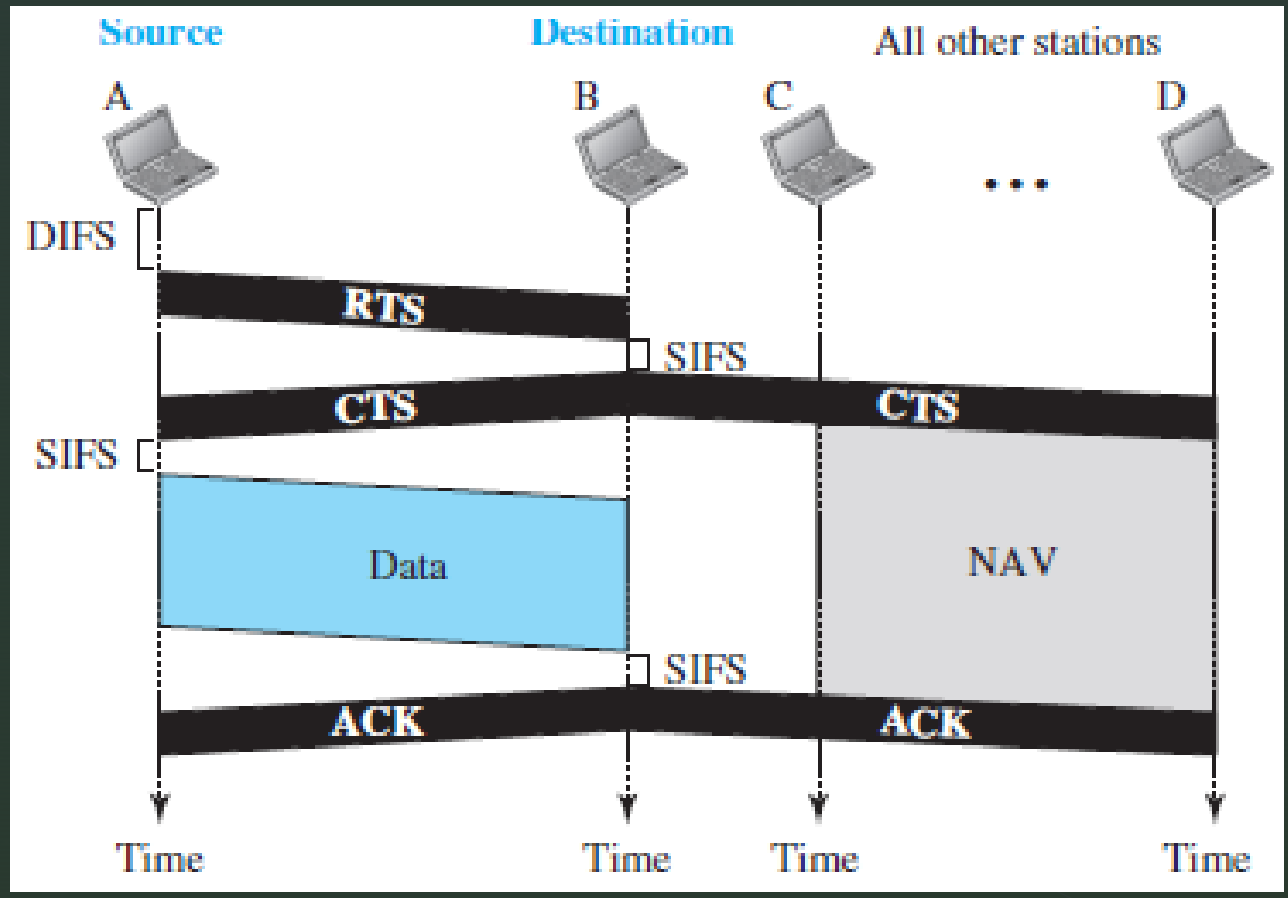


Figure : CSMA/CA and NAV

CSMA/CA -*Frame Exchange Time Line*

- Exchange of data and control frames in time.

1. Before sending a frame, the source station senses the medium by checking the energy level at the carrier frequency

   a. The channel uses a persistence strategy with backoff until the channel is idle

   b. After the station is found to be idle, the station waits for a period of time called the *DCF interframe space (DIFS); then the station sends a control frame called* the *request to send (RTS).*

2. After receiving the RTS and waiting a period of time called the *short interframe space (SIFS), the destination station sends a control frame, called the clear to send (CTS), to the source station. This control frame indicates that the destination* station is ready to receive data.

3. The source station sends data after waiting an amount of time equal to SIFS.

4. The destination station, after waiting an amount of time equal to SIFS, sends an acknowledgment to show that the frame has been received. Acknowledgment is needed in this protocol because the station does not have any means to check for the successful arrival of its data at the destination. On the other hand, the lack of collision is a kind of indication to the source that data have arrived.

CSMA/CA - *Network Allocation Vector :*

1. How do other stations defer sending their data if one station acquires access?
2. how is the *collision avoidance aspect of this protocol accomplished?*

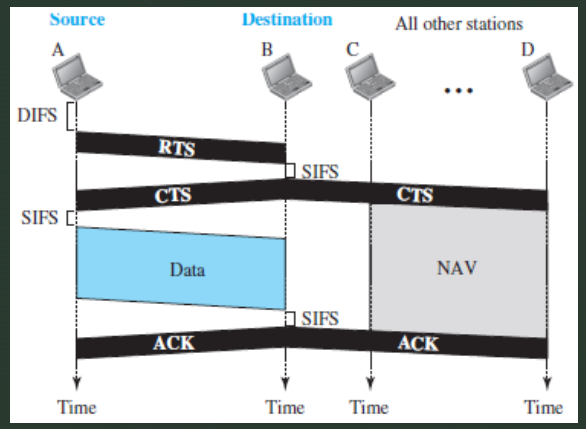
- *Use key* feature *Network Allocation Vector (NAV):*
  - When a station sends an RTS frame, it includes the duration of time that it needs to occupy the channel.
  - The stations that are affected by this transmission create a timer called a **network allocation vector (NAV) that shows how much time must pass before** these stations are allowed to check the channel for idleness.
  - Each time a station accesses the system and sends an RTS frame, other stations start their NAV.

Collision During Handshaking :

- Two or more stations may try to send RTS frames at the same time. These control frames may collide.
- However, because there is no mechanism for collision detection, the sender assumes there has been a collision if it has not received a CTS frame from the receiver.
- The backoff strategy is employed, and the sender tries again.

CSMA/CA - Hidden-Station Problem

- Use of the handshake frames (RTS and CTS)
- The RTS message from B reaches A, but not C.
- However, because both B and C are within the range of A, the CTS message, which contains the duration of data transmission from B to A, reaches C
- Station C knows that some hidden station is using the channel and refrains from transmitting until that duration is over



Controlled Access

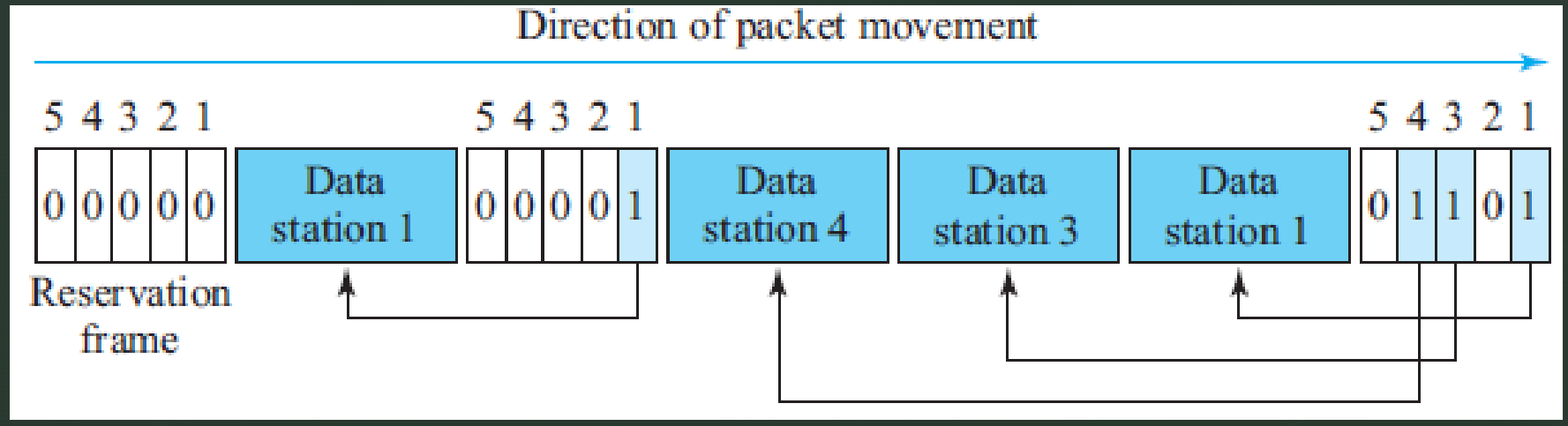
- In controlled access, the stations consult one another to find which station has the right to send.
- A station cannot send unless it has been authorized by other stations.
- Three controlled-access methods:
  - Reservation,
  - Polling, and
  - Token passing.

Reservation :

- In the reservation method, a station needs to make a reservation before sending data.
- Time is divided into intervals. In each interval, a reservation frame precedes the data frames sent in that interval.
- If there are N stations in the system, there are exactly N reservation minislots in the reservation frame.
- Each minislot belongs to a station. When a station needs to send a data frame, it makes a reservation in its own minislot.
- The stations that have made reservations can send their data frames after the reservation frame

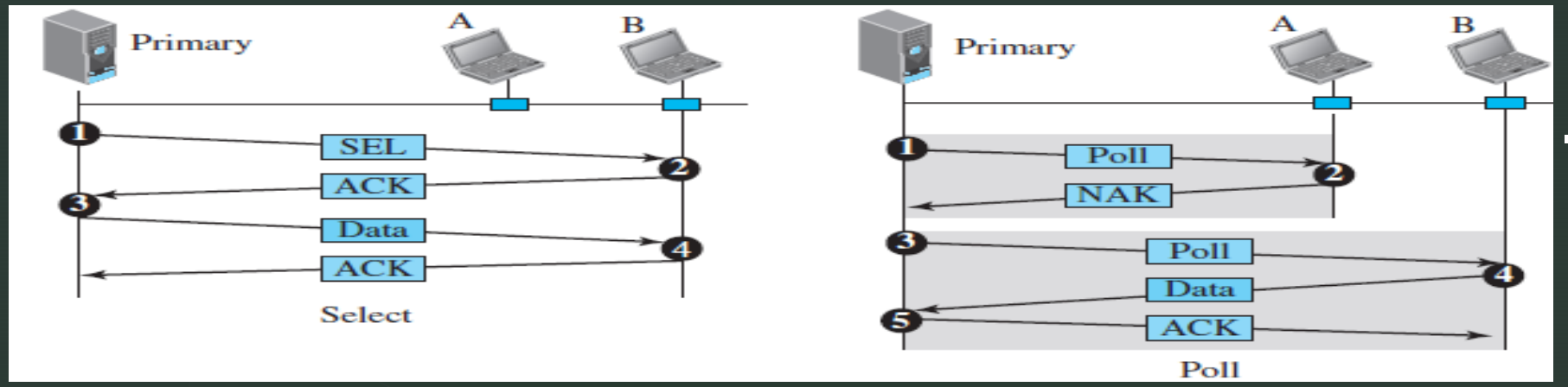
Reservation access method

- Five stations and a five-minislot reservation frame
- In the first interval, only stations 1, 3, and 4 have made reservations
- In the second interval, only station 1 has made a reservation



Polling :

- Works with topologies in which one device is designated as a primary station that controls the link and the other devices are secondary stations that follow its instructions
- All data exchanges must be made through the primary device even when the ultimate destination is a secondary device
- Primary device - determine which device is allowed to use the channel at a given time and always the initiator of a session
- uses poll and select functions to prevent collisions
- Drawback - if the primary station fails, the system goes down



- *used whenever the primary device has something to send.*
- *Primary* does not know whether the target device is prepared to receive
- Primary alert the secondary about the upcoming transmission and wait for an acknowledgment of the secondary's ready status using select (SEL) frame
- One field of SEL includes the address of the intended secondary

- *Used by the primary device to solicit transmissions from the secondary* devices
- When the primary is ready to receive data, it must ask (poll) each device in turn if it has anything to send
- When the first secondary is approached, it responds either with a AK frame if it has nothing to send or with data if it does
- If the response is negative (a NAK frame), then the primary polls the next secondary in the same manner until it finds one with data to send
- When the response is positive (a data frame), the primary reads the frame and returns an acknowledgment (ACK frame), verifying its receipt

Token Passing

- **The stations in a network are organized in a logical ring where** each station has a *predecessor and a successor*
- *The predecessor* is the station which is logically before the station in the ring; 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
- The right to this access has been passed from the predecessor to the current station then to the successor when the current station has no more data to send

how is the right to access the channel passed from one station to another?

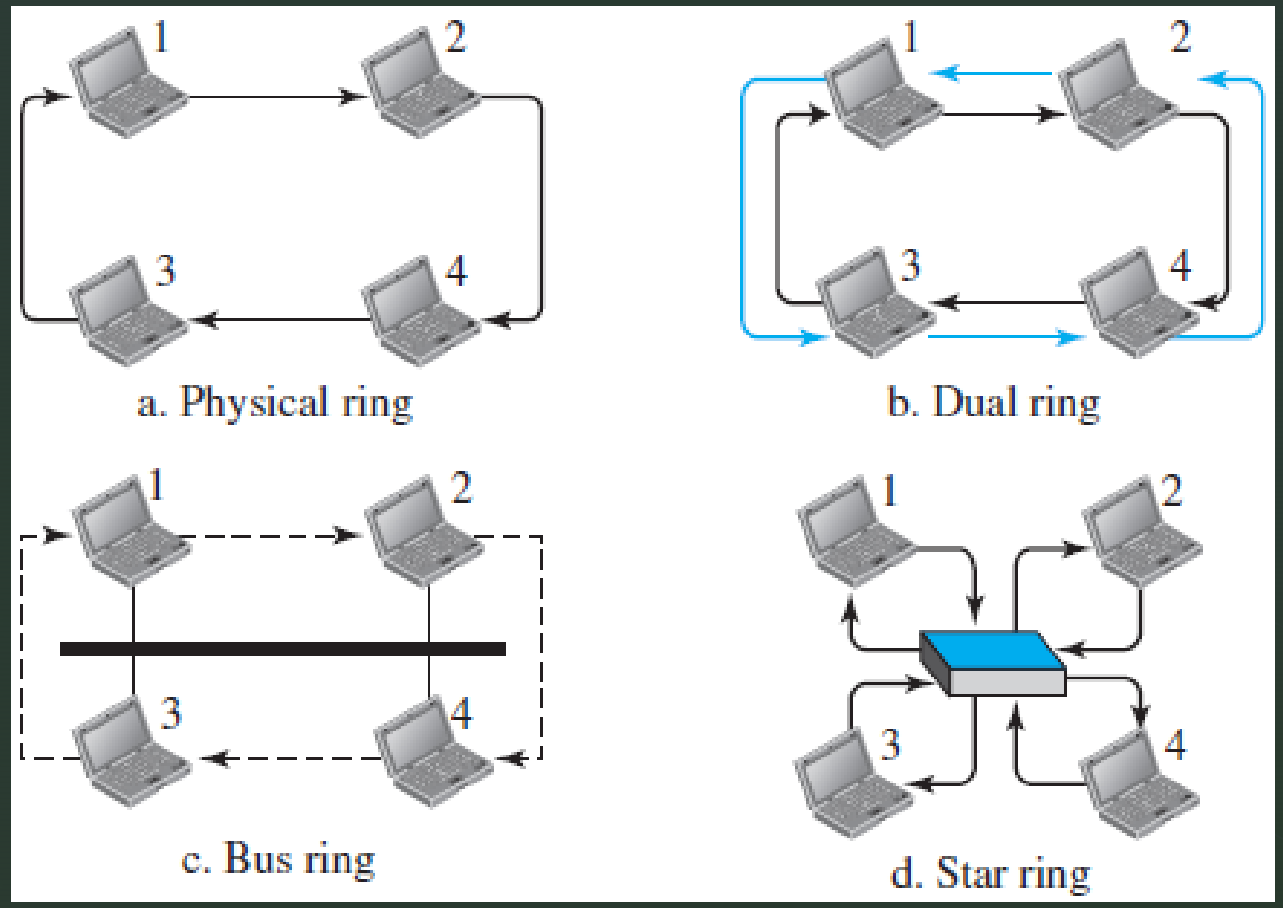
- A special packet called a ***token circulates through the ring***
- ***The possession*** of the token gives the station the right to access the channel and send its data
- When a station has some data to send, it waits until it receives the token from its predecessor
- It then holds the token and sends its data
- When the station has no more data to send, it releases the token, passing it to the next logical station in the ring
- The station cannot send data until it receives the token again in the next round
- When a station receives the token and has no data to send, it just passes the data to the next station

Token management

- Stations must be limited
- The token must be monitored to ensure it has not been lost or destroyed
  - For example, if a station that is holding the token fails, the token will disappear from the network
- Assign priorities to the stations and to the types of data being transmitted
- make low-priority stations release the token to high-priority stations

Logical Ring

- Four different physical topologies that can create a logical ring



END – Unit 4