ANALOG AND DIGITAL SIGNAL

Analog and Digital Data

Data can be analog or digital. The term **analog data** refers to information that is continuous; **digital data** refers to information that has discrete states.

For example, an analog clock that has hour, minute, and second hands gives information in a continuous form; the movements of the hands are continuous. On the other hand, a digital clock that reports the hours and the minutes will change suddenly from 8:05 to 8:06.

Analog data, such as the sounds made by a human voice, take on continuous values. When someone speaks, an analog wave is created in the air. This can be captured by a microphone and converted to an analog signal or sampled and converted to a digital signal.

Digital data take on discrete values. For example, data are stored in computer memory in the form of Os and 1s. They can be converted to a digital signal or modulated into an analog signal for transmission across a medium.

Analog and Digital Signals

Like the data they represent, signals can be either analog or digital. An analog signal has infinitely many levels of intensity over a period of time.

A digital signal, on the other hand, can have only a limited number of defined values. Although each value can be any number, it is often as simple as 1 and O.

Periodic and Non periodic Signals

Both analog and digital signals can take one of two forms: periodic or non periodic.

A periodic signal completes a pattern within a measurable time frame, called a period, and repeats that pattern over subsequent identical periods. The completion of one full pattern is

called a cycle.

A non periodic signal changes without exhibiting a pattern or cycle that repeats over time.

Both analog and digital signals can be periodic or non periodic. In data communications, we commonly use periodic analog signals (because they need less bandwidth) and non periodic digital signals (because they can represent variation in data).

Periodic Analog Signals

Periodic analog signals can be classified as simple or composite.

A simple periodic analog signal, a sine wave, cannot be decomposed into simpler signals.
A composite periodic analog signal is composed of multiple sine waves.

Sine Wave

The sine wave is the most fundamental form of a periodic analog signal. When we visualize it as a simple oscillating curve, its change over the course of a cycle is smooth and consistent, a continuous, rolling flow.

A sine wave can be represented by three parameters: the peak amplitude, the frequency, and the phase. These three parameters fully describe a sine wave.

Peak Amplitude

The peak amplitude of a signal is the absolute value of its highest intensity, proportional to the energy it carries. For electric signals, peak amplitude is normally measured in volts.

Period and Frequency

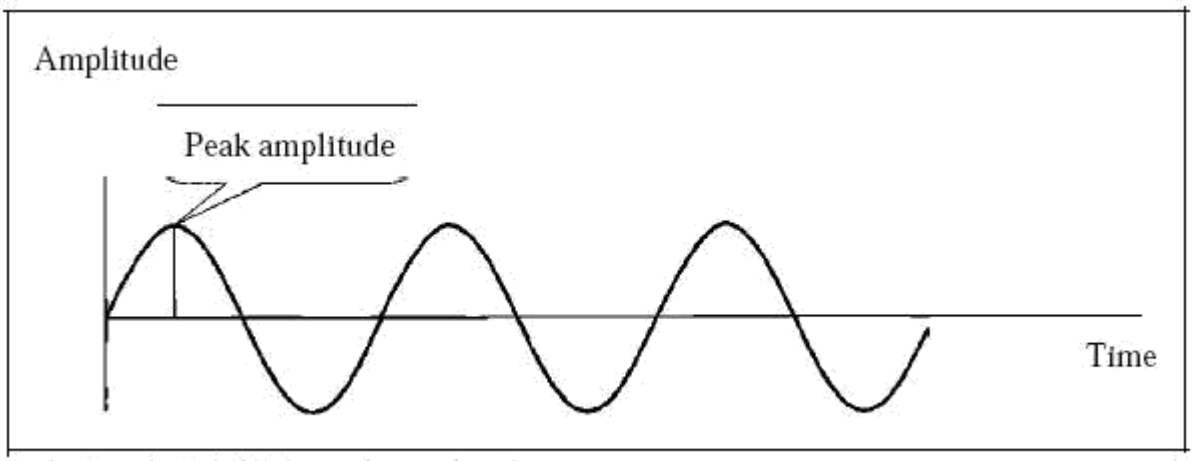
Period refers to the amount of time, in seconds, a signal needs to complete 1 cycle. Frequency refers to the number of periods in 1s.

Period is the inverse of frequency, and frequency is the inverse of period, as the following formulas show.

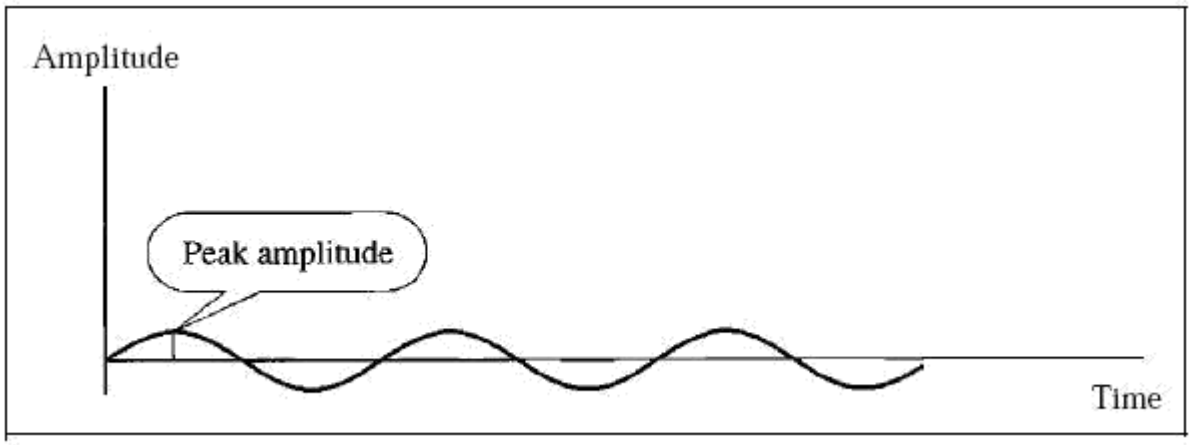
$f = 1/T$ and $T = 1/f$

Unit	Equivalent	Unit	Equivalent
Seconds (s)	1 s	Hertz (Hz)	1 Hz
Milliseconds (ms)	10^{-3} s	Kilohertz (kHz)	10^{3} Hz
Microseconds (µs)	10^{-6} s	Megahertz (MHz)	10^{6} Hz
Nanoseconds (ns)	10^{-9} s	Gigahertz (GHz)	10^{9} Hz

Picoseconds (ps) 10 s Terahertz (THz) 10 Hz



a. A signal with high peak amplitude



b. A signal with low peak amplitude

Composite Signals -

12 12

Simple sine waves have many applications in daily life. We can send a single sine wave to carry electric energy from one place to another. For example, the power company sends a single sine wave with a frequency of 60 Hz to distribute electric energy to houses and businesses.

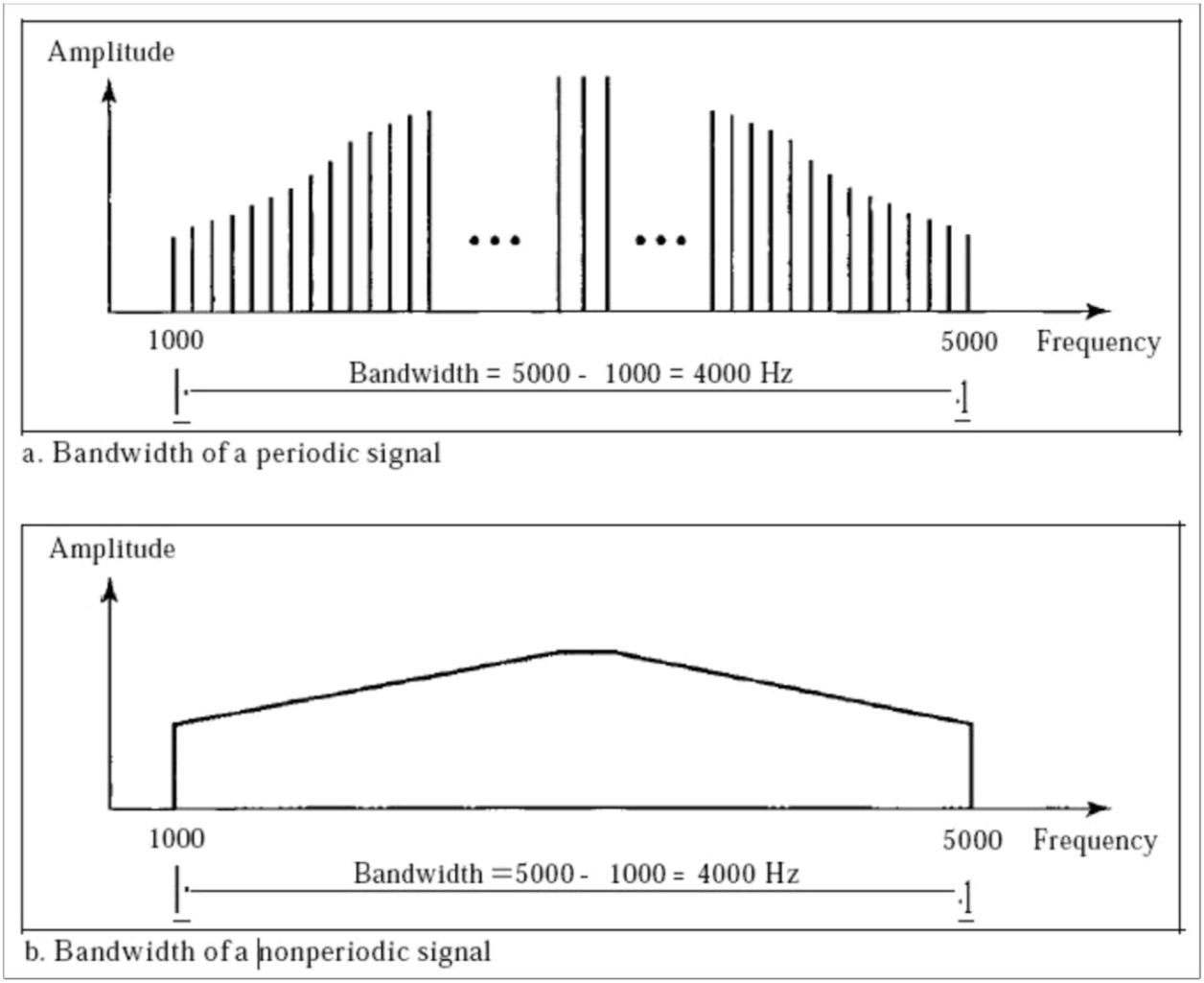
A single frequency sine wave is not useful in data communications; we need to send a composite signal, a signal made of many simple sine waves.

According to Fourier analysis, any composite signal is a combination of simple sine waves with different frequencies, amplitudes, and phases.

Bandwidth

The range of frequencies contained in a composite signal is its **bandwidth**. The bandwidth is normally a difference between two numbers. For example, if a composite signal contains frequencies between 1000 and 5000, its bandwidth is 5000 - 1000, or 4000.

The bandwidth of a composite signal is the difference between the highest and the lowest frequencies contained in that signal.



a. Bandwidth of a periodic signal

b. Bandwidth of a nonperiodic signal

Example :

A periodic signal has a bandwidth of 20 Hz. The highest frequency is 60 Hz. What is the lowest frequency? Draw the spectrum if the signal contains all frequencies of the same amplitude.

Solution:

Digital Signal

In addition to being represented by an analog signal, information can also be represented by a digital signal. For example, 1 can be encoded as a positive voltage and a 0 as zero voltage. A digital signal can have more than two levels.

Bit Rate

Most digital signals are non periodic, and thus period and frequency are not appropriate characteristics. Another term-bit rate is used to describe digital signals.

> Let f_h be the highest frequency, f_z the lowest frequency, and B the bandwidth.
> Then $B = f_h - f_z$
> $20 = 60 - f_z$ or $f_z = 60 - 20 = 40$ Hz

transmission medium. We can define something similar for a digital signal: the bit length. The bit length is the distance one bit occupies on the transmission medium.

Bit length=propagation speed x bit duration

Data Rate Limits

A very important consideration in data communications is how fast we can send data, in bits per second over a channel. Data rate depends on three factors:

1. The bandwidth available
2. The level of the signals we use
3. The quality of the channel (the level of noise)

Two theoretical formulas were developed to calculate the data rate: one by Nyquist for a noiseless channel, another by Shannon for a noisy channel.

Noiseless Channel: Nyquist Bit Rate

For a noiseless channel, the Nyquist bit rate formula defines the theoretical maximum bit rate

BitRate = 2 x bandwidth x l0g2 L

In this formula, bandwidth is the bandwidth of the channel, L is the number of signal levels used to

represent data, and Bit Rate is the bit rate in bits per second.

Example:

Consider a noiseless channel with a bandwidth of 3000 Hz transmitting a signal with two signal levels. The maximum bit rate can be calculated as

BitRate =2 x 3000 x $\log_2$ 2 =6000 bps

Noisy Channel: Shannon Capacity

In reality, we cannot have a noiseless channel; the channel is always noisy. In 1944, Claude Shannon introduced a formula, called the Shannon capacity, to determine the theoretical highest data rate for a noisy channel:

Capacity =bandwidth X $\log_2$ (1 +SNR)

In this formula, bandwidth is the bandwidth of the channel, SNR is the signal-to-noise ratio, and capacity is the capacity of the channel in bits per second. Note that in the Shannon formula there is no indication of the signal level, which means that no matter how many levels we have, we cannot achieve a data rate higher than the capacity of the channel. In other words, the formula defines a characteristic of the channel, not the method of transmission.

Example:

Consider an extremely noisy channel in which the value of the signal -to-noise ratio is almost zero. In other words, the noise is so strong that the signal is faint. For this channel the capacity C is calculated as $C=B \log_2 (1 + SNR) =B \log_2 (1 + 0) =B \log_2 1 = B \times 0 =0$

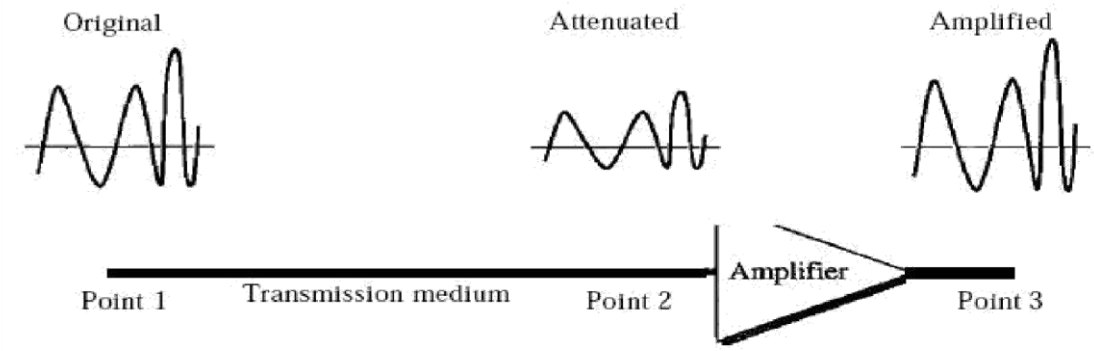
This means that the capacity of this channel is zero regardless of the bandwidth. In other words, we cannot receive any data through this channel.

Transmission Impairment

Signals travel through transmission media, which are not perfect. The imperfection causes signal impairment. This means that the signal at the beginning of the medium is not the same as the signal at the end of the medium. What is sent is not what is received. Three causes of impairment are **attenuation, distortion**, and **noise**.

Attenuation

Attenuation means a loss of energy. When a signal, simple or composite, trav els through a medium, it loses some of its energy in overcoming the resistance of the medium. That i s why a wire carrying electric signals gets warm, if not hot, after a while. Some of the electrical ene rgy in the signal is converted to heat. To compensate for this loss, amplifiers are used to amplify the signal.



Decibel

To show that a signal has lost or gained strength, engineers use the unit of the decibel.

The decibel (dB) measures the relative strengths of two signals or one signal a t two different points.

Note that the decibel is negative if a signal is attenuated and positive if a signal is amplified.

$$\mathbf{dB} = 10 \log_{10} \frac{P_2}{P_1}$$

Variables PI and P2 are the powers of a signal at points 1 and 2, respectively.

Example:

Suppose a signal travels through a transmission medium and its power is reduced to one-half. Find the attenuation (loss of power).

Solution:

dB=10 log (P/2P)= -3 dB

Example:

A signal travels through an amplifier, and its power is increased 10 times. Find the amplification (gain of power).

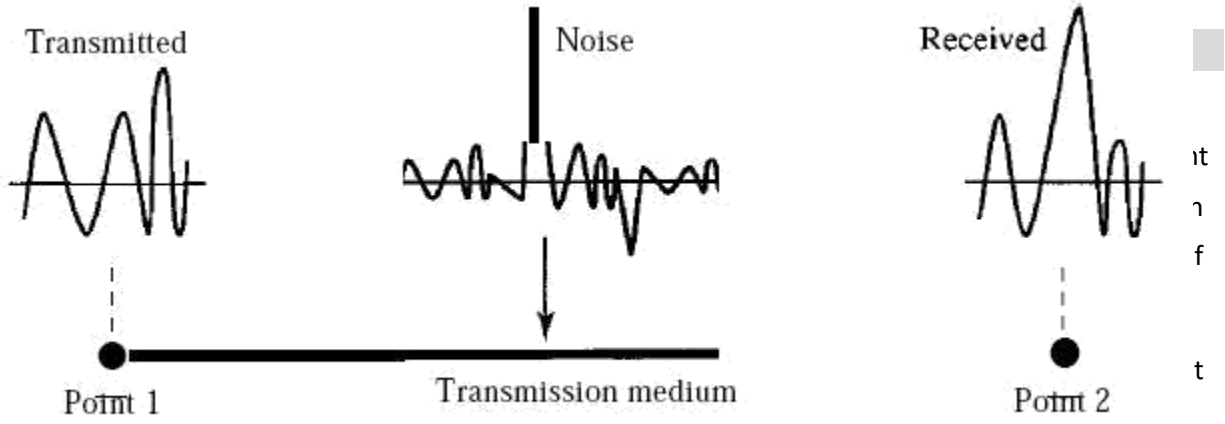
Solution:

dB=10 log (10P/P)= 10 dB

Noise

Noise is another cause of impairment. Several types of noise, such as thermal noise, induced noise, crosstalk, and impulse noise, may corrupt the signal. Thermal noise is the ra ndom motion of electrons in a wire which creates an extra signal not originally sent by the transmitter . Induced noise comes from sources such as motors and appliances.

D



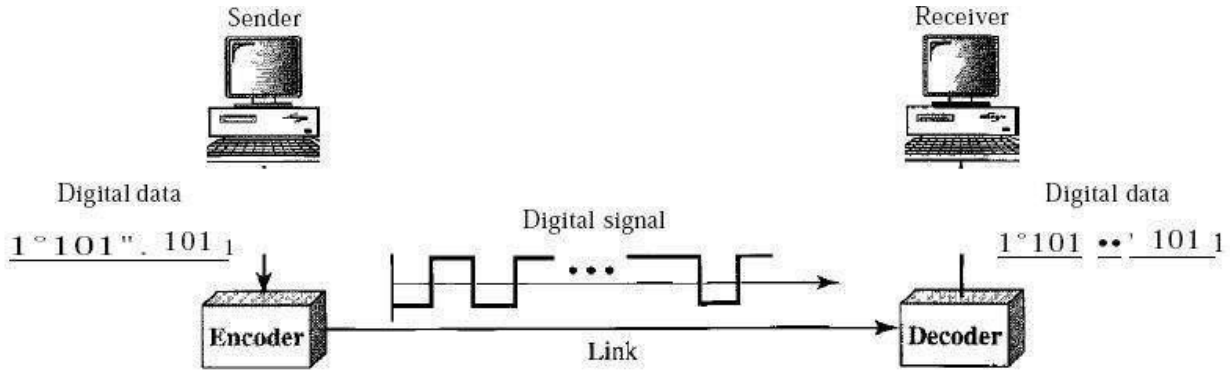
To find the theoretical bit rate limit, we need to know the ratio of the signal power to the noise power. The signal-to-noise ratio is defined as:

SNR = average signal power/ average noise power

Because SNR is the ratio of two powers, it is often described in decibel units, $\mathbf{SNR_{dB}}$, defined as

DIGITAL TRANSMISSION

We can represent digital data by using digital signals. The conversion involves three techniques: line coding, block coding, and scrambling. Line coding is always needed. Block coding and scrambling mayor may not be needed.



Line Coding

Line coding is the process of converting digital data to digital signals.

We assume that data, in the form of text, numbers, graphical images, audio, or video, are stored in computer memory as sequences of bits.

Line coding converts a sequence of bits to a digital signal. At the sender, digital data are

encoded into a digital signal; at the receiver, the digital data are recreated by decoding the digital signal.

We can formulate the relationship between data rate and signal rate as:

S =c xNx 1/r baud

where N is the data rate (bps); c is the case factor, which varies for each case; S is the number of signal elements; and r is the previously defined factor.

Example

A signal is carrying data in which one data element is encoded as one signal element (r = 1). If the bit rate is 100 kbps, what is the average value of the baud rate if c is between 0 and 1?

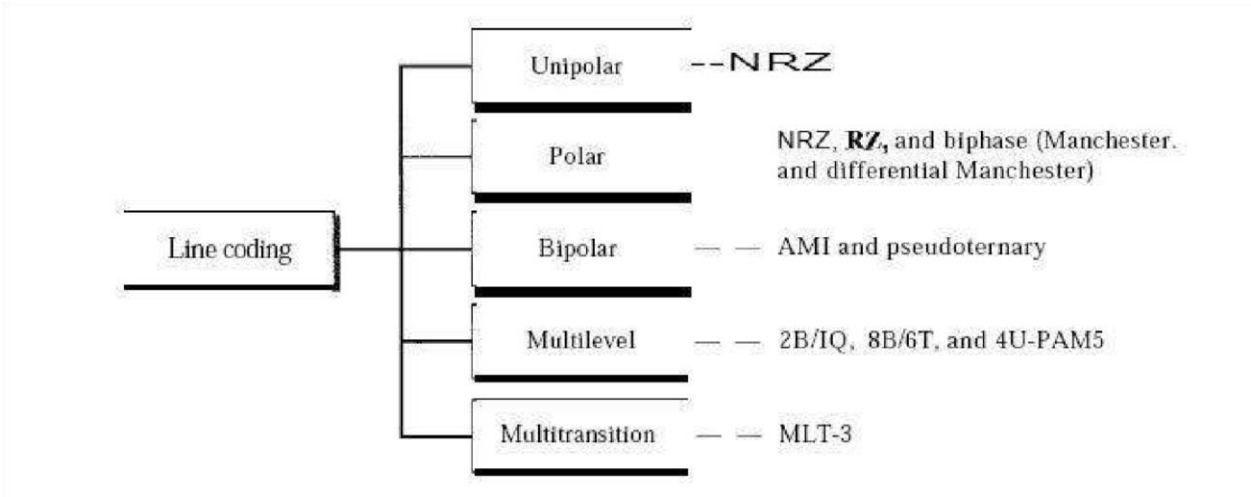
Solution:

We assume that the average value of c is ½. The baud rate is then

S =c x N x1/r = ½ x 100, 000 x 1= 50, 000 = 50 Kbaud

Line Coding Schemes

We can roughly divide line coding schemes into five broad categories, as shown in below:

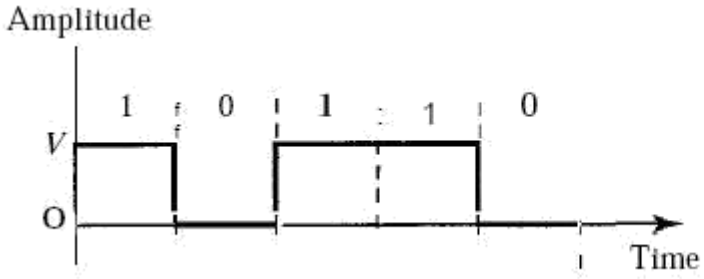


Unipolar Scheme

In a unipolar scheme, all the signal levels are on one side of the time axis, either above or below. **NRZ**

(Non-Return-to-Zero): Traditionally, a unipolar scheme was designed as a non-return-to-zero (NRZ) scheme in which the positive voltage defines bit 1 and the zero voltage defines bit 0. ***It is called NRZ***

because the signal does not return to zero at the middle of the bit.



$$\frac{1}{2}V^2 + \frac{1}{2}(0)^2 = \frac{1}{2}V^2$$

Nonnalized power

Polar Schemes

In polar schemes, the voltages are on the both sides of the time axis. For example, the voltage level for 0 can be positive and the voltage level for 1 can be negative.

Non-Return-to-Zero (NRZ):

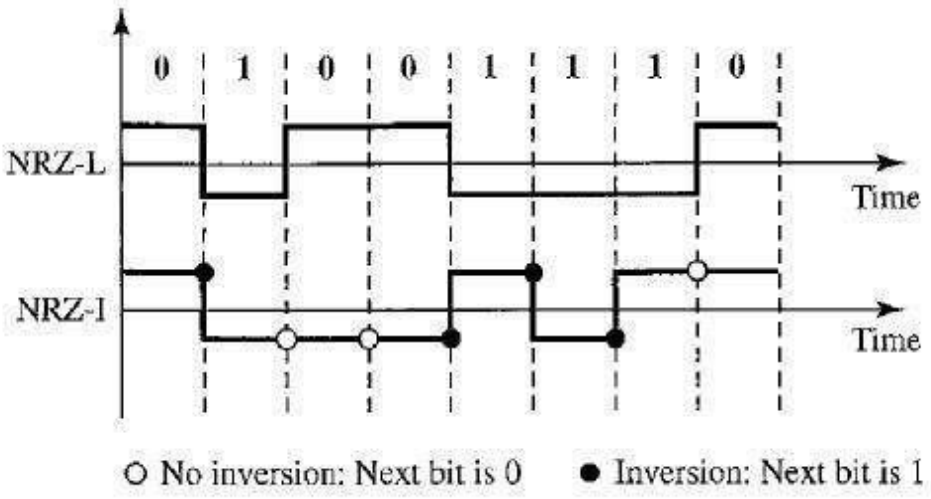
In polar NRZ encoding, we use two levels of voltage amplitude. We can have two versions of polar NRZ: NRZ-Land NRZ-I.

In the first variation, **NRZ-L** (NRZ-Level), the level of the voltage determines the value of the bit. In the second variation, **NRZ-I** (NRZ-Invert), the change or lack of change in the level of the

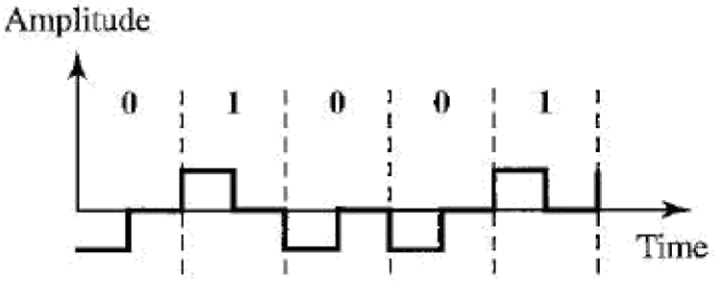
voltage determines the value of the bit**. If there is no change, the bit is 0; if there is a change, the bit is 1.**

The synchronization problem (sender and receiver clocks are not synchronized) also exists in both schemes. Again, this problem is more serious in **NRZ-L** than in **NRZ-I**. While a long sequence of 0's can cause a problem in both schemes, a long sequence of 1s affects only **NRZ-L**.

Another problem with NRZ-L occurs when there is a sudden change of polarity in the system. NRZ-I does not have this problem. Both schemes have an average signal rate of **N/2 Bd**.



Return to Zero (RZ) The main problem with NRZ encoding occurs when the sender and receiver clocks are not synchronized. The receiver does not know when one bit has ended and the next bit is starting. One solution is the return-to-zero (RZ) scheme, which uses three values: positive, negative, and zero**. In RZ, the signal changes not between bits but during the bit**.

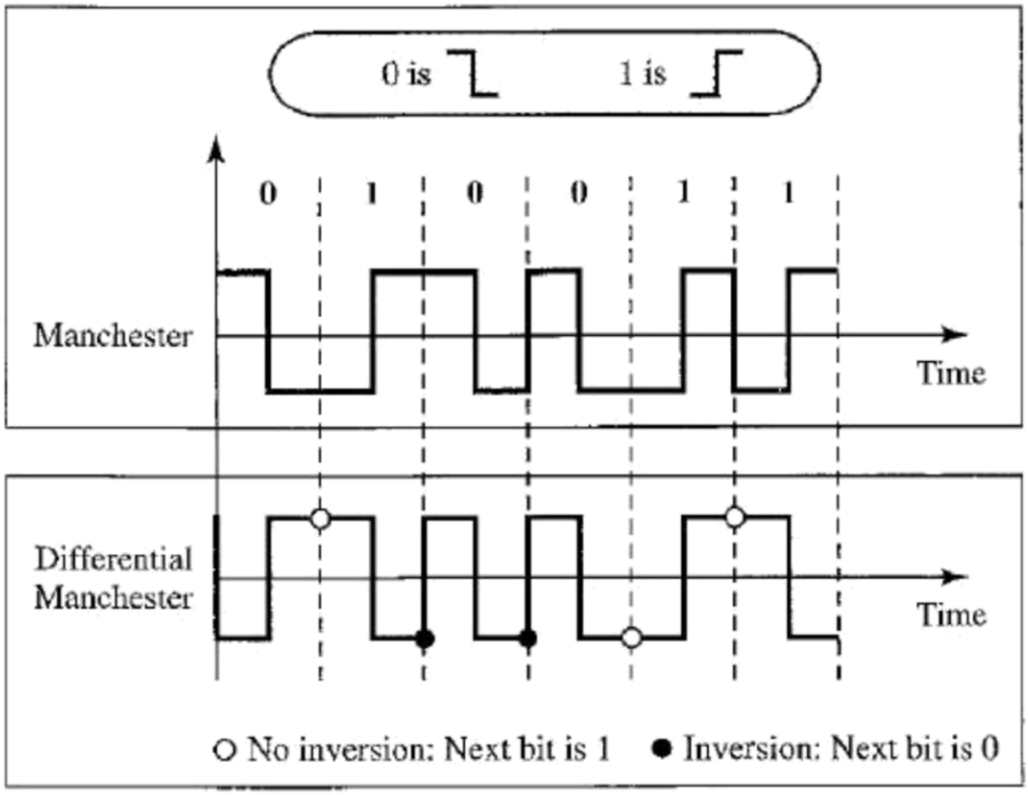


Biphase: Manchester and Differential Manchester

The idea of RZ (transition at the middle of the bit) and the idea of NRZ-L are combined into the Manchester scheme.

In Manchester encoding, the duration of the bit is divided into two halves. The voltage remains at one level during the first half and moves to the other level in the second half. The transition at the middle of the bit provides synchronization.

Differential Manchester, on the other hand, combines the ideas of RZ and NRZ-I. There is always a transition at the middle of the bit, but the bit values are determined at the beginning of the bit. If the next bit is 0, there is a transition; if the next bit is 1, there is none.



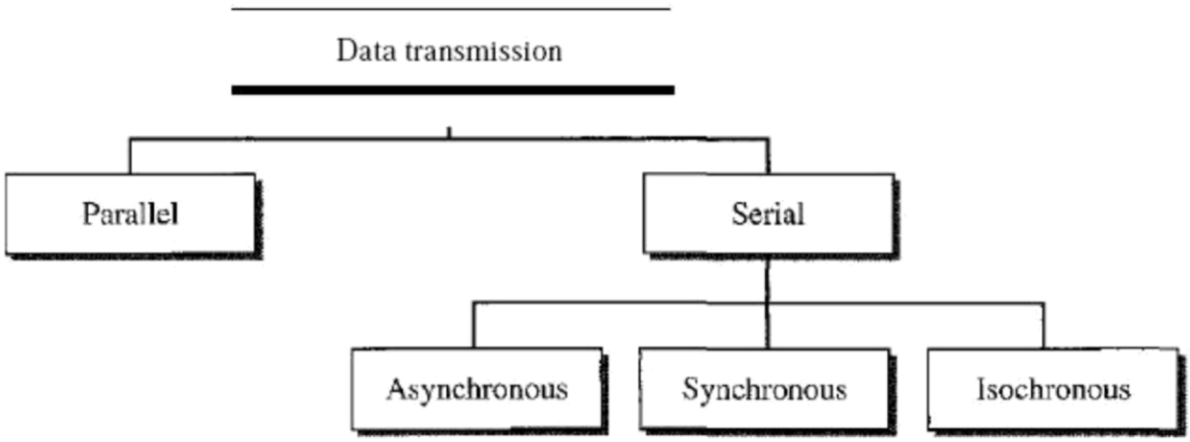
The minimum bandwidth of Manchester and differential Manchester is **2** times that of **NRZ**.

Block Coding

We need redundancy to ensure synchronization and to provide some kind of inherent error de tecting. Block coding can give us this redundancy and improve the performance of line coding. In ge neral, block coding changes a block of m bits into a block of n bits, where n is larger than m. Block coding is referred to as an mB/ nB encoding technique.

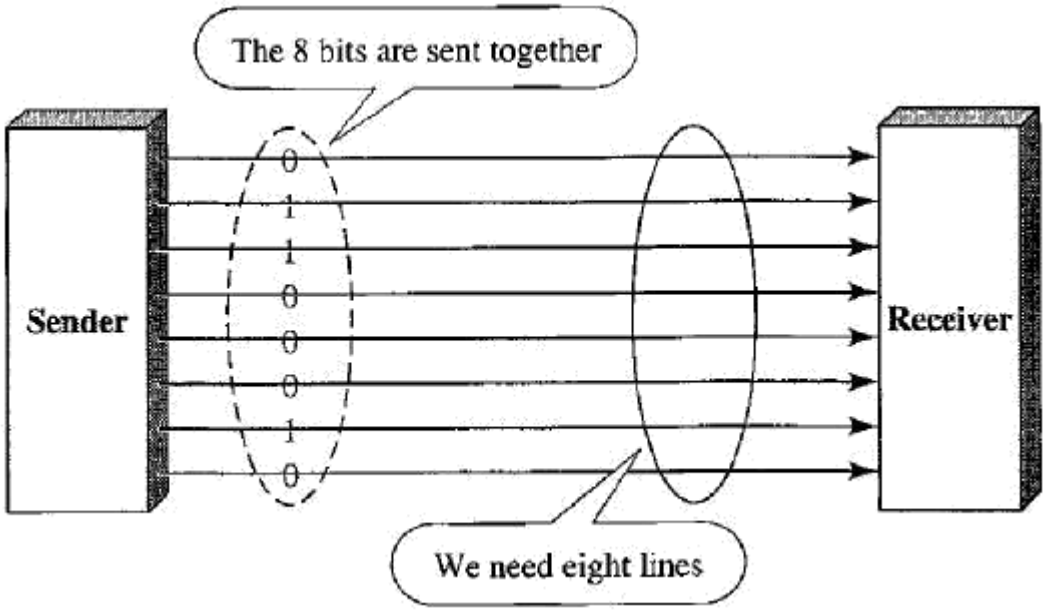
Transmission Modes

Of primary concern when we are considering the transmission of data fro m one device to another is the wiring, and of primary concern when we are considering the wiring is the da ta stream. Do we send 1 bit at a time; or do we group bits into larger groups and, if so, how? The transmission of binary data across a link can be accomplished in either parallel o r serial mode. In parallel mode, multiple bits are sent with each clock tick. In serial mode, 1 bit is sent with each clock tick. While t here is only one way to send parallel data, there are three subclasses of serial transmission: asynchronous, synchronous, and isochronous.



Parallel Transmission

Binary data, consisting of 1s and 0s, may be organized into groups of n bits each. Com puters produce and consume data in groups of bits much as we conceive of and use spoken language in the for m of words rather than letters. By grouping, we can send data n bits at a time instea d of 1. This is called parallel transmission.



Advantage:

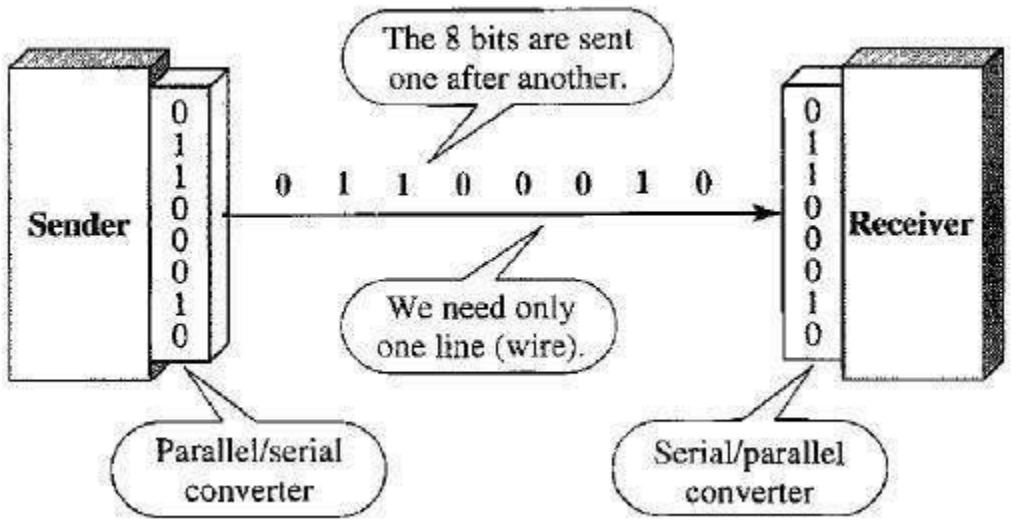
The advantage of parallel transmission is speed. All else being equal, paralle l transmission can increase the transfer speed by a factor of *n* over serial transmission.

Disadvantage:

Parallel transmission requires *n* communication lines just to transmit the data stream. Because this is expensive, parallel transmission is usually limited to short distances.

Serial Transmission

In serial transmission one bit follows another, so we need only one communication channel rather than n to transmit data between two communicating devices.

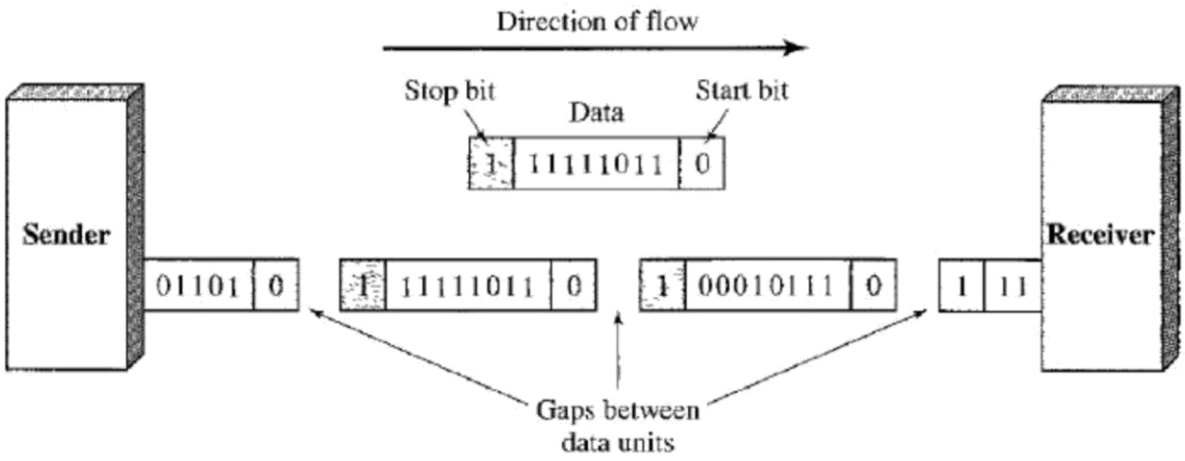


The advantage of serial over parallel transmission is that with only one communication channel, serial transmission reduces the cost of transmission over parallel by roughly a factor of n.

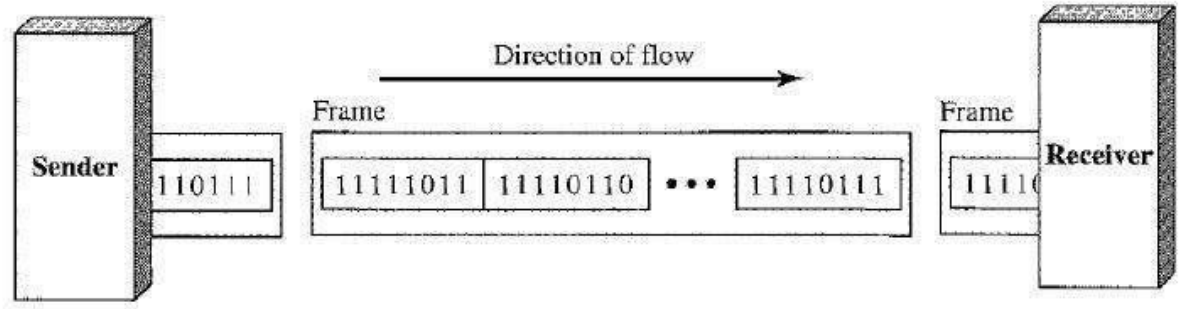
Since communication within devices is parallel, conversion devices are required at the interface between the sender and the line (parallel-to-serial) and between the line and the receiver (serial-to-parallel).

Serial transmission occurs in one of three ways: **asynchronous**, **synchronous**, and **isochronous**.

In asynchronous transmission, we send 1 **start bit (0)** at the beginning and 1 or more **stop bits (1)** at the end of each byte. There may be a gap between each byte.



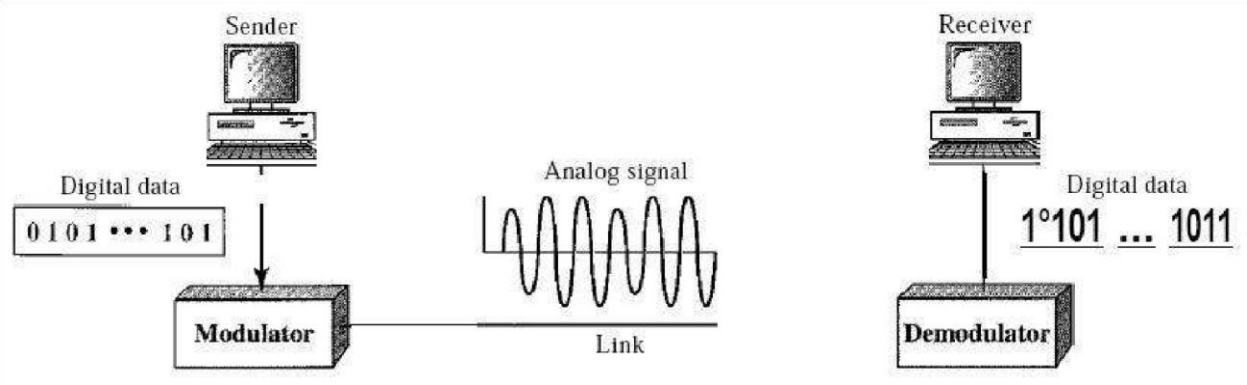
In synchronous transmission , we send bits one after another without start or stop bits or gaps. It is the responsibility of the receiver to group the bits.



The **isochronous transmission** guarantees that the data arrive at a fixed rate. In real -time audio and video, in which uneven delays between frames are not acceptable, synchronous transmission fails. For example, TV images are broadcast at the rate of 30 images per second; they must be viewed at the same rate. If each image is sent by using one or more f rames, there should be no delays between frames.

ANALOG TRANSMISSION

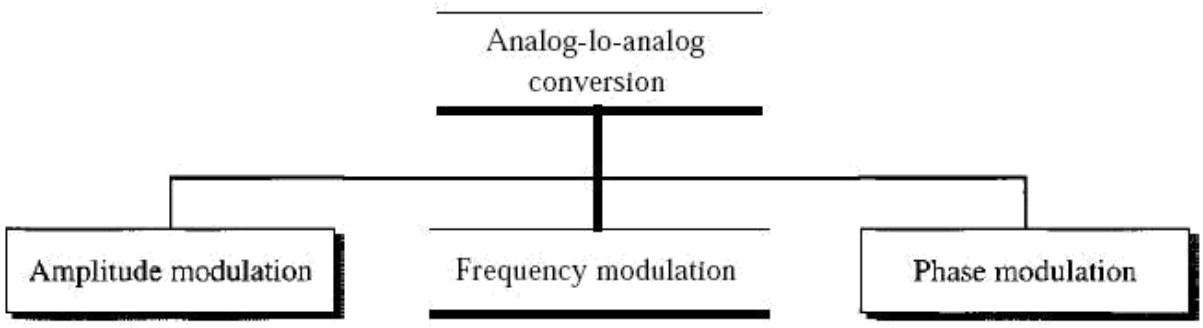
Digital-to-analog conversion is the process of changing one of the characteristics of an analog signal based on the information in digital data.



Analog-to-analog conversion, or analog modulation, is the representation of analog information by an analog signal. One may ask why we need to modulate an analog signal; it is already analog. Modulation is needed if the medium is band pass in nature or if only a band pass channel is available to us. An example is radio. The government assigns a narrow bandwidth to each radio station. The analog signal produced by each station is a low-pass signal, all in the same range. To be able to listen to different stations, the low-pass signals need to be shifted,

each to a different range.

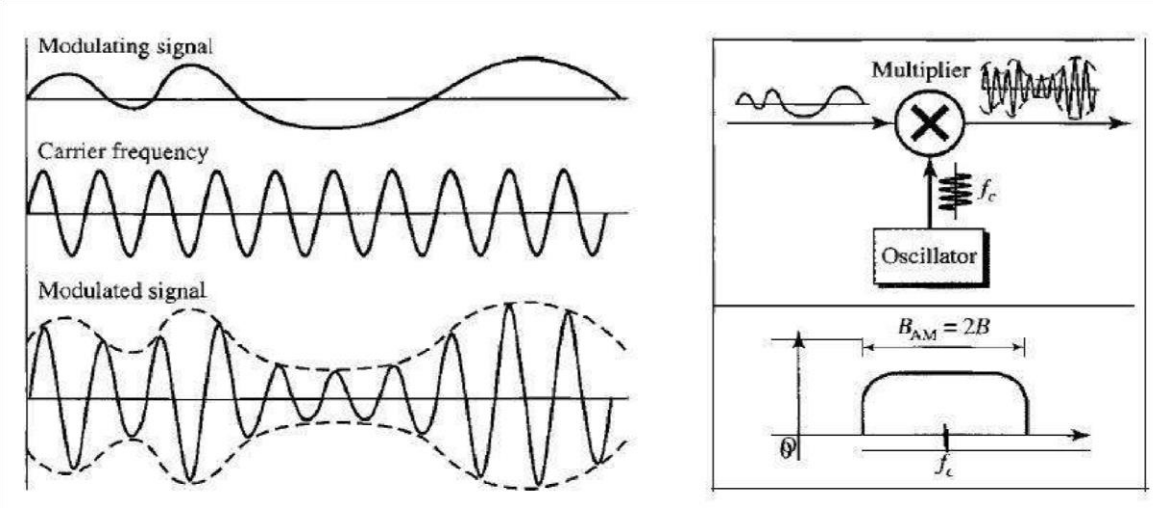
Analog-to-analog conversion can be accomplished in three ways: amplitude modulation (AM), frequency modulation (FM), and phase modulation (PM).



Amplitude Modulation

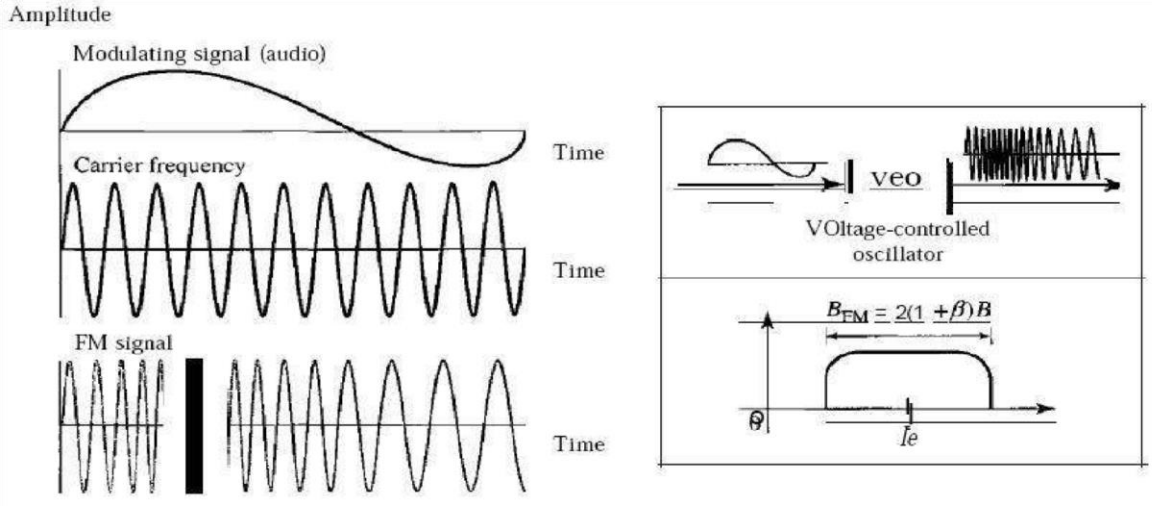
In AM transmission, the carrier signal is modulated so that its amplitude varies with the changing amplitudes of the modulating signal. The frequency and phase of the carrier remain the same; only the amplitude changes to follow variations in the information. Below Figure shows how this concept works.

The modulating signal is the envelope of the carrier.



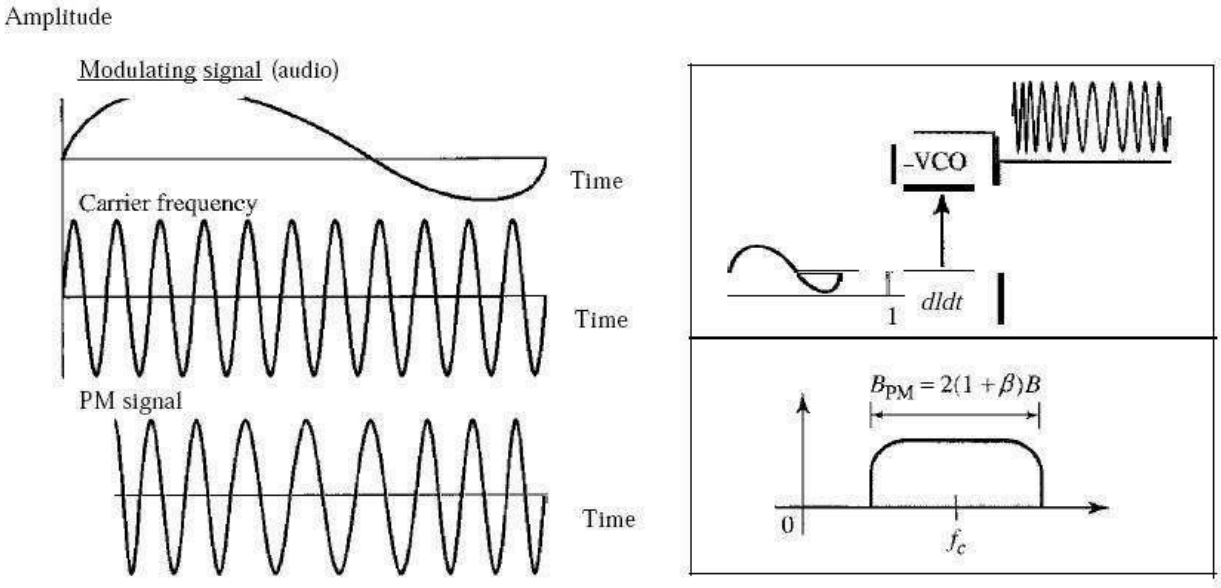
Frequency Modulation

In FM transmission, the frequency of the carrier signal is modulated to follow the changing voltage level (amplitude) of the modulating signal. The peak amplitude and phase of the carrier signal remain constant, but as the amplitude of the information signal changes, the frequency of the carrier changes correspondingly.



Phase Modulation

In PM transmission, the phase of the carrier signal is modulated to follow the changing voltage level (amplitude) of the modulating signal. The peak amplitude and frequency of the carrier signal remain constant, but as the amplitude of the information signal changes, the phase of the carrier changes correspondingly. In FM, the instantaneous change in the carrier frequency is proportional to the amplitude of the modulating signal; in PM the instantaneous change in the carrier frequency is proportional to the derivative of the amplitude of the modulating signal.



Multiplexing

Whenever the bandwidth of a medium linking two devices is greater than the bandwidth needs of the devices, the link can be shared.

Multiplexing is the set of techniques that allows the simultaneous transmission of multiple signals across a single data link.

As data and telecommunications use increases, so does traffic. We can accommodate this increase by continuing to add individual links each time a new channel is needed; or we can

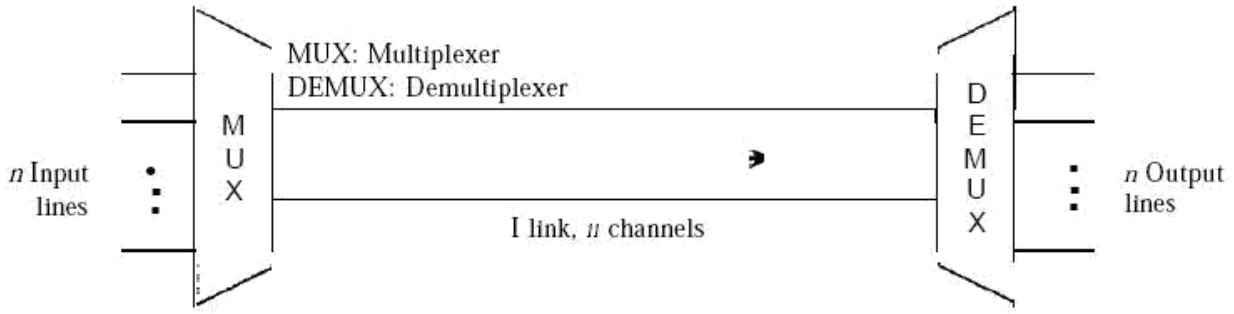
install higher-bandwidth links and use each to carry multiple signals.

In a multiplexed system, n lines share the bandwidth of one link.

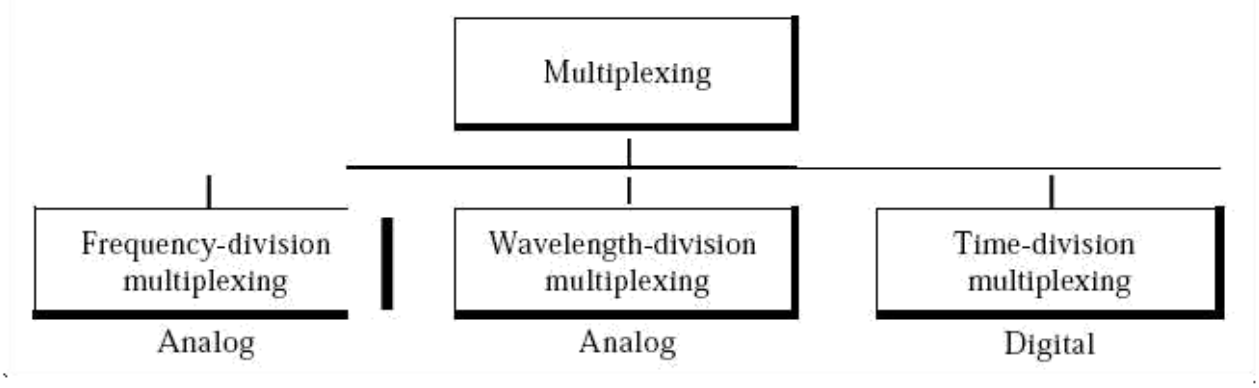
The lines on the left direct their transmission streams to a multiplexer (MUX), which combines them into a single stream (many-to-one).

At the receiving end, that stream is fed into a de-multiplexer (DEMUX), which separates the stream back into its component transmissions (one-to-many) and directs them to their corresponding lines. In the figure, the word link refers to the physical path.

The word channel refers to the portion of a link that carries a transmission between a given pair of lines. One link can have many (n) channels.



There are three basic multiplexing techniques: frequency-division multiplexing, wavelengthdivision multiplexing, and time-division multiplexing. The first two are techniques designed for analog signals, the third, for digital signals.



Frequency-Division Multiplexing

Frequency-division multiplexing (FDM) is an analog technique that can be applied when the bandwidth of a link (in hertz) is greater than the combined bandwidths of the signals to be transmitted.

In FDM, signals generated by each sending device modulate different carrier frequencies. These modulated signals are then combined into a single composite signal that can be transported by the link.

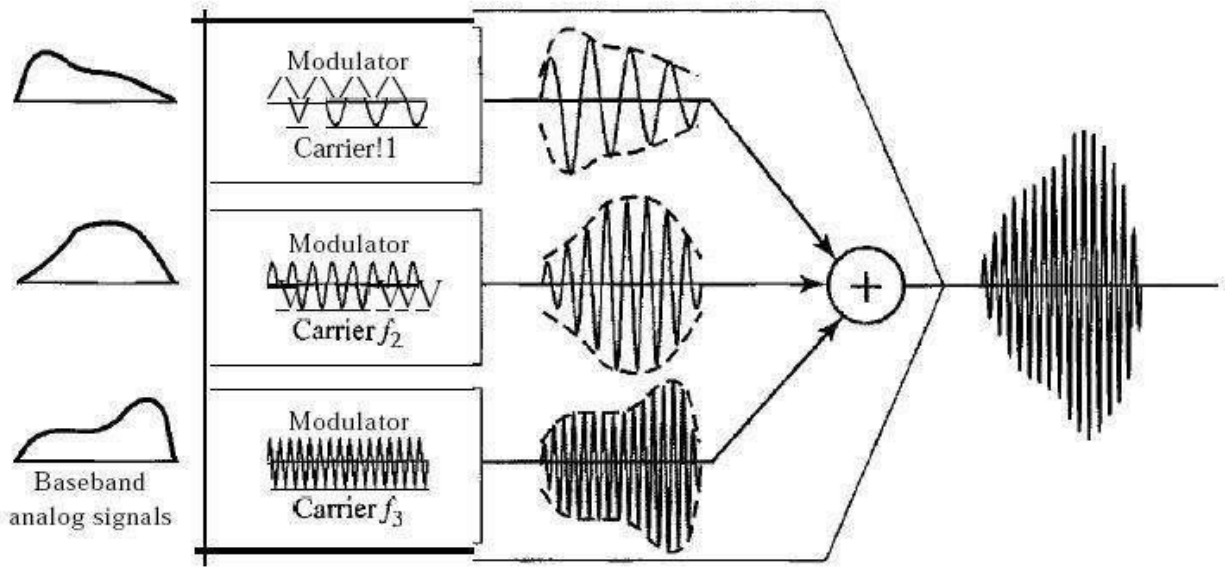
Carrier frequencies are separated by sufficient bandwidth to accommodate the modulated signal.

These bandwidth ranges are the channels through which the various signals travel. Channels can be separated by strips of unused bandwidth-guard bands-to prevent signals from overlapping.

In addition, carrier frequencies must not interfere with the original data frequencies.

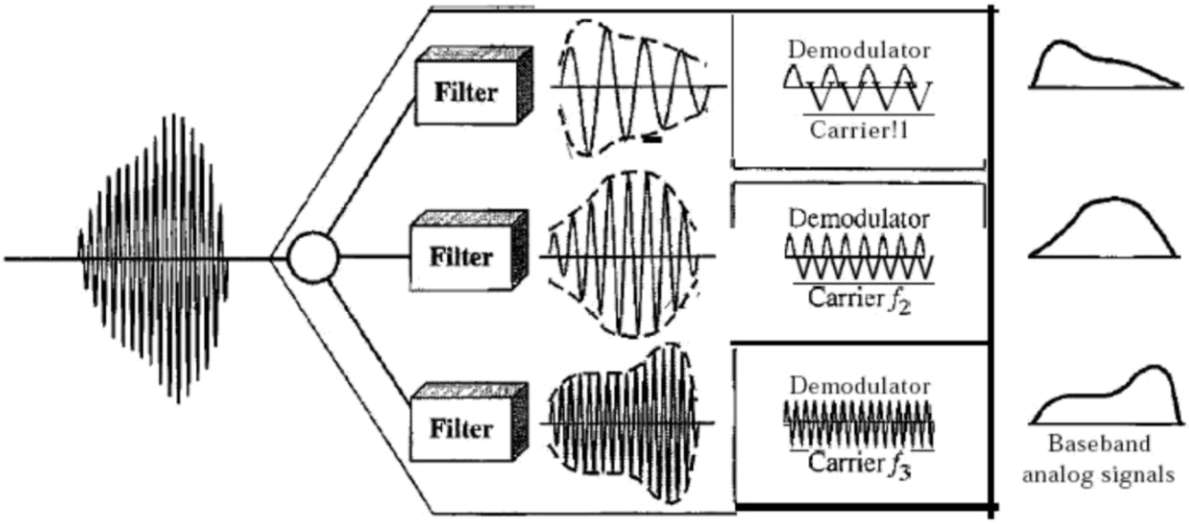
Multiplexing Process

Each source generates a signal of a similar frequency range. Inside the multiplexer, these similar signals modulate different carrier frequencies f1, f2, and f3). The resulting modulated signals are then combined into a single composite signal that is sent out over a media link that has enough bandwidth to accommodate it.



Demultiplexing Process

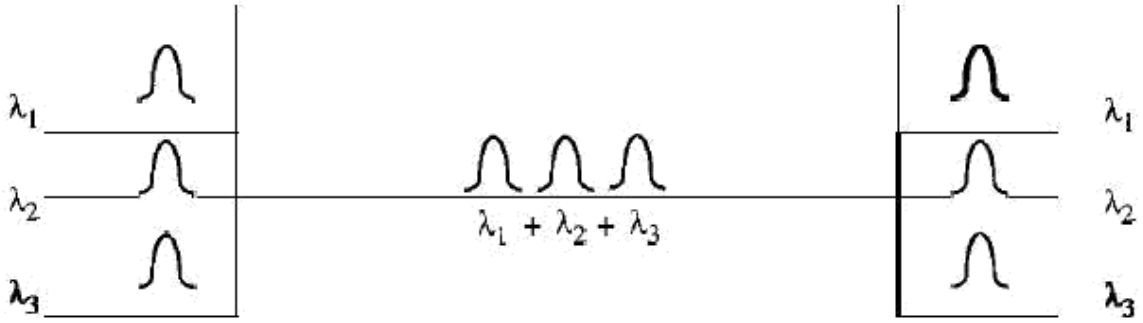
The de-multiplexer uses a series of filters to decompose the multiplexed signal into its constituent component signals. The individual signals are then passed to a demodulator that separates them from their carriers and passes them to the output lines.



Wavelength-Division Multiplexing

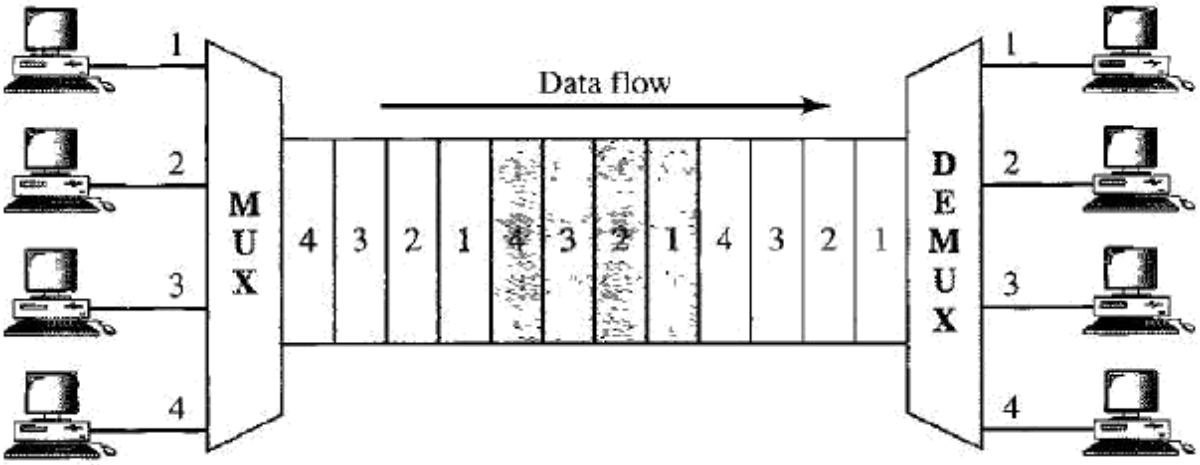
Wavelength-division multiplexing (WDM) is designed to use the high-data-rate capability of fiber-optic cable. The optical fiber data rate is higher than the data rate of metallic transmission cable. Using a fiber -optic cable for one single line wastes the available bandwidth. Multiplexing allows us to combine several lines into one.

WDM is conceptually the same as FDM, except that the multiplexing and de-multiplexing involve optical signals transmitted through fiber-optic channels. The idea is the same: We are combining different signals of different frequencies. The difference is that the frequencies are very high.



Time-Division Multiplexing

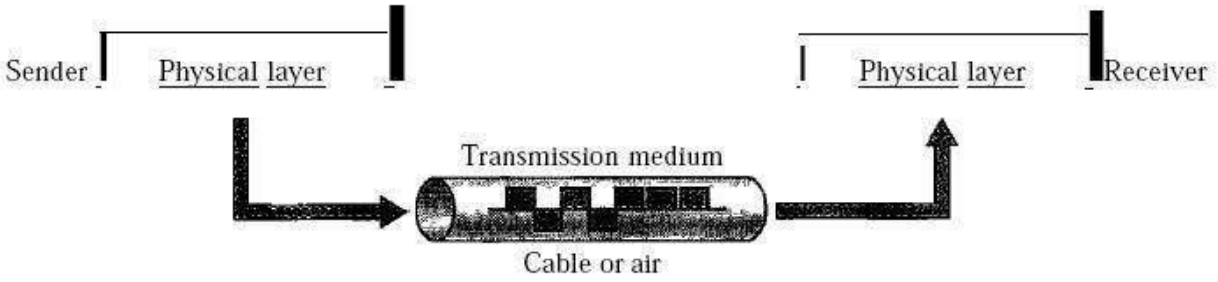
Time-division multiplexing (TDM) is a digital process that allows several connections to share the high bandwidth of a link Instead of sharing a portion of the bandwidth as in FDM, time is shared. Each connection occupies a portion of time in the link.



TRANSMISSION MEDIA

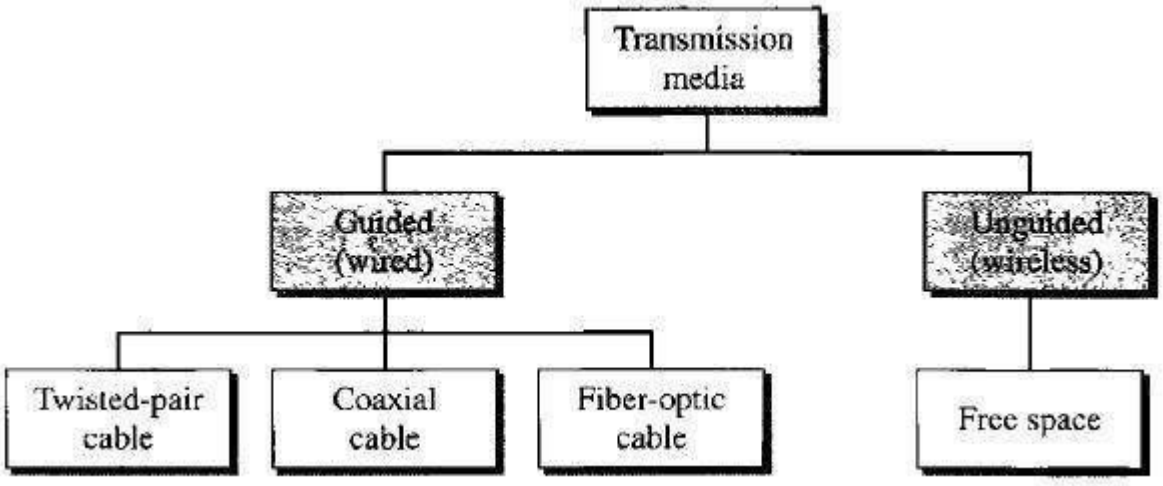
A transmission **medium** can be broadly defined as anything that can carry information from a source to a destination.

For example, the transmission medium for two people having a dinner conversation is the air. The air can also be used to convey the message in a smoke signal or semaphore. For a written message, the transmission medium might be a mail carrier, a truck, or an airplane.



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. Below Figure shows this taxonomy.



Guided Media

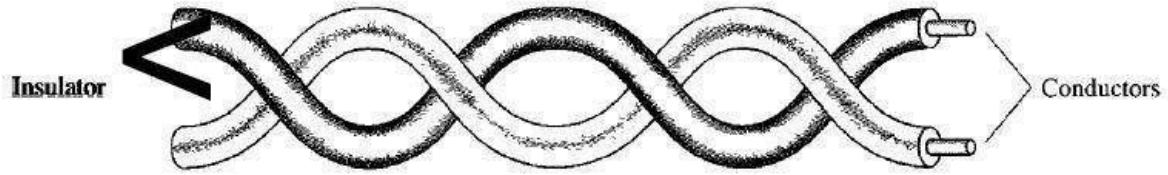
Guided media, which are those that provide a conduit from one device to another, include twisted-pair cable, coaxial cable, and fiber-optic cable.

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.

Optical fiber 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, as shown in below figure.



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.

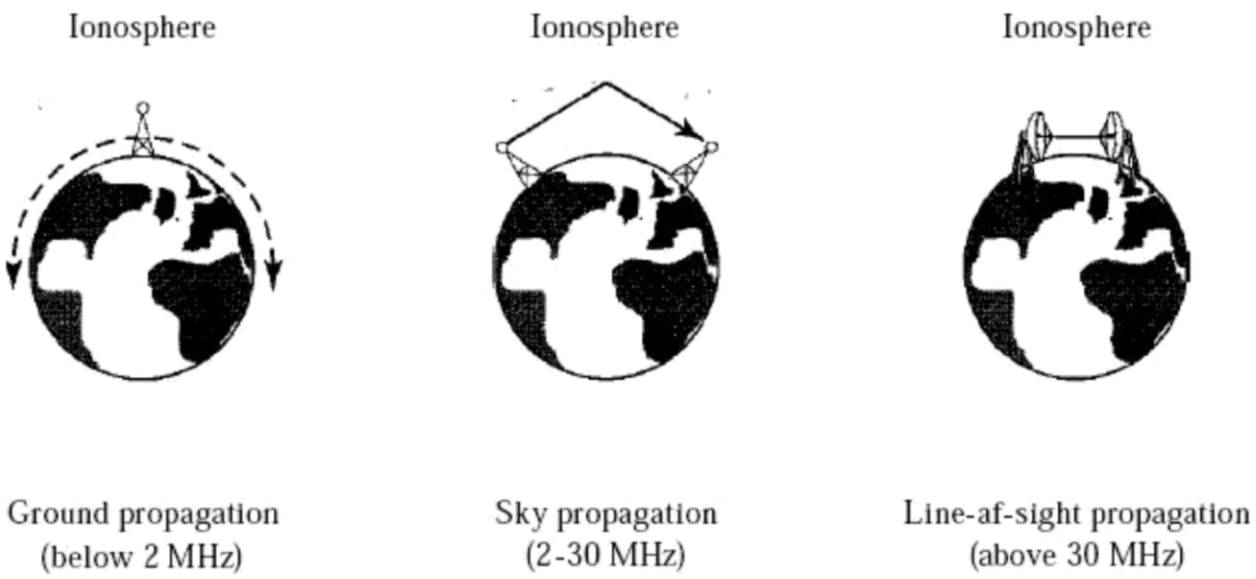
In addition to the signal sent by the sender on one of the wires, interference (noise) and crosstalk may affect both wires and create unwanted signals.

Unshielded Versus Shielded Twisted-Pair Cable

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).

STP cable has a metal foil or braided mesh covering that encases each pair of insulated conductors. Although metal casing improves the quality of cable by preventing the penetration of **noise** or crosstalk, it is **bulkier** and **more expensive**.

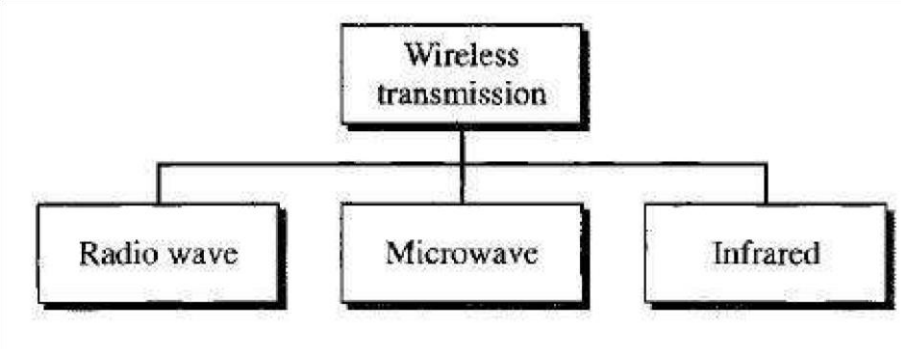


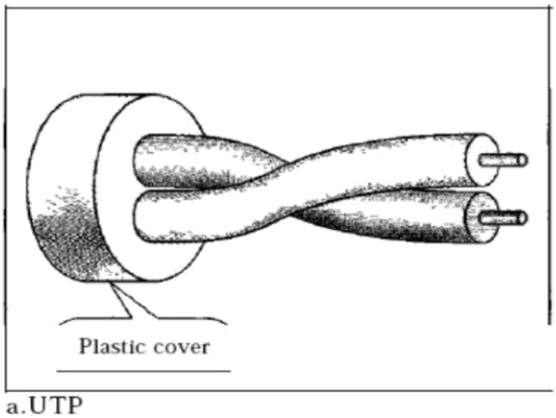
In **ground propagation**, radio waves travel through the lowest portion of the atmosphere, hugging the earth.

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.

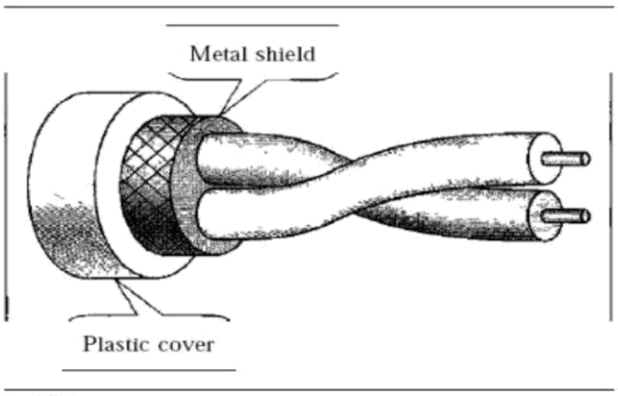
In **line-or-sight propagation**, very high-frequency signals are transmitted in straight lines directly from antenna to antenna.

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





a.UTP



b.STP

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.

Unguided Media: Wireless

Unguided media transport electromagnetic waves without using a physical conduc tor. 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 capa ble of receiving them.

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

Radio Waves

Although there is no clear-cut demarcation between radio waves and microwaves, 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**.

Radio waves, for the most part, are omni-directional. 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 omni-directional 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.

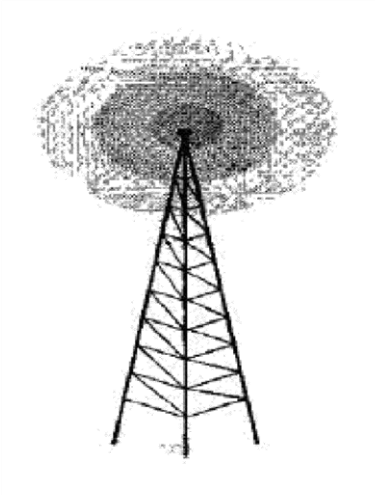
Radio waves, particularly those waves that propagate in the sky mode, can travel long distances. This makes radio waves a good candidate for long-distance broadcasting such as AM radio.

Omni directional Antenna

Radio waves use omni directional antennas that send out signals in all directions. Based on the wavelength, strength, and the purpose of transmission, we can have several types of antennas.

Applications

The omni directional characteristics of radio waves make them useful for multicasting, in which there is one sender but many receivers. AM and FM radio, television, maritime radio, cordless phones, and paging are examples of multicasting.



Microwaves

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

Microwaves are unidirectional.

When an antenna transmits microwave waves, 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.

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 curvatures 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 long distance communication.

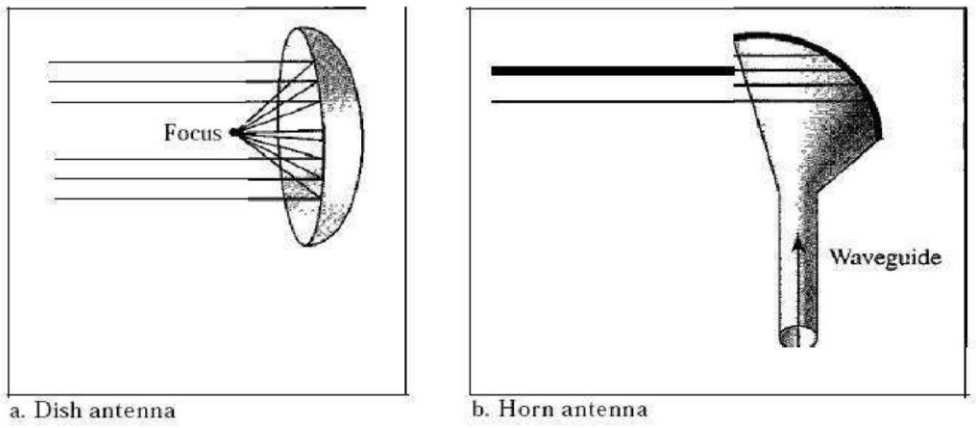
Very high-frequency microwaves cannot penetrate walls. This characteristic can be a disadvantage if receivers are inside buildings.

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

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

Unidirectional Antenna

Microwaves need unidirectional antennas that send out signals in one direction. Two types of antennas are used for microwave communications: the parabolic dish and the home.



a. Dish antenna b. Horn antenna

Applications

Microwaves, due to their unidirectional properties, are very useful when unicast (one-to-one) communication is needed between the sender and the receiver.

They are used in cellular phones, satellite networks and wireless LANs.

Infrared

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.

Applications

The infrared band, almost 400 THz, has an excellent potential for data transmission. Such a wide bandwidth can be used to transmit digital data with a very high data rate.

ERROR DETECTION AND CORRECTION

Data can be corrupted during transmission. Some applications require that errors be detected and corrected.

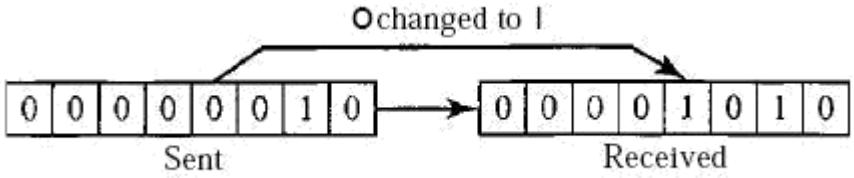
Types of Errors

Whenever bits flow from one point to another, they are subject to unpredictable changes because of interference. This interference can change the shape of the signal. Errors are of two types:

Single-Bit Error

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

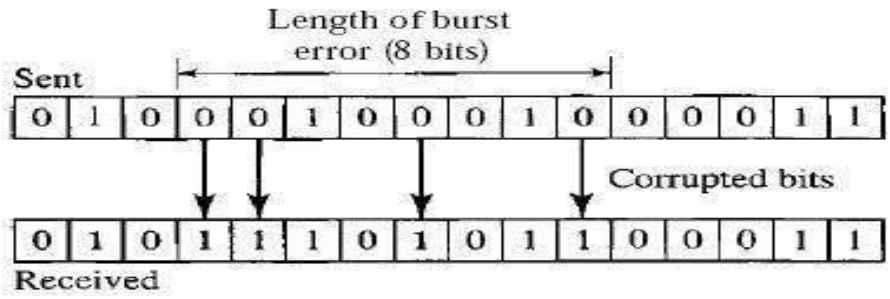
For a single-bit error to occur, the noise must have a duration of only 1) ls, which is very rare; noise normally lasts much longer than this.



Burst Error

The term burst error means that 2 or more bits in the data unit have changed from 1 to 0 o r from 0 to 1.

A burst error is more likely to occur than a single-bit error. The duration of noise is normally longer than the duration of 1 bit, which means that when noise affects data, it affects a set of bits. The number of bits affected depends on the data rate and duration of noise.



Redundancy

The central concept in detecting or correcting errors is redundancy.

To be able to detect or correct errors, we need to send some extra bits with our data.

These redundant bits are added by the sender and removed by the receiver. Their presence allows the receiver to detect or correct corrupted bits.

Detection versus Correction

The correction of errors is more difficult than the detection. In error **detection**, we are looking only to see if any error has occurred. The answer is a simple yes or no.

In error **correction,** we need to know the exact number of bits that are corrupted and more importantly, their location in the message. The number of the errors and the size of the message are important factors.

Forward Error Correction versus Retransmission

There are two main methods of error correction.

Forward error correction is the process in which the receiver tries to guess the message by using redundant bits. This is possible, as we see later, if the number of errors is small.

Correction by **retransmission** is a technique in which the receiver detects the occurrence of an error and asks the sender to resend the message. Resending is repeated unt il a message arrives that the receiver believes is error-free (usually, not all errors can be detected).

Coding

Redundancy is achieved through various coding schemes. The sender adds redundant bits through a process that creates a relationship between the redundant bits and the actual data bits. The receiver checks the relationships between the two sets of bits to detect or correct the errors.

In modular arithmetic, we use only a limited range of integers. We define an upper limit, called a modulus N. We then use only the integers 0 to N -1, inclusive. This is modulo-N arithmetic.

For example, if the modulus is 12, we use only the integers 0 to 11, inclusive.

Of particular interest is modulo-2 arithmetic. In this arithmetic, the modulus N is 2. We can use

only 0 and 1.

Addition	**Subtraction**
0+0=0	0-0=0
1+0=1	1-0=1
1+1=0	1-1=0
0+1=1	0-1=1

Particularly that addition and subtraction give the same results. In this arithmetic we use the XOR (exclusive OR) operation for both addition and subtraction. The result of an XOR operation is 0 if two bits are the same; the result is 1 if two bits are different.

If the modulus is not 2, addition and subtraction are distinct.

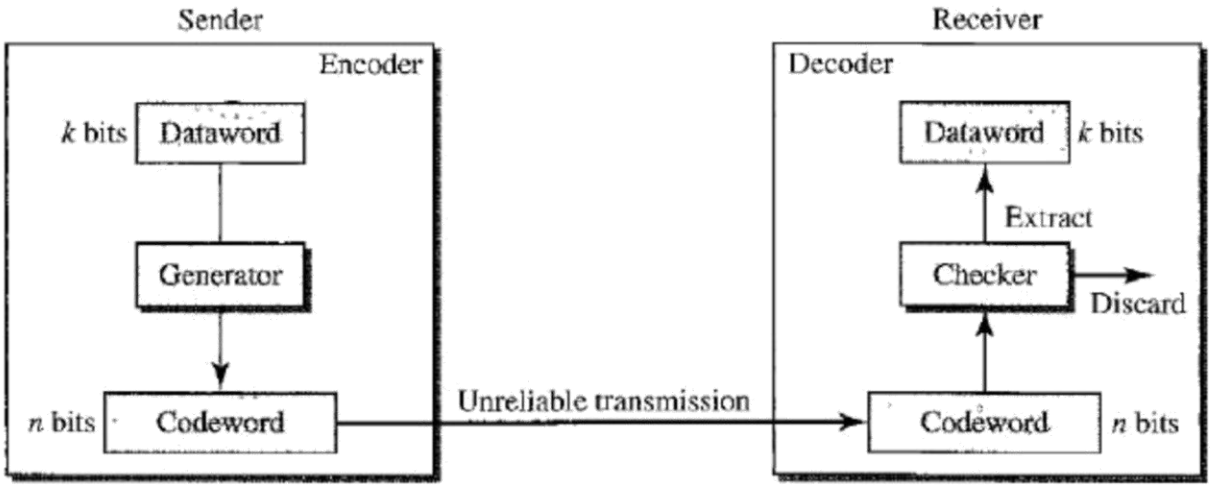
BLOCK CODING

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

With k bits, we can create a combination of 2k data words; with n bits, we can create a combination of 2n code words. Since n > k, the number of possible code words is larger than the number of possible data words. The block coding process is one-to-one; the same data word is always encoded as the same codeword. This means that we have 2n - 2k code words that are not used.

Error Detection

The sender creates code words out of data words by using a generator that applies the rules and procedures of encoding. Each codeword sent to the receiver may change during transmission. If the received codeword is the same as one of the valid code words, the word is accepted; the corresponding data word is extracted for use. If the received codeword is not valid, it is discarded. However, if the codeword is corrupted during transmission but the received word still matches a valid codeword, the error remains undetected. This type of coding can detect only single errors. Two or more errors may remain undetected.



[Diagram: Structure of encoder and decoder for Error detection]

Example

Let us assume that k =2 and n =3. Table 1 shows the list of data words and code words. Later, we will see how to derive a codeword from a data word.

TABLE 1

Data words	Code words
00	000
01	011
10	101
11	110

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

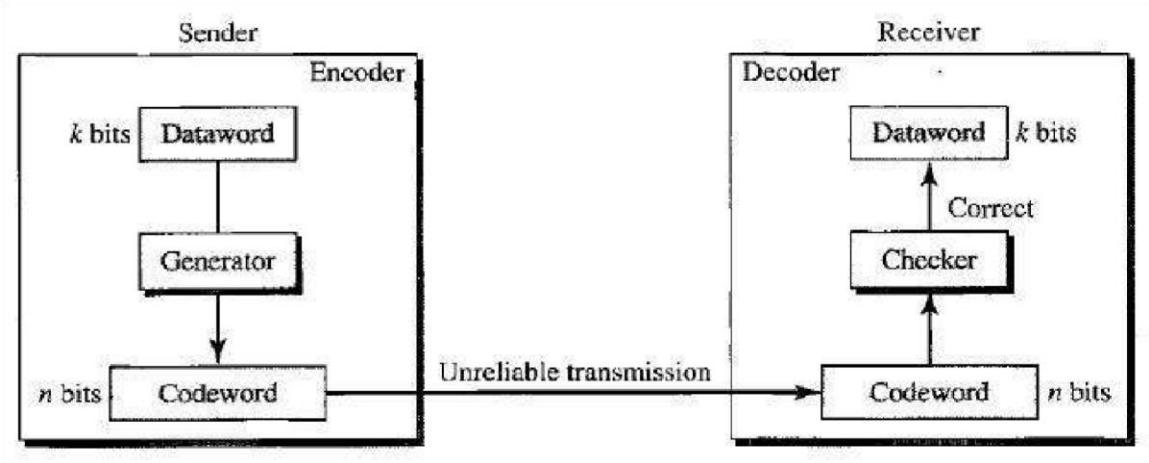
1. The receiver receives 011. It is a valid codeword. The receiver extracts the data word 01 from it.

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 data word 00. Two corrupted bits have made the error undetectable.

Error Correction

As we said before, error correction is much more difficult than error detection. In error detection, the receiver needs to know only that the received codeword is invalid; in error correction the receiver needs to find (or guess) the original codeword sent.



[Diagram: Structure of encoder and decoder for Error correction]

Example

Let us assume that k =2 and n =3. Below table shows the list of data words and code words.

Dataword(k)	Codeword(n=k+r)
00	00000
01	01011
10	10101
11	11110

Assume the dataword is 01. The sender consults the table (or uses an algorithm) to create the codeword 01011. The codeword is corrupted during transmission, and 01001 is received (error in the second bit from the right). First, the receiver finds that the received codeword is not in the table. This means an error has

occurred. (Detection must come before correction.) The receiver, assuming that there is only 1 bit corrupted, uses the following strategy to guess the correct dataword.

1. Comparing the received codeword with the first codeword in the table (01001 versus 00000), the receiver decides that the first codeword is not the one that was sent because there are two different bits.
2. By the same reasoning, the original codeword cannot be the third or fourth one in the table.
3. The original codeword must be the second one in the table because this is the only one that differs from the received codeword by 1 bit. The receiver replaces 01001 with 01011 and consults the table to find the dataword 01.

Hamming Distance

The Hamming distance between two words (of the same size) is the number of differences between the corresponding bits.

The Hamming distance can easily be found if we apply the *XOR operation on the two words and count the number of 1s in the result*. Note that the Hamming distance is a value greater than zero.

The minimum Hamming distance is the smallest Hamming distance between all possible pairs in a set of words.

Example:

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

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

Linear Block Codes

In a linear block code, the exclusive OR (XOR) of any two valid code words creates another valid codeword.

The scheme in Table 1 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 code words creates the fourth one.

Let us now show some linear block codes. These codes are trivial because we can easily find the encoding and decoding algorithms and check their performances.

Simple Parity-Check Code

A simple parity-check code is a single-bit error-detecting code in which n =k + 1 with d_{min} =2.

sender with one exception: The addition is done over all 5 bits. The result, which is called the syndrome, is just 1 bit. *The syndrome is 0 when the number of 1s in the received codeword is even; otherwise, it is 1*.

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

The syndrome is passed to the decision logic analyzer. *If the syndrome is 0,* there is no error in the received codeword; the data portion of the received codeword is accepted as the dataword; *if the syndrome is 1,* the data portion of the received codeword is discarded. The dataword is not created.

Example:

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

We examine five cases:

1. No error occurs; the received codeword is 10111. The syndrome is 0. The dataword 1011 is created.

2. One single-bit error changes a_1. The received codeword is 10011. The syndrome is 1. No dataword is created.

3. One single-bit error changes r_0.The received codeword is 10110. The syndrome is 1. No dataword is created. 4. An error changes r_0 and a second error changes a_3. The received codeword is 00110. The syndrome is 0. The dataword 0011 is created at the receiver. Note that here the dataword is wrongly created due to the syndrome value. *The simple parity-check decoder cannot detect an even number of errors. The errors cancel each other out and give the syndrome a value of 0.*

5. Three bits-a_3, a_2, and a_1-are changed by errors. The received codeword is 01011. The syndrome is 1. The dataword is not created. *This shows that the simple parity check, guaranteed to detect one single error, can also find any odd number of errors.*

Hamming Code

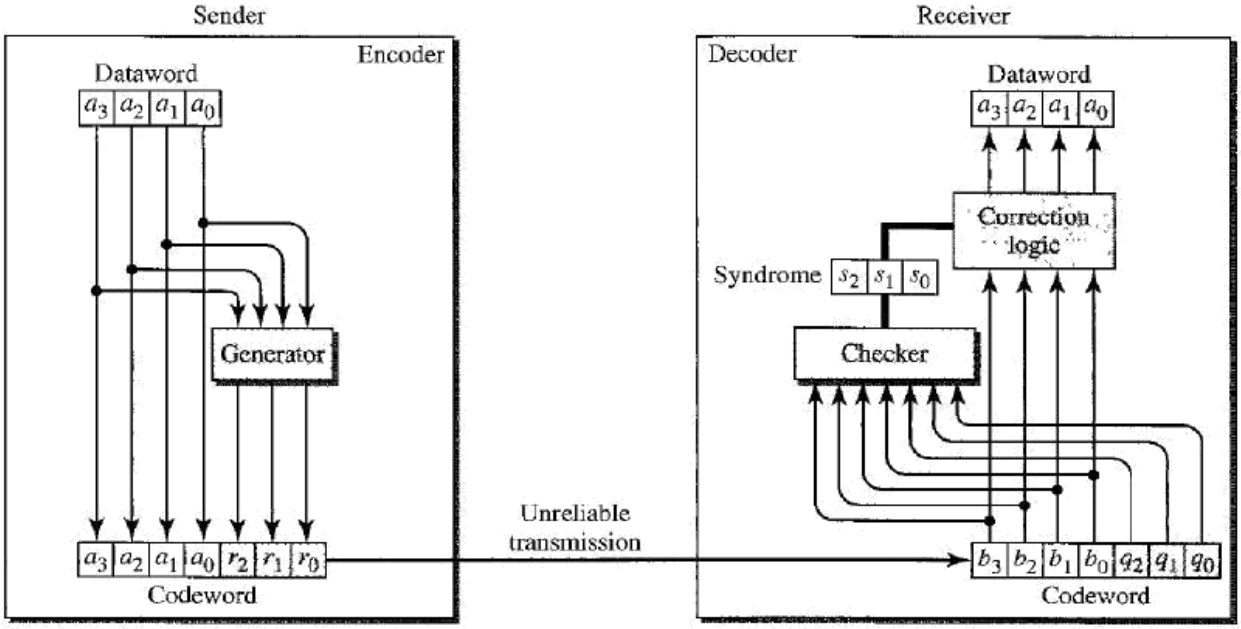
These codes were originally designed with $d_{min} = 3$, which means that they can detect up to two errors or correct one single error. Although there are some Hamming codes that can correct more than one error, our discussion focuses on the single-bit error-correcting code.

Let us find the relationship between n and k in a Hamming code. We need to choose an integer m >= 3.

m

The values of n and k are then calculated from m as n = 2 - 1and k=n - m. The number of check bits r =m.

A Hamming code can only correct a single error or detect a double error.



[Encoder and Decoder for Hamming Code]

A copy of a 4-bit dataword is fed into the generator that creates three parity checks r_0, r_1 and r_2 as

shown below:

$r_0 = a_2 + a_1 + a_o$ modulo-2

$r_1 = a_3 + a_2 + a_1$ modulo-2

$r = a + a + a$ modulo-2

2 1 0 3

In other words, each of the parity-check bits handles 3 out of the 4 bits of the dataword. The total number of 1s in each 4-bit combination (3 dataword bits and 1 parity bit) must be even.

The checker in the **decoder** creates a 3-bit syndrome (s2s1s0) in which each bit is the parity check for 4 out of the 7 bits in the received codeword:

$s_0 = b_2 + b_1 + b_0 + q_0$ modulo-2 $s_1 = b_3 + b_2 + b_1 + q_1$ modulo-2

$s_2 = b_1 + b_0 + b_3 + q_2$ modulo-2

The equations used by the checker are the same as those used by the generator with the parity-check bits added to the right-hand side of the equation. The 3-bit syndrome creates eight different bit patterns

(000 to 111) that can represent eight different conditions. These conditions define a lack of error or an error in 1 of the 7 bits of the received codeword, as shown in Table:

Syndrome	**000**	**001**	**010**	**011**	**100**	**101**	**110**	**111**
Error	**None**	q_0	q_1	**b2**	**q2**	b_0	b_3	b_1

Let us trace the path of three datawords from the sender to the destination:

1. The dataword 0100 becomes the codeword 0100011. The codeword 01 00011 is received. The syndrome is 000 (no error), the final dataword is 0100.

2. The dataword 0111 becomes the codeword 0111001. The codeword 0011001 is received. The syndrome is 011. According to Table, b2 is in error. After flipping b2 (changing the 1 to 0), the final dataword is 0111.

3. The dataword 1101 becomes the codeword 1101000. The codeword 0001000 is received (two errors). The syndrome is 101, which means that b_0 is in error. After flipping b_0, we get 0000, the wrong dataword. This shows that our code cannot correct two errors.

Cyclic Codes

Cyclic codes are special linear block codes with one extra property. In a cyclic code, if a codeword is cyclically shifted (rotated), the result is another codeword. For example, if 1011000

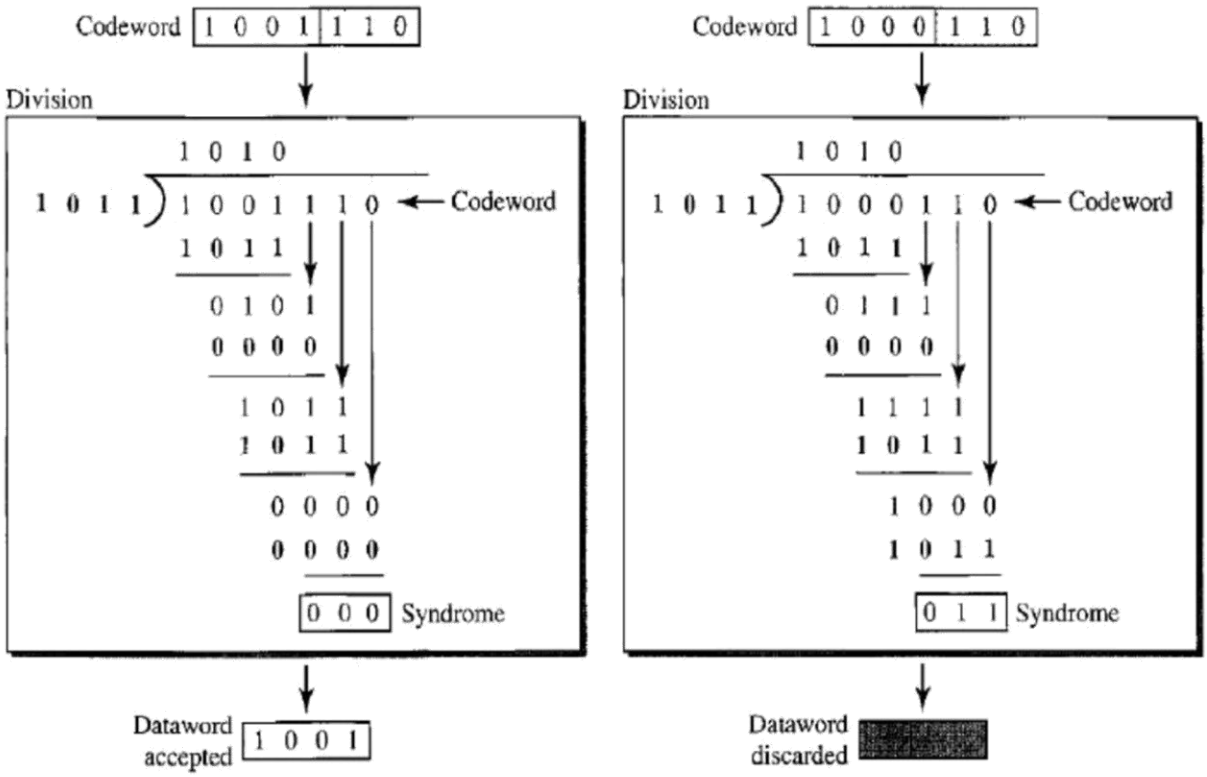
is a codeword and we cyclically left-shift, then 0110001 is also a codeword.

If we call the bits in the first word a_0 to a_6 and the bits in the second word b_0 to b_6, we can shift the bits by using the following:

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

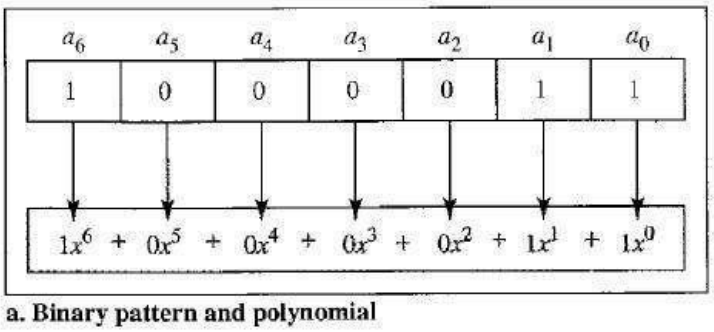
Cyclic Redundancy Check

We can create cyclic codes to correct errors. However, the theoretical background required is beyond the scope of this book. In this section, we simply discuss a category of cyclic codes called the cyclic redundancy check (CRC) that is used in networks such as LANs and WANs.

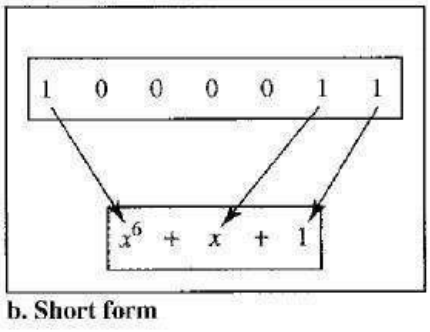


Polynomial

A pattern of 0s and 1s can be represented as a **polynomial** with coefficients of 0 and 1. The power of each term shows the position of the bit; the coefficient shows the val ue of the bit. Below figure shows a binary pattern and its polynomial representation.



a. Binary pattern and polynomial



b. Short form