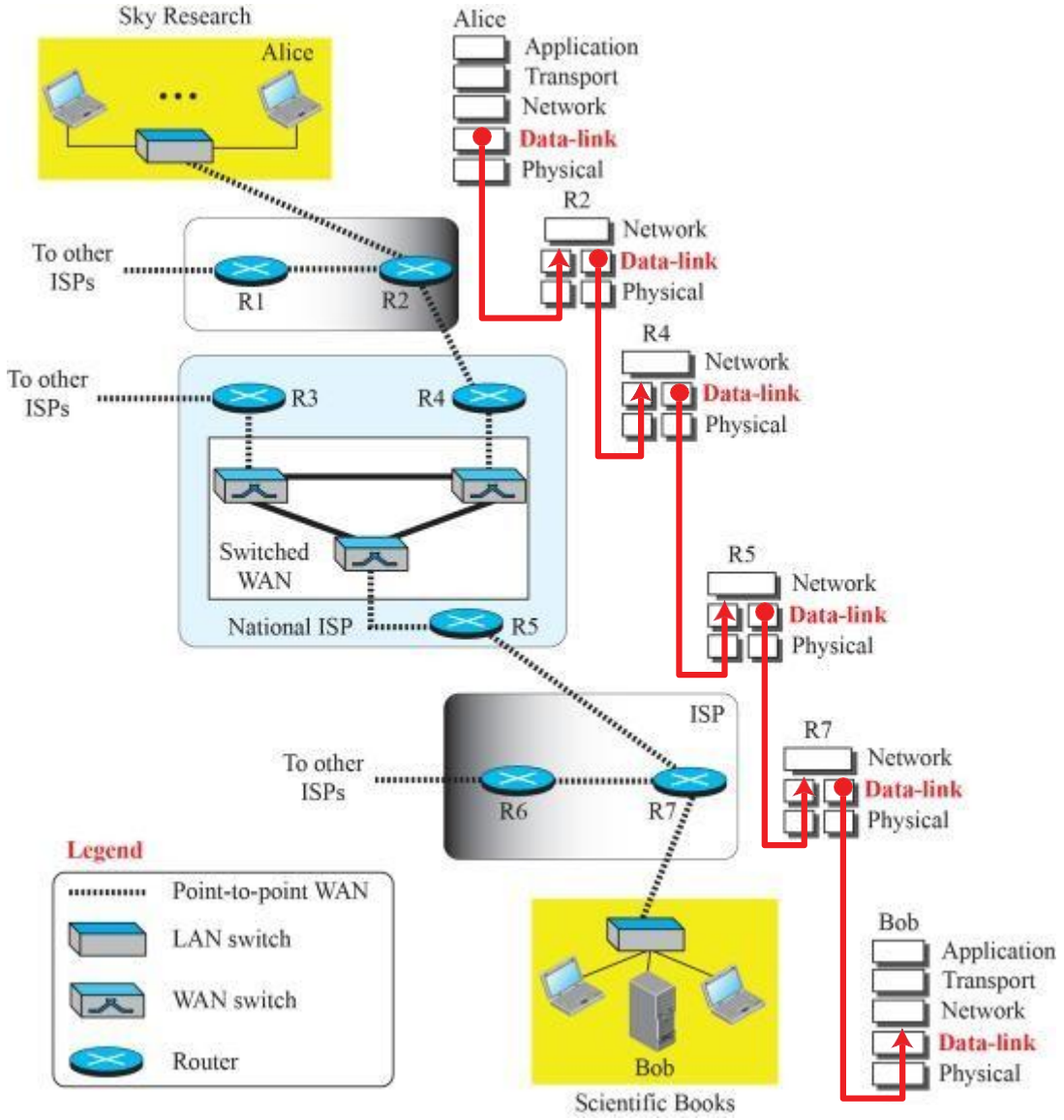
Data-Link Layer: Wired Networks

Chapter 5

INTRODUCTION

In networking, communication at the network layer is like sending postcards from one person to another. These postcards, called datagrams, travel from one host to another worldwide. The Internet is a collection of interconnected networks, and to reach its destination, a datagram has to pass through these networks.

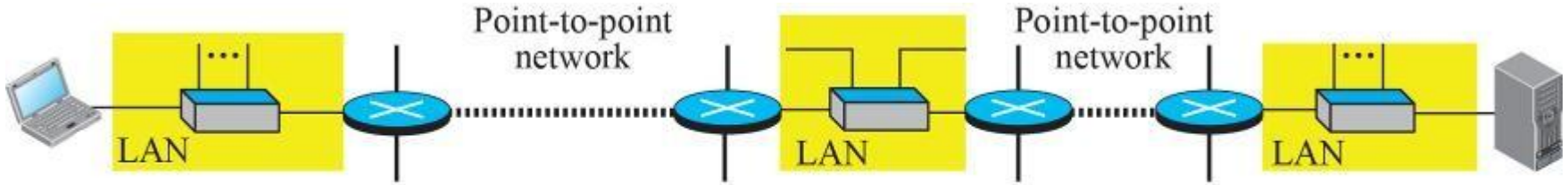
Imagine Alice sending a message to Bob. At the data-link layer, there are five logical connections between devices. Alice's computer talks to router R2, which talks to router R4, and so on, until reaching Bob's computer. Only one data-link layer is involved at the source and destination, but each router in between has two. This is because routers connect different networks, guiding the datagram on its journey.


Sky Research
Alice
Alice
Application
Transport
Network
Data-link
Physical
R2
Network
Data-link
Physical
To other ISPs
R1
R2
R4
Network
Data-link
Physical
To other ISPs
R3
R4
Switched WAN
National ISP
R5
R5
Network
Data-link
Physical
ISP
To other ISPs
R6
R7
R7
Network
Data-link
Physical
Legend
Point-to-point WAN
LAN switch
WAN switch
Router
Bob
Scientific Books
Bob
Application
Transport
Network
Data-link
Physical

Nodes and Links

Alright, think of internet communication like a relay race. At the application, transport, and network layers, it's like passing a baton from one runner directly to the next. But at the data-link layer, it's more like passing the baton from one runner to the next in a chain.

So, your data goes through different networks (like local and wide-area networks) connected by routers, and these are the nodes. The links between them are like the paths. Picture it as a relay with two end runners (your devices) and routers in between, each handling a part of the race. The whole journey involves six nodes, and the links represent the connections between them. Easy enough, right?



a. A small part of the Internet

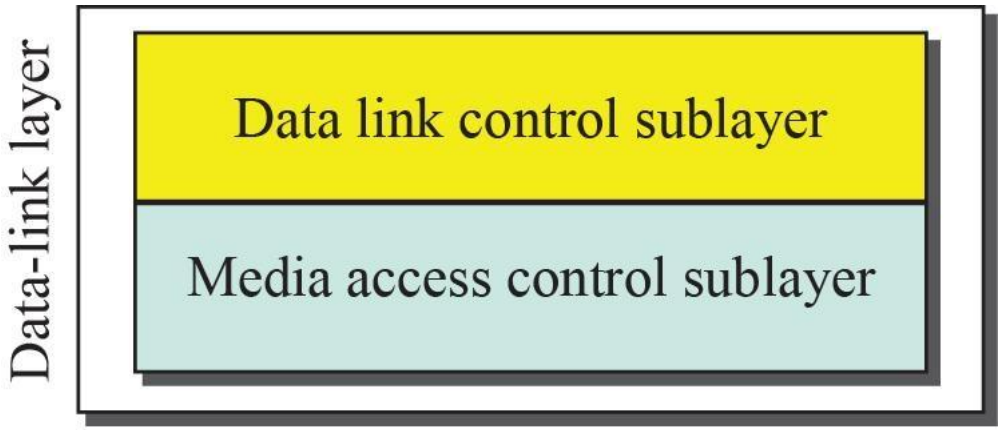
b. Nodes and links

Two Types of Links

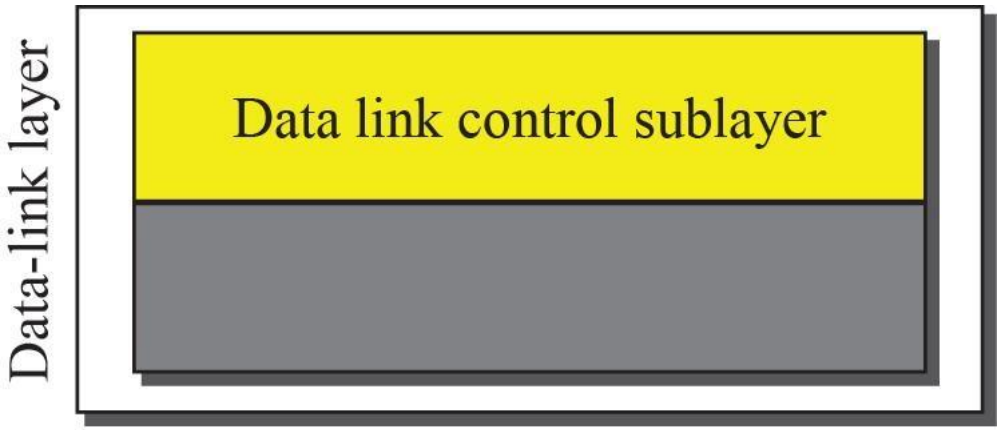
In networking, how data moves between nodes depends on the data-link layer. **This layer controls the use of the transmission medium—whether it's fully utilized or shared**. There are point-to-point links (exclusive to two devices) and broadcast links (shared among multiple pairs). Think of a landline call between friends as point-to-point, while chatting on cell phones involves a broadcast link, shared with others using the airwaves.

Two Sublayers

The data-link layer, which manages how data moves between nodes, has two parts: data link control (DLC) and media access control (MAC). DLC handles common issues for both point-to-point and broadcast links, while MAC focuses on broadcast-specific matters. Think of it like sorting tasks—DLC deals with shared problems, and MAC handles stuff unique to broadcast links. We'll dive into DLC first, then MAC, and explore a protocol from each category.



a. Data-link layer of a broadcast link



b. Data-link layer of a point-to-point link

DATA LINK CONTROL (DLC)

Data link control manages communication between adjacent nodes, focusing on framing, flow and error control, and error detection/correction, regardless of dedicated or broadcast links. **Framing organizes bits, while flow and error control ensure smooth transmission**. Error detection techniques are covered later in this section.

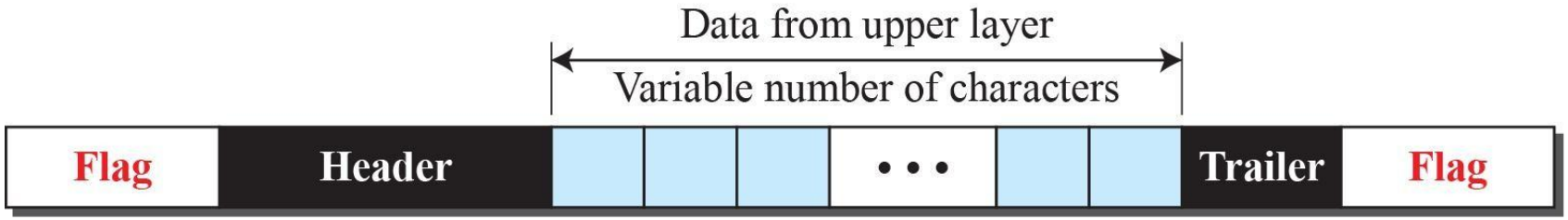
Framing

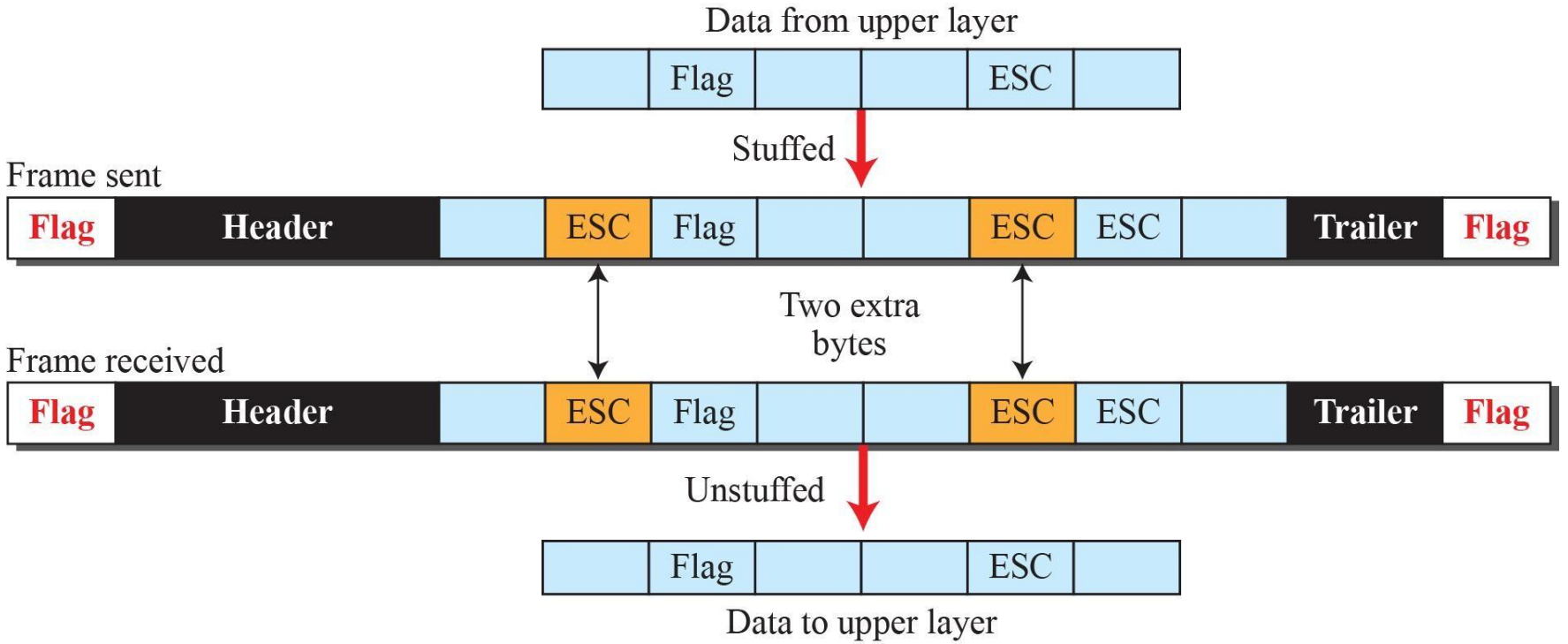
Physical layer moves bits as signals from source to destination, ensuring synchronized bit durations. Data-link layer organizes bits into frames, like putting a letter in an envelope. Frames have sender and receiver addresses, separating messages. It's better to use smaller frames to avoid retransmitting a whole message for a single-bit error.

Frame size - Frames can be fixed or variable in size. Fixed frames use size as a delimiter, like ATM WAN's cells. Variable framing, common in local-area networks, requires defining frame boundaries. Historically, character-oriented and bit-oriented approaches were used for this purpose.

Character-Oriented Framing

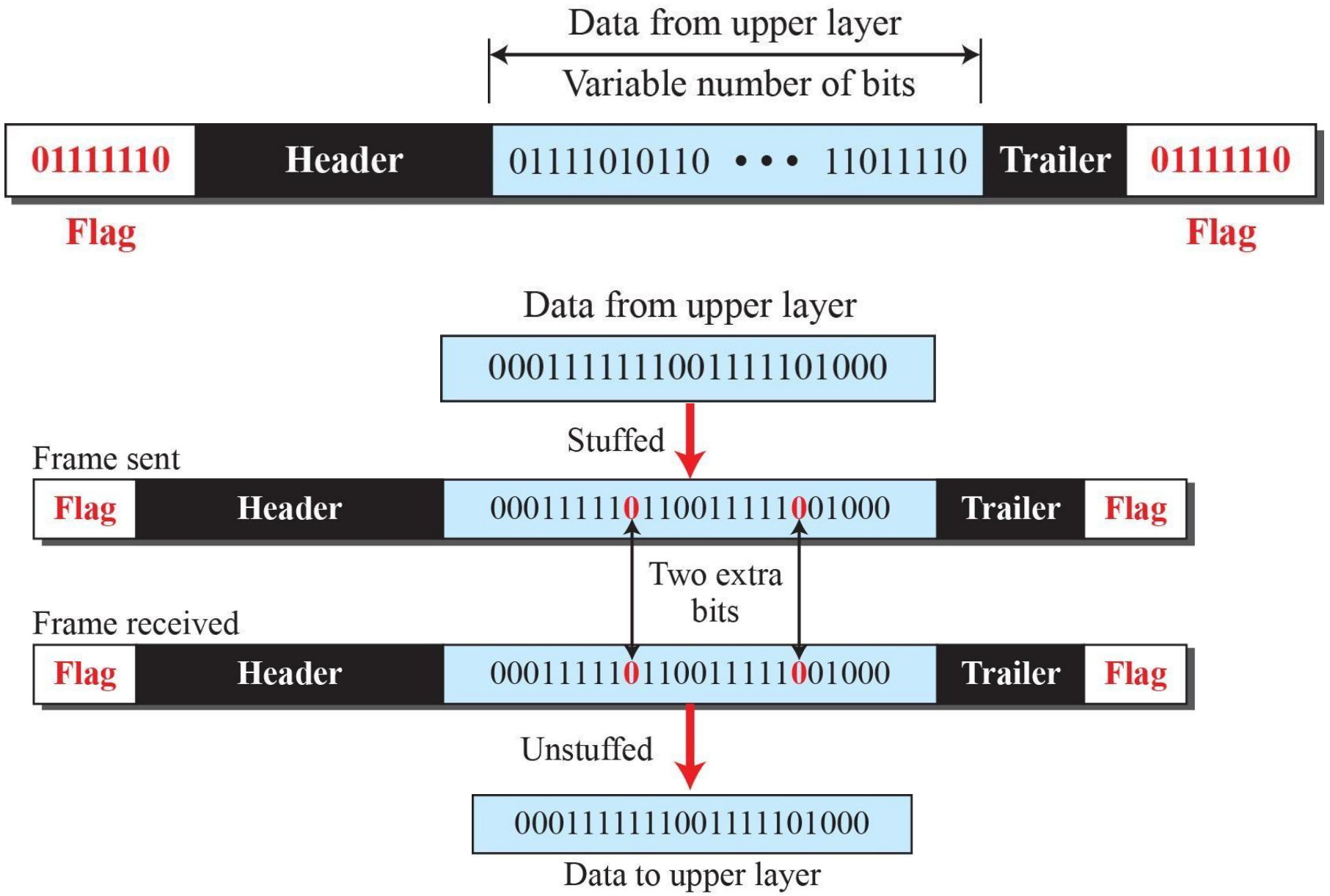
character-oriented framing involves sending data in 8-bit characters, like those in ASCII. Frames start and end with an 8-bit flag. To avoid confusion, an escape character is used when the flag pattern appears in the data. However, this creates a new issue, so escape characters in the data are marked with another escape character. This method helps prevent errors when dealing with various types of information. Despite its past popularity, character-oriented framing faces challenges with modern coding systems like Unicode, leading to a shift toward bit-oriented protocols.


Data from upper layer
Variable number of characters
Flag
Header
Trailer
Flag


Data from upper layer
Flag
ESC
Stuffed
Frame sent
Flag
Header
ESC
Flag
ESC
ESC
Trailer
Flag
Two extra bytes
Frame received
Flag
Header
ESC
Flag
ESC
ESC
Trailer
Flag
Unstuffed
Flag
ESC
Data to upper layer

Bit-Oriented Framing

so in bit-oriented framing, data is a bunch of bits representing text, graphics, audio, etc. Frames begin and end with a special 8-bit pattern (01111110). But if this pattern appears in the data, we use bit stuffing. If there's a 0 followed by five 1s, we add an extra 0 to avoid confusion. The receiver removes this extra bit. Even if there's a 0 after five 1s, we still add a 0, which the receiver later removes. This prevents the actual flag (01111110) from being mistaken for data. So, bit stuffing ensures smooth communication in bit-oriented protocols.


Data from upper layer
Variable number of bits
01111110
Header
01111010110 • • • 11011110
Trailer
01111110
Flag
Flag
Data from upper layer
00011111110011111101000
Stuffed
Frame sent
Flag
Header
00011111011001111101000
Trailer
Flag
Two extra bits
Frame received
Flag
Header
00011111011001111101000
Trailer
Flag
Unstuffed
00011111110011111101000
Data to upper layer

Transmission Media

Transmission media are actually located below the physical layer and are directly controlled by the physical layer.

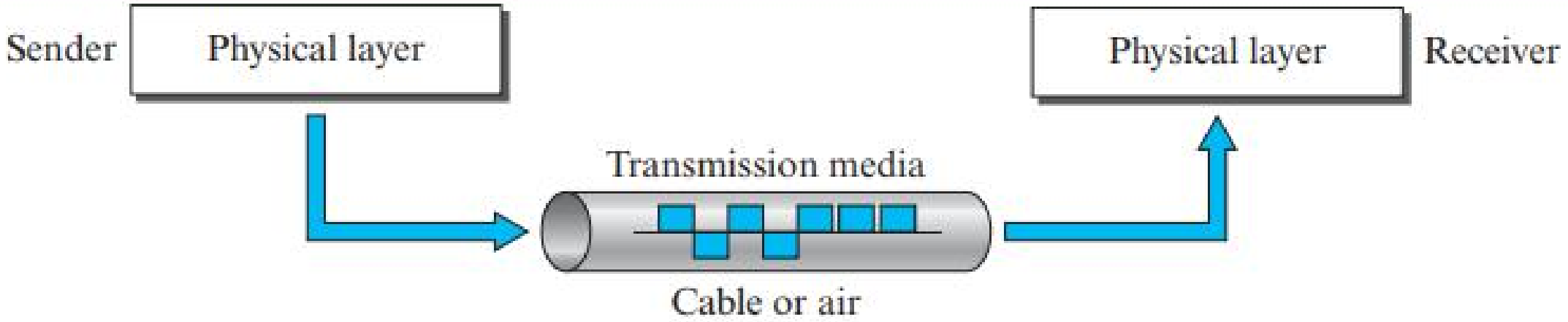
A transmission medium can be broadly defined as anything that can carry information from a source to a destination

In data communications the definition of the information and the transmission medium is more specific. The transmission medium is usually free space, metallic cable, or fiber-optic cable. The information is usually a signal that is the result of a conversion of data from another form.

In telecommunications, transmission media can be divided into two broad categories: guided and unguided. Guided media include twisted-pair cable, coaxial cable, and fiber-optic cable. Unguided medium is free space.

Transmission Media

Figure 7.58 *Transmission media and physical layer*



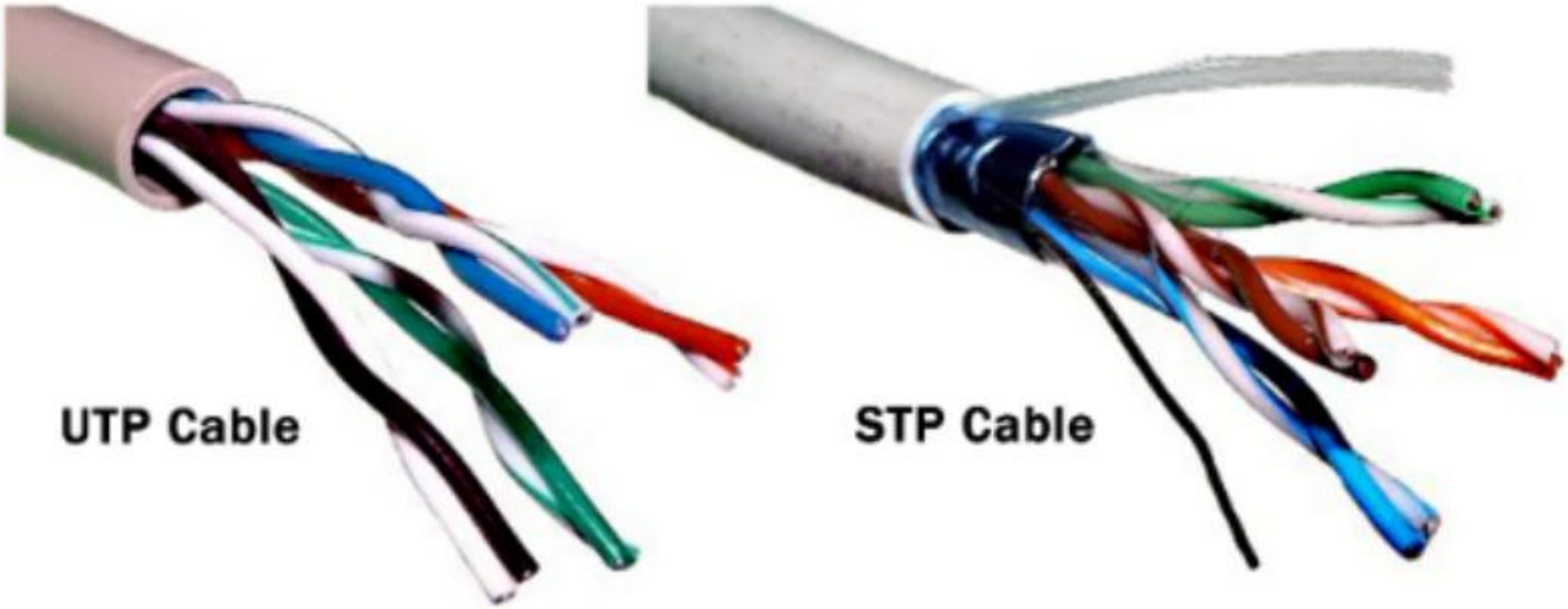
Guided Media

Guided media, which are those that provide a conduit from one device to another, include twisted-pair cable, coaxial cable, and fiber-optic cable.
A signal traveling along any of these media is directed and contained by the physical limits of the medium.
Twisted-pair and coaxial cable use metallic (copper) conductors that accept and transport signals in the form of electric current.
Fiber-optic cable is a cable that accepts and transports signals in the form of light.

Twisted-Pair Cable

A twisted pair consists of two conductors (normally copper), each with its own plastic insulation, twisted together.
One of the wires is used to carry signals to the receiver, and the other is used only as a ground reference. The receiver uses the difference between the two.
In addition to the signal from the sender, interference (noise) and crosstalk may affect both wires and create unwanted signals.
If the two wires are parallel, the effect of these unwanted signals is not the same in both wires because they are at different locations relative to the noise or crosstalk sources. This results in a difference at the receiver. By twisting the pairs, a balance is maintained.
The most common twisted-pair cable used in communications is referred to as unshielded twisted-pair (UTP). IBM has also produced a version of twisted-pair cable for its use called shielded twisted-pair (STP).

Twisted-Pair Cable



Twisted Pair Cables

Performance

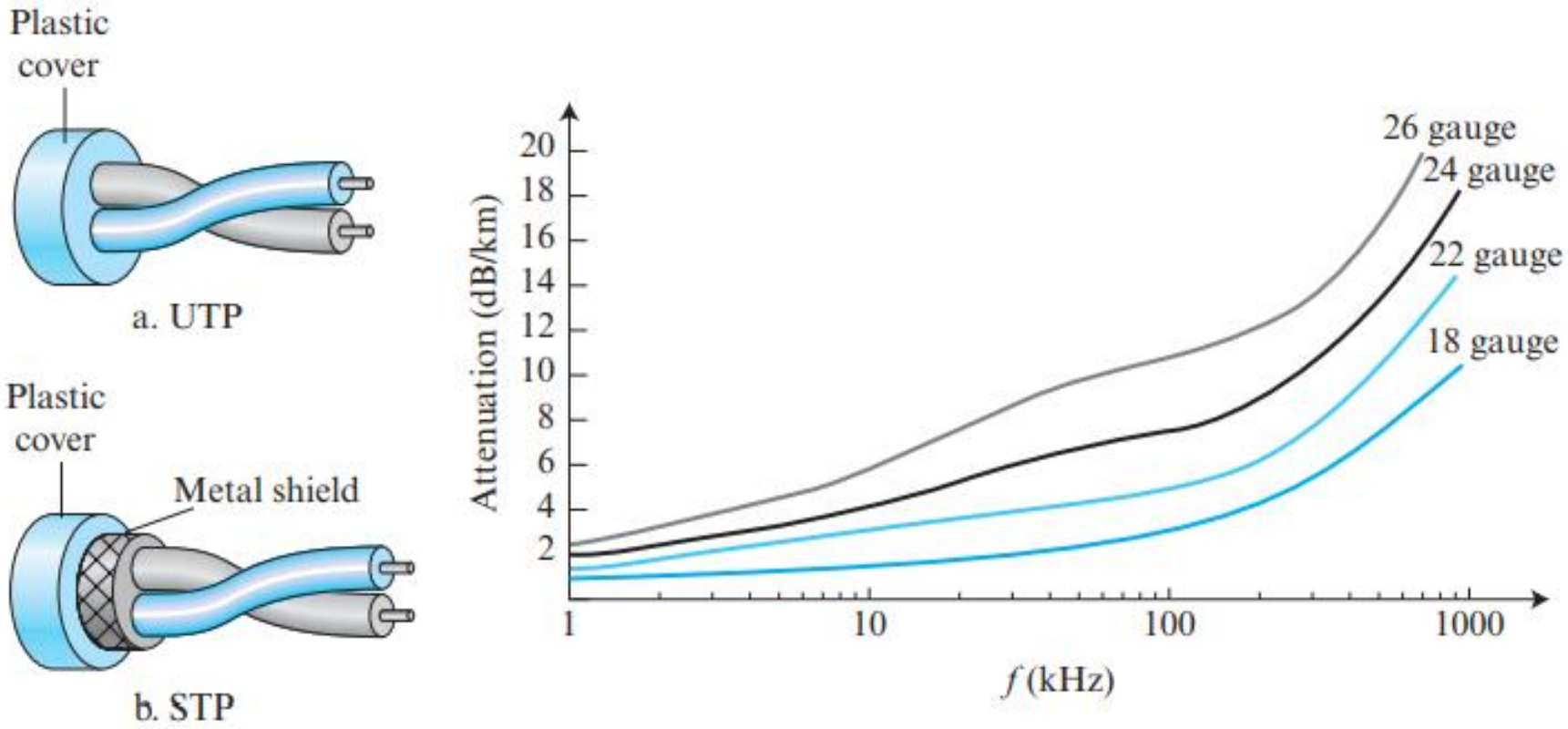
One way to measure the performance of twisted-pair cable is to compare attenuation versus frequency and distance. A twisted-pair cable can pass a wide range of frequencies.
However, Figure 7.59 also shows that with increasing frequency, the attenuation, measured in decibels per kilometer (dB/km), sharply increases with frequencies above 100 kHz. Note that gauge is a measure of the thickness of the wire (inversely).

Applications

Twisted-pair cables are used in telephone lines to provide voice and data channels. The local loop—the line that connects subscribers to the central telephone office—commonly consists of unshielded twisted-pair cables.
The DSL lines that are used by the telephone companies to provide high-data-rate connections also use the high-bandwidth capability of unshielded twisted-pair cables

Twisted Pair Cables

Figure 7.59 *Twisted-pair cable*



Coaxial Cable

Coaxial cable (or coax) carries signals of higher frequency ranges than those in twistedpair cable, in part because the two media are constructed quite differently. Instead of having two wires, coax has a central core conductor of solid or stranded wire (usually copper) enclosed in an insulating sheath, which is, in turn, encased in an outer conductor of metal foil, braid, or a combination of the two.

The outer metallic wrapping serves both as a shield against noise and as the second conductor, which completes the circuit. This outer conductor is also enclosed in an insulating sheath, and the whole cable is protected by a plastic cover

Coaxial Cable

Performance

As we did with twisted-pair cables, we can measure the performance of a coaxial cable. We notice in Figure 7.60 that the attenuation is much higher in coaxial cable than in twisted-pair cable.

In other words, although coaxial cable has a much higher bandwidth, the signal weakens rapidly and requires the frequent use of repeaters.

Applications

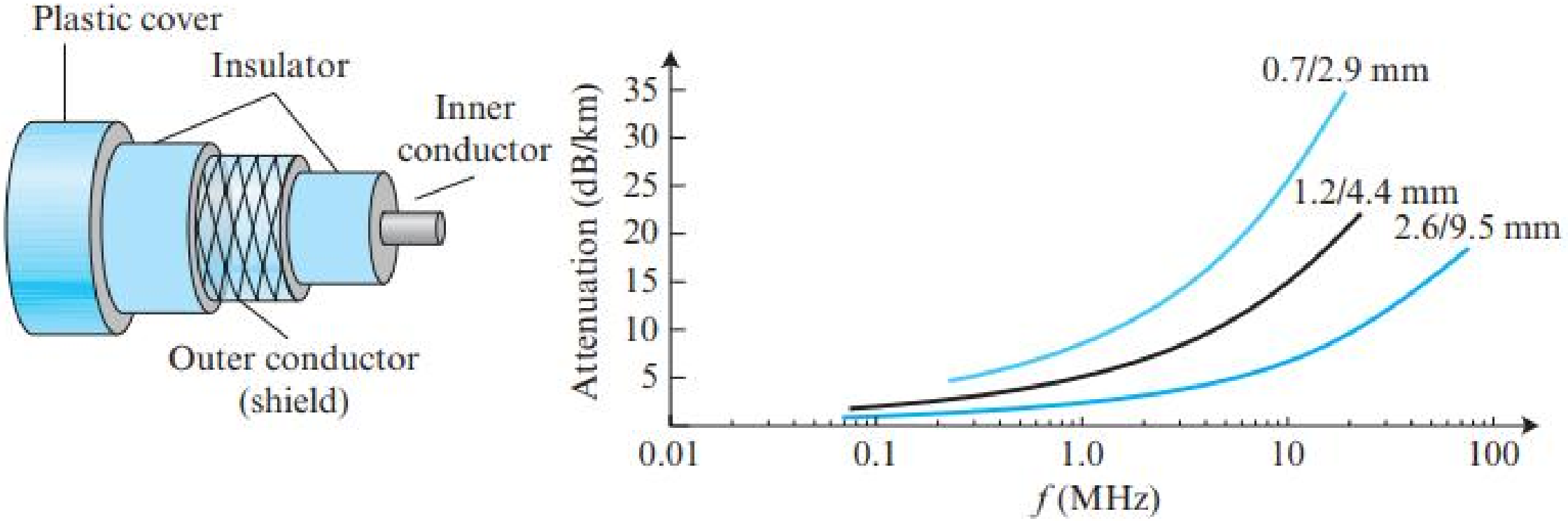
Coaxial cable was widely used in analog telephone networks where a single coaxial network could carry 10,000 voice signals.

Cable TV networks also use coaxial cable. In the traditional cable TV network, the entire network used coaxial cable. Later, however, cable TV providers replaced most of the media with fiber-optic cable.

Another common application of coaxial cable is in traditional Ethernet LANs. Because of its high bandwidth, and consequently high data rate, coaxial cable was chosen for digital transmission in early Ethernet LANs. Thick Ethernet has specialized connectors.

Coaxial Cable

Figure 7.60 *Coaxial cable*



Fiber-Optic Cable

A fiber-optic cable is made of glass or plastic and transmits signals in the form of light. To understand optical fiber, we first need to explore several aspects of the nature of light.

Light travels in a straight line as long as it is moving through a single uniform substance. If a ray of light traveling through one substance suddenly enters another substance (of a different density), the ray changes direction.

Optical fibers use reflection to guide light through a channel. A glass or plastic core is surrounded by a cladding of less dense glass or plastic. The difference in density of the two materials must be such that a beam of light moving through the core is reflected off the cladding instead of being refracted into it.

Fiber-Optic Cable

Figure 7.61 *Bending of light ray*

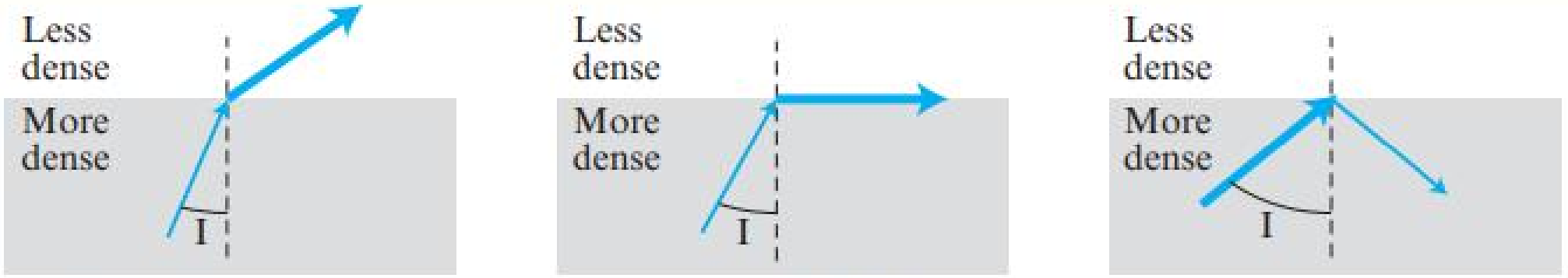
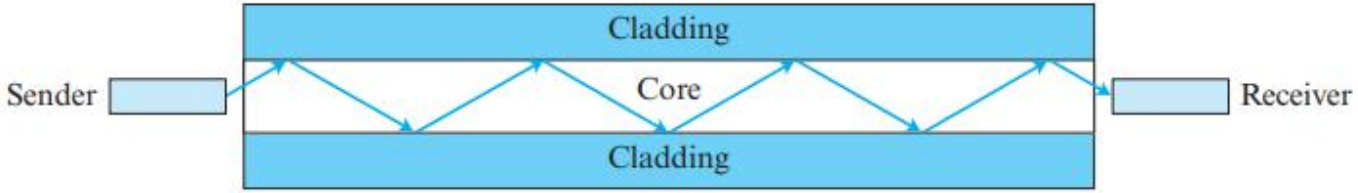


Figure 7.62 *Optical fiber*



Propagation Modes

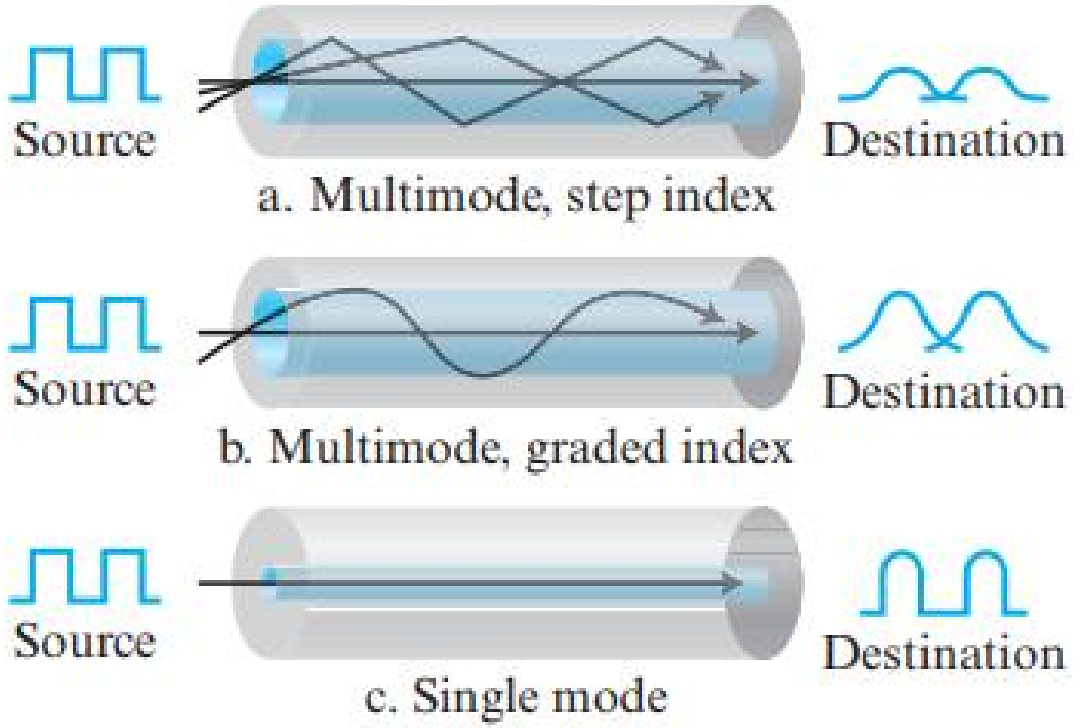
Current technology supports two modes (multimode and single mode) for propagating light along optical channels, each requiring fiber with different physical characteristics.

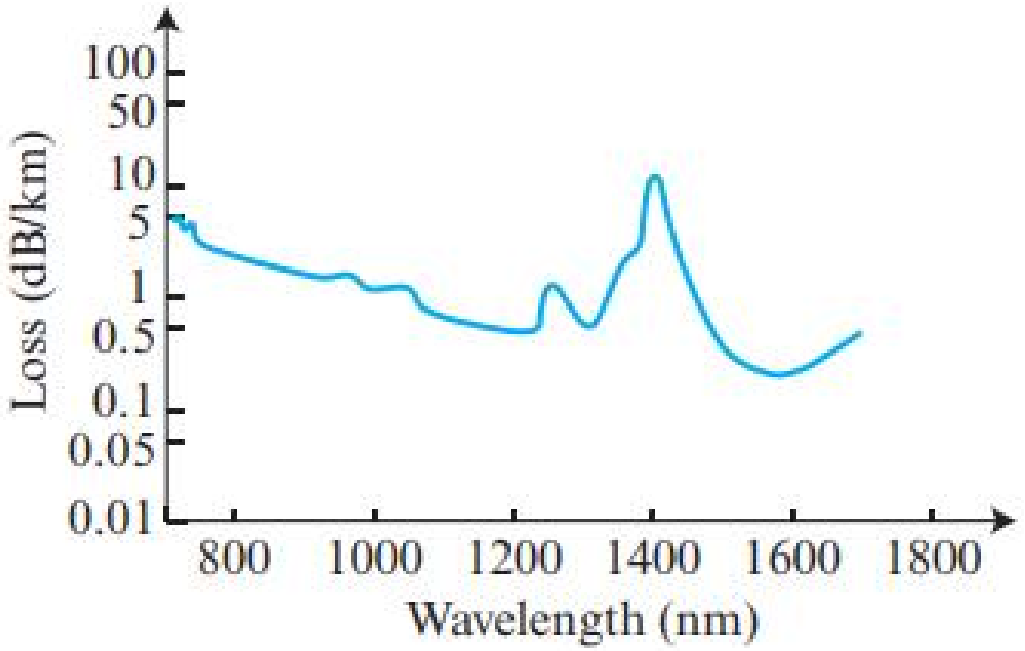
Multimode is so named because multiple beams from a light source move through the core in different paths. How these beams move within the cable depends on the structure of the core.

Multimode can be implemented in two forms: step-index or graded-index.

Propagation Modes

Figure 7.63 *Modes*





Propagation Modes

In multimode step-index fiber, the density of the core remains constant from the center to the edges. A beam of light moves through this constant density in a straight line until it reaches the interface of the core and the cladding. At the interface, there is an abrupt change due to a lower density; this alters the angle of the beam's motion. The term step index refers to the suddenness of this change, which contributes to the distortion of the signal as it passes through the fiber.

A second type of fiber, called multimode graded-index fiber, decreases this distortion of the signal through the cable. The word index here refers to the index of refraction. As we saw above, the index of refraction is related to density. A graded-index fiber, therefore, is one with varying densities. Density is highest at the center of the core and decreases gradually to its lowest at the edge.

Propagation Modes

Single-mode uses step-index fiber and a highly focused source of light that limits beams to a small range of angles, all close to the horizontal. The single-mode fiber itself is manufactured with a much smaller diameter than that of multimode fiber, and with substantially lower density (index of refraction). The decrease in density results in a critical angle that is close enough to 90° to make the propagation of beams almost horizontal. In this case, propagation of different beams is almost identical, and delays are negligible. All the beams arrive at the destination “together” and can be recombined with little distortion to the signal.

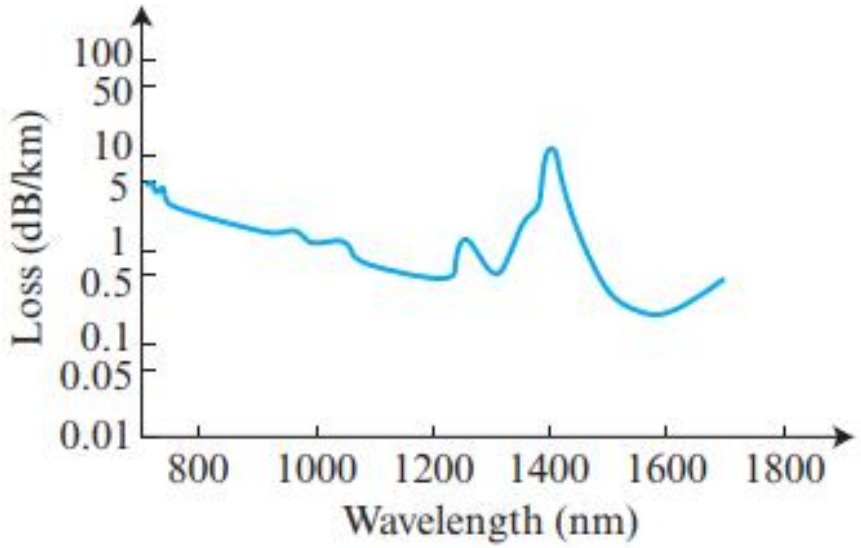
Fibel-Optic Cables

Performance

The plot of attenuation versus wavelength in Figure 7.63 also shows a very interesting phenomenon in fiber-optic cable. Attenuation is flatter than in the case of twisted-pair cable and coaxial cable. The performance is such that we need fewer (actually 10 times fewer) repeaters when we use fiber-optic cable.

Applications

Fiber-optic cable is often found in backbone networks because its wide bandwidth is cost-effective. Today, with wavelength-division multiplexing (WDM), we can transfer data at a rate of 1600 Gbps. Some cable TV companies use a combination of optical fiber and coaxial cable, thus creating a hybrid network.



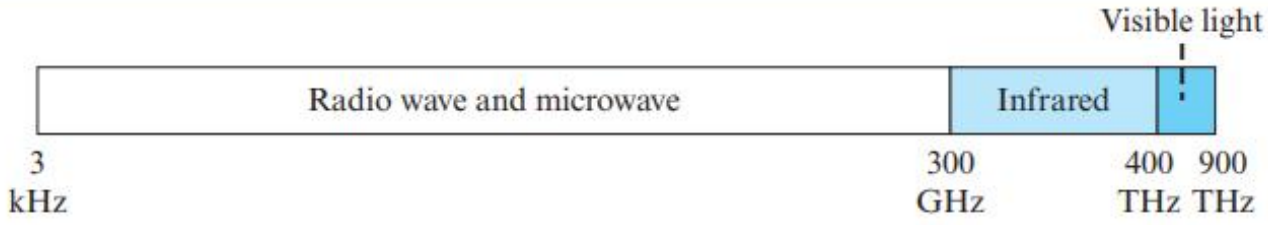
Unguided Media: Wireless

Unguided media transport electromagnetic waves without using a physical conductor.

This type of communication is often referred to as wireless communication.

Signals are normally broadcast through free space and thus are available to anyone who has a device capable of receiving them.

Figure 7.64 *Electromagnetic spectrum for wireless communication*



Unguided Media

Unguided signals can travel from source to destination in several ways: ground propagation, sky propagation, and line-of-sight propagation.

In ground propagation, radio waves travel through the lowest portion of the atmosphere, hugging the earth. These low-frequency signals emanate in all directions from the transmitting antenna and follow the curvature of the planet. Distance depends on the amount of power in the signal: The greater the power, the greater the distance.

In sky propagation, higher-frequency radio waves radiate upward into the ionosphere (the layer of atmosphere where particles exist as ions) where they are reflected back to earth. This type of transmission allows for greater distances with lower output power.

Unguided Media

In line-of-sight propagation, very high-frequency signals are transmitted in straight lines directly from antenna to antenna. Antennas must be directional, facing each other, and either tall enough or close enough together not to be affected by the curvature of the earth. Line-of-sight propagation is tricky because radio transmissions cannot be completely focused.

The section of the electromagnetic spectrum defined as radio waves and microwaves is divided into eight ranges, called bands, each regulated by government authorities. These bands are rated from very low frequency (VLF) to extremely high frequency (EHF).

We can divide wireless transmission into three broad groups: radio waves, microwaves, and infrared waves.

Unguided Media

Table 7.1 *Bands*

Band	*Range*	*Propagation*	*Application*
VLF (very low frequency)	3–30 kHz	Ground	Long-range radio
LF (low frequency)	30–300 kHz	Ground	Radio beacons
MF (middle frequency)	300 kHz–3 MHz	Sky	AM radio
HF (high frequency)	3–30 MHz	Sky	Citizens band (CB), ship/aircraft communication
VHF (very high frequency)	30–300 MHz	Sky and line-of-sight	VHF TV, FM radio
UHF (ultrahigh frequency)	300 MHz–3 GHz	Line-of-sight	UHF TV, cellular phones, paging, satellite
SHF (superhigh frequency)	3–30 GHz	Line-of-sight	Satellite communication
EHF (extremely high frequency)	30–300 GHz	Line-of-sight	Radar, satellite

Radio Waves

Electromagnetic waves ranging in frequencies between 3 kHz and 1 GHz are normally called radio waves; waves ranging in frequencies between 1 and 300 GHz are called microwaves. However, the behavior of the waves, rather than the frequencies, is a better criterion for classification.

Radio waves, for the most part, are omnidirectional. When an antenna transmits radio waves, they are propagated in all directions. This means that the sending and receiving antennas do not have to be aligned.

The omnidirectional property has a disadvantage, too. The radio waves transmitted by one antenna are susceptible to interference by another antenna that may send signals using the same frequency or band.

Radio waves, particularly those waves that propagate in the sky mode, can travel long distances. This makes radio waves a good candidate for long-distance broadcasting such as AM radio.

Radio Waves

Radio waves, particularly those of low and medium frequencies, can penetrate walls. This characteristic can be both an advantage and a disadvantage. It is an advantage because, for example, an AM radio can receive signals inside a building. It is a disadvantage because we cannot isolate a communication to just inside or outside a building.
The radio wave band is relatively narrow, just under 1 GHz, compared to the microwave band. When this band is divided into subbands, the subbands are also narrow, leading to a low data rate for digital communications.
Almost the entire band is regulated by authorities. Using any part of the band requires permission from the authorities.

Microwaves

Electromagnetic waves having frequencies between 1 and 300 GHz are called microwaves.

Microwaves are unidirectional. When an antenna transmits microwaves, they can be narrowly focused. This means that the sending and receiving antennas need to be aligned.

The unidirectional property has an obvious advantage. A pair of antennas can be aligned without interfering with another pair of aligned antennas.

Microwaves

The following describes some characteristics of microwave propagation:

Microwave propagation is line-of-sight. Since the towers with the mounted antennas need to be in direct sight of each other, towers that are far apart need to be very tall. The curvature of the earth as well as other blocking obstacles do not allow two short towers to communicate by using microwaves. Repeaters are often needed for longdistance communication.

Very high-frequency microwaves cannot penetrate walls. This characteristic can be a disadvantage if receivers are inside buildings.

The microwave band is relatively wide, almost 299 GHz. Therefore wider subbands can be assigned, and a high data rate is possible.

Use of certain portions of the band requires permission from authorities.

Applications

Microwaves, due to their unidirectional properties, are very useful when unicast (oneto-one) communication is needed between the sender and the receiver. They are used in cellular phones, satellite networks, and wireless LANs.

Microwaves

Infrared Waves

Infrared waves, with frequencies from 300 GHz to 400 THz (wavelengths from 1 mm to 770 nm), can be used for short-range communication.

Infrared waves, having high frequencies, cannot penetrate walls. This advantageous characteristic prevents interference between one system and another; a short-range communication system in one room cannot be affected by another system in the next room.

When we use our infrared remote control, we do not interfere with the use of the remote by our neighbors. However, this same characteristic makes infrared signals useless for long-range communication.

In addition, we cannot use infrared waves outside a building because the sun's rays contain infrared waves that can interfere with the communication.

Flow Control

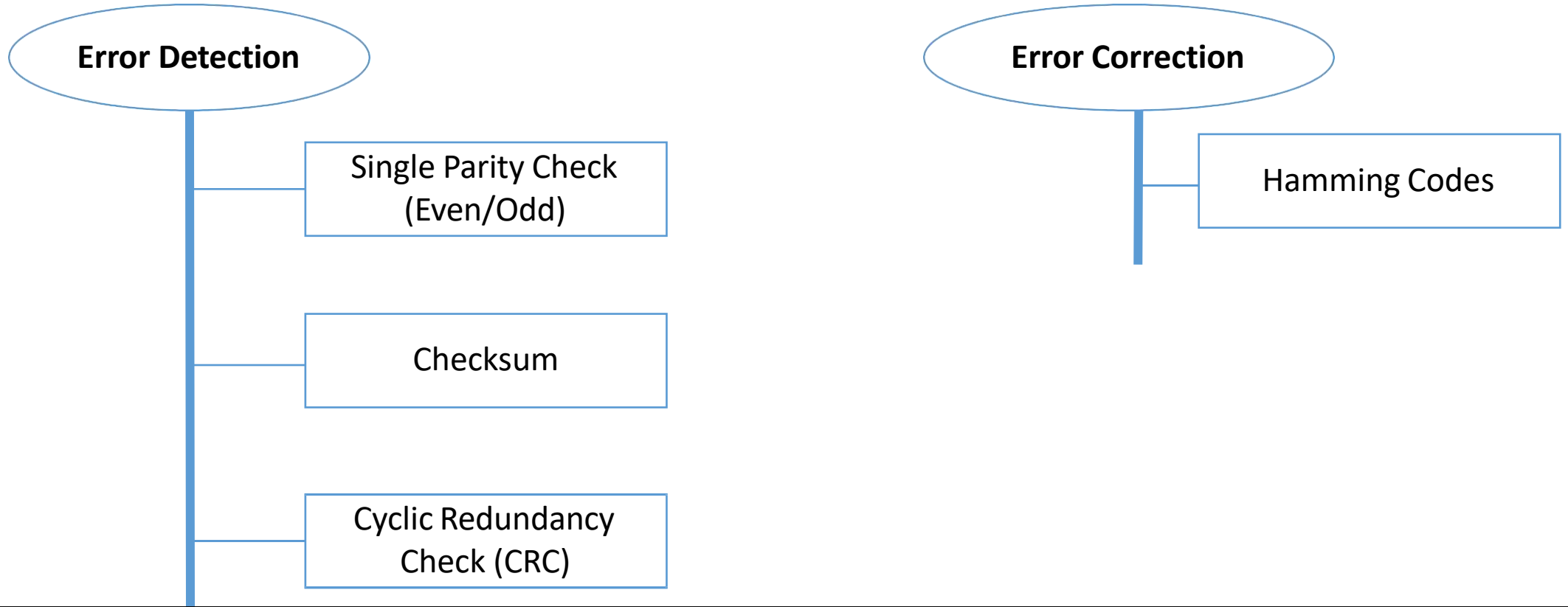
As we discussed in the transport layer (Chapter 3), flow control coordinates the amount of data that can be sent before receiving an acknowledgment. In the data-link layer, flow control is one of the duties of the data link control sublayer. The idea of flow control in the data-link layer follows the same principle we discuss for the transport layer. Although at the transport layer flow control is end-to-end (host-to-host), flow control at the data-link layer is node-to-node, across the link.

Error Control

Error control in **the data-link layer involves both detecting and correcting errors.** It lets the receiver tell the sender about lost or damaged frames, leading to their retransmission. **Unlike the end-to-end error control in the transport layer, data-link layer error control is node-to-node, ensuring frames are not corrupted as they pass through each link. In simpler terms, it's about making sure data gets across the link without hiccups.**

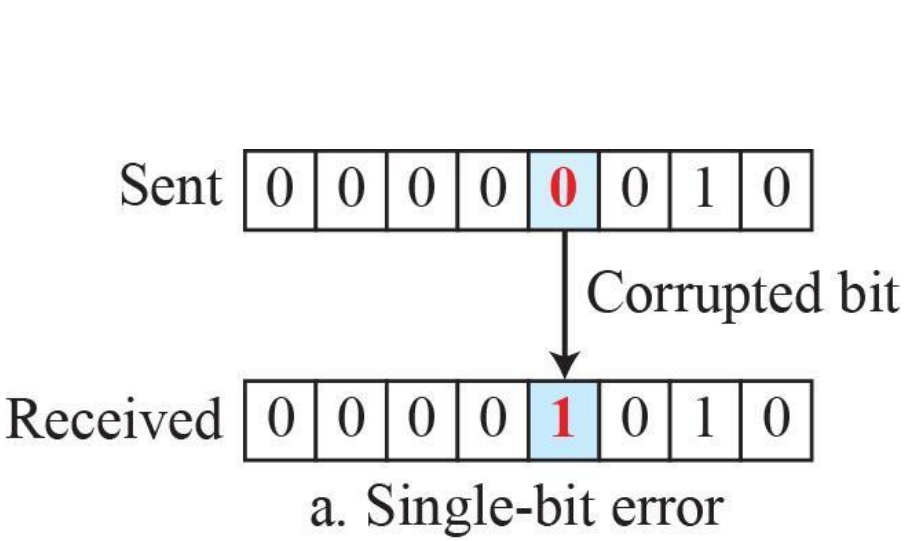
Error Detection and Correction

At the data-link layer, if a frame is corrupted between the two nodes, it needs to be corrected before it continues its journey to other nodes. However, **most link-layer protocols simply discard the frame and let the upper-layer protocols handle the retransmission of the frame.** Some wireless protocols, however, try to correct the corrupted frame.

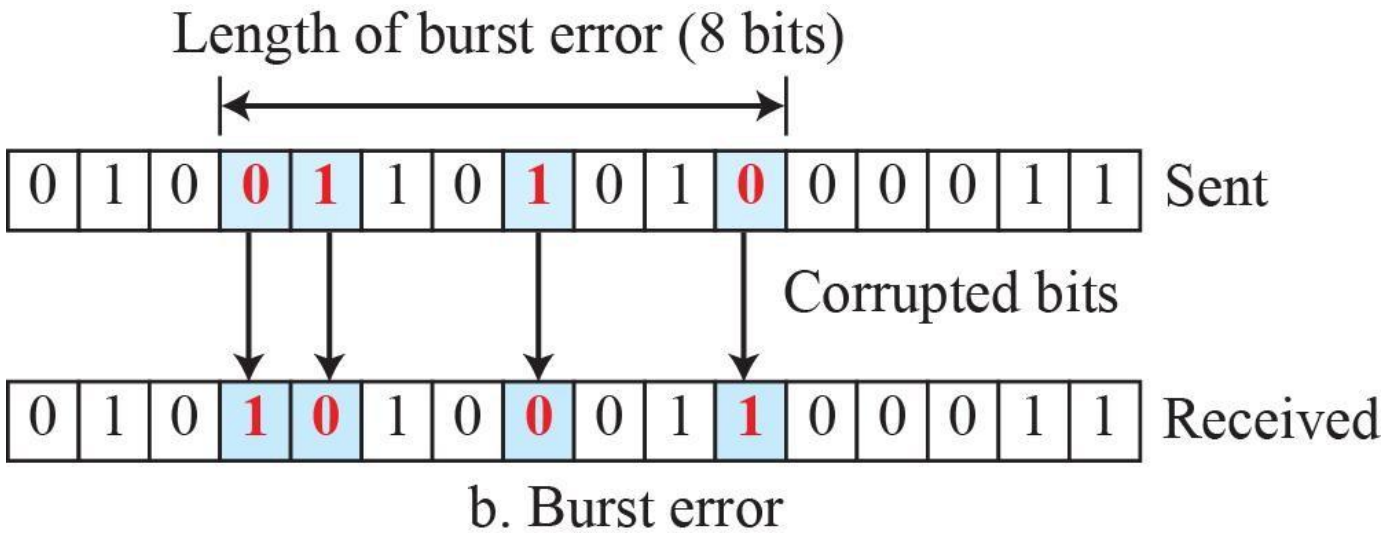


Types of Errors

When bits travel, interference can cause unpredictable changes in the signal. A **single-bit error** means one bit flips from 0 to 1 or vice versa. On the other hand, a burst error involves two or more bits changing. **Burst errors** are more likely because the duration of noise is usually longer than one bit, affecting a group of bits. The number of affected bits depends on the data rate and noise duration—higher rates mean more affected bits. In a nutshell, single-bit errors are isolated flips, while burst errors involve consecutive bit changes due to longer interference.



a. Single-bit error



b. Burst error

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

Fixing errors is tougher than just spotting them. Error detection gives a simple yes or no, without caring about the number of corrupted bits. In error correction, we must know how many bits are wrong and where. The more errors and the larger the message, the harder it gets. Correcting one error in an 8-bit unit means considering eight possible locations, and for two errors, it's 28 possibilities. Imagine finding 10 errors in a 1000-bit unit—it's quite a challenge. So, while we focus on detecting errors, correcting them is a trickier task.

Coding

To add redundancy for error detection, coding schemes use various methods. **The sender includes extra bits related to the actual data bits. The receiver then checks these relationships to spot errors. The coding scheme's effectiveness depends on the ratio of redundant bits to data bits** and the process's robustness.

There are two main types:

1. Block coding
2. Convolution coding

We focus on block coding in this book, as convolution coding is more complex and not covered here.

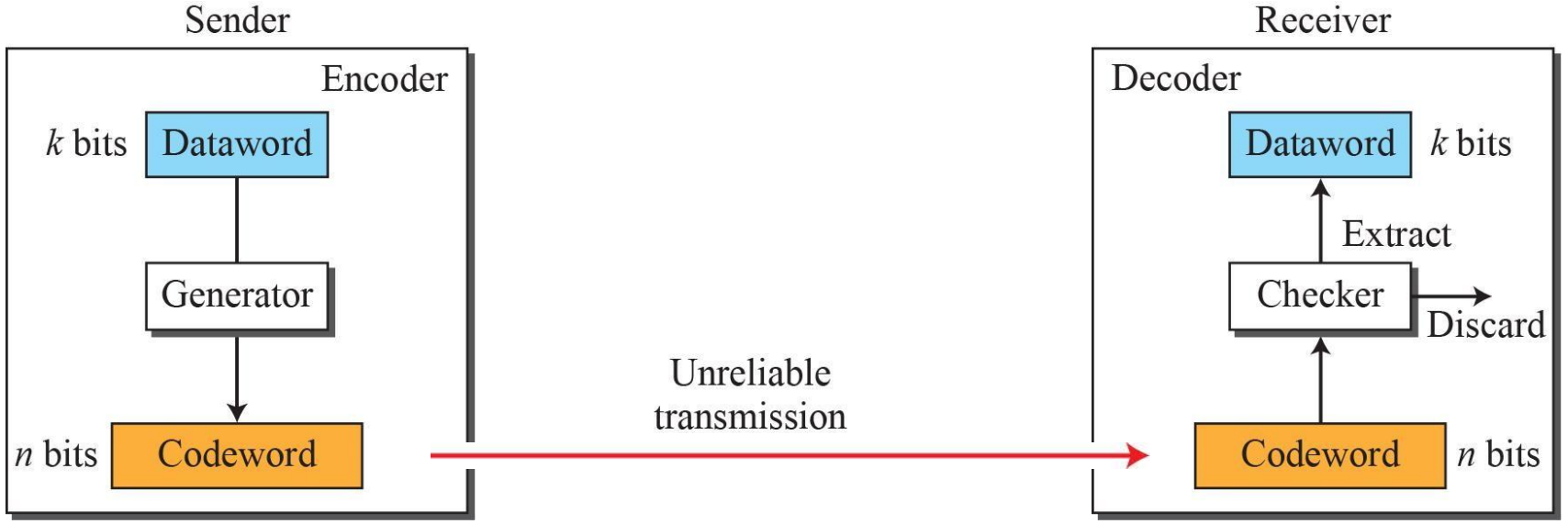
Block Coding

Block coding involves dividing a message into k-bit blocks (datawords) and adding r redundant bits to create n-bit blocks (codewords). The extra bits' selection method will be explained later. There are 2k possible datawords and 2n possible codewords, where n > k. Each dataword corresponds to a unique codeword, resulting in 2n - 2k unused codewords known as invalid or illegal codes. Detecting errors relies on spotting these invalid codes; if the receiver gets one, it means the data got corrupted during transmission.

Error Detection

How can errors be detected by using block coding? 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.



Hamming Distance

In coding, Hamming distance measures how different two sets of bits are. It's crucial for finding errors during data transmission. The distance is the number of bit differences between sent and received data. For instance, if 00000 is sent and 01101 is received, the Hamming distance is 3, meaning 3 bits got messed up. We use XOR and count the 1s in the result to find Hamming distance—basically, how many bits went wrong during communication.

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

The sender uses a generator to turn datawords into codewords through encoding rules. During transmission, codewords may change. If the received codeword matches a valid one, it's accepted, and the corresponding dataword is used. Invalid codewords are discarded. However, if corruption happens but the received word still matches a valid codeword, the error goes undetected.

Example 5.1

Let us assume that $k = 2$ and $n = 3$. Table 5.1 shows the list of datawords and codewords. Later, we will see how to derive a codeword from a dataword.

Table 5.1 *A code for error detection in Example 5.1*

Datawords	*Codewords*	*Datawords*	*Codewords*
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.

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

Example

Let us find the Hamming distance between two pairs of words.

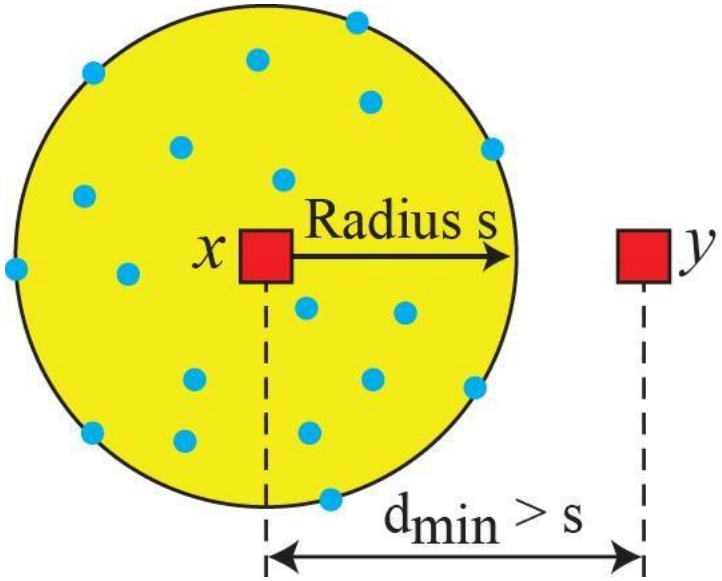
1. The Hamming distance d(000, 011) is 2 because (000 $\oplus$ 011) is 011 (two 1s).
2. The Hamming distance d(10101, 11110) is 3 because (10101 $\oplus$ 11110) is 01011 (three 1s).

Minimum Hamming Distance for Error Detection

In a group of code words, the minimum Hamming distance is the smallest difference between any two codes. To detect up to s errors, the minimum distance between valid codes must be (s + 1). This way, if s errors happen during transmission, the received code won't match any valid code. Imagine the sent code as the center of a circle, and errors create points inside or on the circle's edge. All valid codes must be outside the circle. So, dmin (minimum distance) has to be an integer greater than s, or simply dmin = s + 1, to ensure errors are detected.

To guarantee the detection of up to s errors in all cases, the minimum

Hamming distance in a block code must be d_min = s + 1.



Legend

- Any valid codeword
- Any corrupted codeword with 1 to s errors

Example

The minimum Hamming distance for our first code scheme (Table page 23) is 2. This code guarantees detection of only a single error. For example, 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.

Example

A code scheme has a Hamming distance dmin = 4. This code guarantees the detection of up to three errors (d = s + 1 or s= 3).

Linear Block Codes

Most block codes used today are linear block codes. Nonlinear block codes are less common because they're tricky to analyze and implement. Linear block codes, which we focus on, follow a rule: when you XOR (add modulo-2) two valid codes, you get another valid code. The technical definition involves abstract algebra, but for now, just know that linear block codes play nice with XOR.

Example- The code in Table (page 23) 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.

Minimum Distance for Linear Block Codes

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

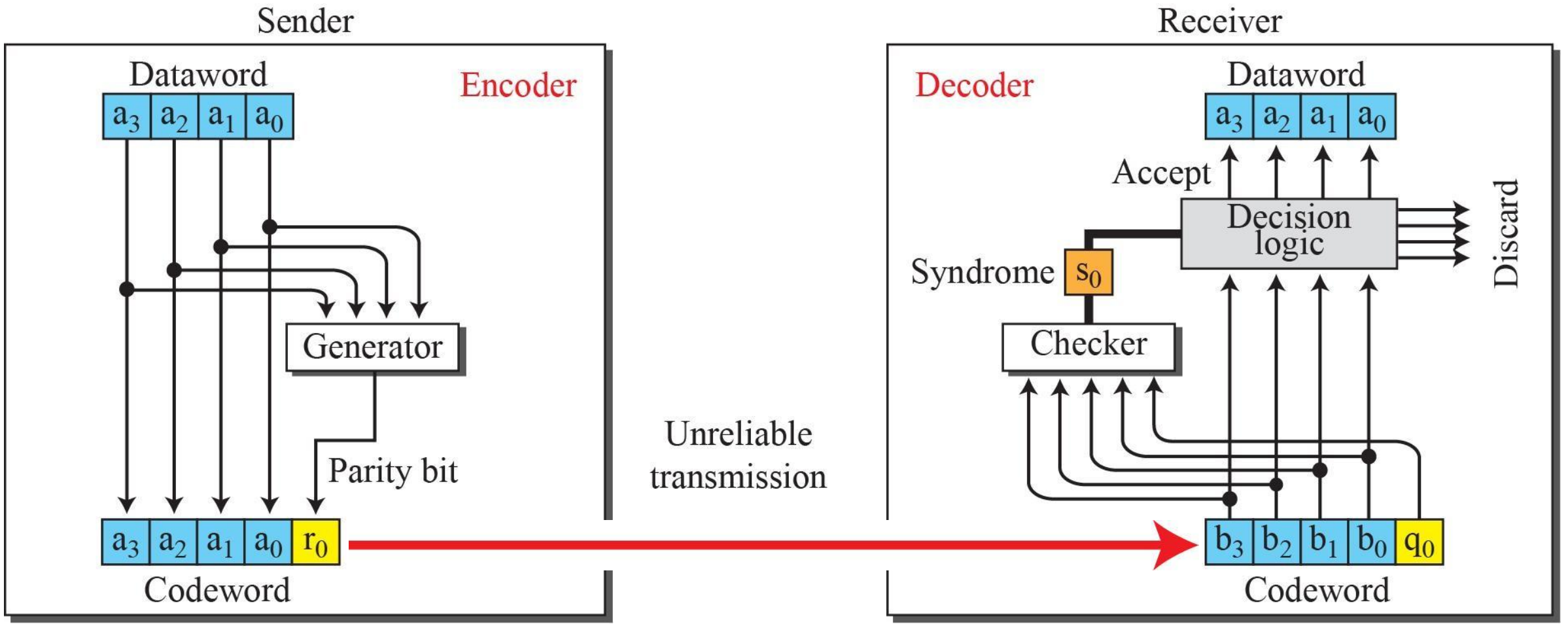
Example- In our first code (Table page 23), the numbers of 1s in the nonzero codewords are 2, 2, and 2. So the minimum Hamming distance is dmin = 2.

Parity-Check Code

A well-known error-detecting code is the parity-check code. It's a linear block code that adds a parity bit to a k-bit dataword, making it n bits (n = k + 1). The parity bit is chosen to ensure an even total of 1s in the codeword. This makes the minimum Hamming distance dmin = 2, meaning it can detect single-bit errors. For example, a parity-check code with k = 2 and n = 3 is in Table (page 23) and another with k = 4 and n = 5 is in Table (next page).

Simple parity-check code (5,4)

Datawords	**Codewords**	*Datawords*	**Codewords**
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**


Sender
Receiver
Dataword
Encoder
Decoder
Dataword
Accept
Decision logic
Discard
Syndrome
Generator
Checker
Parity bit
Unreliable transmission
Codeword
Codeword

The encoder uses a generator to add a parity bit (r0) to a 4-bit dataword. The trick is to make the total number of 1s in the 5-bit codeword even. This is done by adding the data bits (modulo-2), making r0 = a3 + a2 + a1 + a0 (modulo-2). If the sum is even, r0 is 0; if odd, r0 is 1.

During transmission, the codeword might get corrupted. The receiver uses a checker to create a syndrome by adding all 5 bits (modulo-2). The syndrome is 0 if the total 1s are even; otherwise, it's 1 (s0 = b3 + b2 + b1 + b0 + q0, modulo-2). The syndrome goes to a decision logic analyzer: if it's 0, no detectable error, and the data part is accepted; if it's 1, the data part is ignored because there might be an error.

Example

Let us look at some transmission scenarios. Assume the sender sends the datawo 1011. The codeword created from this dataword is 10111, which is sent to th 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 10**0**11. The syndrome is 1. No dataword is created.
3. One single-bit error changes r_0. The received codeword is 1011**0**. 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 **0**011**0**. 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 **010**11. 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.

Limitation-

This technique can not detect an even number of bit errors (two, four, six and so on).
If even number of bits flip during transmission, then receiver can not catch the error.

Example:
Consider the data unit to be transmitted is 10010001 and even parity is used.
Then, code word transmitted to the receiver = 100100011
Consider during transmission, code word modifies as 10**1**100**1**11. (2 bits flip)
On receiving the modified code word, receiver finds the number of 1's is even and even parity is used.
So, receiver assumes that no error occurred in the data during transmission though the data is corrupted.

Cyclic Codes

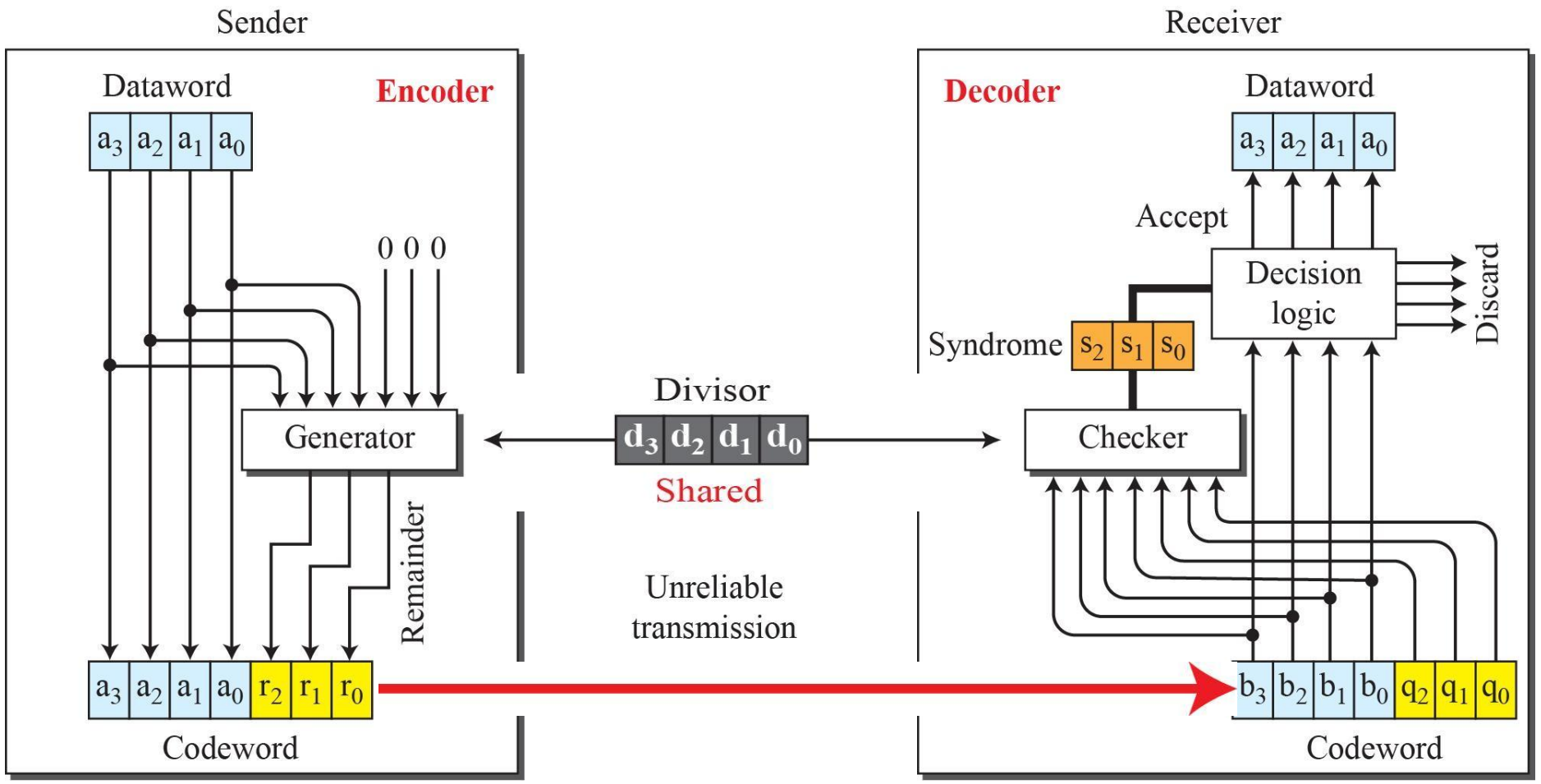
Cyclic codes are special linear block codes that have a neat trick: if you shift a codeword in a circular way (like a rotation), you get another valid codeword. For instance, if 1011000 is a codeword, left-shifting it gives 0110001, which is also a codeword. This shifting works like this: b1 = a0, b2 = a1, and so on, until b0 = a6. The last bit of the first word wraps around to become the first bit of the second word. It's like a circular dance of bits!

Cyclic Redundancy Check

Cyclic codes, like CRC (cyclic redundancy check), are used to correct errors in networks. Here's a simple example: In the encoder, if the dataword has 4 bits, the codeword has 7 bits. We add 3 zeros to the dataword's right side, making it 7 bits. This 7-bit result goes through a generator that uses a predefined divisor of size 4. The generator does modulo-2 division, and the remainder (r2r1r0) is added to the dataword to create the codeword.

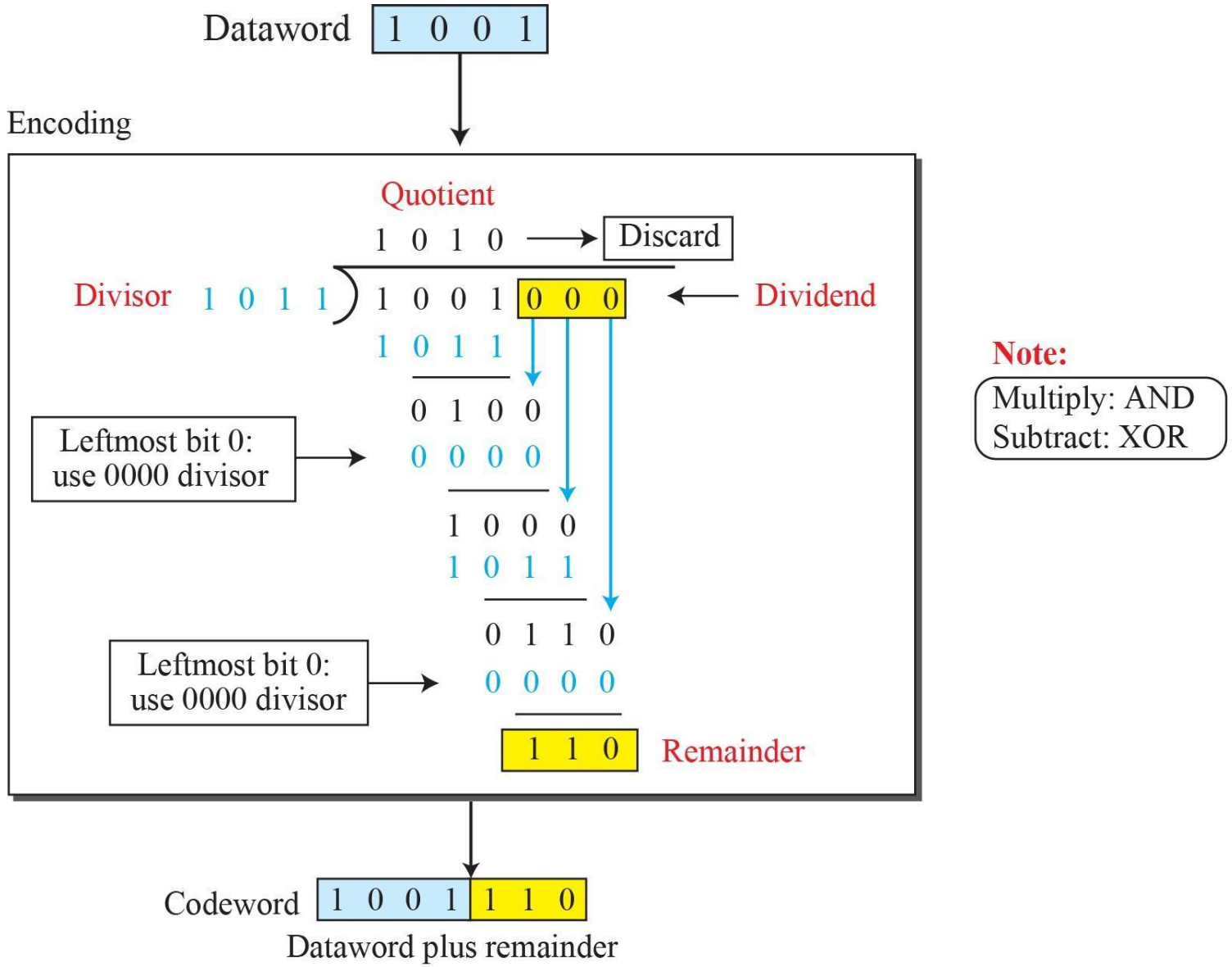
The decoder gets the possibly corrupted codeword and feeds a copy of all 7 bits to the checker, a replica of the generator. The checker's remainder, a 3-bit syndrome (n − k), goes to a decision logic analyzer. If all syndrome bits are 0s, the left 4 bits of the codeword are accepted as the dataword (no error); otherwise, they're discarded (error).

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


Sender
Receiver
Dataword
Encoder
a_3 a_2 a_1 a_0
0 0 0
Generator
Remainder
a_3 a_2 a_1 a_0 r_2 r_1 r_0
Codeword
Divisor
d_3 d_2 d_1 d_0
Shared
Unreliable transmission
Decoder
Dataword
a_3 a_2 a_1 a_0
Accept
Decision logic
Discard
Syndrome
s_2 s_1 s_0
Checker
b_3 b_2 b_1 b_0 q_2 q_1 q_0
Codeword

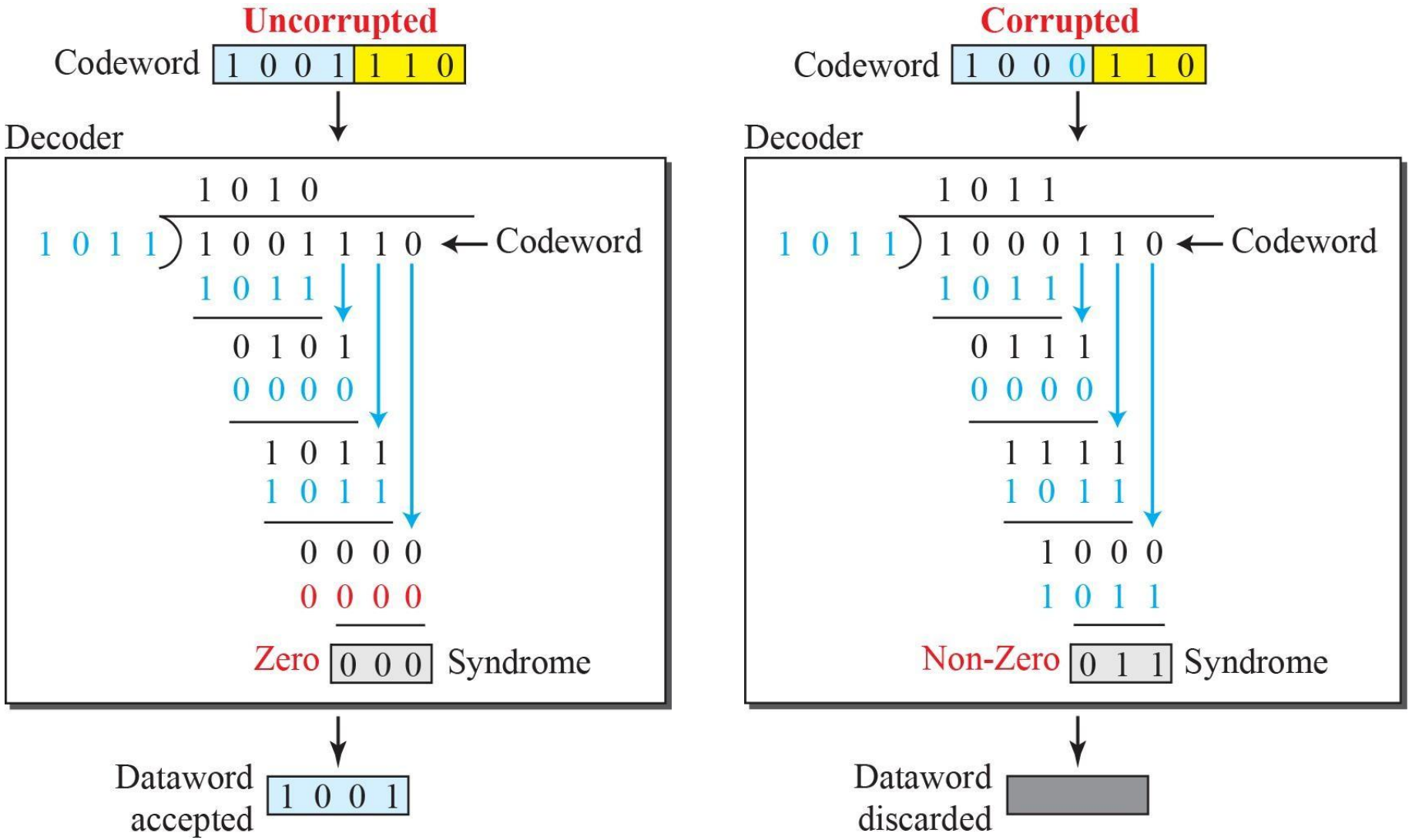
Encoder

The encoder pads the dataword with zeros, divides it by the divisor using XOR for addition and subtraction. The modulo-2 binary division is similar to decimal division. Each step involves XORing the divisor with 4 bits of the dividend, creating a 3-bit remainder. If the leftmost bit is 0, an all-0s divisor is used. Repeat until no bits remain, and the 3-bit remainder becomes check bits (r2, r1, r0) added to the dataword for the codeword.


Dataword
1 0 0 1
Encoding
Quotient
1 0 1 0
Discard
Divisor 1 0 1 1
1 0 0 1 0 0 0
Dividend
1 0 1 1
0 1 0 0
0 0 0 0
Leftmost bit 0:
use 0000 divisor
1 0 0 0
1 0 1 1
0 1 1 0
0 0 0 0
Leftmost bit 0:
use 0000 divisor
1 1 0
Remainder
Codeword
1 0 0 1 1 1 0
Dataword plus remainder
Note:
Multiply: AND
Subtract: XOR

Decoder

During transmission, the codeword may change. The decoder uses the same division process as the encoder. The remainder is the syndrome. If the syndrome is all 0s, the dataword is likely error-free, and it's accepted. Otherwise, if the syndrome has any non-zero values, the data is discarded. In the absence of errors, the syndrome is 000; if there's a single error, the syndrome is not all 0s, like in the case shown with 011 in Figure next page.


Uncorrupted
Codeword 1 0 0 1 1 1 0
Decoder
1 0 1 0
1 0 1 1) 1 0 0 1 1 1 0 ← Codeword
1 0 1 1
0 1 0 1
0 0 0 0
1 0 1 1
1 0 1 1
0 0 0 0
0 0 0 0
Zero 0 0 0 Syndrome
Dataword accepted 1 0 0 1
Corrupted
Codeword 1 0 0 0 1 1 0
Decoder
1 0 1 1
1 0 1 1) 1 0 0 0 1 1 0 ← Codeword
1 0 1 1
0 1 1 1
0 0 0 0
1 1 1 1
1 0 1 1
1 0 0 0
1 0 1 1
Non-Zero 0 1 1 Syndrome
Dataword discarded

Divisor

Choosing the divisor, like 1011, depends on the desired code properties. Criteria are discussed on the book website. Standard divisors in networking, like CRC-8 or CRC-32, are named based on the polynomial degree, with bits one more than the degree. For instance, CRC-8 has 9 bits, and CRC-32 has 33 bits. Check Table for common divisors.

Name	*Binary*	*Application*
CRC-8	100000111	ATM header
CRC-10	11000110101	ATM AAL
CRC-16	1000100000100001	HDLC
CRC-32	100000100110000010001110110110111	LANs

- **Important Notes-**
- **If the CRC generator is chosen according to the above rules, then-**
- CRC can detect all single-bit errors
- CRC can detect all double-bit errors provided the divisor contains at least three logic 1’s.
- CRC can detect any odd number of errors provided the divisor is a factor of x+1.
- CRC can detect all burst error of length less than the degree of the polynomial.
- CRC can detect most of the larger burst errors with a high probability.

- **Burst Errors.** If we assume the length of the burst error is L bits and r is the length of the remainder (r is the length of the generator minus 1; it is also the value of the highest power in the polynomial representing the generator):

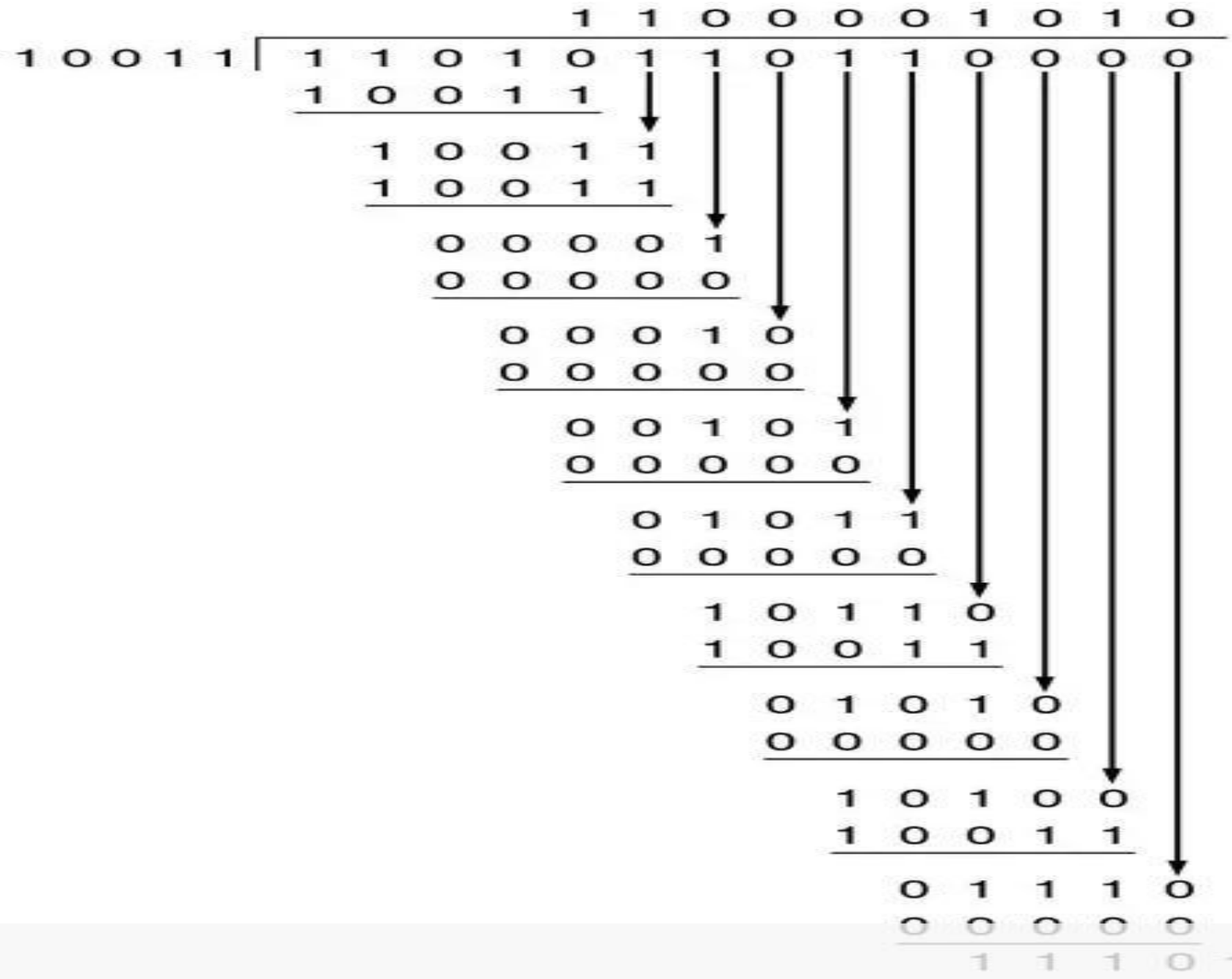
1. All burst errors of the size L ≤ r are detected.
2. All burst errors of the size L = r + 1 are detected with probability 1 − (0.5)^(r−1).
3. All burst errors of the size L > r + 1 are detected with probability 1 − (0.5)^(r)

Advantages of Cyclic Codes

Cyclic codes can easily be implemented in hardware and software. They are especially fast when implemented in hardware. This has made cyclic codes a good candidate for many networks. In the book website, we show how division can be done by a shift register that is included in the hardware of the node.

- **Q.** A bit stream 1101011011 is transmitted using the standard CRC method. The generator polynomial is x^4+x+1. What is the actual bit string transmitted?

- **Solution:**
- The generator polynomial $G(x) = x^4 + x + 1$ is encoded as 10011.
- Clearly, the generator polynomial consists of 5 bits.
- So, a string of **4 zeroes** is appended to the bit stream to be transmitted.
- The resulting bit stream is 11010110110000.
- Now, the binary division is performed as-



From here, CRC = 1110.
Now,
The code word to be transmitted is obtained by replacing the last 4 zeroes of 11010110110000 with the CRC.
Thus, the code word transmitted to the receiver = 11010110111110.

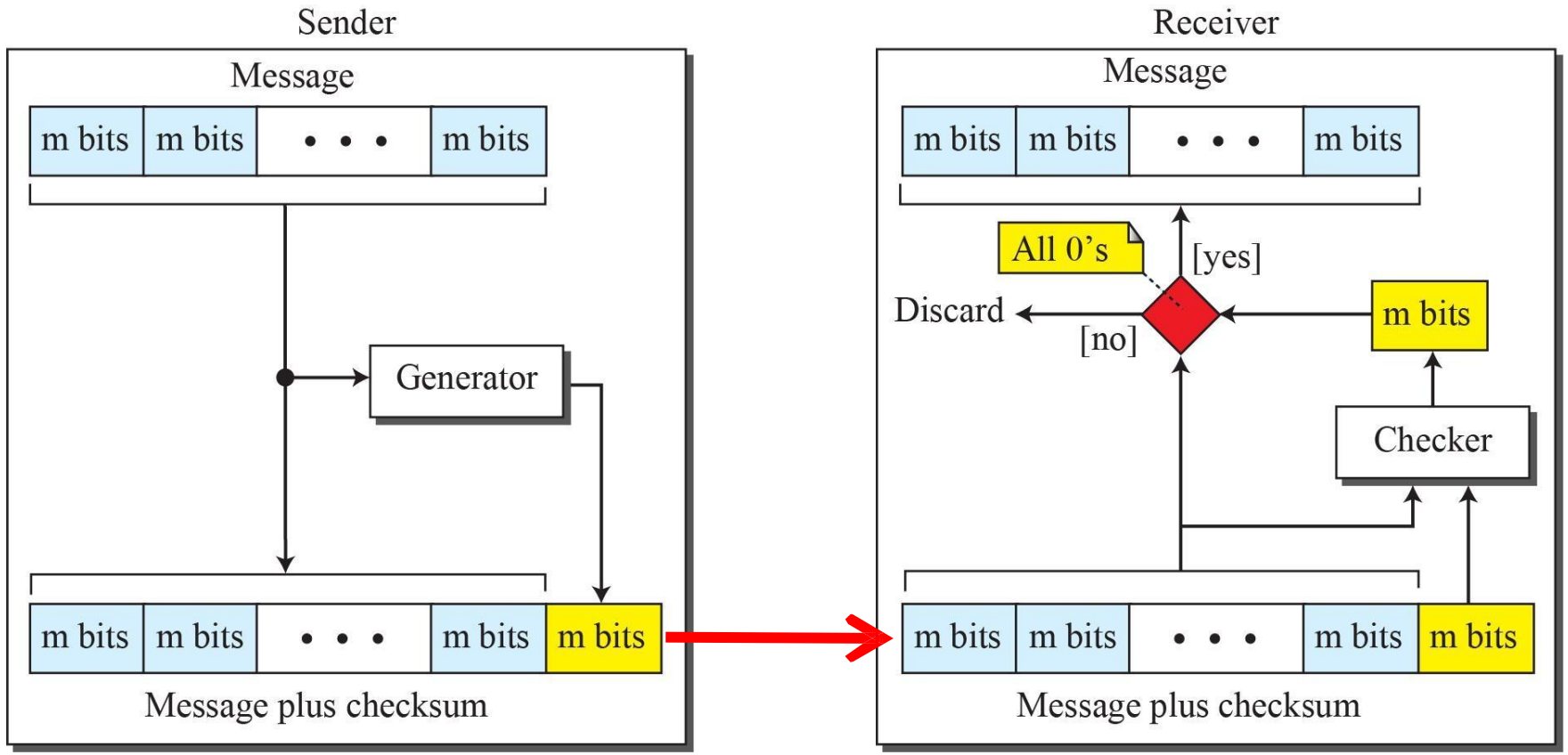
- **Q.** A bit stream 10011101 is transmitted using the standard CRC method. The generator polynomial is x^3+1.
- What is the actual bit string transmitted?
- Suppose the third bit from the left is inverted during transmission. How will receiver detect this error?
- **Solution:**
- **Part-01:**
- The generator polynomial $G(x) = x^3 + 1$ is encoded as 1001.
- Clearly, the generator polynomial consists of 4 bits.
- So, a string of 3 zeroes is appended to the bit stream to be transmitted.
- The resulting bit stream is 10011101000.
- Now, the binary division is performed and the remainder is obtained as 0100.
- From here, CRC = 100. (As 3 zeros were intially appended)

- Now,
- The code word to be transmitted is obtained by replacing the last 3 zeroes of 10011101000 with the CRC.
- Thus, the code word transmitted to the receiver = 10011101100.
- **Part-02:**
- According to the question,
- Third bit from the left gets inverted during transmission.
- So, the bit stream received by the receiver = 10111101100.
- Now,
- Receiver performs the binary division with the same generator polynomial which gives the remainder as **0100.**
- The remainder obtained on division is a **non-zero value** [i.e., 100 (Taking 3 bits).
- This indicates to the receiver that an error occurred in the data during the transmission.
- Therefore, receiver rejects the data and asks the sender for retransmission.

- **Q.** Data word to be sent - 100100 and the generator polynomial is $x^3 + x^2 + 1$. Perform the following:
- Case 1: When the receiver receives the correct codeword.
- Case 2: Fourth bit from the left gets inverted during transmission.

Checksum

Checksum is a method for detecting errors in a message. It's commonly used in Internet communication at the network and transport layers. The process involves dividing the message into m-bit units at the source, creating a checksum, and sending it along. At the destination, a new checksum is generated using the received message and original checksum. If the new checksum is all 0s, the message is accepted; otherwise, it's discarded. In practice, the checksum unit doesn't have to be added at the message's end; it can be inserted anywhere.


Sender
Message
m bits
m bits
m bits
Generator
m bits
m bits
m bits
m bits
m bits
Message plus checksum
Receiver
Message
m bits
m bits
m bits
All 0's
[yes]
Discard
[no]
m bits
Checker
m bits
m bits
m bits
m bits
m bits
Message plus checksum

Example

Suppose the message 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, if the set of numbers is (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 two are the same, the receiver assumes no error, accepts the five numbers, and discards the sum. Otherwise, there is an error somewhere and the message not accepted.

One’s Complement Addition

The issue with the previous example is that the sum may require more than 4 bits. To address this, one's complement arithmetic can be used, allowing representation of unsigned numbers between 0 and 2^(m) − 1 using only m bits. If a number exceeds m bits, the extra leftmost bits are added to the m rightmost bits through wrapping.

Example

In the previous example, the decimal number 36 in binary is (100100)2. To change it to a 4-bit number we add the extra leftmost bit to the right four bits as shown below.

$$(10)_2 + (0100)_2 = (0110)_2 \rightarrow (6)_{10}$$

Instead of sending 36 as the sum, we can send 6 as the sum (7, 11, 12, 0, 6, 6). The receiver can add the first five numbers in one’s complement arithmetic. If the result is 6, the numbers are accepted; otherwise, they are rejected.

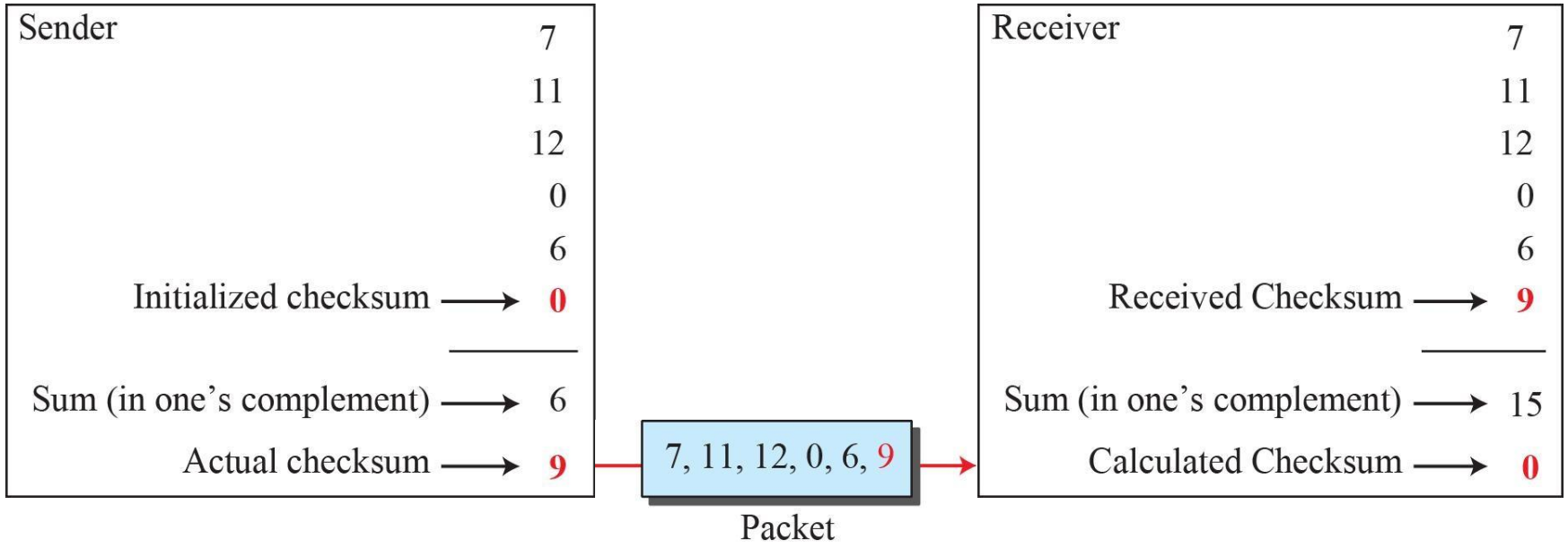
Checksum

To simplify the receiver's task, we can send the complement of the sum, known as the checksum. In one's complement arithmetic, finding the complement involves flipping all bits. This is equivalent to subtracting the number from 2^m − 1. In one's complement arithmetic, there are two zeros: a positive zero with all bits set to 0 and a negative zero with all bits set to 1 (2^m − 1). Adding a number to its complement results in a negative zero (all bits set to 1). When the receiver adds all numbers, including the checksum, it gets a negative zero, which can be complemented again to obtain a positive zero.

Example

Let us use the idea of the checksum in Example 5.9. The sender adds all five numbers in one's complement to get the sum = 6. The sender then complements the result to get the checksum = 9, which is 15 − 6. Note that 6 = (0110)2 and 9= (1001)2; they are complements of each other. The sender sends the five data numbers and the checksum (7, 11, 12, 0, 6, 9). If there is no corruption in transmission, the receiver receives (7, 11, 12, 0, 6, 9) and adds them in one's complement to get 15.

The sender complements 15 to get 0. This shows that data have not been corrupted. Figure shows the process.



Internet Checksum

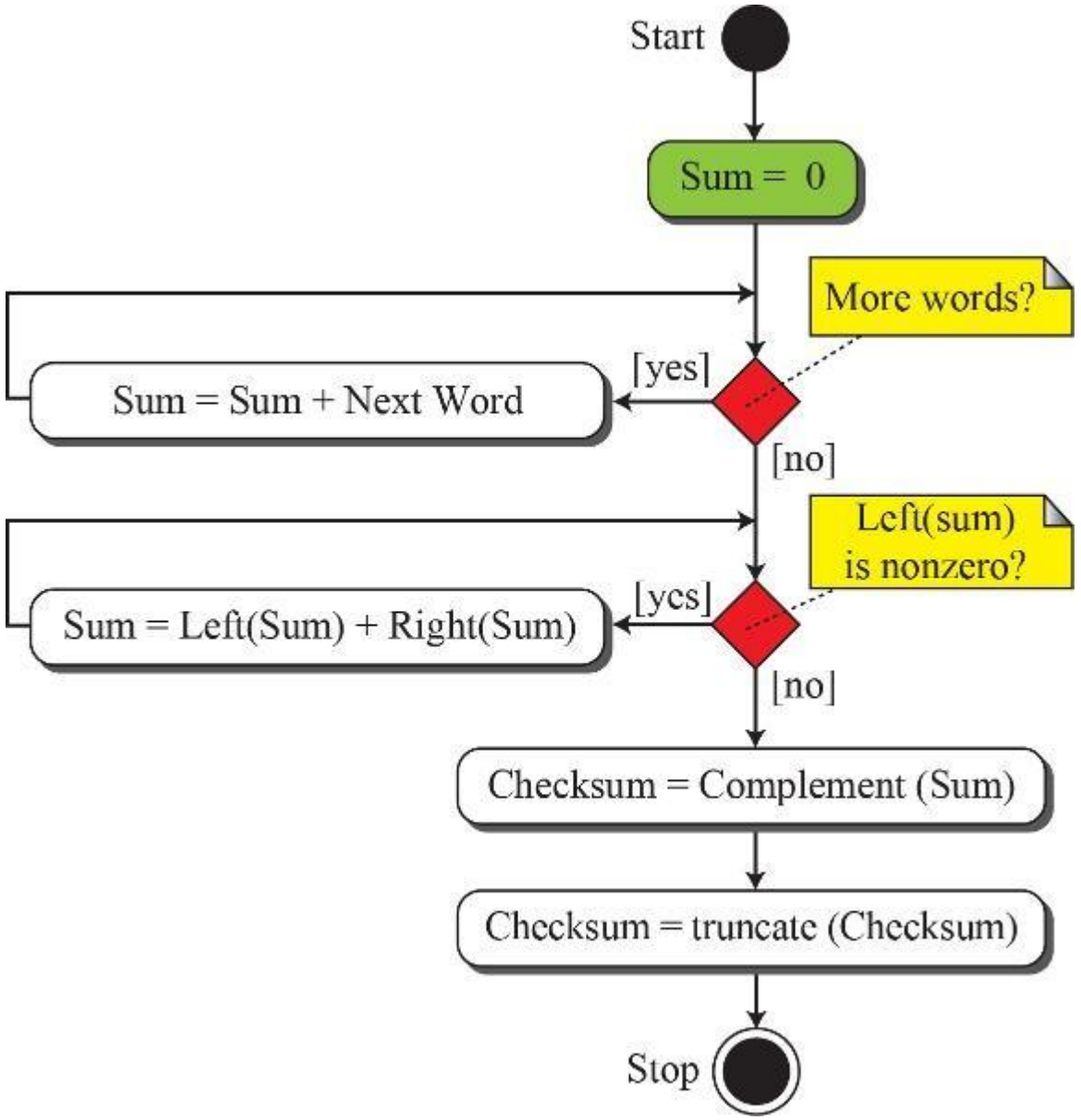
Traditionally, the Internet has used a 16-bit checksum. The sender and the receiver follow the steps depicted

Sender	*Receiver*
1. The message is divided into 16-bit words.	1. The message and the checksum is 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.

Algorithm

Notes:

a. Word and Checksum are each 16 bits, but Sum is 32 bits.
b. Left(Sum) can be found by shifting Sum 16 bits to the right.
c. Right(Sum) can be found by ANDing Sum with $(0000FFFF)_{16}$.
d. After Checksum is found, truncate it to 16 bits.



Other Approaches to the Checksum

The traditional checksum method has a significant flaw—it fails to detect errors if two 16-bit items are swapped during transmission. This is because the traditional checksum doesn't consider the order of data items; it treats each item equally. To address this issue, weighted checksums like Fletcher and Adler have been introduced.

Fletcher Checksum

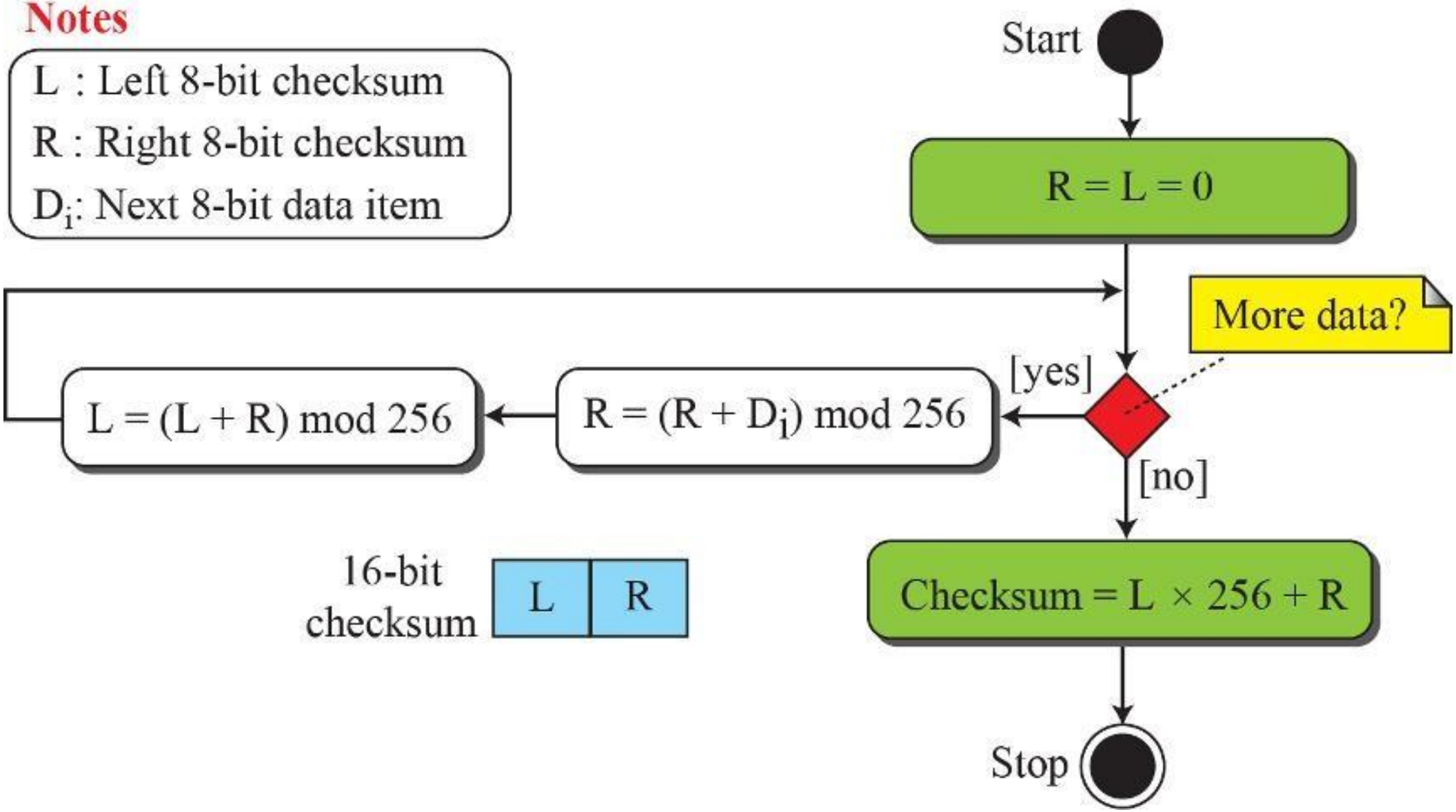
Fletcher checksum provides weighted calculations for data items based on their position. Two algorithms, 8-bit and 16-bit, were proposed by Fletcher. The 8-bit version operates on 8-bit data items, generating a 16-bit checksum. It uses two accumulators, L and R, with R adding a weighted component. The calculation is done modulo 256. The 16-bit Fletcher works similarly but operates on 16-bit data items, creating a 32-bit checksum, and the calculation is done modulo 65,536.

Notes

L : Left 8-bit checksum

R : Right 8-bit checksum

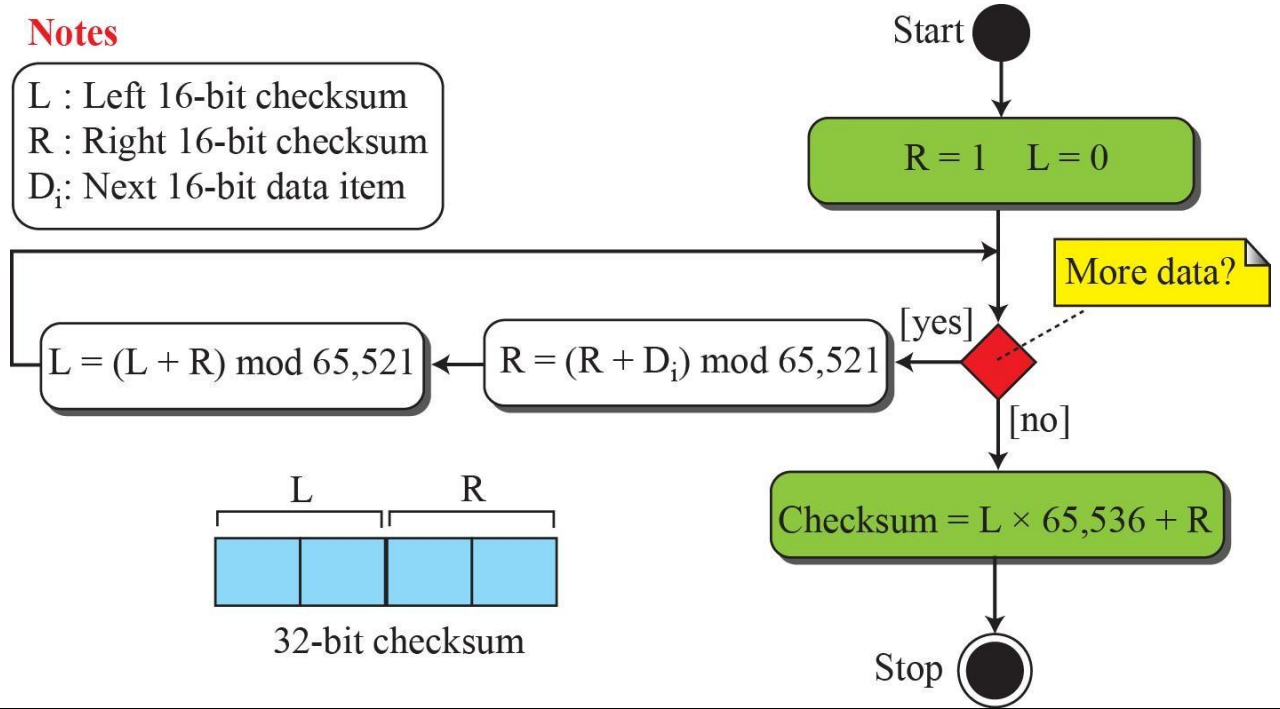
D_i: Next 8-bit data item



16-bit checksum

Adler Checksum

The Adler checksum is a 32-bit checksum with a simple algorithm. In contrast to the 16-bit Fletcher, it operates on single bytes, uses a prime modulus (65,521), and initializes L to 1 instead of 0. The use of a prime modulo is shown to enhance error detection in certain data combinations.



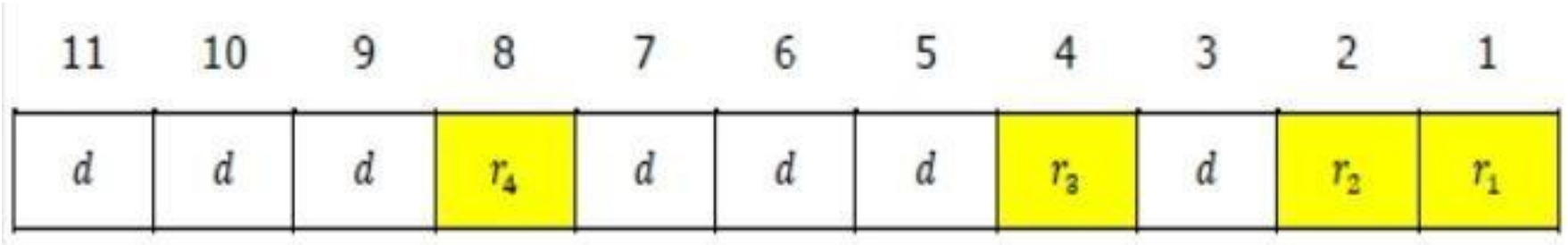
HAMMING CODE

Hamming code is a block code that is capable of detecting up to two simultaneous bit errors and correcting single-bit errors. It was developed by R.W. Hamming for error correction. In this coding method, the source encodes the message by inserting redundant bits within the message. These redundant bits are extra bits that are generated and inserted at specific positions in the message itself to enable error detection and correction. When the destination receives this message, it performs recalculations to detect errors and find the bit position that has error.

- **Encoding a message by Hamming Code:**
- Step 1 − Calculation of the number of redundant bits.
- Step 2 − Positioning the redundant bits.
- Step 3 − Calculating the values of each redundant bit.
- Once the redundant bits are embedded within the message, this is sent to the receiver.

- **Step 1 − Calculation of the number of redundant bits.**
- If the message contains m number of data bits, r number of redundant bits are added to it so that is able to indicate at least (m + r + 1) different states. Here, (m + r) indicates location of an error in each of bit positions and one additional state indicates no error. Since, r bits can indicate 2^r states, 2^r must be at least equal to (m + r + 1). Thus the following equation should hold −
- $$2^r \geq + \quad + 1$$
- **Example −** If the data is of 7 bits, i.e. m = 7, the minimum value of r that will satisfy the above equation is 4, ($24 \geq 7 + 4 + 1$). The total number of bits in the encoded message, (m + r) = 11. This is referred as (11,4) code.

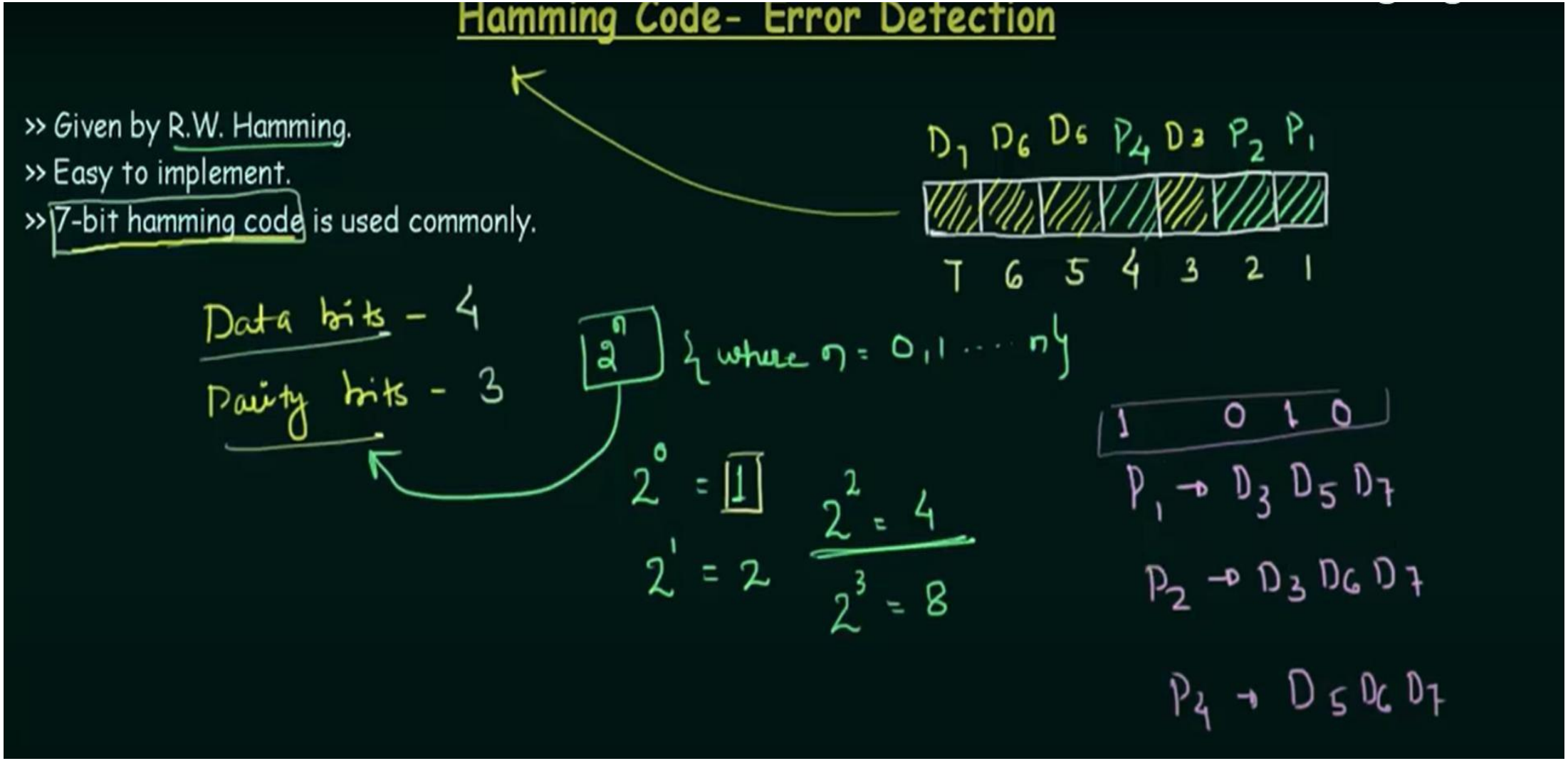
- Step 2 − Positioning the redundant bits.
- The r redundant bits placed at bit positions of powers of 2, i.e. 1, 2, 4, 8, 16 etc. They are referred in the rest of this text as r1 (at position 1), r2 (at position 2), r3 (at position 4), r4 (at position 8) and so on.
- **Example** − If, m = 7 comes to 4, the positions of the redundant bits are as follows −



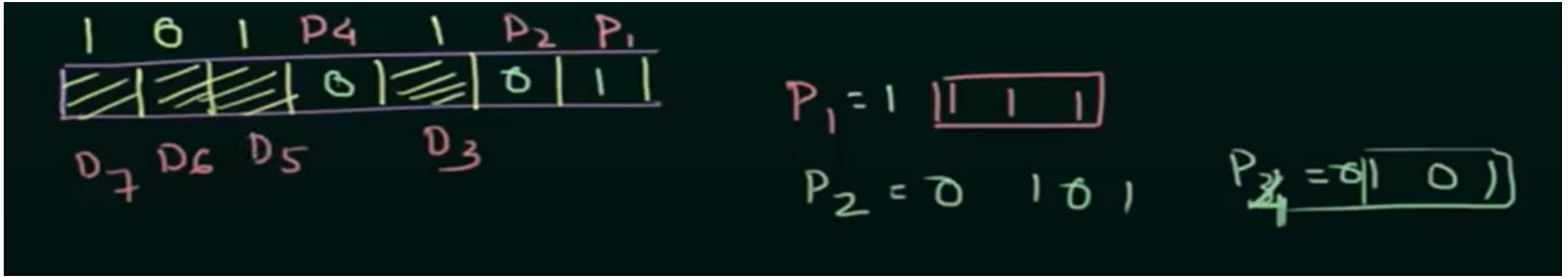
- **Step 3 − Calculating the values of each redundant bit.**
- The redundant bits are parity bits. A parity bit is an extra bit that makes the number of 1s either even or odd. The two types of parity are −

- Even Parity − Here the total number of bits in the message is made even.
- Odd Parity − Here the total number of bits in the message is made odd.
- Each redundant bit, r_i, is calculated as the parity, generally even parity, based upon its bit position. It covers all bit positions whose binary representation includes a 1 in the i^{th} position except the position of r_i. Thus −
- r1 is the parity bit for all data bits in positions whose binary representation includes a 1 in the least significant position excluding 1 (3, 5, 7, 9, 11 and so on)
- r2 is the parity bit for all data bits in positions whose binary representation includes a 1 in the position 2 from right except 2 (3, 6, 7, 10, 11 and so on)
- r3 is the parity bit for all data bits in positions whose binary representation includes a 1 in the position 3 from right except 4 (5-7, 12-15, 20-23 and so on)

Hamming Code for Error Detection and Correction



- **Q.** Suppose the data to be transmitted is 1011. Construct the 7-bit Hamming code for this data.



- Therefore, the 7-bit Hamming code which will be transmitted from the sender to the receiver is: **1010101**.

Single-Error correction, Double-Error detection:

We know that hamming code can correct and detect one bit error only. For Double error detection follow the below steps:

- After generating hamming code we add an extra bit denoted as 'P_{n+1}' which is even parity of that hamming code. i.e; p_{n+1} = xor of all bits in hamming code generated.

Now calculate error position C and also calculate P = XOR of received hamming code

CONDITIONS FOR ERROR DETECTION AND CORRECTION:

1. If $C = 0$ and $P = 0$, no error occurred
2. If $C \neq 0$ and $P = 1$, a single error occurred that can be corrected
3. If $C \neq 0$ and $P = 0$, a double error occurred that is detected but that cannot be corrected
4. If $C = 0$ and $P = 1$, an error occurred in the P_{n+1} bit

Cont..

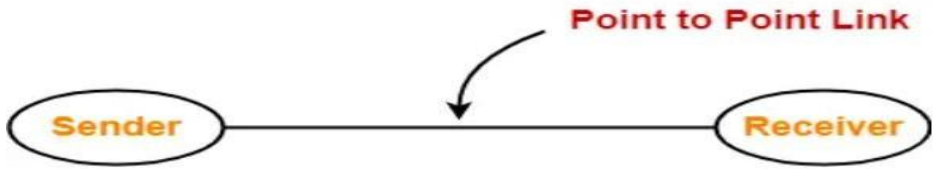
- **Q.** Consider the hamming code 1001100. Suppose during transmission the first two bits from left is fliped. How the receiver will detect this error?
- **Solution**:

MULTIPLE ACCESS PROTOCOLS

The data-link layer comprises two sublayers: data link control (DLC) and media access control (MAC). DLC, discussed earlier, manages data transfer on dedicated links like dial-up telephone lines. In contrast, when sharing media with others, a MAC protocol is needed to coordinate access. For instance, in cellular communication, where the channel is not dedicated, multiple users share the same band. In multipoint or broadcast links, a multiple-access protocol is essential to regulate access, preventing collisions and ensuring fair usage. Various protocols address this issue, categorized into three groups for shared link access.

ACCESS CONTROL

- **Types of Communication Links-** Communication links enable the stations to communicate with each other. Stations may communicate using the following types of links-
- **1. Point to Point Link-** Point to Point link is a dedicated link that exists between the two stations. The entire capacity of the link is used for transmission between the two connected stations only. Depending upon the Type Of Channel, the data flow takes place between the stations.
- **2. Broadcast Link-** Broadcast link is a common link to which multiple stations are connected. The capacity of the link is shared among the connected stations for transmission.

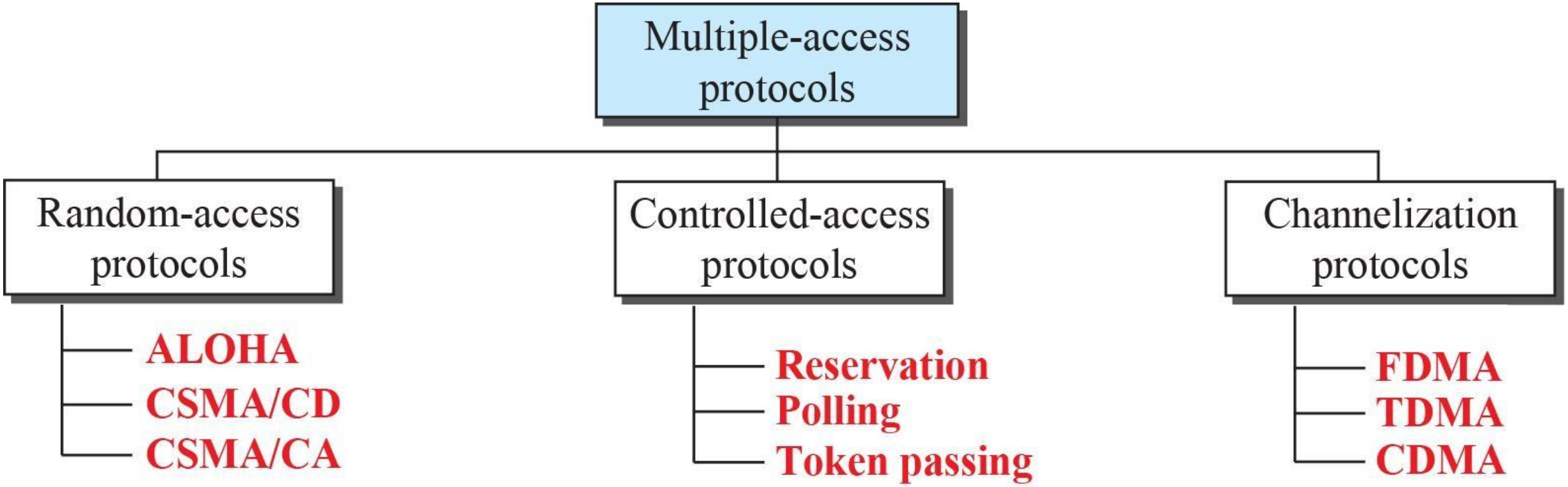


- **Access Control** is a mechanism that controls the access of stations to the transmission link.Broadcast links require the access control.This is because the link is shared among several stations.

- **Need of Access Control-** To prevent the occurrence of collision or if the collision occurs, to deal with it.

- **Consider a situation where-** Multiple stations place their data packets on the link and starts transmitting simultaneously. Such a situation gives rise to a collision among the data packets. **Collision of data packets causes the data to get corrupt.**

- **Access Control Methods-** Access control methods are the methods used for providing access control. **They prevent the collision or deal with it and ensures smooth flow of traffic on the network.** They are implemented at the data link layer of the OSI reference model.


Multiple-access protocols
Random-access protocols
ALOHA
CSMA/CD
CSMA/CA
Controlled-access protocols
Reservation
Polling
Token passing
Channelization protocols
FDMA
TDMA
CDMA

- **RANDOM ACCESS:** In **random access** or **contention methods**, no station is superior to another station and none is assigned the control over another. No station permits, or does not permit, another station to send. **At each instance, a station that has data to send uses a procedure defined by the protocol to make a decision on whether or not to send.** This decision depends on the state of the medium (idle or busy). In other words, each station can transmit when it desires on the condition that it follows the predefined procedure.
- **Two features give this method its name:**

1. 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**.
2. 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 as **contention methods.**

- In a random access method, each station has the right to the medium without being controlled by any other station. However, 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?

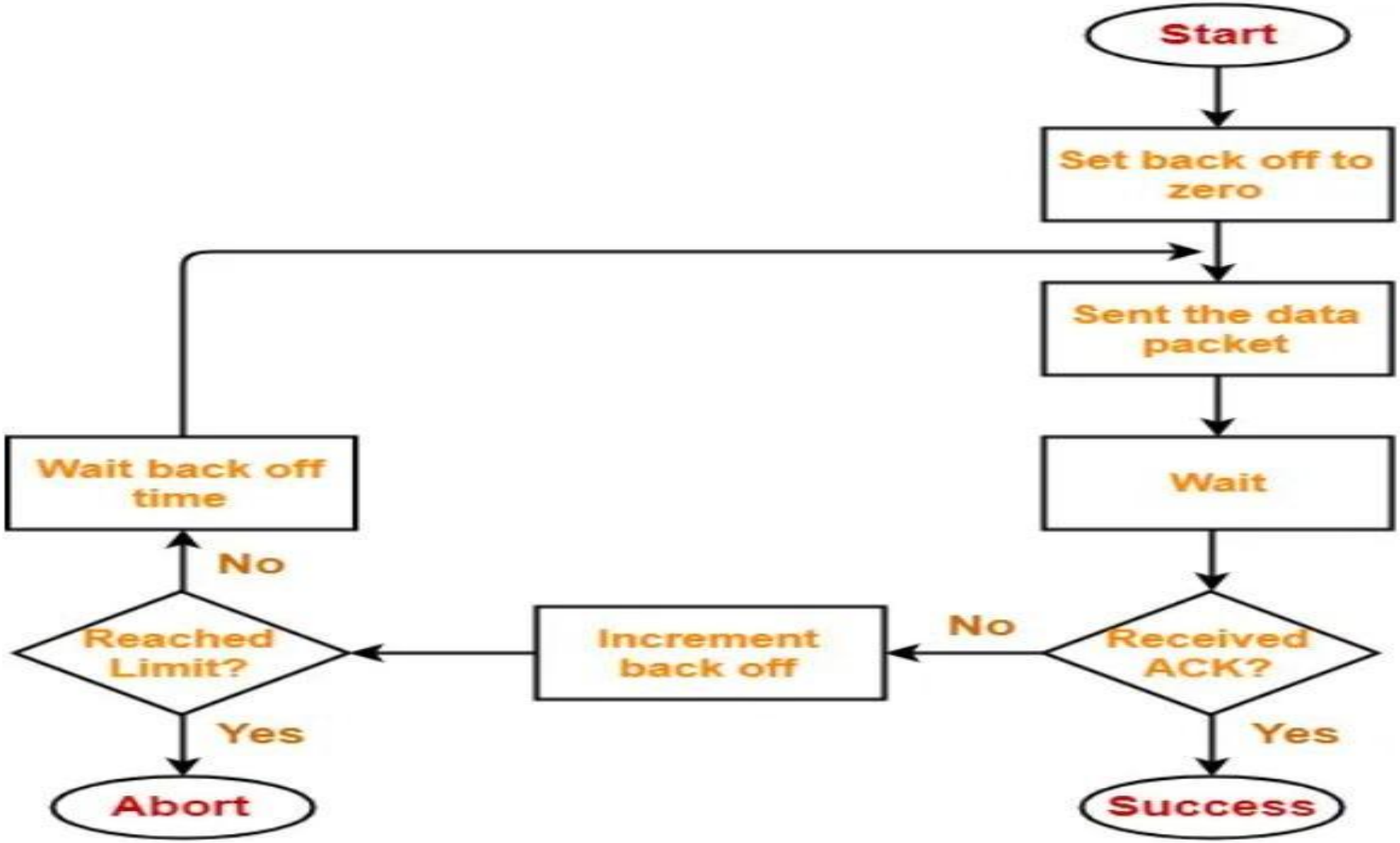
ALOHA: ALOHA, the earliest random access method, was developed at the University of Hawaii in early 1970. It was designed for a radio (wireless) LAN, but it can be used on any shared medium.

Versions of ALOHA:

1. PURE ALOHA:

It allows the stations to transmit data at any time whenever they want. After transmitting the data packet, station waits for some time. Then, following 2 cases are possible-

- **Case-01:** Transmitting station receives an acknowledgement from the receiving station.In this case, transmitting station assumes that the transmission is successful.
- **Case-02:Transmitting station does not receive any acknowledgement within specified time from the receiving station.** In this case, transmitting station assumes that the transmission is unsuccessful. Then, Transmitting station uses a **Back Off Strategy** and waits for some random amount of time. After back off time, it transmits the data packet again. It keeps trying until the back off limit is reached after which it aborts the transmission.


Start
Set back off to zero
Sent the data packet
Wait
Received ACK?
No
Increment back off
Reached Limit?
No
Wait back off time
Yes
Abort
Yes
Success

- **Efficiency-**

Efficiency of Pure Aloha **(η) = G x e^{-2G}**

where G = Number of stations willing to transmit data

- **Maximum Efficiency-**

Maximum value of η occurs at G = 1/2

Substituting G = 1/2 in the above expression, we get-

Maximum efficiency of Pure Aloha

$= 1/2 \times e^{-2 \times 1/2}$

$= 1 / 2e$

$= 0.184$

$=$ **18.4%**

The maximum efficiency of Pure Aloha is very less due to large number of collisions.

Vulnerable time: Let us find the length of time, the vulnerable time, in which there is a possibility of collision.

Pure ALOHA vulnerable time = 2 x Tfr,

Where**, Tfr = frame transmission time.**

Q.1. 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?

Ans: Average frame transmission time Tfr = 200 bits/200 kbps or 1 ms.

The vulnerable time is 2 x 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 1 ms period that this station is sending.

Q.2. 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 200bit/200 kbps or 1 ms.

- **a**. If the system creates 1000 frames per second, this is 1 frame per millisecond. The load is 1. In this case $S = G \times e^{-2G}$ or $S = 1 \times e^{-2 \times 1}$ or **S = 0.135** (13.5 percent). This means that the **throughput is 1000 X 0.135 =135 frames**. Only 135 frames out of 1000 will probably survive.
- **b.** If the system creates 500 frames per second, this is (1/2) frame per millisecond. The load is (1/2). In this case $S = G \times e^{-2G}$ or or $S = 1/2 \times e^{-2 \times 1/2}$ or **S = 0.184** (18.4 percent). This means that the **throughput is 500 x 0.184 =92** and that only 92 frames out of 500 will probably survive.
- **c.** Throughput is 250 x 0.152 = 38. Only 38 frames out of 250 will probably survive.

2. Slotted Aloha-

Slotted Aloha divides the time of shared channel into discrete intervals called as time slots. Any station can transmit its data in any time slot. The only condition is that station must start its transmission from the beginning of the time slot. If the beginning of the slot is missed, then station has to wait until the beginning of the next time slot. A collision may occur if two or more stations try to transmit data at the beginning of the same time slot.

Efficiency of Slotted Aloha (η) = G x e^{-G}

where G = Number of stations willing to transmit data at the beginning of the same time slot

Maximum Efficiency- Maximum value of η occurs at G = 1, Substituting G = 1 in the above expression, we get Maximum efficiency of Slotted Aloha

= 1 x e^{-1}

= 1 / e

= 0.368

= **36.8%**

SlottedALOHA vulnerable time = Tfr

Pure Aloha	Slotted Aloha
Any station can transmit the data at any time.	Any station can transmit the data at the beginning of any time slot.
The time is continuous and not globally synchronized.	The time is discrete and globally synchronized.
Vulnerable time in which collision may occur $= 2 \times T_t$	Vulnerable time in which collision may occur $= T_t$
Probability of successful transmission of data packet $= G \times e^{-2G}$	Probability of successful transmission of data packet $= G \times e^{-G}$
Maximum efficiency = 18.4% (Occurs at G = 1/2)	Maximum efficiency = 36.8% (Occurs at G = 1)
The main advantage of pure aloha is its simplicity in implementation.	The main advantage of slotted aloha is that it reduces the number of collisions to half and doubles the efficiency of pure aloha.

- **Q.** A group of N stations share 100 Kbps slotted ALOHA channel. Each station output a 500 bits frame on an average of 5000 ms even if previous one has not been sent. What is the required value of N?

Throughput of each station

= Number of bits sent per second

= 500 bits / 5000 ms

= 500 bits / (5000 x 10^{-3} sec)

= 100 bits/sec

Throughput of slotted aloha

= Efficiency x Bandwidth

= 0.368 x 100 Kbps

= 36.8 Kbps

Throughput of slotted aloha = Total number of stations x Throughput of each station

Total number of stations = Throughput of slotted aloha / Throughput of each station

∴ N = 368.

Carrier Sense Multiple Access (CSMA)

To minimize the chance of collision and, therefore, increase the performance, the CSMA method was developed. The chance of collision can be reduced if a station senses the medium before trying to use it. Carrier sense multiple access (CSMA) requires that each station first listen to the medium (or check the state of the medium) before sending. In other words, CSMA is based on the principle "**sense before transmit**".

Vulnerable Time:

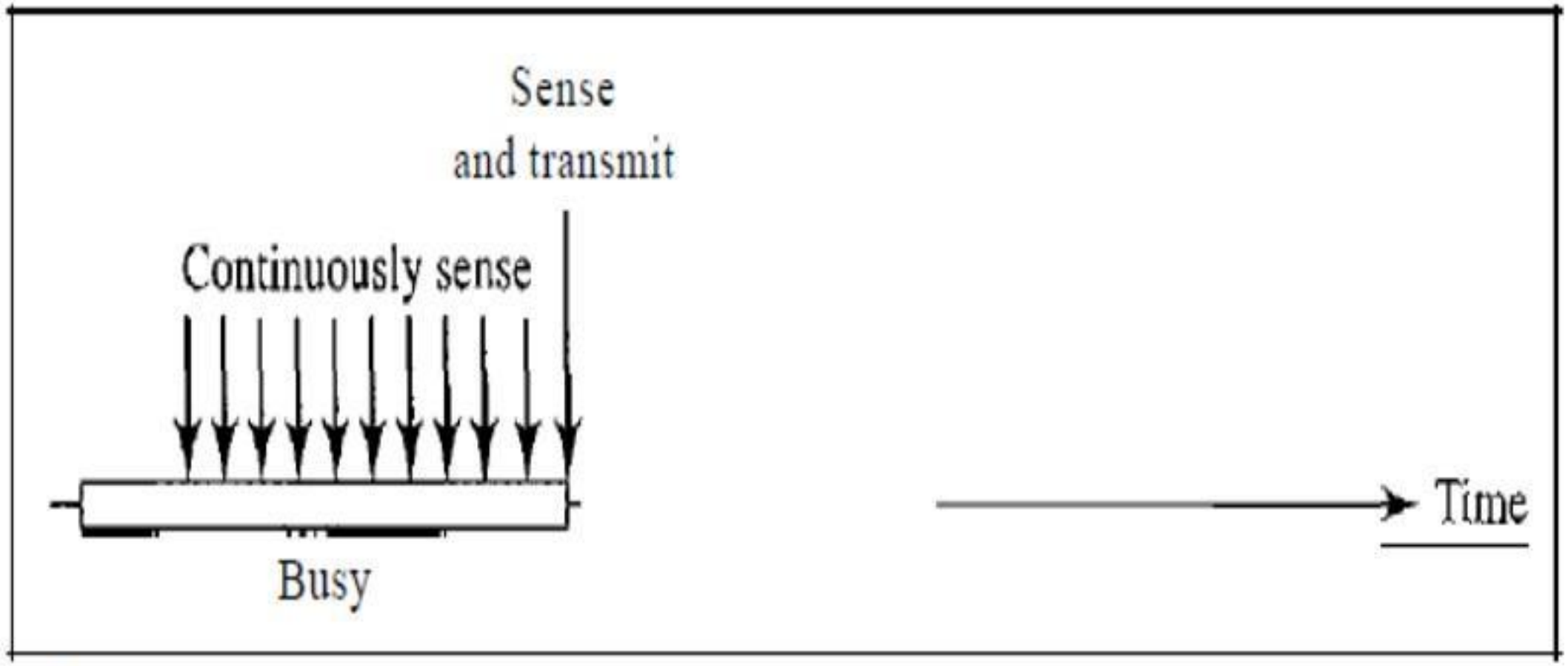
The vulnerable time for CSMA is the **propagation time Tp**. This is the time needed for a signal to propagate from one end of the medium to the other. When a station sends a frame, and any other station tries to send a frame during this time, a collision will result. But if the first bit of the frame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

Persistence Methods:

- What should a station do if the channel is busy? What should a station do if the channel is idle?
- Three methods have been devised to answer these questions: the **I-persistent method**, the **nonpersistent method**, and the **p-persistent method**.

I-Persistent: The I-persistent method is simple and straightforward.

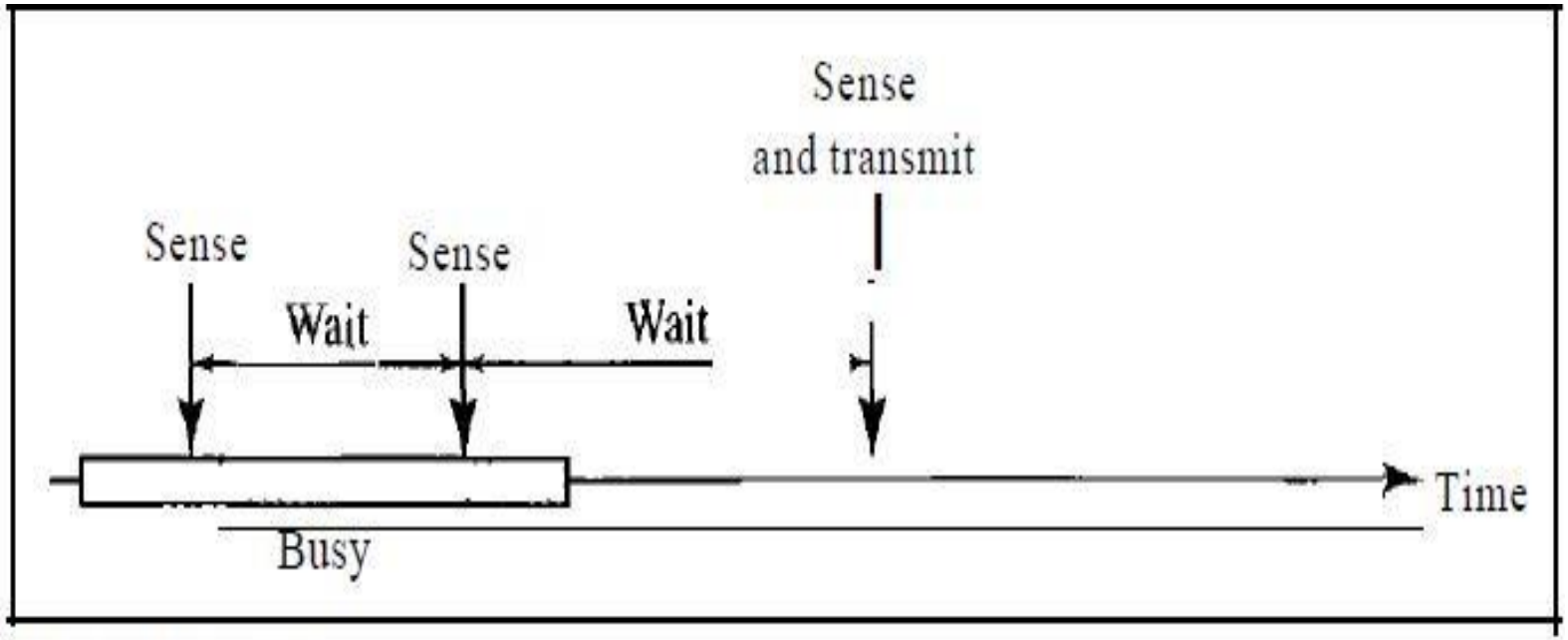
- In this method, after the station finds the line idle, it sends its frame immediately (with probability I).
- This method has the highest chance of collision because two or more stations may find the line idle and send their frames immediately.



3. I-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.



b. 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.

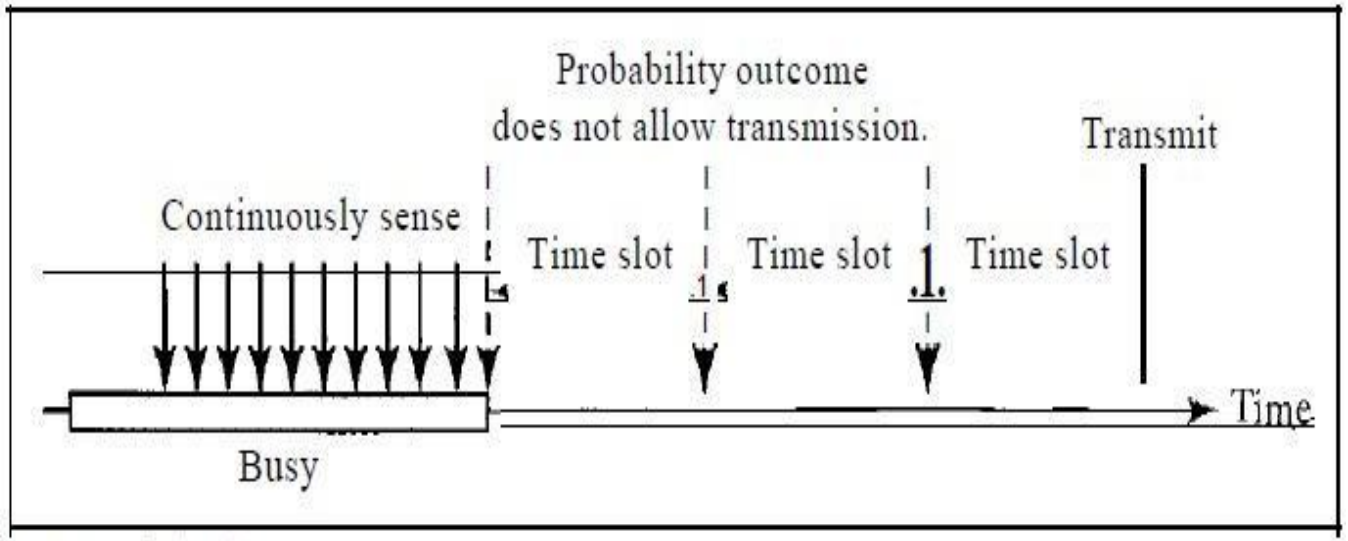
In this method, after the station finds the line idle it follows these steps:

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.

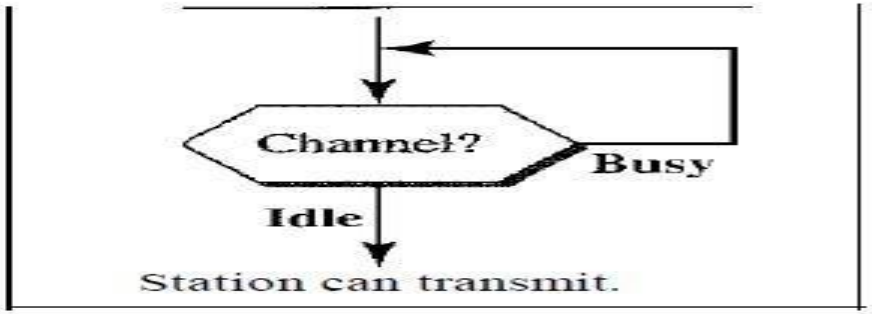
a. If the line is idle, it goes to step 1.

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

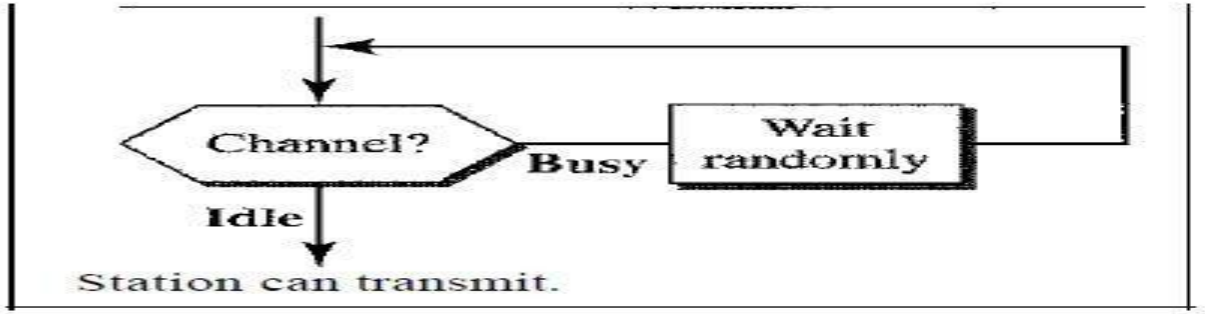


c. p-persistent

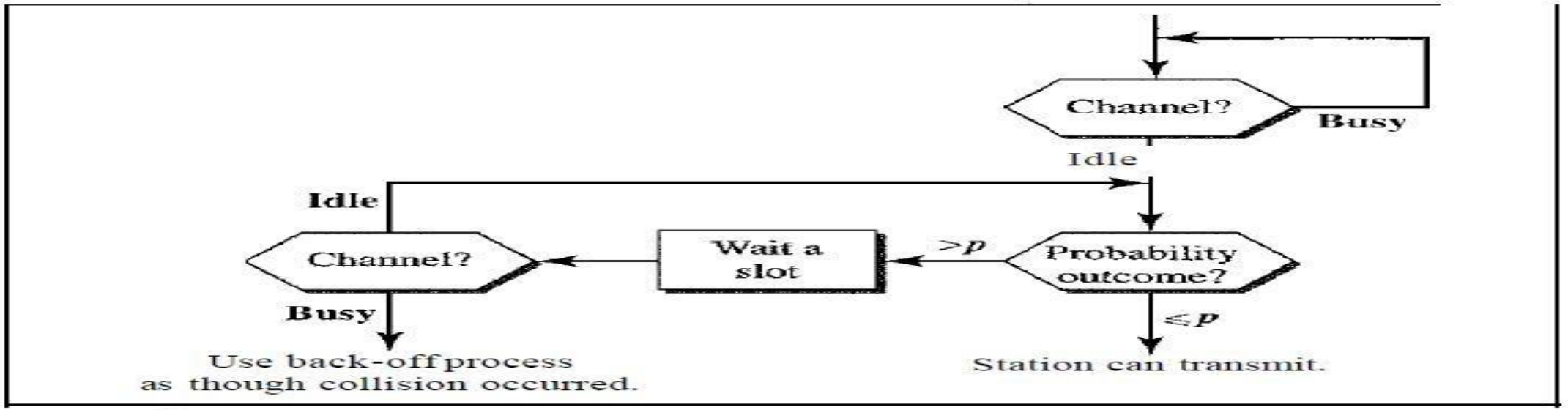
11 *Flow diagram for three persistence methods*



a. 1-persistent



b. Nonpersistent



c. p-persistent

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

The CSMA method does not specify the procedure following a collision. CSMA/CD augments the algorithm to handle the collision.

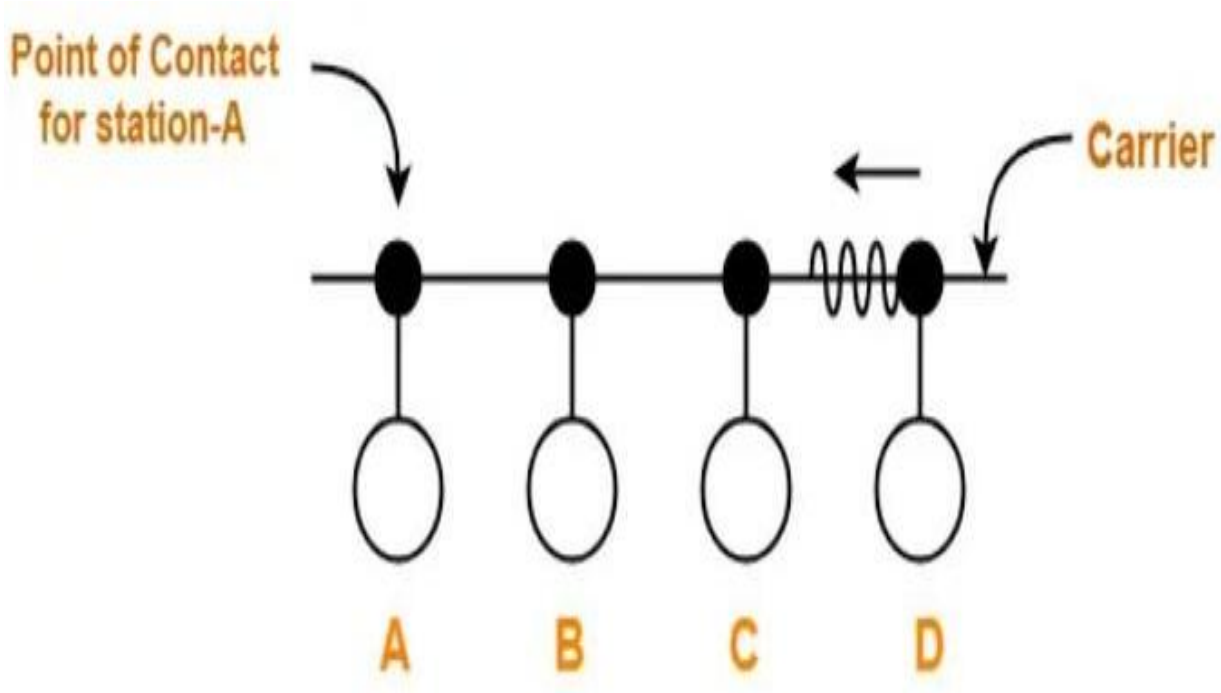
This access control method works as follows-

Step-01: Sensing the Carrier- Any station willing to transmit the data senses the carrier. If it finds the carrier free, it starts transmitting its data packet otherwise not.

How?

- Each station can sense the carrier only at its **point of contact** with the carrier.
- It is not possible for any station to sense the entire carrier.
- Thus, there is a huge possibility that a station might sense the carrier free even when it is actually not.

- **Example-**
- Consider the following scenario-



At the current instance,

- If station A senses the carrier at its point of contact, then it will find the carrier free.
- But the carrier is actually not free because station D is already transmitting its data.
- If station A starts transmitting its data now, then it might lead to a collision with the data transmitted by station D.

Step-02: Detecting the Collision-

In CSMA / CD, It is the responsibility of the transmitting station to detect the collision.For detecting the collision, CSMA / CD implements the following condition. This condition is followed by each station-

Transmission delay >= 2 x Propagation delay

According to this condition,

Each station must transmit the data packet of size whose transmission delay is at least twice its propagation delay.

If the size of data packet is smaller, then collision detection would not be possible.

Length Of Data Packet-

Transmission delay = Length of data packet (L) / Bandwidth (B)

Propagation delay = Distance between the two stations (D) / Propagation speed (V)

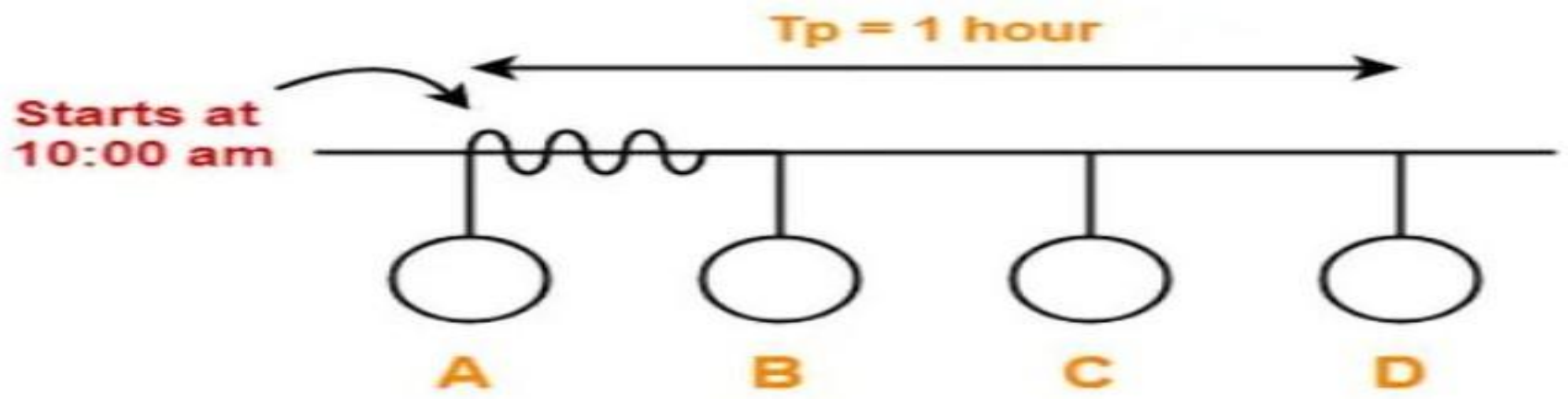
Substituting values in the above condition, we get-

$L / B >= 2 \times D / V$

$$\mathbf{L >= 2 \times B \times D / V}$$

Understanding the Condition To Detect Collision With Example:

- Consider at time 10:00 am, station A senses the carrier.
- It finds the carrier free and starts transmitting its data packet to station D.
- Let the propagation delay be 1 hour.
- (We are considering station D for the worst case)

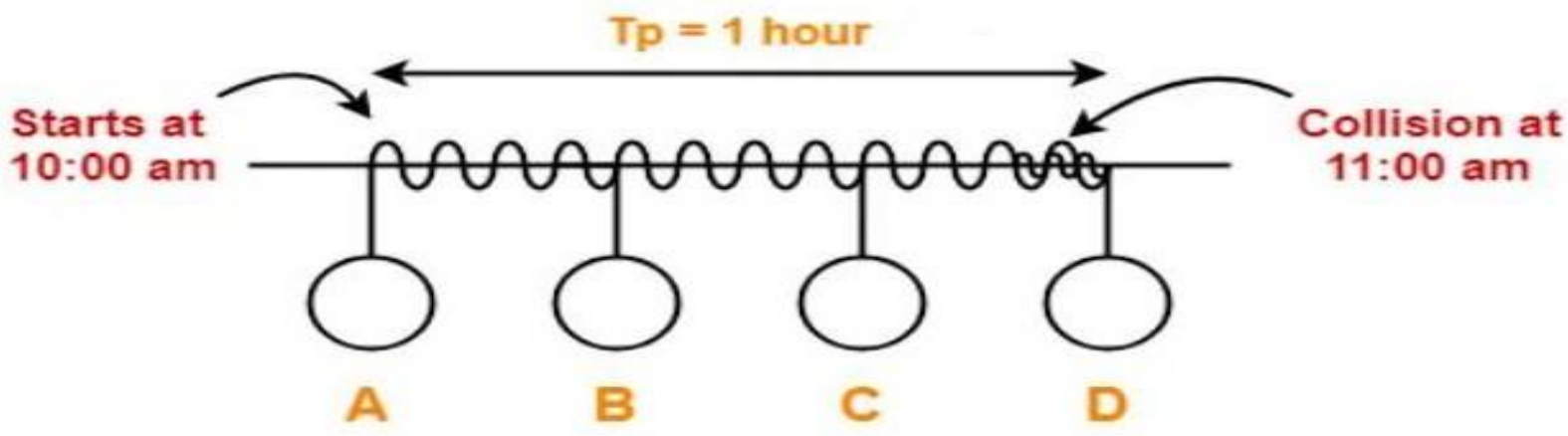


Let us consider the scenario at time 10:59:59:59 when the packet is about to reach the station D.

At this time, station D senses the carrier.

It finds the carrier free and starts transmitting its data packet.

Now, as soon as station D starts transmitting its data packet, a collision occurs with the data packet of station A at time **11:00 am**.



- **After collision occurs, the collided signal starts travelling in the backward direction.**
- The collided signal takes 1 hour to reach the station A after the collision has occurred.
- For station A to detect the collided signal, it must be still transmitting the data.
- So, transmission delay of station A must be >= 1 hour + 1 hour >= 2 hours to detect the collision.
- **That is why, for detecting the collision, condition is Tt >= 2Tp.**

Two cases are possible-

Case-01: If no collided signal comes back during the transmission,

- It indicates that no collision has occurred.
- The data packet is transmitted successfully.

Case-02: If the collided signal comes back during the transmission,

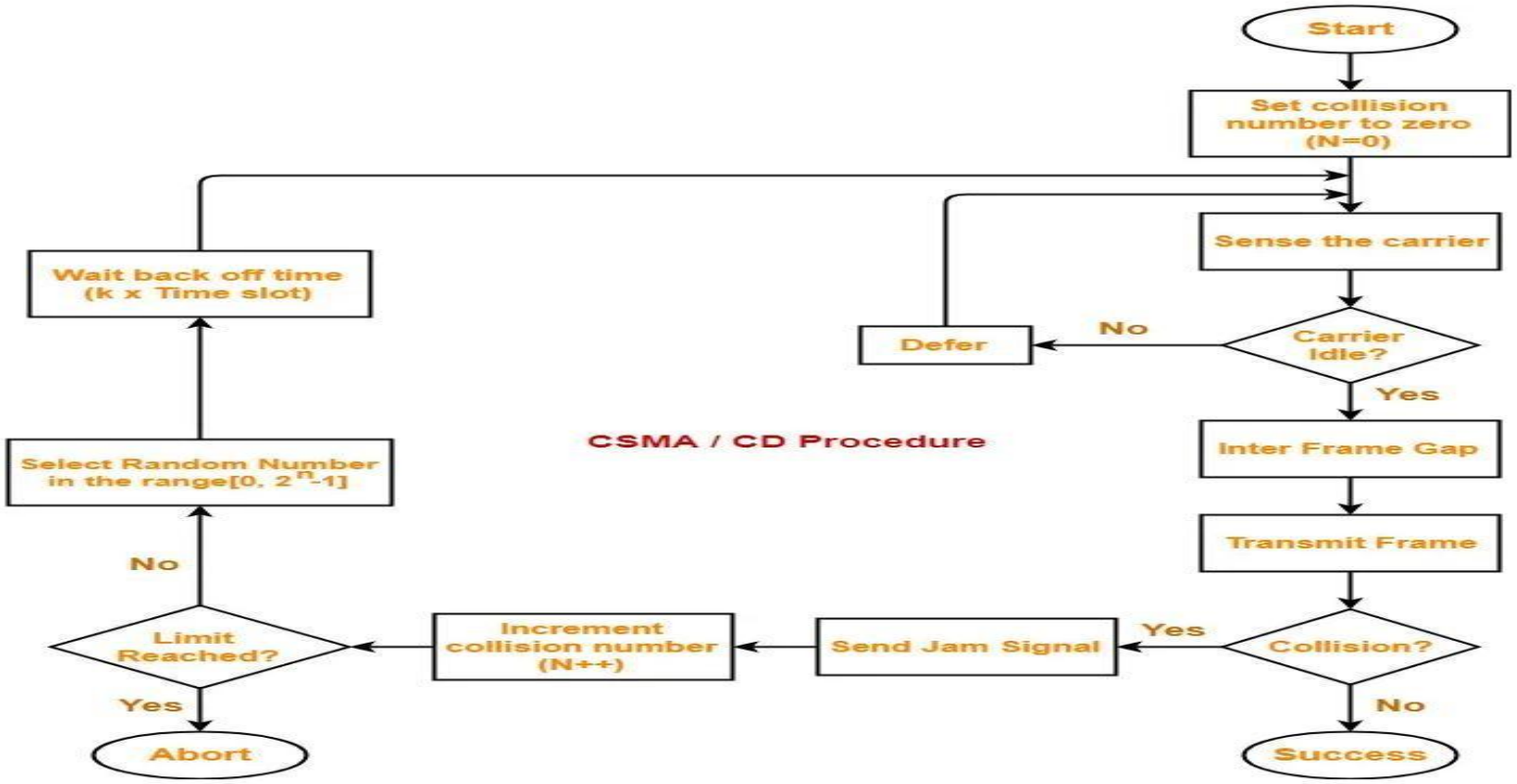
- It indicates that the collision has occurred.
- The data packet is not transmitted successfully.
- Step-03 is followed.

Step-03: Releasing Jam Signal-

- Jam signal is a 48 bit signal.
- It is released by the transmitting stations as soon as they detect a collision.
- It alerts the other stations not to transmit their data immediately after the collision.
- Otherwise, there is a possibility of collision again with the same data packet.
- Ethernet sends the jam signal at a frequency other than the frequency of data signals.
- This ensures that jam signal does not collide with the data signals undergone collision.

Step-04: Waiting For Back Off Time-

- After the collision, the transmitting station waits for some random amount of time called as **back off time**.
- After back off time, it tries transmitting the data packet again.
- If again the collision occurs, then station again waits for some random back off time and then tries again.
- The station keeps trying until the back off time reaches its limit.
- After the limit is reached, station aborts the transmission.



Efficiency:

$$\text{Efficiency } (\eta) = \frac{1}{1 + 6.44 \times a}, \text{ where } a = T_p / T_t$$

Q. In a CSMA / CD network running at 1 Gbps over 1 km cable with no repeaters, the signal speed in the cable is 200000 km/sec. What is minimum frame size?

Solution-

Given-

- Bandwidth = 1 Gbps
- Distance = 1 km
- Speed = 200000 km/sec

- Solution Cont..

Propagation delay (Tp)
= Distance / Propagation speed
= 1 km / (200000 km/sec)
= 0.5×10^{-5} sec
= 5×10^{-6} sec

Calculating Minimum Frame Size-
Minimum frame size
= 2 x Propagation delay x Bandwidth
= $2 \times 5 \times 10^{-6}$ sec x 10^{9} bits per sec
= 10000 bits

Q. Suppose nodes A and B are on same 10 Mbps Ethernet segment and the propagation delay between two nodes is 225 bit times. Suppose A and B send frames at t=0, the frames collide then at what time, they finish transmitting a jam signal. Assume a 48 bit jam signal.

Propagation delay (Tp)

= 225 bit times

= 225 bit / 10 Mbps

= 22.5×10^{-6} sec

= 22.5 μsec

At t = 0,

- Nodes A and B start transmitting their frame.
- Since both the stations start simultaneously, so collision occurs at the mid way.
- Time after which collision occurs = Half of propagation delay.
- So, time after which collision occurs = 22.5 μsec / 2 = 11.25 μsec.

At t = 11.25 μsec,

- After collision occurs at t = 11.25 μsec, collided signals start travelling back.
- Collided signals reach the respective nodes after time = Half of propagation delay
- Collided signals reach the respective nodes after time = 22.5 μsec / 2 = 11.25 μsec.
- Thus, at t = 22.5 μsec, collided signals reach the respective nodes.

At t = 22.5 μsec,

- As soon as nodes discover the collision, they immediately release the jam signal.
- Time taken to finish transmitting the jam signal = 48 bit time = 48 bits/ 10 Mbps = 4.8 μsec.
- Thus,
- Time at which the jam signal is completely transmitted
- = 22.5 μsec + 4.8 μsec
- = 27.3 μsec or 273 bit times
- More practise numericals can be found at the link:
- https://www.gatevidyalay.com/tag/csma-protocol/

Q.

Consider a CSMA/CD network that transmits data at a rate of 100 Mbps (10^8 bits per second) over 1 Km cable with no repeaters. If the minimum frame size required for this network is 1250 bytes What is the signal speed (Km/sec) in the cable ?

- In this question, also find the efficiency.

Ans: 20000 KM/Sec.

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA):

The basic idea behind CSMA/CD is that a station needs to be able to receive while transmitting to detect a collision. When there is no collision, the station receives one signal: its own signal. When there is a collision, the station receives two signals: its own signal and the signal transmitted by a second station. To distinguish between these two cases, the received signals in these two cases must be significantly different. In other words, the signal from the second station needs to add a significant amount of energy to the one created by the first station.

In a wired network, the received signal has almost the same energy as the sent signal. This means that in a collision, the detected energy almost doubles.

However, in a wireless network, much of the sent energy is lost in transmission. The received signal has very little energy. Therefore, a collision may add only 5 to 10 percent additional energy. This is not useful for effective collision detection.

We need to avoid collisions on wireless networks because they cannot be detected. Carrier sense multiple access with collision avoidance (CSMA/CA) was invented for this network. Collisions are avoided through the use of CSMAICA's three strategies: **the interframe space, the contention window,** and **acknowledgments.**

Interframe Space (IFS):

First, collisions are avoided by deferring transmission even if the channel is found idle. When an idle channel is found, the station does not send immediately. It waits for a period of time called the interframe space or IFS.

Even though the channel may appear idle when it is sensed, a distant station may have already started transmitting. The distant station's signal has not yet reached this station.

The IFS time allows the front of the transmitted signal by the distant station to reach this station.If after the IFS time the channel is still idle, the station can send, but it still needs to wait a time equal to the **contention time**.

Contention Window:

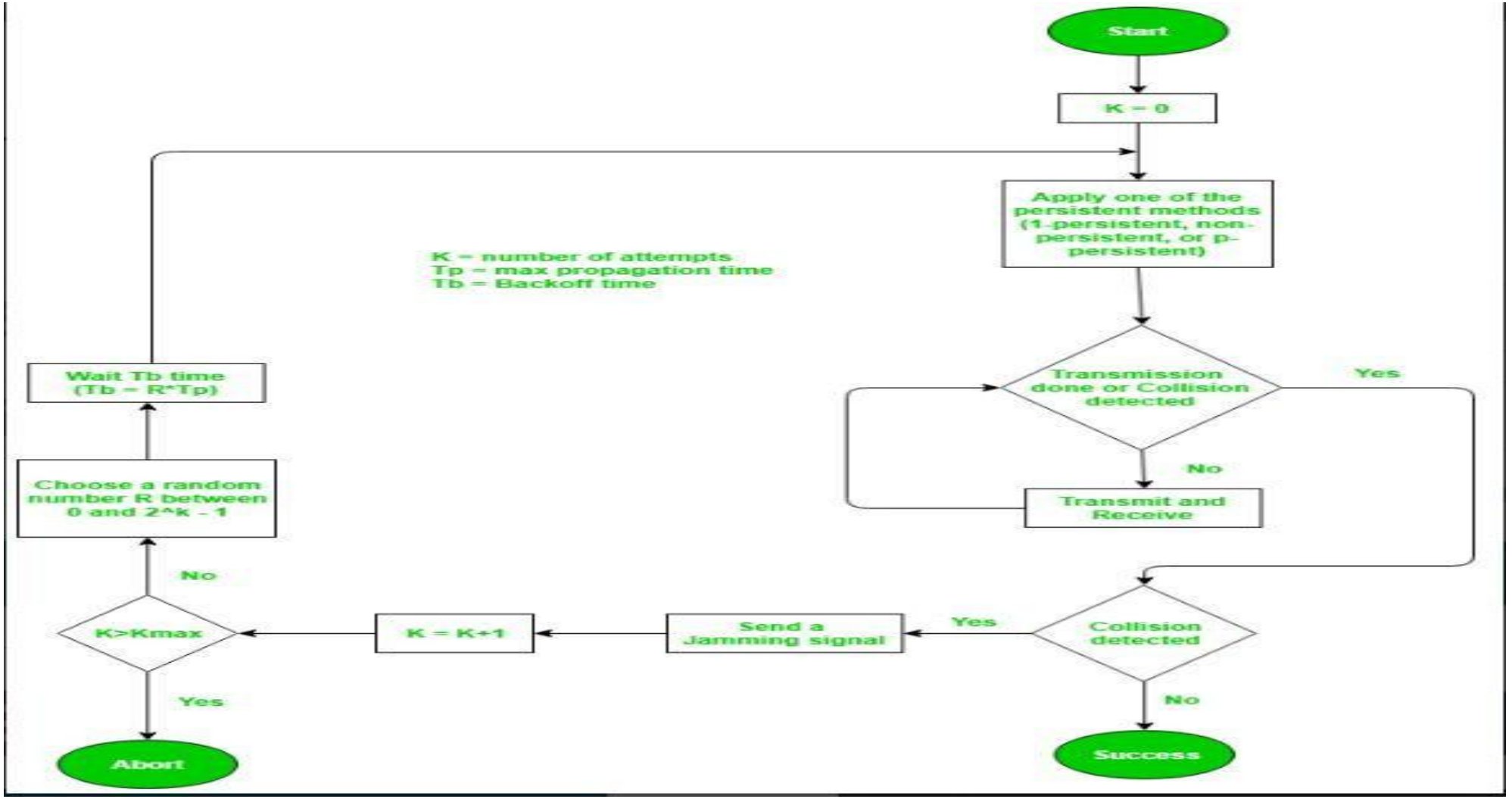
The contention window is an amount of time divided into slots.

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 back-off strategy. This means that 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.One interesting point about the contention window is that the station needs to sense thechannel 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.

- **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.

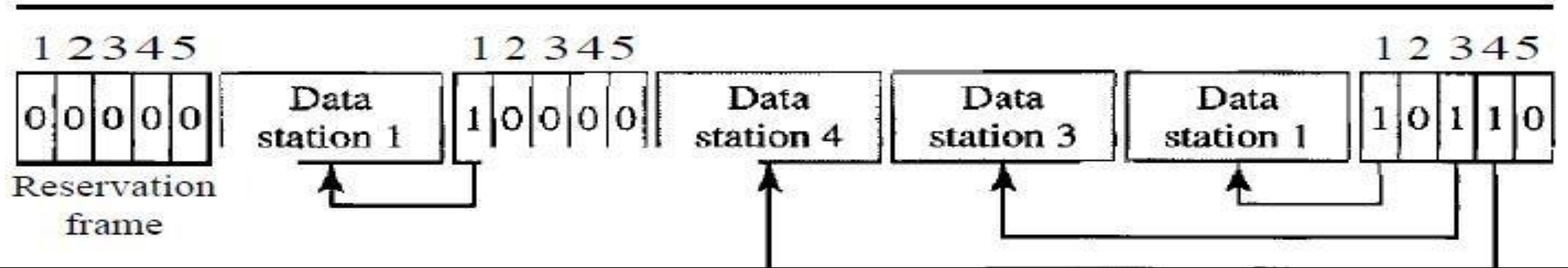


CONTROLLED ACCESS:

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.

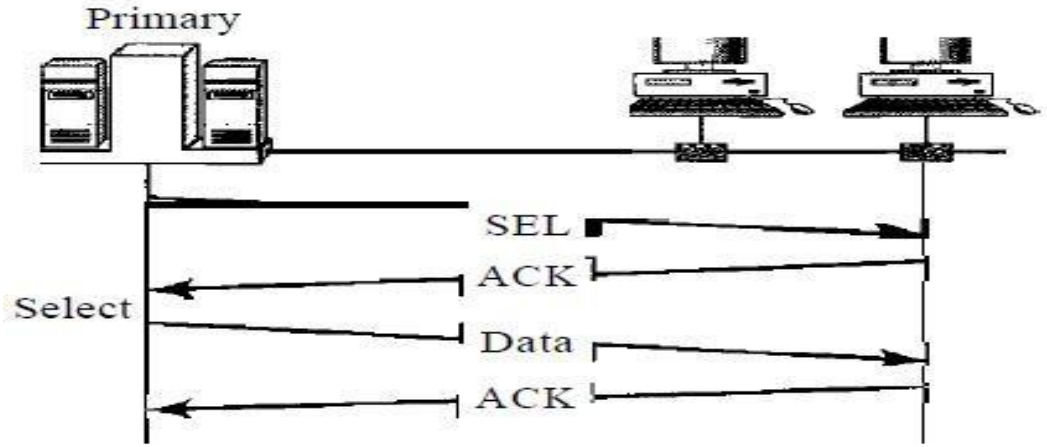
Reservation:

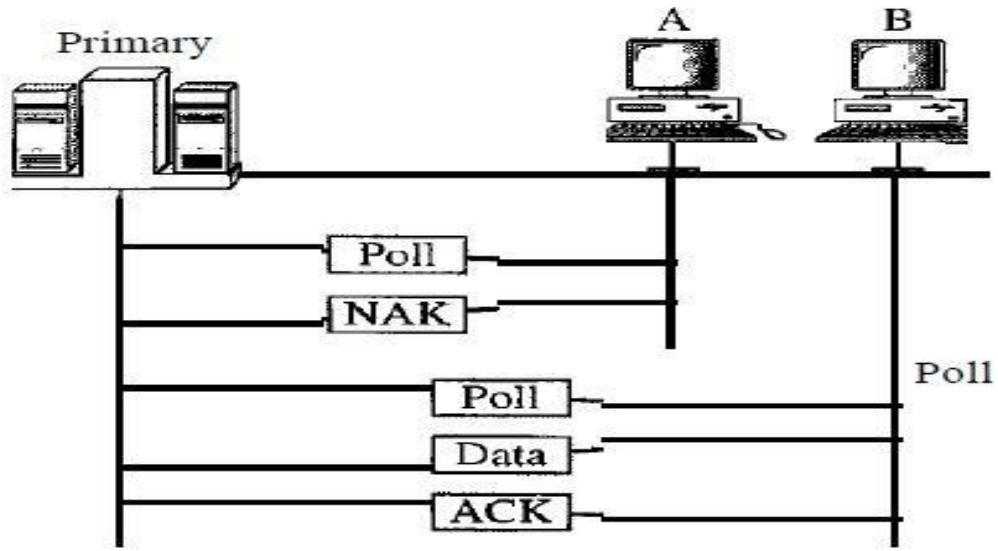
- In the reservation method, a station needs to make a reservation before sending data. Time is divided into **intervals**. In each interval, a reservation frame precedes the data frames sent in that interval. If there are N stations in the system, there are exactly N reservation minislots in the reservation 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.In the figure below, a situation with five stations and a five-minislot reservation frame.In the first interval, only stations 1, 3, and 4 have made reservations. In the second interval, only station 1 has made a reservation.



Polling:

- Polling works with topologies in which one device is designated as a primary station and the other devices are secondary stations. All data exchanges must be made through the primary device even when the ultimate destination is a secondary device. The primary device controls the link; the secondary devices follow its instructions. It is up to the primary device to determine which device is allowed to use the channel at a given time. The primary device, therefore, is always the initiator of a session.If the primary wants to receive data, it asks the secondaries if they have anything to send; this is called **poll function**. If the primary wants to send data, it tells the secondary to get ready to receive; this is called **select function**.





Select:

The select function is used whenever the primary device has something to send. Remember that the primary controls the link. If the primary is neither sending nor receiving data, it knows the link is available. If it has something to send, the primary device sends it. What it does not know, however, is whether the target device is prepared to receive. So the primary must alert the secondary to the upcoming transmission and wait for an acknowledgment of the secondary's ready status. Before sending data, the primary creates and transmits a select (SEL) frame, one field of which includes the address of the intended secondary.

Poll:

The poll function is 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 NAK frame if it has nothing to send or with data (in the form of a data frame) 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.

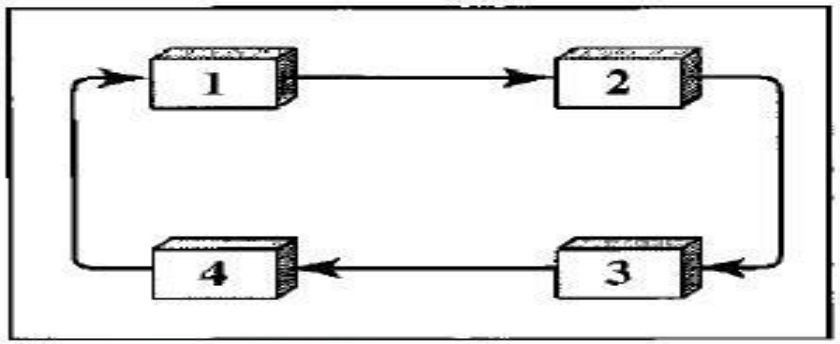
Token Passing:

In the token-passing method, the stations in a network are organized in a **logical ring**. In other words, for each station, there is 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. The right will be passed to the successor when the current station has no more data to send.

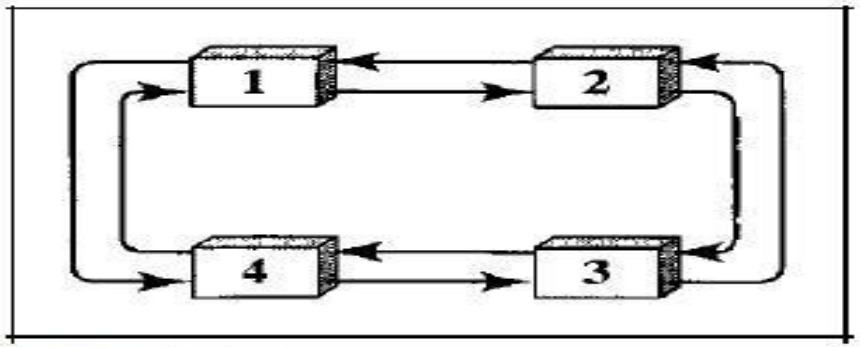
In this method, a **special packet** called a **token** circulates through the ring. The possession of the token gives the station the right to access the channel and send its data. When a station has some data to send, it waits until it receives the token from its predecessor. It then holds the token 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.

In this process, when a station receives the token and has no data to send, it just passes the token to the next station.

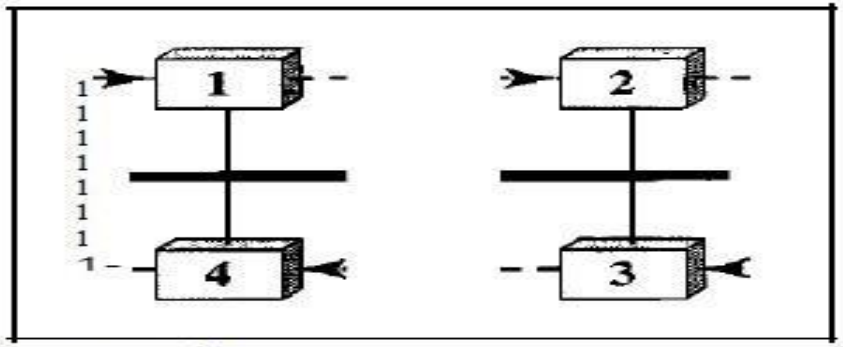
- **Logical Ring:**
- In a token-passing network, stations do not have to be physically connected in a ring; the ring can be a logical one.
- Figure below show four different physical topologies that can create a logical ring.



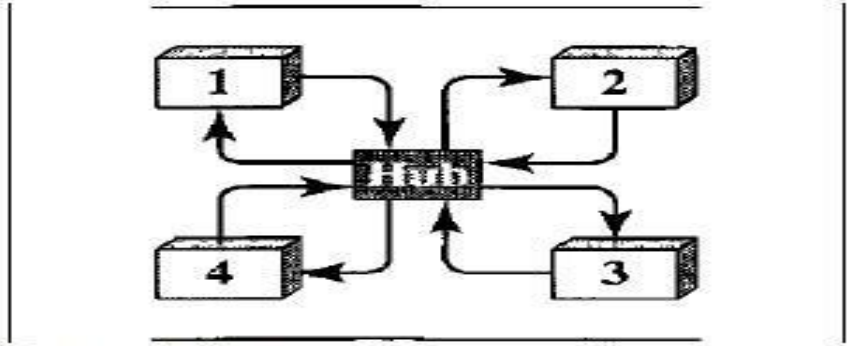
a. Physical ring



b. Dual ring



c. Bus ring



d. Star ring

In the **physical ring topology**, when a station sends the token to its successor, the token cannot be seen by other stations; the successor is the next one in line. This means that the token does not have to have the address of the next successor. The problem with this topology is that if one of the links-the medium between two adjacent stations fails, the whole system fails.

The **dual ring topology** uses a **second (auxiliary) ring** which operates in the reverse direction compared with the main ring.

The second ring is for emergencies only. If one of the links in the main ring fails, the system automatically combines the two rings to form a temporary ring. After the failed link is restored, the auxiliary ring becomes idle again. Note that for this topology to work, each station needs to have two transmitter ports and two receiver ports. The high-speed Token Ring networks called FDDI (Fiber Distributed Data Interface) and CDDI (Copper Distributed Data Interface) use this topology.

In the **bus ring topology**, also called a **token bus**, the stations are connected to a single cable called a **bus**.

They, however, make a logical ring, because each station knows the address of its successor (and also predecessor for token management purposes).

When a station has finished sending its data, it releases the token and inserts the address of its successor in the token.

Only the station with the address matching the destination address of the token gets the token to access the shared media.

In a **star ring topology**, the physical topology is a star. There is a hub, however, that acts as the connector. The wiring inside the hub makes the ring; the stations are connected to this ring through the two wire connections. This topology makes the network less prone to failure because if a link goes down, it will be bypassed by the hub and the rest of the stations can operate.

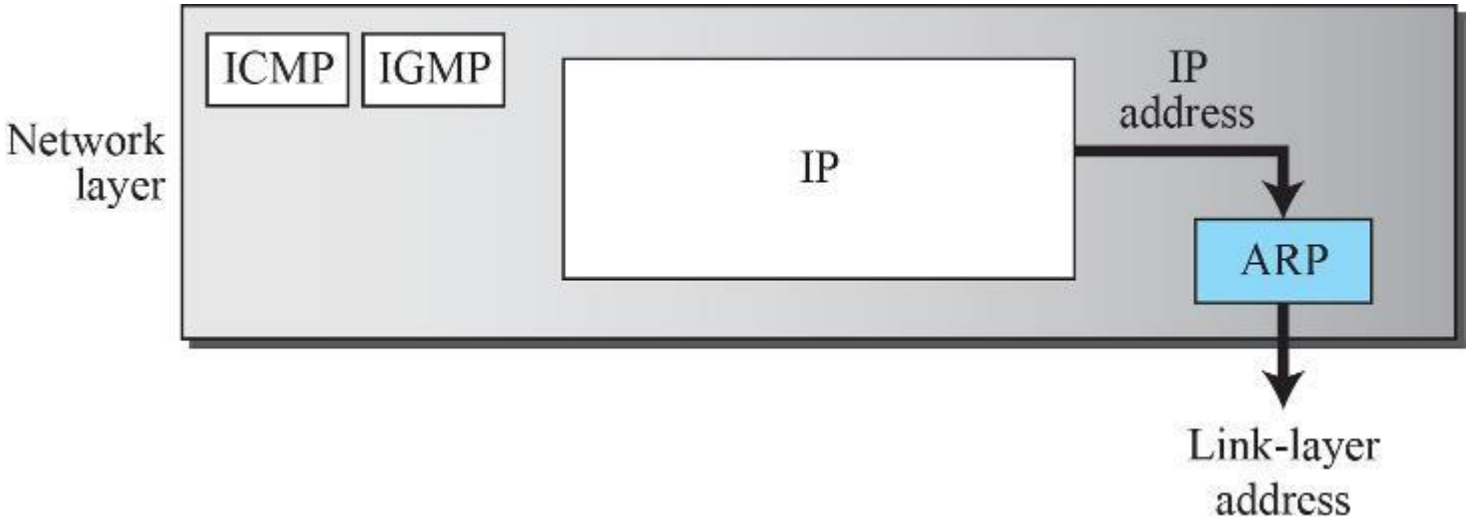
S. No	CSMA CD	CSMA CA
1.	It is the type of CSMA to detect the collision on a shared channel.	It is the type of CSMA to avoid collision on a shared channel.
2.	It is the collision detection protocol.	It is the collision avoidance protocol.
3.	It is used in 802.3 Ethernet network cable.	It is used in the 802.11 Ethernet network.
4.	It works in wired networks.	It works in wireless networks.
5.	It is effective after collision detection on a network.	It is effective before collision detection on a network.
6.	Whenever a data packet conflicts in a shared channel, it resends the data frame.	Whereas the CSMA CA waits until the channel is busy and does not recover after a collision.
7.	It minimizes the recovery time.	It minimizes the risk of collision.
8.	The efficiency of CSMA CD is high as compared to CSMA.	The efficiency of CSMA CA is similar to CSMA.

Protocol	Transmission behavior	Collision detection method	Efficiency	Use cases
Pure ALOHA	Sends frames immediately	No collision detection	Low	Low-traffic networks
Slotted ALOHA	Sends frames at specific time slots	No collision detection	Better than pure ALOHA	Low-traffic networks
CSMA/CD	Monitors medium after sending a frame, retransmits if necessary	Collision detection by monitoring transmissions	High	Wired networks with moderate to high traffic
CSMA/CA	Monitors medium while transmitting, adjusts behavior to avoid collisions	Collision avoidance through random backoff time intervals	High	Wireless networks with moderate to high traffic and high error rates

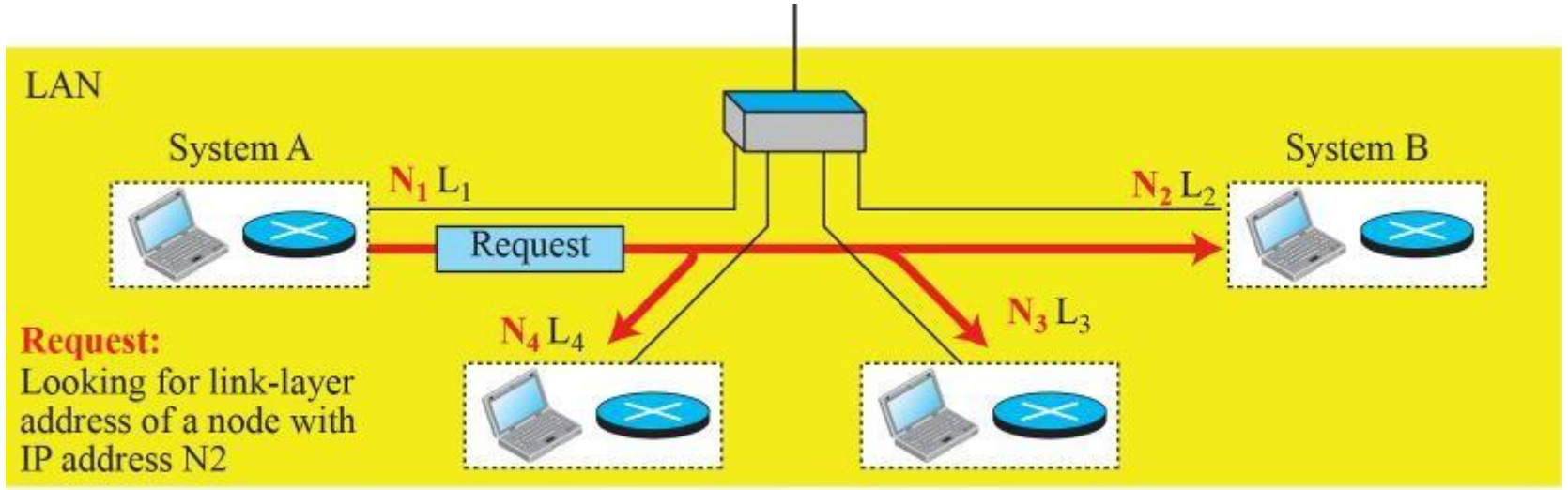
ARP

Address Resolution Protocol (ARP)

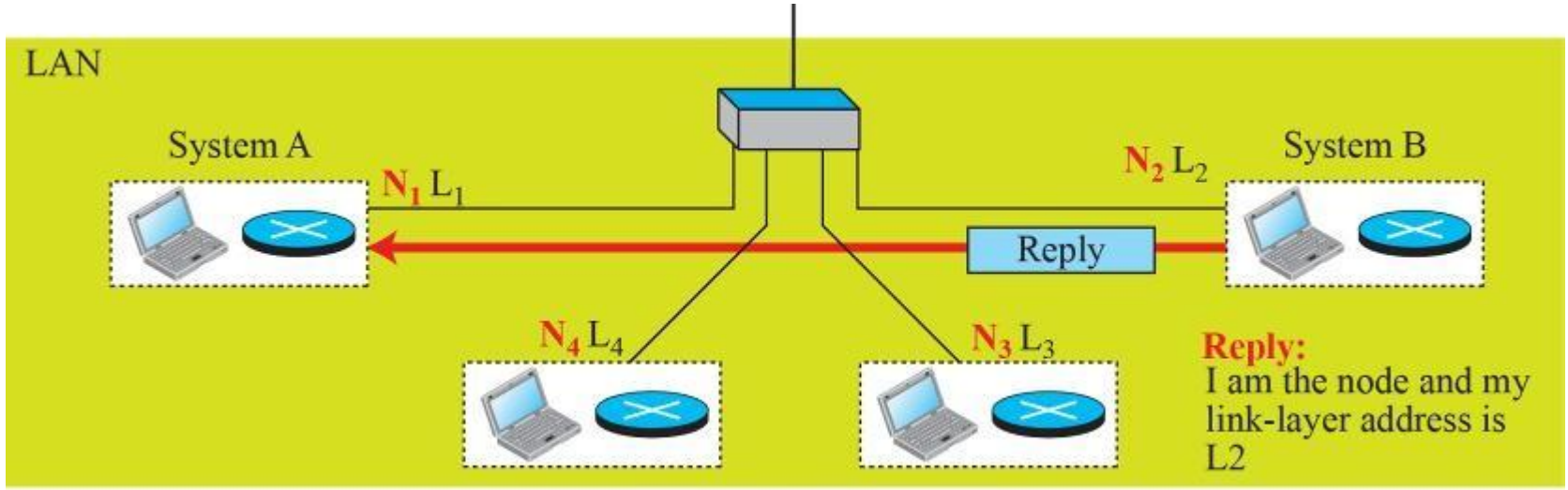
When a node wants to send an IP datagram to another node on the same link, it has the recipient's IP address. The source host also knows the IP address of the default router, and each router in the path obtains the next router's IP address from its forwarding table. However, to move a frame through a link, we need the link-layer address of the next node, not just its IP address. This is where the Address Resolution Protocol (ARP) comes in. ARP, an auxiliary protocol in the network layer, maps an IP address to a logical-link address. It takes an IP address from the IP protocol, translates it to the corresponding link-layer address, and passes it to the data-link layer.



When a host or router needs the link-layer address of another node on its network, it sends an ARP request packet. This packet contains the sender's link-layer and IP addresses, as well as the receiver's IP address. Since the sender doesn't know the receiver's link-layer address, the query is broadcast using the link-layer broadcast address. All nodes on the network receive and process the ARP request, but only the intended recipient replies with an ARP response containing its IP and link-layer addresses. The response is unicast directly to the requesting node. This process allows a node (e.g., A) with an IP packet for another node (e.g., B) to discover B's link-layer address dynamically. After receiving the ARP reply, A can use B's link-layer address for subsequent packet delivery.



a. ARP request is broadcast



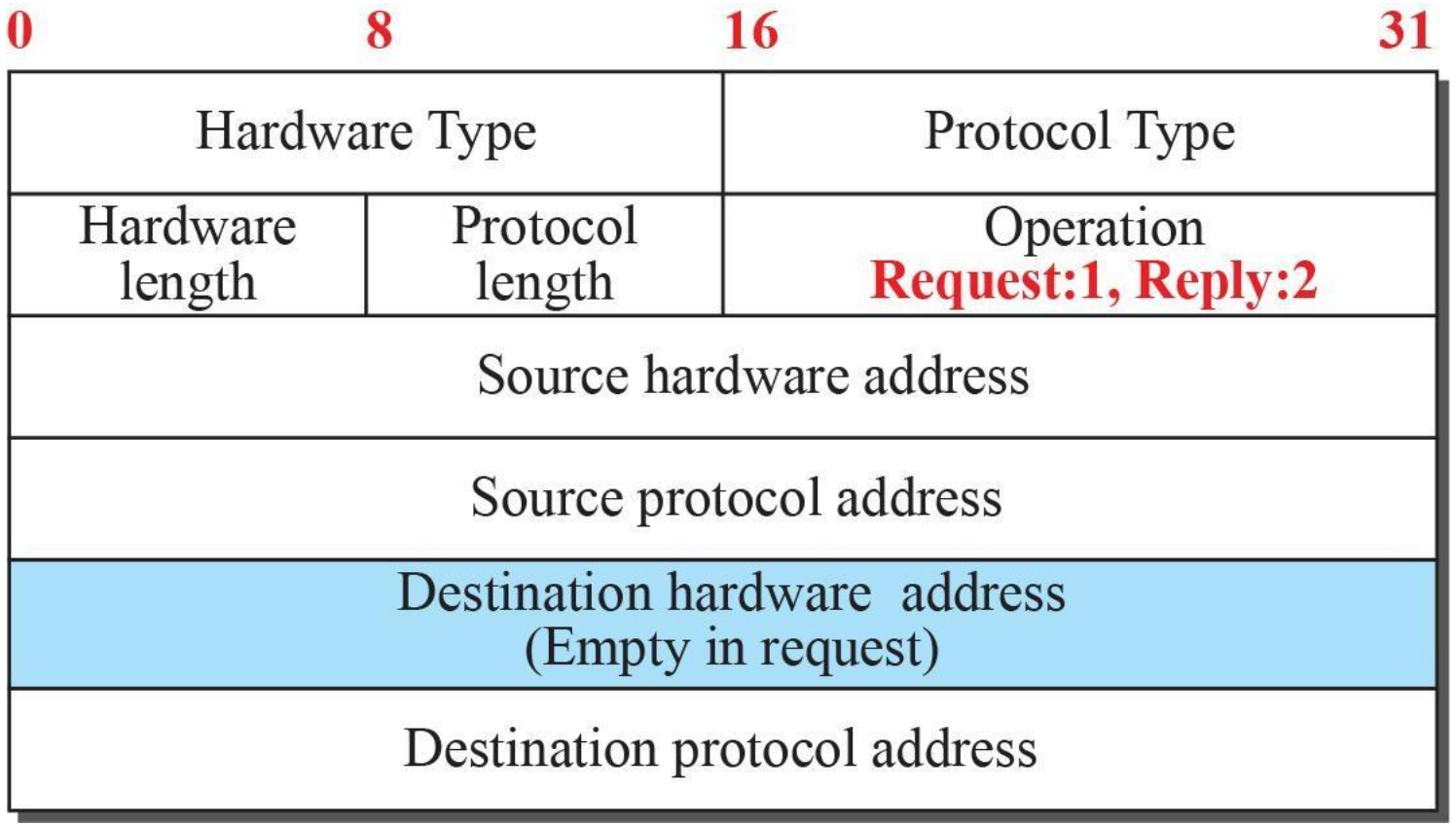
b. ARP reply is unicast

Packet Format

The ARP packet format, illustrated in Figure 5.48, comprises several fields. The hardware type field specifies the link-layer protocol type, with Ethernet assigned the type 1. The protocol type field indicates the network-layer protocol, with IPv4 represented by (0800)_16. The source hardware and source protocol addresses contain variable-length fields representing the sender's link-layer and network-layer addresses. Similarly, the destination hardware and destination protocol address fields define the receiver's link-layer and network-layer addresses. An ARP packet is directly encapsulated into a data-link frame, which requires a field indicating that the payload belongs to ARP, not the network-layer datagram.

Hardware: LAN or WAN protocol

Protocol: Network-layer protocol

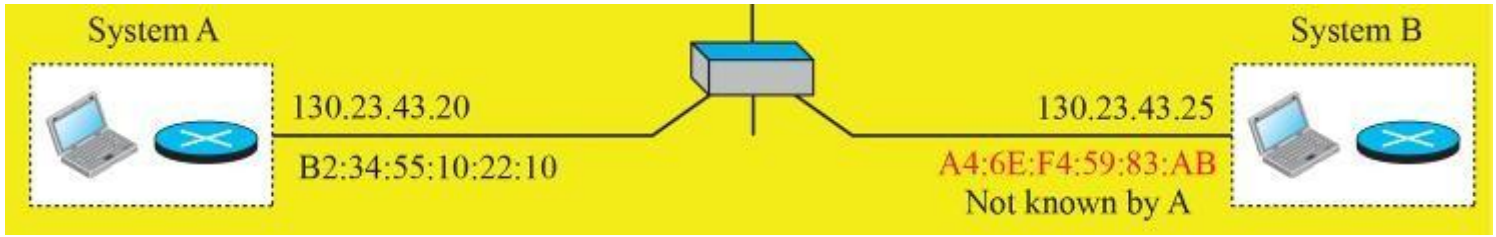


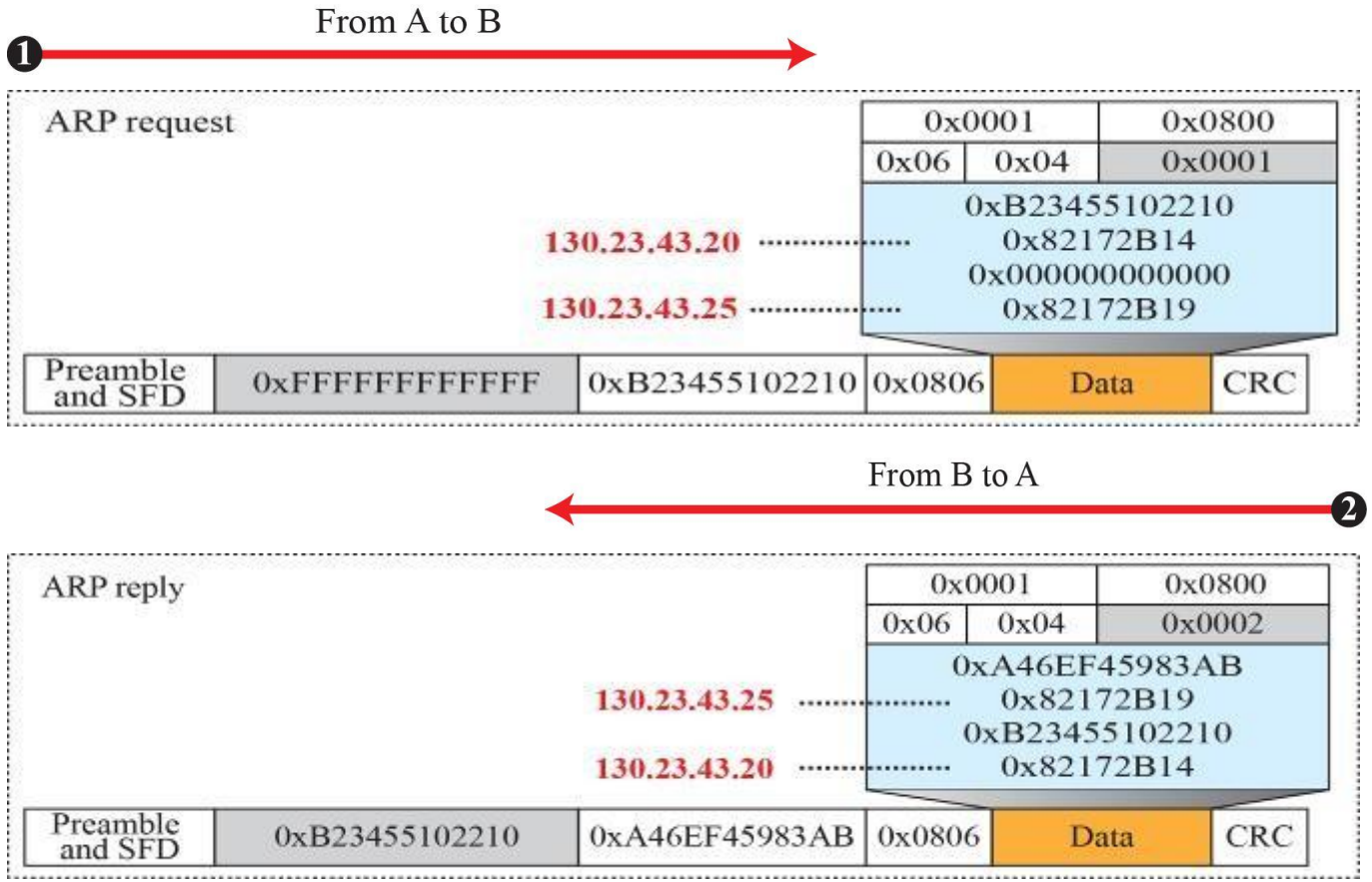
Example

A host with IP address N1 and MAC address L1 has a packet to send to another host with IP address N2 and physical address L2 (which is unknown to the first host). The two hosts are on the same network. Show the ARP request and reply packets encapsulated in Ethernet frames (see Figure on next page).

Solution

Figure on next page shows the ARP request and reply packets. Note that the ARP data field in this case is 28 bytes, and that the individual addresses do not fit in the 4-byte boundary. That is why we do not show the regular 4-byte boundaries for these addresses. Also note that the IP addresses are shown in hexadecimal.


System A
System B
130.23.43.20
B2:34:55:10:22:10
130.23.43.25
A4:6E:F4:59:83:AB
Not known by A


From A to B
1
ARP request
0x0001
0x0800
0x06
0x04
0x0001
0xB23455102210
0x82172B14
0x000000000000
0x82172B19
130.23.43.20
130.23.43.25
Preamble and SFD
0xFFFFFFFFFFFF
0xB23455102210
0x0806
Data
CRC
From B to A
2
ARP reply
0x0001
0x0800
0x06
0x04
0x0002
0xA46EF45983AB
0x82172B19
0xB23455102210
0x82172B14
130.23.43.25
130.23.43.20
Preamble and SFD
0xB23455102210
0xA46EF45983AB
0x0806
Data
CRC

Ethernet Frame format

WIRED LANS: ETHERNET PROTOCOL

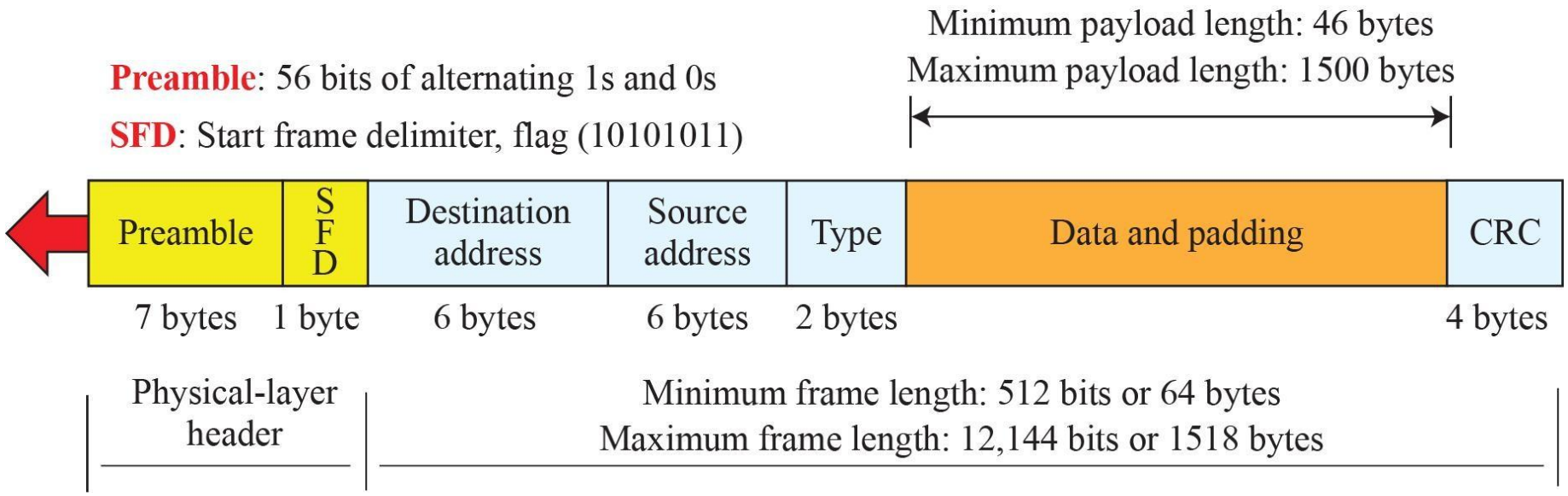
This chapter explores wired local and wide area networks (LANs and WANs) within the TCP/IP protocol suite. While LANs are designed for limited areas, they often connect to wider networks like the Internet. In the past, various LAN technologies existed, but Ethernet, with its adaptability and evolution across generations, became the dominant choice. Ethernet, developed in the 1970s, has undergone four generations, accommodating increasing transmission rates from 10 Mbps to 10 Gbps.

Standard Ethernet

We call the initial Ethernet technology, operating at 10 Mbps, Standard Ethernet. Despite subsequent advancements in Ethernet, certain features of Standard Ethernet have remained unchanged. Exploring this standard version will provide a foundation for understanding the evolution of the other three Ethernet technologies.

Frame Format

The Ethernet frame contains seven fields, as shown in Figure



- Preamble - The preamble is a 56-bit field with alternating 0s and 1s added at the physical layer of an Ethernet frame. It serves to **alert the receiving system about an incoming frame and helps synchronize its clock if needed.** This pattern provides an alert and a timing pulse, allowing stations to miss some bits at the frame's beginning. Importantly, the preamble is not formally considered part of the frame itself.
- Start frame delimiter (SFD) - The Start Frame Delimiter (SFD) is a 1-byte field with the pattern 10101011. **It signals the start of an Ethernet frame, serving as a final opportunity for synchronization. The last 2 bits (11)_2 indicate that the next field is the destination address.** Essentially, the SFD functions as a flag, marking the beginning of the variable-length Ethernet frame. Like the preamble, the SFD is added at the physical layer.

- Destination address (DA) - The Destination Address field is six bytes (48 bits) in an Ethernet frame, **holding the link-layer address of the intended recipient station or stations.** This address is crucial for addressing purposes, determining which station(s) should receive the packet. When the receiver identifies its own link-layer address, a multicast address for a group it belongs to, or a broadcast address, it extracts the data from the frame and forwards it to the upper-layer protocol specified by the type field's value.
- Source address (SA) - This field is also six bytes and contains the link-layer address of the sender of the packet. We will discuss addressing shortly.
- Type - The **Type field in an Ethernet frame specifies the upper-layer protocol encapsulated within the frame**, such as IP, ARP, OSPF, etc. Similar to the protocol field in a datagram or a port number in a segment or user datagram, it facilitates multiplexing and demultiplexing—determining how to handle the encapsulated data based on the specified upper-layer protocol.

- Data - The Data **field in an Ethernet frame carries data from upper-layer protocols, ranging from a minimum of 46 to a maximum of 1500 bytes.** If the upper-layer data exceeds 1500 bytes, it must be fragmented across multiple frames. Conversely, if the data is less than 46 bytes, it's padded with extra 0s. A padded data frame is delivered to the upper-layer protocol without removing the padding, placing the responsibility on the upper layer to manage padding. The upper-layer protocol needs to be aware of the data length, often facilitated by a length-defining field, as seen in a datagram.
- CRC - The **last field contains error detection information**, in this case a CRC-32. The CRC is calculated over the addresses, types, and data field. If the receiver calculates the CRC and finds that it is not zero (corruption in transmission), it discards the frame.