MODULE 2: DIGITAL TRANSMISSION

2.1 ANALOG TO DIGITAL CONVERSION

- An analog-signal may created by a microphone or camera.
- To change an analog-signal to digital-data, we use PCM (pulse code modulation).
- After the digital-data are created (digitization), then we convert the digital-data to a digital-signal.

2.1.1 PCM

- PCM is a technique used to change an analog signal to digital data (digitization).
- PCM has encoder at the sender and decoder at the receiver.
- The encoder has 3 processes (Figure 4.21):

  1) Sampling
  2) Quantization &
  3) Encoding.

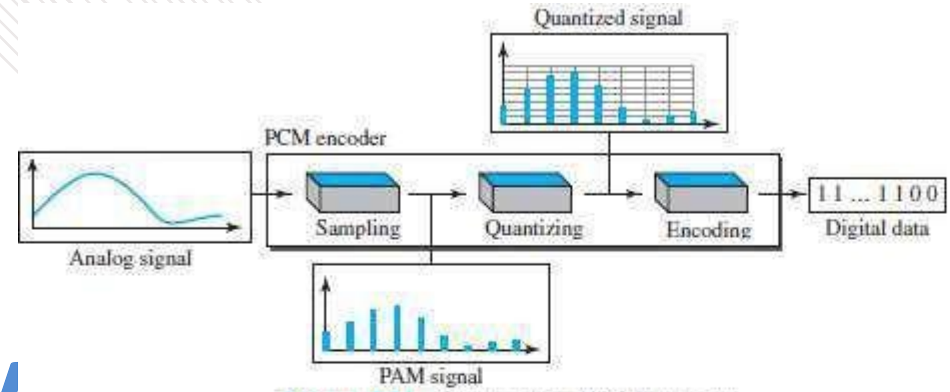


Figure 4.21 *Components of PCM encoder*

2.1.1.1 Sampling

- We convert the continuous time signal (analog) into the discrete time signal (digital). • Pulses from the analog-signal are sampled every T_s sec where T_s is the sample-interval or period.
- The inverse of the sampling-interval is called the sampling-frequency (or sampling-rate).
- Sampling-frequency is given by

$$f_s = 1/T_s$$

- Three sampling methods (Figure 4.22):

  1) Ideal Sampling

  □ This method is difficult to implement.

  2) Natural Sampling

  □ A high-speed switch is turned ON for only the small period of time when the sampling occurs. □ The result is a sequence of samples that retains the shape of the analog-signal. **3) Flat Top Sampling**

  □ The most common sampling method is sample and hold. □ Sample and hold method creates flat-top samples.

  □ This method is sometimes referred to as *PAM* (pulse amplitude modulation).

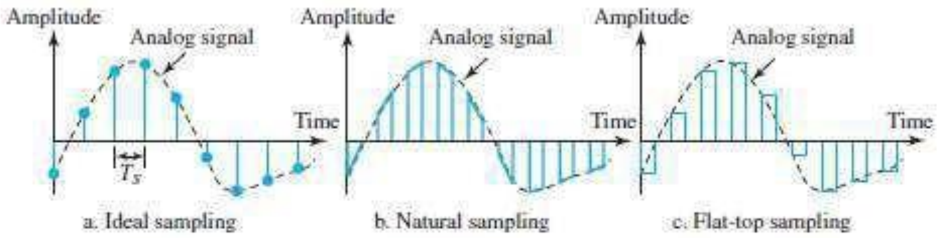


Figure 4.22 *Three different sampling methods for PCM*

2.1.1.1.1 Sampling Rate

• According to Nyquist theorem,

"The sampling-rate must be at least 2 times the highest frequency, not the bandwidth".

i) If the analog-signal is **low-pass**, the bandwidth and the highest frequency are the same value (Figure 4.23a). ii) If the analog-signal is **bandpass**, the bandwidth value is lower than the value of the maximum frequency (Figure 4.23b).

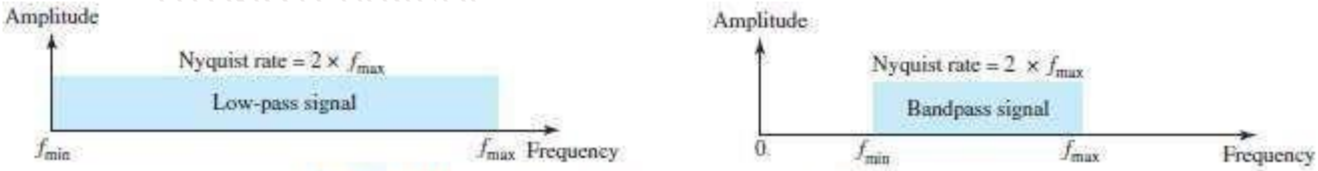


Figure 4.23 *Nyquist sampling rate for low-pass and bandpass signals*

2.1.2 Quantization

- The sampled-signal is quantized.
- Result of sampling is a set of pulses with amplitude-values b/w max & min amplitudes of the signal.
- Four steps in quantization:
  1) We assume that the original analog-signal has amplitudes between V_{min} & V_{max}. 2) We divide the range into L zones, each of height Δ(delta).

$$\Delta = \frac{V_{max} - V_{min}}{L}$$

where L = number of levels.

  3) We assign quantized values of 0 to (L-1) to the midpoint of each zone.
  4) We approximate the value of the sample amplitude to the quantized values.
- For example: Let V_{min}=-20 V_{max} =+20 V L = 8 Therefore, Δ = [+20-(-20)]/8= 5 V
- In the chart (Figure 4.26),
  1) First row is normalized-PAM-value for each sample.
  2) Second row is normalized-quantized-value for each sample.
  3) Third row is normalized error (which is diff. b/w normalized PAM value & quantized values).
  4) Fourth row is quantization code for each sample.
  5) Fifth row is the encoded words (which are the final products of the conversion).

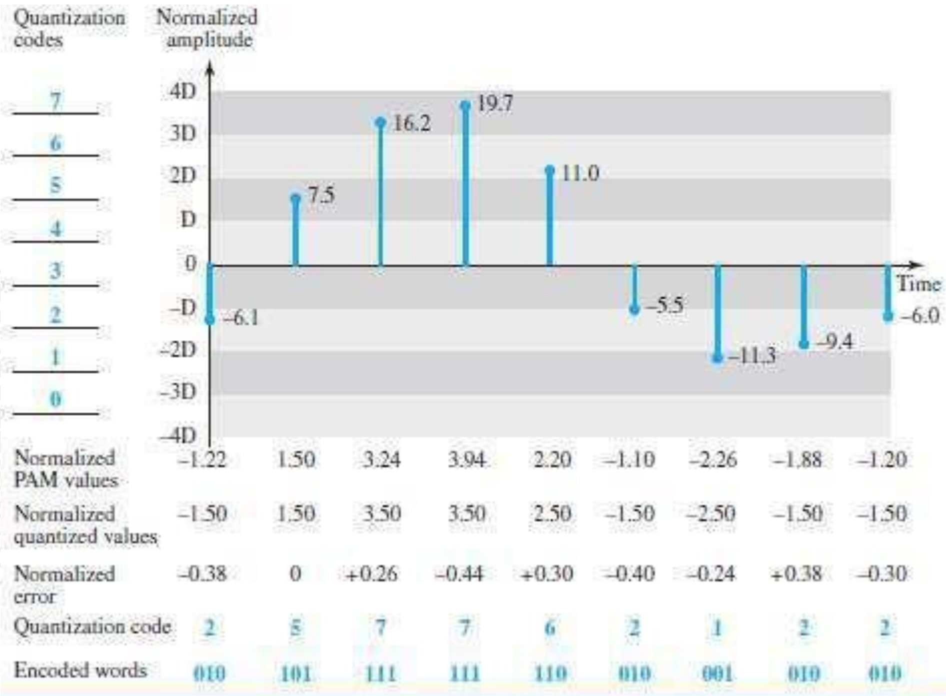


Figure 4.26 *Quantization and encoding of a sampled signal*

2.1.2.1 Quantization Level

- Let L = number of levels.
- The choice of L depends on

 → range of the amplitudes of the analog-signal and → how accurately we need to recover the signal.

- If the signal has only 2 amplitude values, we need only 2 quantization-levels.

 If the signal (like voice) has many amplitude values, we need more quantization-levels.

- In audio digitizing, L is normally chosen to be 256. In video digitizing, L is normally thousands.
- Choosing lower values of L increases the quantization-error.

2.1.2.2 Quantization Error

- Quantization-error is the difference b/w normalized PAM value & quantized values
- Quantization is an approximation process.
- The input values to the quantizer are the real values.

 The output values from the quantizer are the approximated values.

- The output values are chosen to be the middle value in the zone.
- If the input value is also at the middle of the zone, Then, there is no error.

 Otherwise, there is an error.

- In the previous example,

 The normalized amplitude of the third sample is 3.24, but the normalized quantized value is 3.50. This means that there is an error of +0.26.

2.1.2.3 Uniform vs. Non Uniform Quantization

- Non-uniform quantization can be done by using a process called companding and expanding.

 1) The signal is companded at the sender before conversion. 2) The signal is expanded at the receiver after conversion.

- Companding means reducing the instantaneous voltage amplitude for large values.
  Expanding means increasing the instantaneous voltage amplitude for small values.
- It has been proved that non-uniform quantization effectively reduces the SNR_{dB} of quantization.

2.1.3 Encoding

- The quantized values are encoded as n-bit code word.
- In the previous example,
  A quantized value 2 is encoded as 010.
  A quantized value 5 is encoded as 101.
- Relationship between number of quantization-levels (L) & number of bits (n) is given by $n=\log_2 L$ or $2^n=L$
- The bit-rate is given by:

$$\text{Bit rate} = \text{sampling rate} \times \text{number of bits per sample} = f_s \times n$$

Example 2.1

A complex low-pass signal has a bandwidth of 200 kHz. What is the minimum sampling rate for this signal?

Solution

The bandwidth of a low-pass signal is between 0 and f, where f is the maximum frequency in the signal. Therefore, we can sample this signal at 2 times the highest frequency (200 kHz). The sampling rate is therefore 400,000 samples per second.

Example 2.2

What is the SNR_{dB} in the example of Figure 4.26?

Solution

We can use the formula to find the quantization. We have eight levels and 3 bits per sample, so $SNR_{dB} = 6.02(3) + 1.76 = 19.82$ dB. Increasing the number of levels increases the SNR.

Example 2.3

A telephone subscriber line must have an SNR_{dB} above 40. What is the minimum number of bits per sample?

Solution

We can calculate the number of bits as

$$SNR_{dB} = 6.02n_b + 1.76 = 40 \rightarrow n = 6.35$$

Telephone companies usually assign 7 or 8 bits per sample.

Example 2.4

We want to digitize the human voice. What is the bit rate, assuming 8 bits per sample?

Solution

The human voice normally contains frequencies from 0 to 4000 Hz. So the sampling rate and bit rate are calculated as follows:

$$\text{Sampling rate} = 4000 \times 2 = 8000 \text{ samples/s}$$
$$\text{Bit rate} = 8000 \times 8 = 64{,}000 \text{ bps} = 64 \text{ kbps}$$

2.1.3.1 Original Signal Recovery

- PCM decoder is used for recovery of the original signal.
- Here is how it works (Figure 4.27):
  1) The decoder first uses circuitry to convert the code words into a pulse that holds theamplitude until the next pulse.

2) Next, the staircase-signal is passed through a low-pass filter to smooth the staircase signal into an analog-signal.

- The filter has the same cut-off frequency as the original signal at the sender.
- If the signal is sampled at the Nyquist sampling-rate, then the original signal will be re-created.
- The maximum and minimum values of the original signal can be achieved by using amplification.

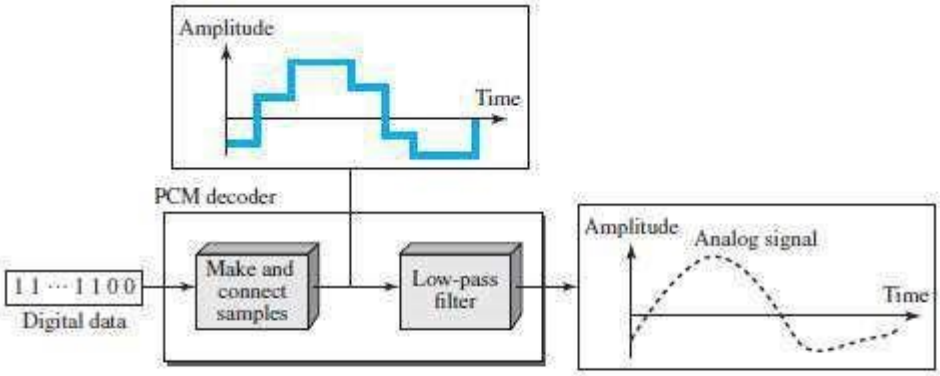


Figure 4.27 *Components of a PCM decoder*

2.1.3.2 PCM Bandwidth

- The minimum bandwidth of a line-encoded signal is

$$B_{\min} = c \times N \times \frac{1}{r}$$

- We substitute the value of N in above formula:

$$B_{\min} = c \times N \times \frac{1}{r} = c \times n_b \times f_s \times \frac{1}{r} = c \times n_b \times 2 \times B_{\text{analog}} \times \frac{1}{r}$$

- When 1/r = 1 (for a NRZ or bipolar signal) and c = (1/2) (the average situation), the minimumbandwidth is

$$B_{\min} = n_b \times B_{\text{analog}}$$

- This means the minimum bandwidth of the digital-signal is n_b times greater than the bandwidth of the analog-signal.

2.1.3.3 Maximum Data Rate of a Channel

- The Nyquist theorem gives the data-rate of a channel as

$$N_{\max} = 2 \times B \times \log_2 L$$

- We can deduce above data-rate from the Nyquist sampling theorem by using the following arguments.

1) We assume that the available channel is low-pass with bandwidth B.
2) We assume that the digital-signal we want to send has L levels, where each level is a signalelement. This means r = 1/log$_2$L.
3) We first pass digital-signal through a low-pass filter to cut off the frequencies above B Hz. 4) We treat the resulting signal as an analog-signal and sample it at 2 x B samples per second and quantize it using L levels. 5) The resulting bit-rate is

$$N = f_s \times n_b = 2 \times B \times \log_2 L$$

This is the maximum bandwidth; if the case factor c increases, the data-rate is reduced.

$$N_{\max} = 2 \times B \times \log_2 L \quad \text{bps}$$

2.1.3.4 Minimum Required Bandwidth

• The previous argument can give us the minimum bandwidth if the data-rate and the number of signal-levels are fixed. We can say

$$B_{min} = \frac{N}{(2 \times \log_2) L} \quad \text{Hz}$$

2.2 TRANSMISSION MODES

• Two ways of transmitting data over a link (Figure 4.31): 1) Parallel mode & 2) Serial mode.

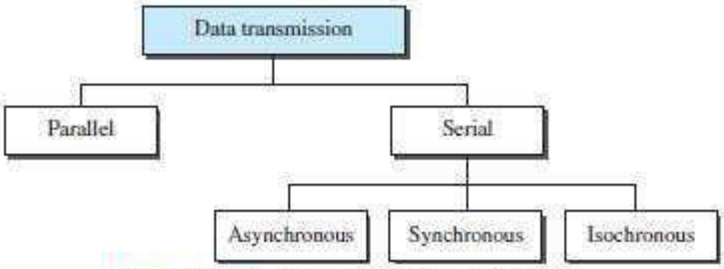


Figure 4.31 *Data transmission and modes*

2.1.1 PARALLEL TRANSMISSION

- Multiple bits are sent with each clock-tick (Figure 4.32).
- „n" bits in a group are sent simultaneously.
- „n" wires are used to send „n" bits at one time.
- Each bit has its own wire.
- Typically, the 8 wires are bundled in a cable with a connector at each end.

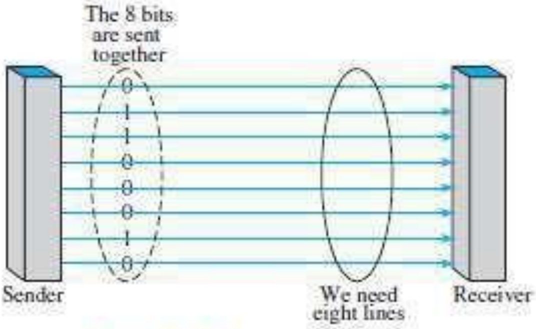


Figure 4.32 *Parallel transmission*

- Advantage:
  1) Speed: Parallel transmission can increase the transfer speed by a factor of n over serialtransmission.
- Disadvantage:
  1) Cost: Parallel transmission requires n communication lines just to transmit the data-stream. Because this is expensive, parallel transmission is usually limited to short distances.

2.2.2 SERIAL TRANSMISSION

- One bit is sent with each clock-tick using only a single link (Figure 4.33).

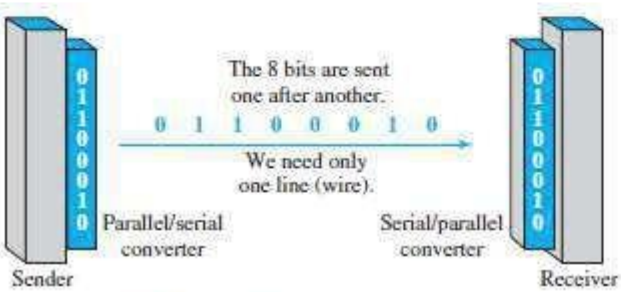


Figure 4.33 *Serial transmission*

- Advantage:

1) Cost: Serial transmission reduces cost of transmission over parallel by a factor of n.

- Disadvantage:
  1) Since communication within devices is parallel, following 2 converters are required at interface: i) Parallel-to-serial converter ii) Serial-to-parallel converter
- Three types of serial transmission: asynchronous, synchronous, and isochronous.

2.2.2.1 Asynchronous Transmission

- Asynchronous transmission is so named because the timing of a signal is not important (Figure 4.34).
- Prior to data transfer, both sender & receiver agree on pattern of information to be exchanged.
- Normally, patterns are based on grouping the bit-stream into bytes.
- The sender transmits each group to the link without regard to a timer.
- As long as those patterns are followed, the receiver can retrieve the info. without regard to a timer.
- There may be a gap between bytes.
- We send
  → 1 start bit (0) at the beginning of each byte → 1
  stop bit (1) at the end of each byte.
- Start bit alerts the receiver to the arrival of a new group.
  Stop bit lets the receiver know that the byte is finished.
- Here, the term asynchronous means "asynchronous at the byte level".
  However, the bits are still synchronized & bit-durations are the same.

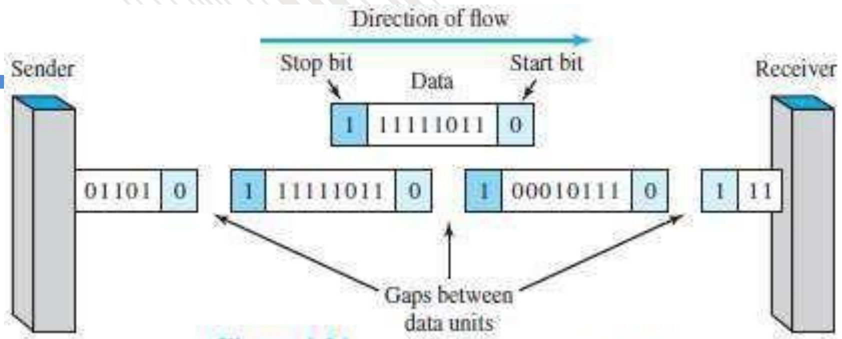


Figure 4.34 *Asynchronous transmission*

- Disadvantage:
  1) Slower than synchronous transmission. (Because of stop bit, start bit and gaps)
- Advantages:
  1) Cheap & effective.
  2) Useful for low-speed communication.

2.2.2.2 Synchronous Transmission

- We send bits one after another without start or stop bits or gaps (Figure 4.35).
- The receiver is responsible for grouping the bits.
- The bit-stream is combined into longer "frames," which may contain multiple bytes.
- If the sender wants to send data in separate bursts, the gaps b/w bursts must be filled with a special sequence of 0s & 1s (that means idle).

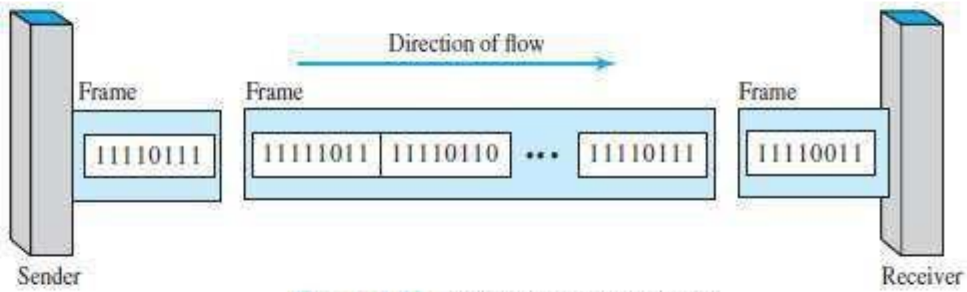


Figure 4.35 *Synchronous transmission*

- Advantages:
  1) Speed: Faster than asynchronous transmission. („.‟ of no stop bit, start bit and gaps).
  2) Useful for high-speed applications such as transmission of data from one computer to another.

2.2.2.3 Isochronous

- Synchronization between characters is not enough; the entire stream of bits must be synchronized.
- The isochronous transmission guarantees that the data arrive at a fixed rate.
- In real-time audio/video, jitter is not acceptable. Therefore, synchronous transmission fails.
- For example: TV images are broadcast at the rate of 30 images per second. The images must beviewed at the same rate.

ANALOG TRANSMISSION

2.3 DIGITAL TO ANALOG CONVERSION

- Digital-to-analog conversion is the process of changing one of the characteristics of an analog-signal based on the information in digital-data (Figure 5.1).

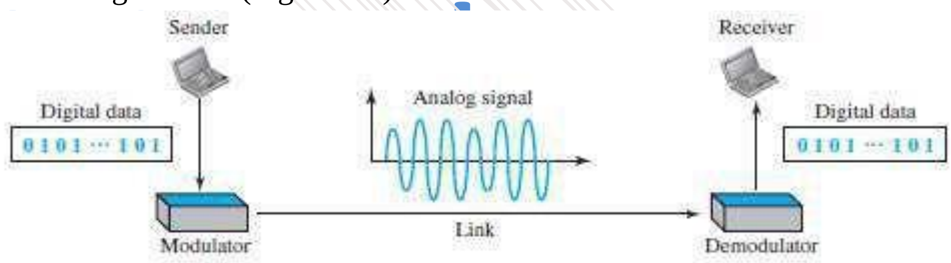


Figure 5.1 *Digital-to-analog conversion*

- A sine wave can be defined by 3 attributes:
  1) Amplitude 2)
  Frequency & 3) Phase.
- When anyone of the 3 attributes of a wave is varied, a different version of the wave will be created.
- So, by changing one attribute of an analog signal, we can use it to represent digital-data.
- Four methods of digital to analog conversion (Figure 5.2):
  1) Amplitude shift keying (ASK) 2) Frequency shift keying (FSK)
  3) Phase shift keying (PSK)
  4) Quadrature amplitude modulation (QAM).
- QAM is a combination of ASK and PSK i.e. QAM combines changing both the amplitude and phase. QAM is the most efficient of these 4 methods.

QAM is the method commonly used today.

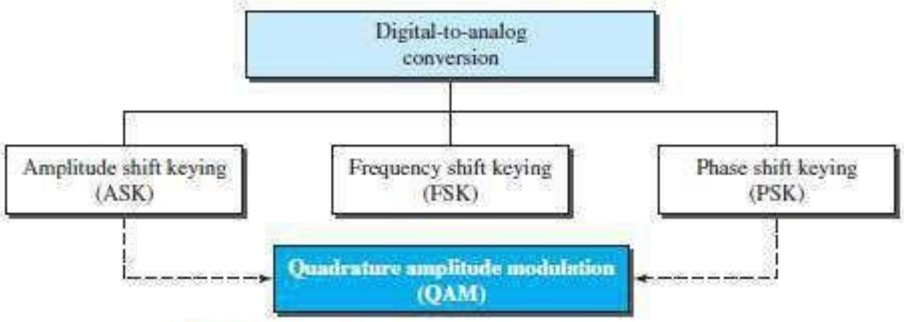


Figure 5.2 *Types of digital-to-analog conversion*

2.3.1 Aspects of Digital-to-Analog Conversion

1) Data Element vs. Signal Element

- A data-element is the smallest piece of information to be exchanged i.e. the bit.
- A signal-element is the smallest unit of a signal that is transmitted.

2) Data Rate vs. Signal Rate

- Data rate (Bit rate) is the number of bits per second.
- Signal-rate (Baud rate) is the number of signal elements per second.
- The relationship between data-rate(N) and the signal-rate(S) is

$$S = N \times \frac{1}{r} \quad \text{baud}$$

where r = number of data-elements carried in one signal-element.

- The value of r is given by $r = \log_2 L$ or $2^r = L$ where L = type of signal-element (not the level)

(In transportation,
→ a baud is analogous to a vehicle, and → a bit is analogous to a passenger.
We need to maximize the number of people per car to reduce the traffic).

3) Carrier Signal

- The sender produces a high-frequency signal that acts as a base for the information-signal.
- This base-signal is called the carrier-signal (or carrier-frequency).
- The receiver is tuned to the frequency of the carrier-signal that it expects from the sender.
- Then, digital-information changes the carrier-signal by modifying its attributes (amplitude, frequency, or phase). This kind of modification is called modulation (shift keying).

4) Bandwidth

- In both ASK & PSK, the bandwidth required for data transmission is proportional to the signal-rate.
- In FSK, the bandwidth required is the difference between the two carrier-frequencies.

Example 2.5

An analog signal carries 4 bits per signal element. If 1000 signal elements are sent per second, find the bit rate.

Solution

In this case, $r = 4$, $S = 1000$, and N is unknown. We can find the value of N from

$$S = N \times (1/r) \quad \text{or} \quad N = S \times r = 1000 \times 4 = 4000 \text{ bps}$$

Example 2.6

An analog signal has a bit rate of 8000 bps and a baud rate of 1000 baud. How many data elements are carried by each signal element? How many signal elements do we need?

Solution

In this example, $S = 1000$, $N = 8000$, and r and L are unknown. We first find the value of r and then the value of L.

$$S = N \times 1/r \longrightarrow r = N / S = 8000 /10{,}000 = 8 \text{ bits/baud}$$
$$r = \log_2 L \longrightarrow L = 2^r = 2^8 = 256$$

2.3.2 Amplitude Shift Keying (ASK)

•The amplitude of the carrier-signal is varied to represent different signal-elements.

• Both frequency and phase remain constant for all signal-elements.

2.3.2.1 Binary ASK (BASK)

• BASK is implemented using only 2 levels. (Figure 5.3)

• This is also referred to as OOK (On-Off Keying).

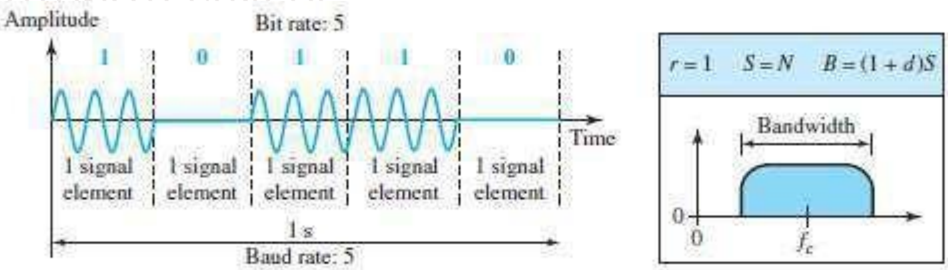


Figure 5.3 *Binary amplitude shift keying*

2.3.2.1.1 Implementation of BASK

• Here, line coding method used = unipolar NRZ (Figure 5.4).

• The unipolar NRZ signal is multiplied by the carrier-frequency coming from an oscillator. 1) When amplitude of the NRZ signal = 0, amplitude of the carrier-signal = 0.

2) When amplitude of the NRZ signal = 1, the amplitude of the carrier-signal is held.

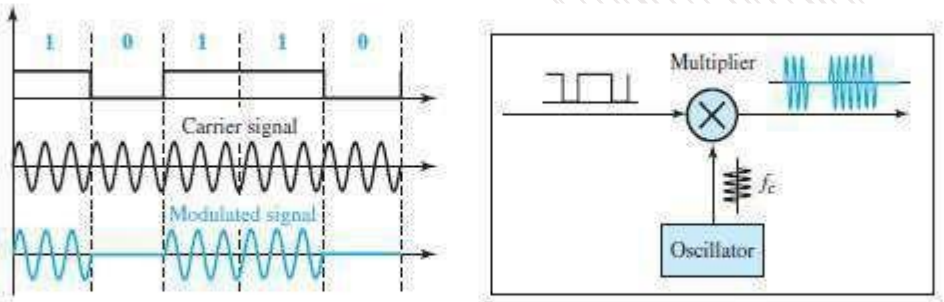


Figure 5.4 *Implementation of binary ASK*

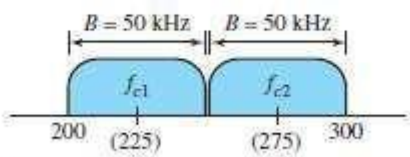


Figure 5.5 *Bandwidth of full-duplex ASK*

2.3.2.1.2 Bandwidth for ASK

• Here, the bandwidth (B) is proportional to the signal-rate (S) (Figure 5.5) • The bandwidth is given by

$$B = (1 + d) \times S$$

where d(0<d<1)= this factor depends on modulation and filtering-process. **Example 2.7**

We have an available bandwidth of 100 kHz which spans from 200 to 300 kHz. What are the carrier frequency and the bit rate if we modulated our data by using ASK with $d = 1$?

Solution

The middle of the bandwidth is located at 250 kHz. This means that our carrier frequency can be at $f_c = 250$ kHz. We can use the formula for bandwidth to find the bit rate (with $d = 1$ and $r = 1$).

$$B = (1 + d) \times S = 2 \times N \times (1/r) = 2 \times N = 100 \text{ kHz} \longrightarrow N = 50 \text{ kbps}$$

2.3.3 Frequency Shift Keying (FSK)

- The frequency of the carrier-signal is varied to represent different signal-elements.
- The frequency of the modulated-signal is constant for the duration of one signal-element, but changes for the next signal-element if the data-element changes. • Both amplitude and phase remain constant for all signal-elements.

2.3.3.1 Binary FSK (BFSK)

• This uses 2 carrier-frequencies: f1 and f2. (Figure 5.6)

1) When data-element = 1, first carrier frequency(f1) is used.
2) When data-element = 0, second carrier frequency(f2) is used.

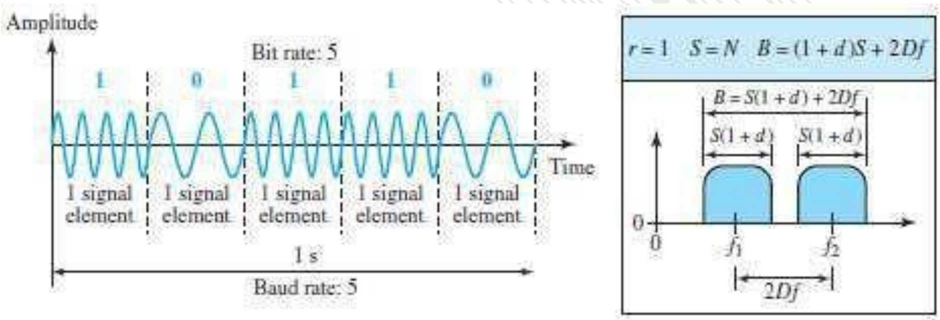


Figure 5.6 *Binary frequency shift keying*

2.3.3.1.1 Implementation

- Here, line coding method used = unipolar NRZ.
- Two implementations of BFSK: i) Coherent and ii) Non-Coherent.

Coherent BFSK	Non Coherent BFSK
The phase continues through the boundary of two signal-elements (Figure 5.7).	There may be discontinuity in the phase when one signal-element ends and the next begins.
This is implemented by using one voltage-controlled oscillator (VCO). VCO changes frequency according to the input voltage.	This is implemented by → treating BFSK as 2 ASK modulations and → using 2 carrier-frequencies

When the amplitude of NRZ signal = 0, the VCO keeps its regular frequency. When the amplitude of NRZ signal = 0, the VCO increases its frequency.	

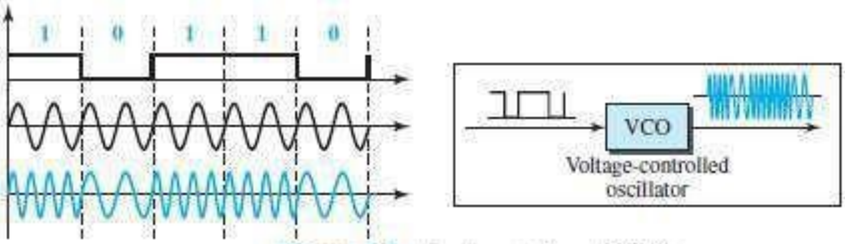


Figure 5.7 *Implementation of BFSK*

2.3.3.1.2 Bandwidth for BFSK

• FSK has two ASK signals, each with its own carrier-frequency f1 or f2. (Figure 5.6) • The bandwidth is given by $B = (1 + d) \times S + 2\Delta f$ where 2Δf is the difference between f1 and f2, **Example 2.8**

We have an available bandwidth of 100 kHz which spans from 200 to 300 kHz. What should be the carrier frequency and the bit rate if we modulated our data by using FSK with $d = 1$?

Solution

This problem is similar to Example 5.3, but we are modulating by using FSK. The midpoint of the band is at 250 kHz. We choose $2\Delta_f$ to be 50 kHz; this means

$$B = (1 + d) \times S + 2\Delta_f = 100 \longrightarrow 2S = 50 \text{ kHz} \longrightarrow S = 25 \text{ kbaud} \longrightarrow N = 25 \text{ kbps}$$

Example 2.9

We need to send data 3 bits at a time at a bit rate of 3 Mbps. The carrier frequency is 10 MHz. Calculate the number of levels (different frequencies), the baud rate, and the bandwidth.

Solution

We can have $L = 2^3 = 8$. The baud rate is $S = 3$ MHz/3 = 1 Mbaud. This means that the carrier frequencies must be 1 MHz apart ($2\Delta_f = 1$ MHz). The bandwidth is $B = 8 \times 1 = 8$ MHz. Figure 5.8 shows the allocation of frequencies and bandwidth.

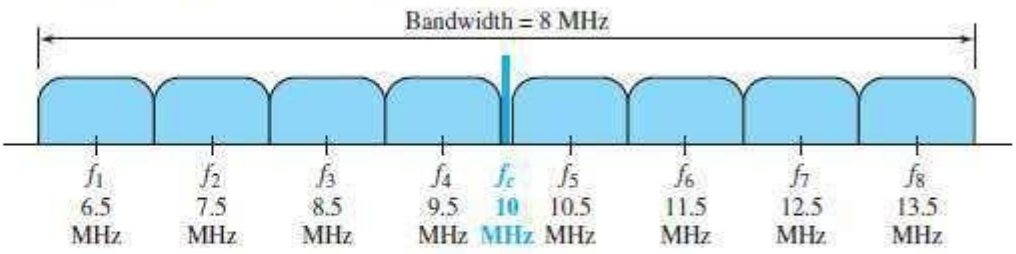


Figure 5.8 *Bandwidth of MFSK used*

2.3.4 Phase Shift Keying (PSK)

- The phase of the carrier-signal is varied to represent different signal-elements.
- Both amplitude and frequency remain constant for all signal-elements.

2.3.4.1 Binary PSK (BPSK)

- We have only two signal-elements:
  1) First signal-element with a phase of 0°.
  2) Second signal-element with a phase of 180° (Figure 5.9).
- ASK vs. PSK
   In ASK, the criterion for bit detection is the amplitude of the signal. In PSK, the criterion for bit detection is the phase.
- Advantages:
  1) PSK is less susceptible to noise than ASK.
  2) PSK is superior to FSK because we do not need 2 carrier-frequencies.
- Disadvantage:
  1) PSK is limited by the ability of the equipment to distinguish small differences in phase.

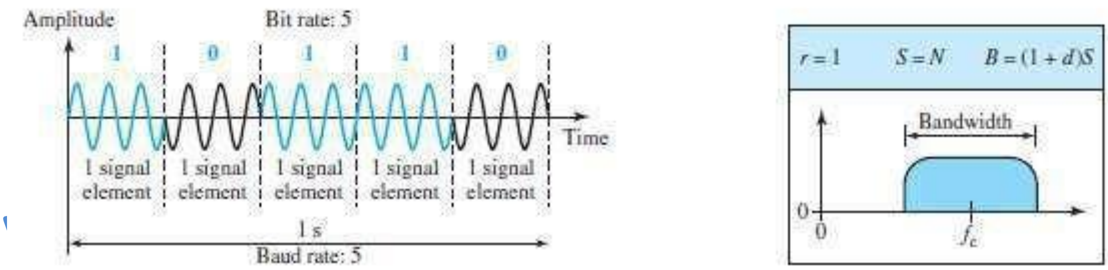


Figure 5.9 *Binary phase shift keying*

2.3.4.1.1 Implementation

- The implementation of BPSK is as simple as that for ASK. (Figure 5.10).
- The signal-element with phase 180° can be seen as the complement of the signal-element with phase 0°.
- Here, line coding method used: polar NRZ.
- The polar NRZ signal is multiplied by the carrier-frequency coming from an oscillator.
  1) When data-element = 1, the phase starts at 0°.
  2) When data-element = 0, the phase starts at 180°.

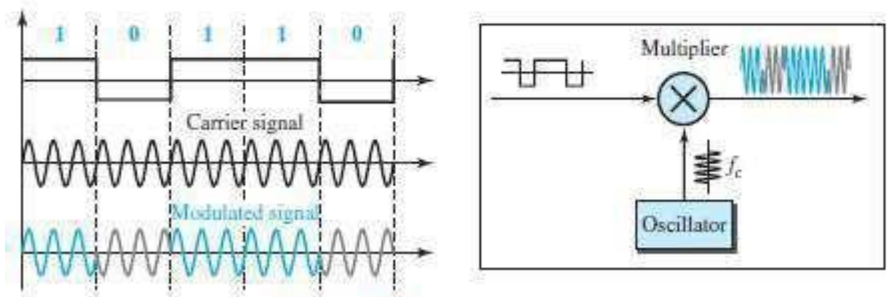


Figure 5.10 *Implementation of BASK*

2.3.4.1.2 Bandwidth for BPSK

- The bandwidth is the same as that for BASK, but less than that for BFSK. (Figure 5.9b)
- No bandwidth is wasted for separating 2 carrier-signals.

2.3.4.2 Quadrature PSK (QPSK)

- The scheme is called QPSK because it uses 2 separate BPSK modulations (Figure 5.11):
  1) First modulation is in-phase,
  2) Second modulation is quadrature (out-of-phase).
- A serial-to-parallel converter
  → accepts the incoming bits
  → sends first bit to first modulator and
  → sends second bit to second modulator.
- The bit to each BPSK signal has one-half the frequency of the original signal.
- Advantages:
  1) Decreases the baud rate.
  2) Decreases the required bandwidth.

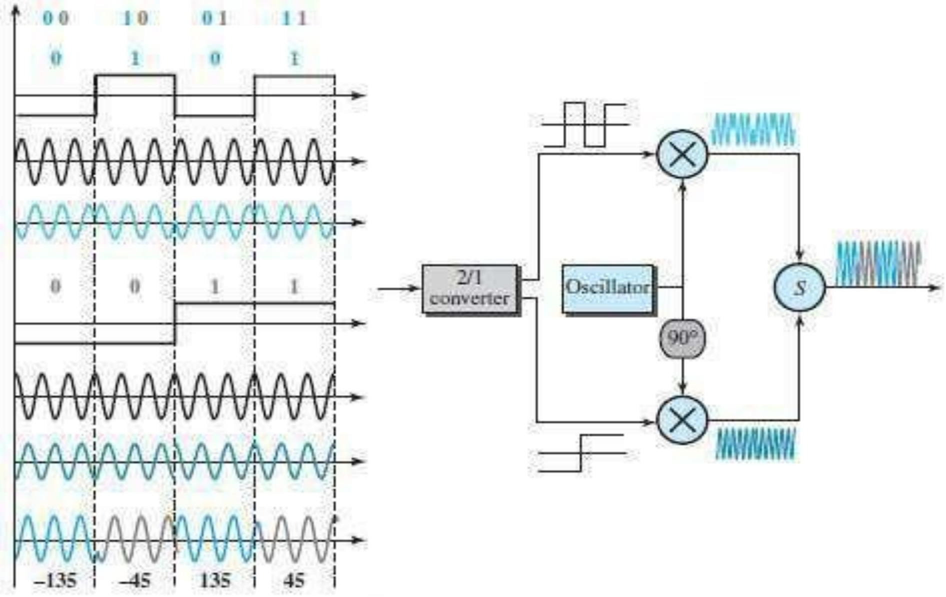


Figure 5.11 *QPSK and its implementation*

- As shown in Figure 5.11, the 2 composite-signals created by each multiplier are 2 sine waves with the same frequency, but different phases.
- When the 2 sine waves are added, the result is another sine wave, with 4 possible phases: 45°, -45°, 135°, and -135°.
- There are 4 kinds of signal-elements in the output signal (L=4), so we can send 2 bits per signalelement (r=2).

Example 2.10

Find the bandwidth for a signal transmitting at 12 Mbps for QPSK. The value of $d = 0$.

Solution

For QPSK, 2 bits are carried by one signal element. This means that $r = 2$. So the signal rate (baud rate) is $S = N \times (1/r) = 6$ Mbaud. With a value of $d = 0$, we have $B = S = 6$ MHz.

2.3.4.3 Constellation Diagram

- A constellation diagram can be used to define the amplitude and phase of a signal-element.
- This diagram is particularly useful
  → when 2 carriers (one in-phase and one quadrature) are used. → when dealing with multilevel ASK, PSK, or QAM.

- In a constellation diagram, a signal-element type is represented as a dot.
- The diagram has 2 axes (Figure 5.12):
    1) The horizontal X axis is related to the in-phase carrier. 2) The vertical Y axis is related to the quadrature carrier.
- For each point on the diagram, 4 pieces of information can be deduced.
    1) The projection of point on the X axis defines the peak amplitude of the in-phase component.
    2) The projection of point on Y axis defines peak amplitude of the quadrature component. 3) The length of the line that connects the point to the origin is the peak amplitude of the signal-element (combination of the X and Y components);
    4) The angle the line makes with the X axis is the phase of the signal-element.

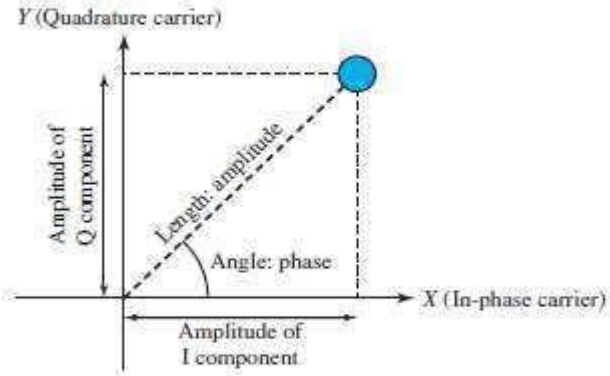


Figure 5.12 *Concept of a constellation diagram*

Example 2.11

Show the constellation diagrams for ASK (OOK), BPSK, and QPSK signals.

Solution

Figure 5.13 shows the three constellation diagrams. Let us analyze each case separately:

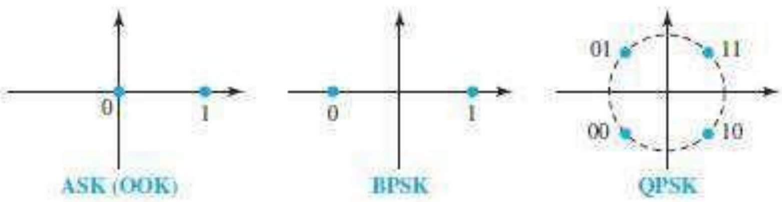


Figure 5.13 *Three constellation diagrams*

2.3.5 Quadrature Amplitude Modulation (QAM) •

This is a combination of ASK and PSK.

- Main idea: Using 2 carriers, one in-phase and the other quadrature, with different amplitude levels for each carrier.
- There are many variations of QAM (Figure 5.14).
    A) Figure 5.14a shows the 4-QAM scheme using a unipolar NRZ signal. This is same as BASK.
    B) Figure 5.14b shows another QAM using polar NRZ. This is the same as QPSK.
    C) Figure 5.14c shows another 4-QAM in which we used a signal with 2 positive levels to modulate each of the 2 carriers.
    D) Figure 5.14d shows a 16-QAM constellation of a signal with 8 levels, 4 positive & 4 negative.

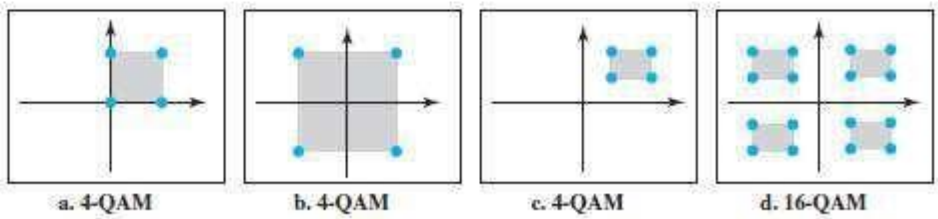


Figure 5.14 *Constellation diagrams for some QAMs*

2.3.5.1 Bandwidth for QAM

- The bandwidth is same as in ASK and PSK transmission.
- QAM has the same advantages as PSK over ASK.

BANDWIDTH UTILIZATION -- MULTIPLEXING AND SPREADING

2.4 MULTIPLEXING

- When bandwidth of a medium is greater than bandwidth needs of the devices, the link can be shared.
- *Multiplexing* allows simultaneous transmission of multiple signals across a single data-link (Fig 4.21).
- The traffic increases, as data/telecommunications use increases.
- We can accommodate this increase by
  → adding individual links, each time a new channel is needed or →
  installing higher-bandwidth links to carry multiple signals.
- Today's technology includes high-bandwidth media such as optical-fiber and satellite microwaves.
- Each has a bandwidth far in excess of that needed for the average transmission-signal.
- If the bandwidth of a link is greater than the bandwidth needs of the devices connected to it, the bandwidth is wasted.
- An efficient system maximizes the utilization of all resources; bandwidth is one of the most precious resources we have in data communications.

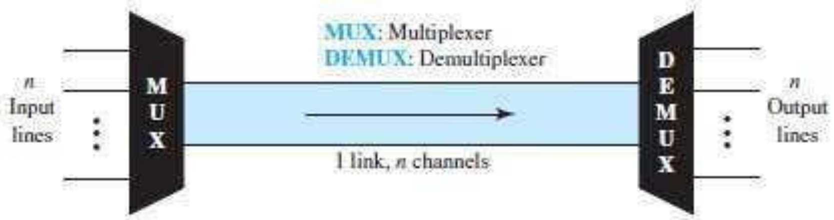


Figure 6.1 *Dividing a link into channels*

- In a multiplexed-system, „n" lines share the bandwidth of one link.
- MUX combines transmission-streams from different input-lines into a single stream (many-to-one).
- At the receiving-end, that stream is fed into a demultiplexer (DEMUX).
- DEMUX
  → separates the stream back into its component-transmissions (one-to-many) and → directs the transmission-streams to different output-lines.
- Link vs. Channel:
  1) The link refers to the physical path.
  2) The channel refers to the portion of a link that carries a transmission between a given pair oflines. One link can have many channels.
- Three multiplexing techniques (Figure 6.2):
  1) Frequency-division multiplexing (FDM) 2) Wavelength-division multiplexing (WDM) and 3) Time-division multiplexing (TDM).
- The first two techniques are used for analog-signals.
  The third one technique is used for digital-signals.

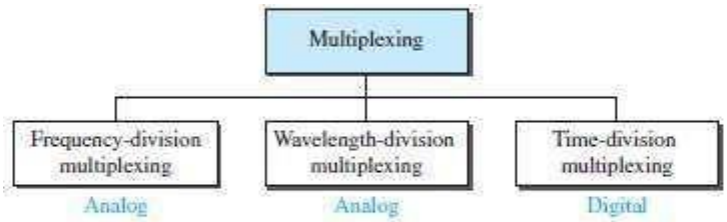


Figure 6.2 *Categories of multiplexing*

2.4.1 Frequency Division Multiplexing (FDM)

- FDM is an analog multiplexing technique that combines analog signals (Figure 6.3).
- FDM can be used when the bandwidth of a link is greater than the combined bandwidths of the signals to be transmitted. (Bandwidth measured in hertz).

Figure 6.3 *Frequency-division multiplexing*

2.4.1.1 Multiplexing Process

- Here is how it works (Figure 6.4):
  1) Each sending-device generates modulated-signals with different carrier-frequencies (f1, f2, & f3).
  2) Then, these modulated-signals are combined into a single multiplexed-signal. 3) Finally, the multiplexed-signal is transported by the link.

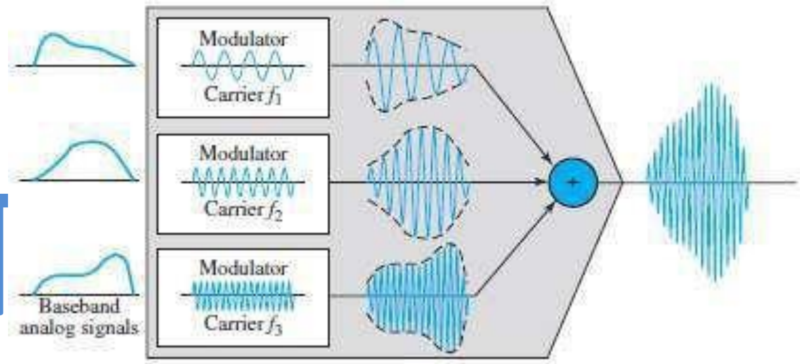


Figure 6.4 *FDM process*

- Carrier-frequencies are separated by sufficient bandwidth to accommodate the modulated-signal.
- Channels can be separated by strips of unused bandwidth called guard bands.
- Guard bands prevent signals from overlapping.
- In addition, carrier-frequencies must not interfere with the original data frequencies.
- Although FDM is considered as analog multiplexing technique, the sources can produce digital-signal.
- The digital-signal can be sampled, changed to analog-signal, and then multiplexed by using FDM.

2.4.1.2 Demultiplexing Process

• Here is how it works (Figure 6.5):

1) The demultiplexer uses filters to divide the multiplexed-signal into individual-signals.
2) Then, the individual signals are passed to a demodulator. 3) Finally, the demodulator
→ separates the individual signals from the carrier signals and → passes the individual signals to the output-lines.

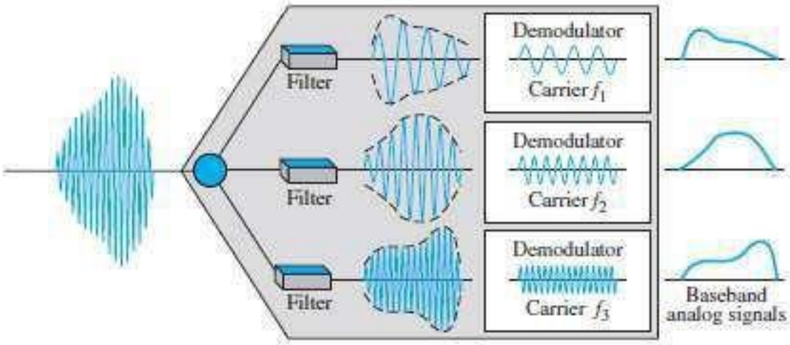


Figure 6.5 *FDM demultiplexing example*

Example 2.12

Assume that a voice channel occupies a bandwidth of 4 kHz. We need to combine three voice channels into a link with a bandwidth of 12 kHz, from 20 to 32 kHz. Show the configuration, using the frequency domain. Assume there are no guard bands.

Solution

We shift (modulate) each of the three voice channels to a different bandwidth, as shown in Figure 6.6. We use the 20- to 24-kHz bandwidth for the first channel, the 24- to 28-kHz bandwidth for the second channel, and the 28- to 32-kHz bandwidth for the third one. Then we combine them as shown in Figure 6.6.

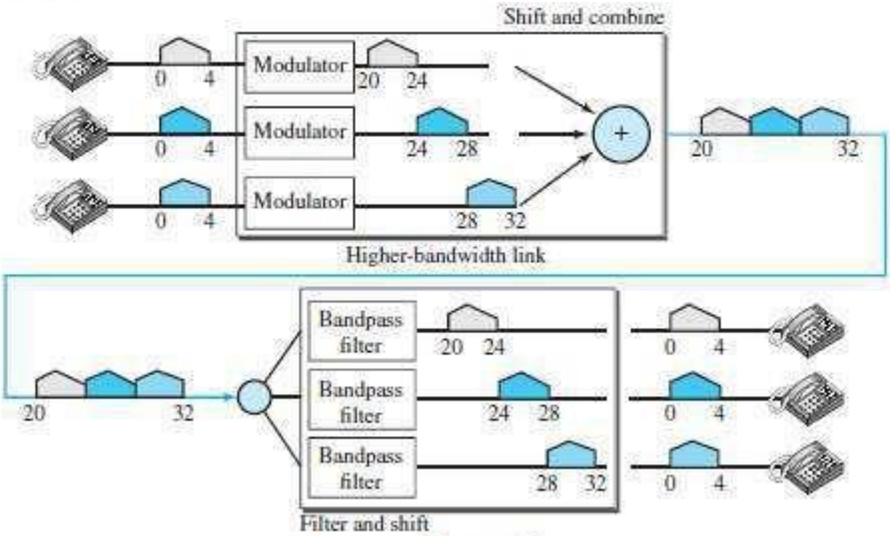


Figure 6.6

Example 2.13

Five channels, each with a 100-kHz bandwidth, are to be multiplexed together. What is the minimum bandwidth of the link if there is a need for a guard band of 10 kHz between the channels to prevent interference?

Solution

For five channels, we need at least four guard bands. This means that the required bandwidth is at least $5 \times 100 + 4 \times 10 = 540$ kHz, as shown in Figure 6.7.

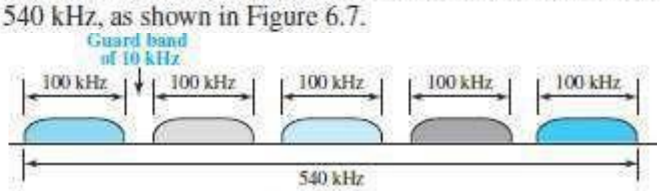


Figure 6.7

Example 2.14

Four data channels (digital), each transmitting at 1 Mbps, use a satellite channel of 1 MHz. Design an appropriate configuration, using FDM.

Solution

The satellite channel is analog. We divide it into four channels, each channel having a 250-kHz bandwidth. Each digital channel of 1 Mbps is modulated so that each 4 bits is modulated to 1 Hz. One solution is 16-QAM modulation. Figure 6.8 shows one possible configuration.

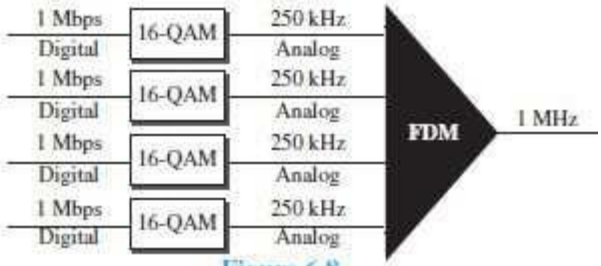


Figure 6.8

2.4.1.3 Applications of FDM

1) To maximize the efficiency of their infrastructure, Telephone-companies have traditionally multiplexed signals from lower-bandwidth lines onto higher-bandwidth lines.
2) A very common application of FDM is AM and FM radio broadcasting.
3) The first generation of cellular telephones (still in operation) also uses FDM.

2.4.1.4 Analog Carrier System

- To maximize the efficiency, telephone-companies have multiplexed-signals from lower-bandwidth lines onto higher-bandwidth lines.
- Many switched or leased lines are combined into bigger channels.
- For analog lines, FDM is used.
- One of these hierarchical systems used by AT&T is made up of (Figure 6.9):
  1) Groups
  2) Super groups
  3) Master groups, and
  4) Jumbo groups

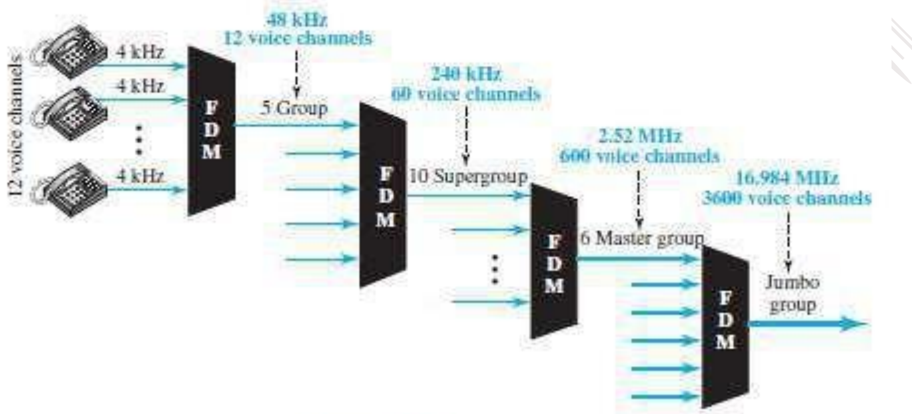


Figure 6.9 *Analog hierarchy*

1) Group: In the analog hierarchy, 12 voice channels are multiplexed onto a higher-bandwidth line to create a group.

- A group has 48 kHz of bandwidth and supports 12 voice channels.

2) Super Group: At the next level, up to five groups can be multiplexed to create a compositesignal called a supergroup.

- A supergroup has a bandwidth of 240 kHz and supports up to 60 voice channels.
- Supergroups can be made up of either five groups or 60 independent voice channels.

3) Master Groups: At the next level, 10 supergroups are multiplexed to create a master group.

- A master group must have 2.40 MHz of bandwidth, but the need for guard bands between the supergroups increases the necessary bandwidth to 2.52 MHz.
- Master groups support up to 600 voice channels.

4) Jumbo Group: Finally, six master groups can be combined into a jumbo group.

- A jumbo group must have 15.12 MHz (6 x 2.52 MHz) of bandwidth, but the need for guard bands b/w the master groups increases the necessary bandwidth to 16.984 MHz

Example 2.15

The Advanced Mobile Phone System (AMPS) uses two bands. The first band of 824 to 849 MHz is used for sending, and 869 to 894 MHz is used for receiving. Each user has a bandwidth of 30 kHz in each direction. The 3-kHz voice is modulated using FM, creating 30 kHz of modulated signal. How many people can use their cellular phones simultaneously?

Solution

Each band is 25 MHz. If we divide 25 MHz by 30 kHz, we get 833.33. In reality, the band is divided into 832 channels. Of these, 42 channels are used for control, which means only 790 channels are available for cellular phone users.

2.4.2 Wavelength Division Multiplexing (WDM)

- WDM is an analog multiplexing technique that combines analog signals (Figure 6.10).
- WDM is designed to use the high-data-rate capability of fiber optical-cable.
- The data-rate of optical-cable is higher than the data-rate of metallic-cable.
- Using an optical-cable for one single line wastes the available bandwidth.
- Multiplexing allows combining several lines into one line.
- WDM is same as FDM with 2 exceptions:
  1) Multiplexing & demultiplexing involve optical-signals transmitted through optical-cable.
  2) The frequencies are very high.

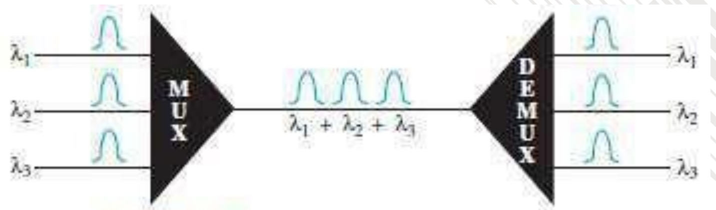


Figure 6.10 *Wavelength-division multiplexing*

- Here is how it works (Figure 6.11):
  - A multiplexer combines several narrow-bands of light into a wider-band of light.
  - A demultiplexer divides a wider-band of light into several narrow-bands of light.
  - A prism is used for combining and splitting of light sources
  - A prism bends a beam of light based on → angle of incidence and → frequency.

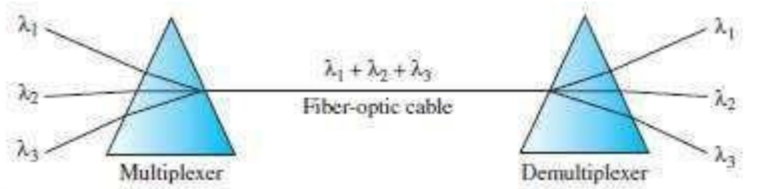


Figure 6.11 *Prisms in wavelength-division multiplexing and demultiplexing*

- Applications of WDM:
    1) SONET network: Multiple optical-fiber lines can be multiplexed and demultiplexed.
    2) Dense WDM (DWDM) can multiplex a very large number of channels by spacing channels very close to one another. DWDM achieves even greater efficiency

2.4.3 Time Division Multiplexing (TDM)

- TDM is a digital multiplexing technique that combines digital signals (Figure 6.12).
- TDM combines several low-rate channels into one high-rate one.
- FDM vs. TDM
    1) In FDM, a portion of the bandwidth is shared. 2) In TDM, a portion of the time is shared.
- Each connection occupies a portion of time in the link.
- Several connections share the high bandwidth of a line.

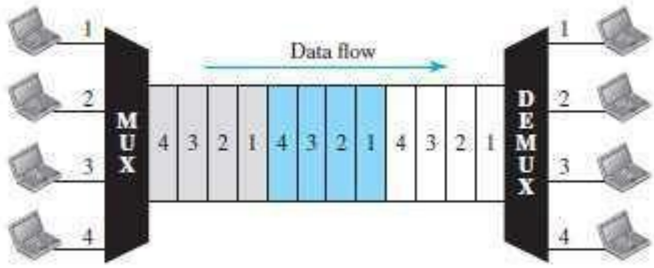


Figure 6.12 *TDM*

- As shown in Figure 6.12, the link is divided by time.
- Portions of signals 1, 2, 3, and 4 occupy the link sequentially.
- Digital-data from different sources are combined into one timeshared link.
- Although TDM is considered as digital multiplexing technique, the sources can produce analog-signal.
- The analog data can be sampled, changed to digital-data, and then multiplexed by using TDM.
- Two types of TDM: 1) Synchronous and 2) Statistical.

2.4.3.1 Synchronous TDM

2.4.3.1.1 Time Slots & Frames

- Each input-connection has an allotment in the output-connection even if it is not sending data.
- The data-flow of input-connection is divided into units (Figure 6.13).
- A unit can be 1 bit, 1 character, or 1 block of data.
- Each input-unit occupies one input-time-slot.
- Each input-unit

  → becomes one output-unit and

  → occupies one output-time-slot.
- However, duration of output-time-slot is „n" times shorter than duration of input-time-slot. • If an input-time-slot is T s, the output-time-slot is T/n s where n = No. of connections.
- In the output-connection, a unit has a shorter duration & therefore travels faster.

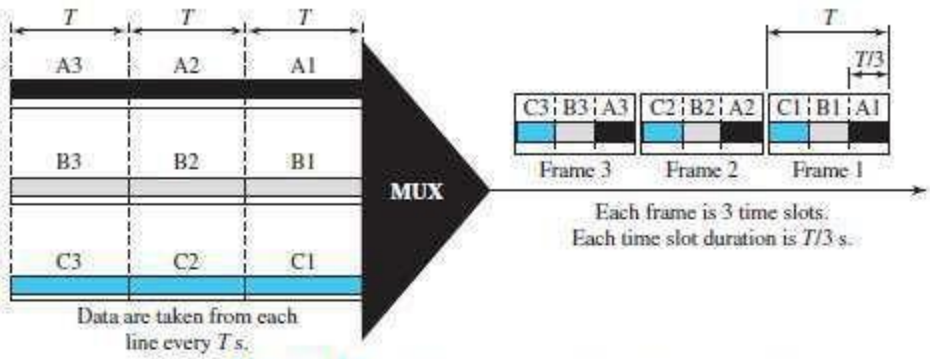


Figure 6.13 *Synchronous time-division multiplexing*

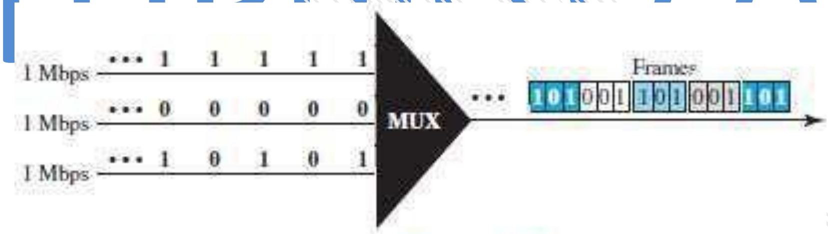


Figure 6.14

- In Figure 6.14, n = 3.
- A set of data-units from each input-connection is grouped into a frame.
- For example:

  If there are 3 connections, a frame is divided into 3 time-slots.

  One slot is allocated for each data-unit.

  One data-unit is used for each input-line.

Example 2.16

In Figure 6.13, the data rate for each input connection is 1 kbps. If 1 bit at a time is multiplexed (a unit is 1 bit), what is the duration of

1. each input slot,
2. each output slot, and
3. each frame?

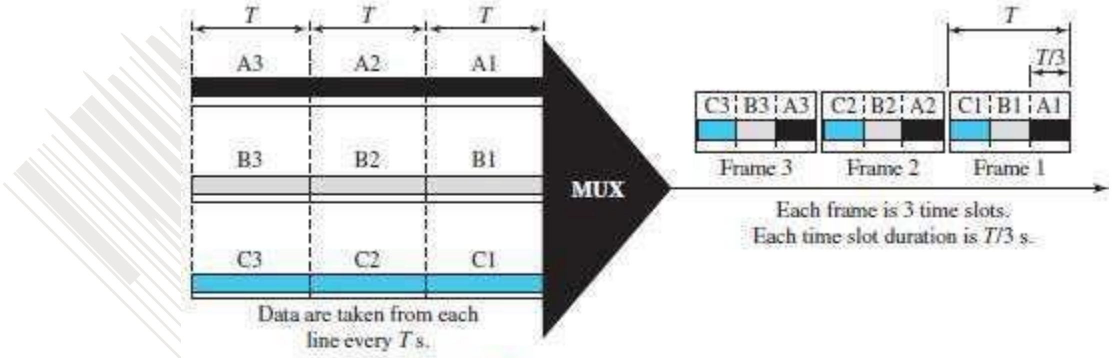


Figure 6.13 *Synchronous time-division multiplexing*

Solution

We can answer the questions as follows:

1. The data rate of each input connection is 1 kbps. This means that the bit duration is 1/1000 s or 1 ms. The duration of the input time slot is 1 ms (same as bit duration).
2. The duration of each output time slot is one-third of the input time slot. This means that the duration of the output time slot is 1/3 ms.
3. Each frame carries three output time slots. So the duration of a frame is $3 \times 1/3$ ms, or 1 ms. The duration of a frame is the same as the duration of an input unit.

Example 2.17

Figure 6.14 shows synchronous TDM with a data stream for each input and one data stream for the output. The unit of data is 1 bit. Find (1) the input bit duration, (2) the output bit duration, (3) the output bit rate, and (4) the output frame rate.

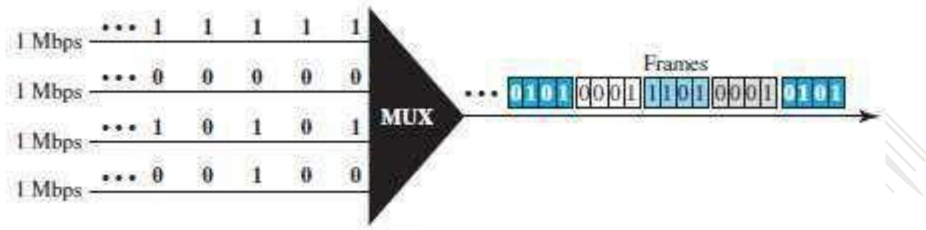


Figure 6.14

Solution

We can answer the questions as follows:

1. The input bit duration is the inverse of the bit rate: 1/1 Mbps = 1 µs.
2. The output bit duration is one-fourth of the input bit duration, or 1/4 µs.
3. The output bit rate is the inverse of the output bit duration, or 1/4 µs, or 4 Mbps. This can also be deduced from the fact that the output rate is 4 times as fast as any input rate; so the output rate = 4×1 Mbps = 4 Mbps.
4. The frame rate is always the same as any input rate. So the frame rate is 1,000,000 frames per second. Because we are sending 4 bits in each frame, we can verify the result of the previous question by multiplying the frame rate by the number of bits per frame.

Example 2.18

Four 1-kbps connections are multiplexed together. A unit is 1 bit. Find (1) the duration of 1 bit before multiplexing, (2) the transmission rate of the link, (3) the duration of a time slot, and (4) the duration of a frame.

Solution

We can answer the questions as follows:

1. The duration of 1 bit before multiplexing is 1/1 kbps, or 0.001 s (1 ms).
2. The rate of the link is 4 times the rate of a connection, or 4 kbps.
3. The duration of each time slot is one-fourth of the duration of each bit before multiplexing, or 1/4 ms or 250 μs. Note that we can also calculate this from the data rate of the link, 4 kbps. The bit duration is the inverse of the data rate, or 1/4 kbps or 250 μs.
4. The duration of a frame is always the same as the duration of a unit before multiplexing, or 1 ms. We can also calculate this in another way. Each frame in this case has four time slots. So the duration of a frame is 4 times 250 μs, or 1 ms.

2.4.3.1.2 Interleaving

- TDM can be seen as 2 fast-rotating switches (Figure 6.15): 1) First switch on the multiplexing-side and

  2) Second switch on the demultiplexing-side.
- The switches are synchronized and rotate at the same speed, but in opposite directions.

  1) On the multiplexing-side (Figure 6.16)

   As the switch opens in front of a connection, that connection has the opportunity to send a unit onto the path. This process is called *interleaving*.

  2) On the demultiplexing-side

   As the switch opens in front of a connection, that connection has the opportunity to receive a unit from the path.

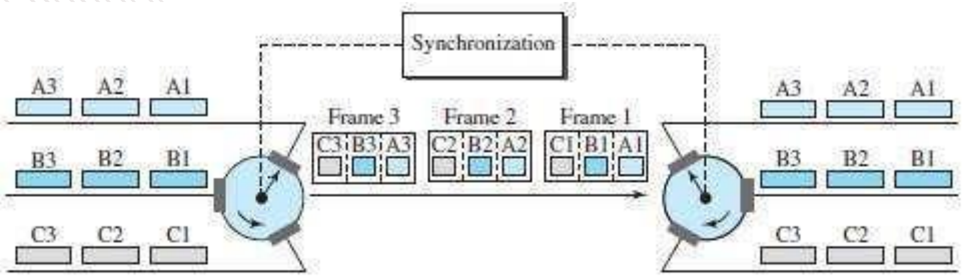


Figure 6.15 *Interleaving*

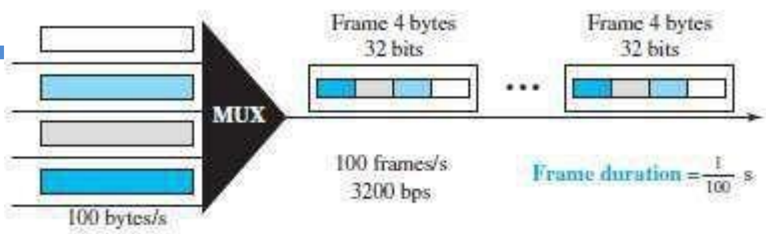


Figure 6.16

Example 2.19

Four channels are multiplexed using TDM. If each channel sends 100 bytes/s and we multiplex 1 byte per channel, show the frame traveling on the link, the size of the frame, the duration of a frame, the frame rate, and the bit rate for the link.

Solution

The multiplexer is shown in Figure 6.16. Each frame carries 1 byte from each channel; the size of each frame, therefore, is 4 bytes, or 32 bits. Because each channel is sending 100 bytes/s and a frame carries 1 byte from each channel, the frame rate must be 100 frames per second. The duration of a frame is therefore 1/100 s. The link is carrying 100 frames per second, and since each frame contains 32 bits, the bit rate is 100 × 32, or 3200 bps. This is actually 4 times the bit rate of each channel, which is 100 × 8 = 800 bps.

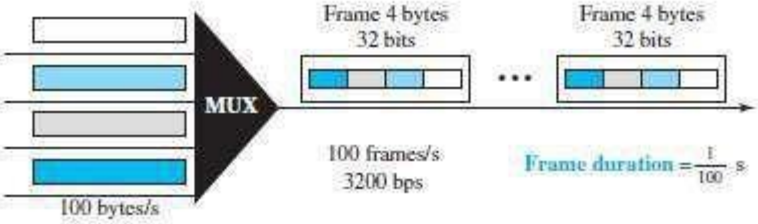


Figure 6.16

Example 2.20

A multiplexer combines four 100-kbps channels using a time slot of 2 bits. Show the output with four arbitrary inputs. What is the frame rate? What is the frame duration? What is the bit rate? What is the bit duration?

Solution

Figure 6.17 shows the output for four arbitrary inputs. The link carries 50,000 frames per second since each frame contains 2 bits per channel. The frame duration is therefore 1/50,000 s or 20 μs. The frame rate is 50,000 frames per second, and each frame carries 8 bits; the bit rate is 50,000 × 8 = 400,000 bits or 400 kbps. The bit duration is 1/400,000 s, or 2.5 μs. Note that the frame duration is 8 times the bit duration because each frame is carrying 8 bits.

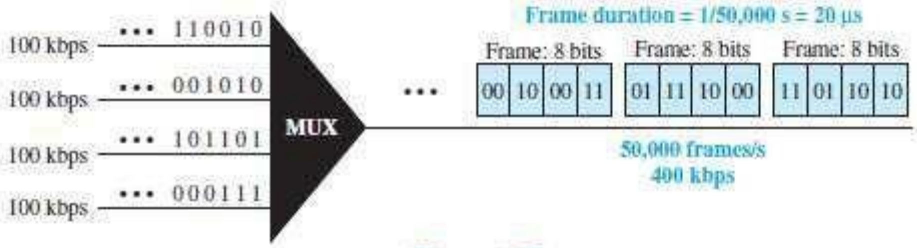


Figure 6.17

2.4.3.1.3 Empty Slots

- Problem: Synchronous TDM is not efficient.

  For example: If a source does not have data to send, the corresponding slot in the output-frame is empty.

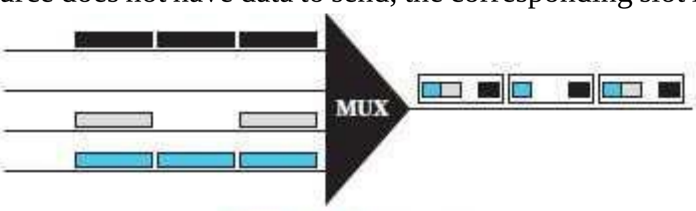


Figure 6.18 *Empty slots*

- As shown in Figure 6.18,

Second input-line has no data to send Third input-line has discontinuous data.

- The first output-frame has 3 slots filled.
  The second frame has 2 slots filled.
  The third frame has 3 slots filled.
  No frame is full.
- Solution: Statistical TDM can improve the efficiency by removing the empty slots from the frame.

2.4.3.1.4 Data Rate Management

- Problem in TDM: How to handle differences in the input data-rates?
- If data-rates are not the same, three strategies can be used.
- Three different strategies: 1) Multilevel multiplexing 2) Multiple-slot allocation and 3) Pulse stuffing

1) Multilevel Multiplexing

- This technique is used when the data-rate of an input-line is a multiple of others.

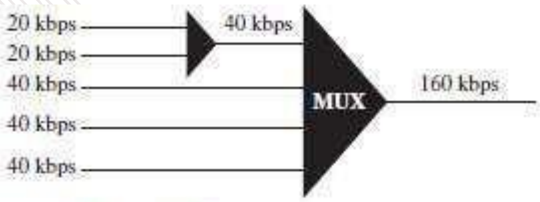


Figure 6.19 *Multilevel multiplexing*

- For example:
- As shown in Figure 6.19, we have 2 inputs of 20 kbps and 3 inputs of 40 kbps. The first 2 input-lines can be multiplexed to provide a data-rate of 40 kbps.

2) Multiple Slot Allocation

- Sometimes it is more efficient to allot more than 1 slot in a frame to a single input-line.

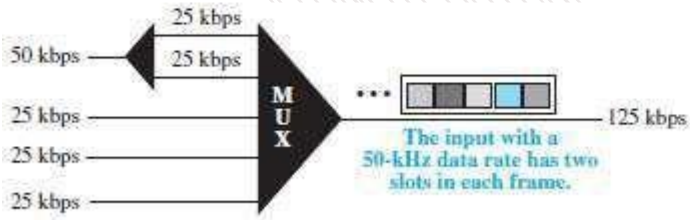


Figure 6.20 *Multiple-slot multiplexing*

- For example:
  Data-rate of multiple input-lines can be data-rate of one input-line.
- As shown in Figure 6.20, the input-line with a 50-kbps data-rate can be given 2 slots in the output-line.
- In first input line, serial-to-parallel converter is used. The converter creates two 25 kbps input lines out of one 50 kbps input line.

3) Pulse Stuffing

- Sometimes the bit-rates of sources are not multiple integers of each other. ∴ above 2 techniques cannot be used.
- Solution:

→ Make the highest input data-rate the dominant data-rate. → Then, add dummy bits to the input-lines with lower rates. → This will increase data rates of input-line.
→ This technique is called pulse stuffing, bit padding, or bit stuffing.

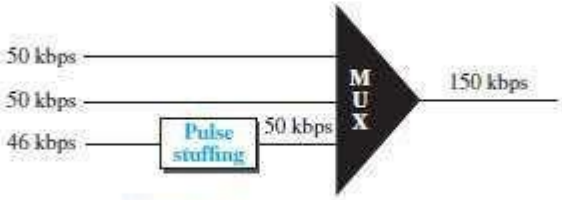


Figure 6.21 *Pulse stuffing*

- As shown in Figure 6.21, the input-line with a lower data-rate = 46kbps is pulse-stuffed to increase the data-rate to 50 kbps.
- Now, multiplexing can take place.

2.4.3.1.5 Frame Synchronizing

• Problem: Synchronization between the multiplexer and demultiplexer is a major issue.
If the multiplexer and the demultiplexer are not synchronized, a bit belonging to one channel may be received by the wrong channel.

Solution: Usually, one or more synchronization-bits are added to the beginning of each frame. These bits are called *framing-bits*.

The framing-bits follow a pattern (frame-to-frame) that allows multiplexer and demultiplexer to synchronize.

As shown in Figure 6.22, the synchronization-information
→ consists of 1 bit per frame and → alternates between 0 & 1.

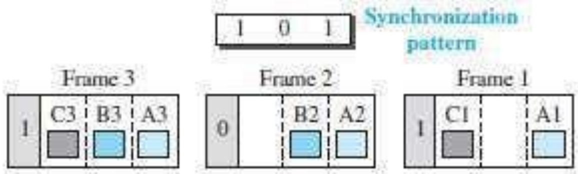


Figure 6.22 *Framing bits*

Example 2.21

We have four sources, each creating 250 characters per second. If the interleaved unit is a character and 1 synchronizing bit is added to each frame, find (1) the data rate of each source, (2) the duration of each character in each source, (3) the frame rate, (4) the duration of each frame, (5) the number of bits in each frame, and (6) the data rate of the link.

Solution

We can answer the questions as follows:

1. The data rate of each source is $250 \times 8 = 2000$ bps $= 2$ kbps.
2. Each source sends 250 characters per second; therefore, the duration of a character is 1/250 s, or 4 ms.
3. Each frame has one character from each source, which means the link needs to send 250 frames per second to keep the transmission rate of each source.
4. The duration of each frame is 1/250 s, or 4 ms. Note that the duration of each frame is the same as the duration of each character coming from each source.
5. Each frame carries 4 characters and 1 extra synchronizing bit. This means that each frame is $4 \times 8 + 1 = 33$ bits.
6. The link sends 250 frames per second, and each frame contains 33 bits. This means that the data rate of the link is 250×33, or 8250 bps. Note that the bit rate of the link is greater than the combined bit rates of the four channels. If we add the bit rates of four channels, we get 8000 bps. Because 250 frames are traveling per second and each contains 1 extra bit for synchronizing, we need to add 250 to the sum to get 8250 bps.

Example 2.22

Two channels, one with a bit rate of 100 kbps and another with a bit rate of 200 kbps, are to be multiplexed. How this can be achieved? What is the frame rate? What is the frame duration? What is the bit rate of the link?

Solution

We can allocate one slot to the first channel and two slots to the second channel. Each frame carries 3 bits. The frame rate is 100,000 frames per second because it carries 1 bit from the first channel. The frame duration is 1/100,000 s, or 10 ms. The bit rate is 100,000 frames/s $\times$ 3 bits per frame, or 300 kbps. Note that because each frame carries 1 bit from the first channel, the bit rate for the first channel is preserved. The bit rate for the second channel is also preserved because each frame carries 2 bits from the second channel.

2.4.3.2 Statistical TDM

- Problem: Synchronous TDM is not efficient.

  For ex: If a source does not have data to send, the corresponding slot in the output-frame is empty.

  Solution: Use statistical TDM.

  Slots are dynamically allocated to improve bandwidth-efficiency.

  Only when an input-line has data to send, the input-line is given a slot in the output-frame.
- The number of slots in each frame is less than the number of input-lines.
- The multiplexer checks each input-line in round robin fashion.

  If the line has data to send;

  Then, multiplexer allocates a slot for an input-line;

  Otherwise, multiplexer skips the line and checks the next line.

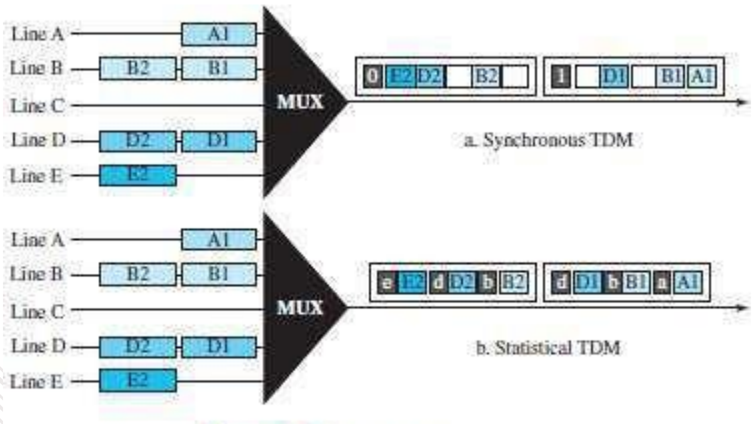


Figure 6.26 *TDM slot comparison*

- In synchronous TDM (Figure 6.26a), some slots are empty because the corresponding line does nothave data to send.
- In statistical TDM (Figure 6.26b), no slot is left empty.

1) Addressing

Synchronous TDM	Statistical TDM
An output-slot needs to carry only data of the destination (Figure 6.26a).	An output-slot needs to carry both data & address of the destination (Figure 6.26b).
There is no need for addressing. Synchronization and pre-assigned relationships between the inputs and outputs serve as an address.	There is no fixed relationship between the inputs and outputs because there are no pre- assigned or reserved slots. We need to include the address of the receiver inside each slot to show where it is to be delivered.

2) Slot Size

- Usually, a block of data is many bytes while the address is just a few bytes. A slot carries both data and address.
- Therefore, address-size must be very small when compared to data-size. This results in efficient transmission.
- For example:

  It will be inefficient to send 1 bit per slot as data, when the address is 3 bits.

  This means an overhead of 300%.

3) No Synchronization Bit

- In statistical TDM, the frames need not be synchronized, so synchronization-bits are not needed.

4) Bandwidth

- Normally, the capacity of the link is less than the sum of the capacities of each channel. The designers define the capacity of the link based on the statistics of the load for each channel.

2.5 SPREAD SPECTRUM

- Spread-spectrum is used in wireless applications (Figure 6.27).
- In wireless applications, all stations use air (or a vacuum) as the medium for communication.
- Goal: Stations must be able to share the air medium without interception by an attacker.

Solution: Spread-spectrum techniques add redundancy i.e. they spread the original spectrum needed for each station.

- If the required bandwidth for each station is B, spread-spectrum expands it to B_{ss} such that $B_{ss}>>B$.
- The expanded-bandwidth allows the source to place its message in a protective envelope for a more secure transmission.

  (An analogy is the sending of a delicate, expensive gift. We can insert the gift in a special box to prevent it from being damaged during transportation).

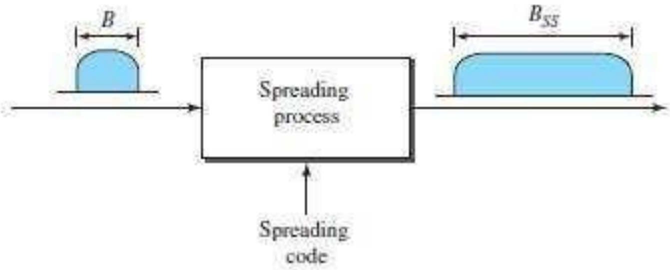


Figure 6.27 *Spread spectrum*

- Spread-spectrum achieves its goal through 2 principles:
  1) The bandwidth allocated to each station needs to be, by far, larger than what is needed. This allows redundancy.
  2) The spreading process must occur after the signal is created by the source.
- After the signal is created by the source, the spreading process → uses a spreading-code and → spreads the bandwidth.
- The spreading-code is a series of numbers that look random, but are actually a pattern.
- Two types of spread-spectrum:
  1) Frequency hopping spread-spectrum (FHSS) and 2) Direct sequence spread-spectrum (DSSS).

2.5.1 Frequency Hopping Spread Spectrum (FHSS)

- This technique uses „M" different carrier-frequencies that are modulated by the source-signal.
- At one moment, the signal modulates one carrier-frequency.

  At the next moment, the signal modulates another carrier-frequency.
- Although the modulation is done using one carrier-frequency at a time, 'M" frequencies are used in the long run.
- The bandwidth occupied by a source is given by

$$B_{FHSS} >> B$$

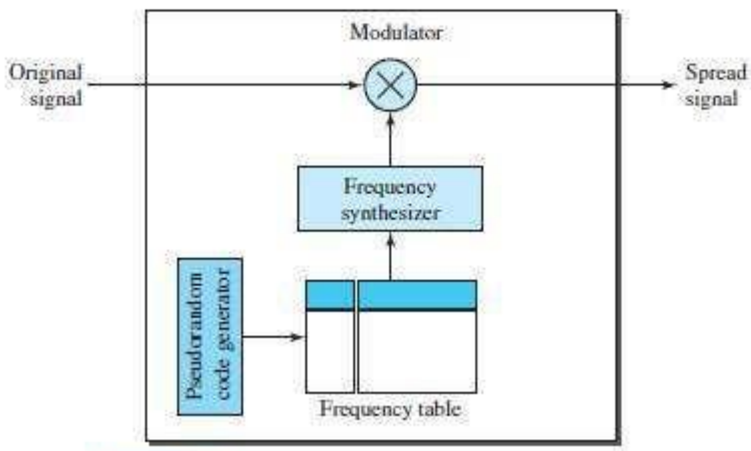


Figure 6.28 *Frequency hopping spread spectrum (FHSS)*

- As shown in Figure 6.28.
  - A pseudorandom code generator (PN) creates a k-bit pattern for every hopping period T_h.
  - The frequency-table
    - → uses the pattern to find the frequency to be used for this hopping period and → passes the frequency to the frequency-synthesizer.
  - The frequency-synthesizer creates a carrier-signal of that frequency.
  - The source-signal modulates the carrier-signal.

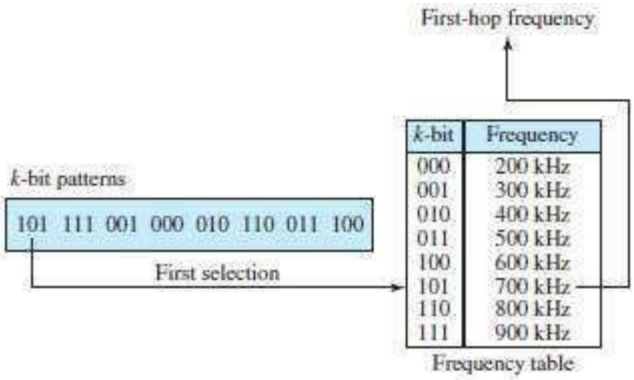


Figure 6.29 *Frequency selection in FHSS*

- As shown in Figure 6.29, assume we have 8 hopping frequencies.
  - Here, M = 8 and k = 3.
  - The pseudorandom code generator will create 8 different 3-bit patterns.
  - These are mapped to 8 different frequencies in the frequency table (see Figure 6.29).
  - The pattern for this station is 101, 111, 001, 000, 010, 111 & 100.
    1) At hopping-period 1, the pattern is 101.
       The frequency selected is 700 kHz; the source-signal modulates this carrier-frequency.
    2) At hopping-period 2, the pattern is 111.
       The frequency selected is 900 kHz; the source-signal modulates this carrier-frequency.

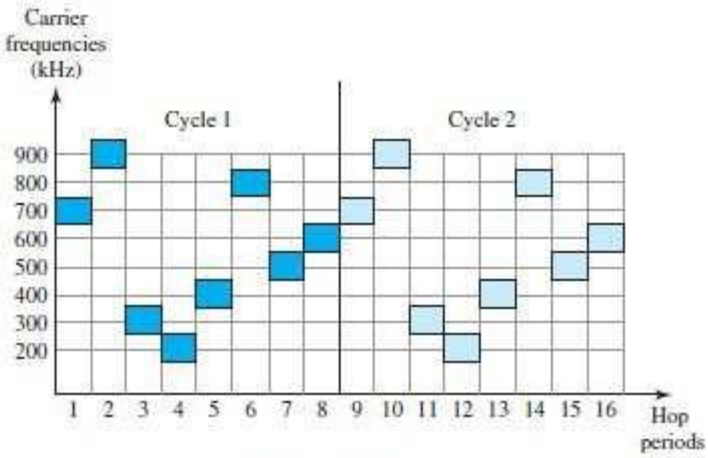


Figure 6.30 *FHSS cycles*

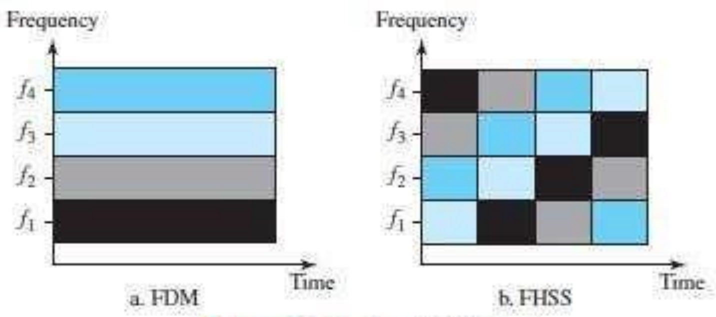


Figure 6.31 *Bandwidth sharing*

- If there are many k-bit patterns & the hopping period is short, a sender & receiver can have privacy.
  If an attacker tries to intercept the transmitted signal, he can only access a small piece of data because he does not know the spreading sequence to quickly adapt himself to the next hop.
- The scheme has also an anti-jamming effect.
  A malicious sender may be able to send noise to jam the signal for one hopping period (randomly), but not for the whole period.

2.5.1.1 Bandwidth Sharing

- If the number of hopping frequencies is M, we can multiplex M channels into one by using the same B_{ss} bandwidth.
- This is possible because
  1) A station uses just one frequency in each hopping period. 2)
  Other M-1 stations uses other M-1 frequencies.
- In other words, M different stations can use the same B_{ss} if a multiple FSK (MFSK) is used.

2.5.2 Direct Sequence Spread Spectrum (DSSS)

- This technique expands the bandwidth of the original signal.
- Each data-bit is replaced with „n" bits using a spreading-code.
- Each bit is assigned a code of „n" bits called chips.
- The chip-rate is „n" times that of the data-bit (Figure 6.32).

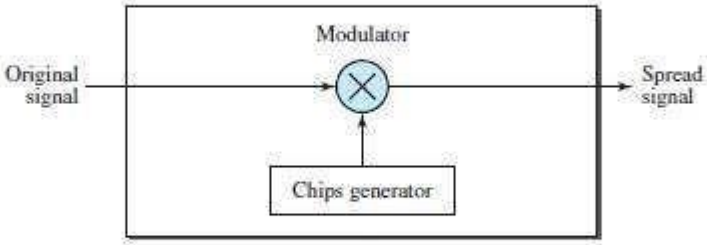


Figure 6.32 *DSSS*

- For example (Figure 6.33):
  - Consider the Barker sequence used in a wireless LAN. Here n =11.
  - Assume that the original signal and the chips in the chip-generator use polar NRZ encoding. □ The spreading-code is 11 chips having the pattern 10110111000.
  - If the original signal-rate is N, the rate of the spread signal is 1/N.
  - This means that the required bandwidth for the spread signal is 11 times larger than the bandwidth of the original signal.

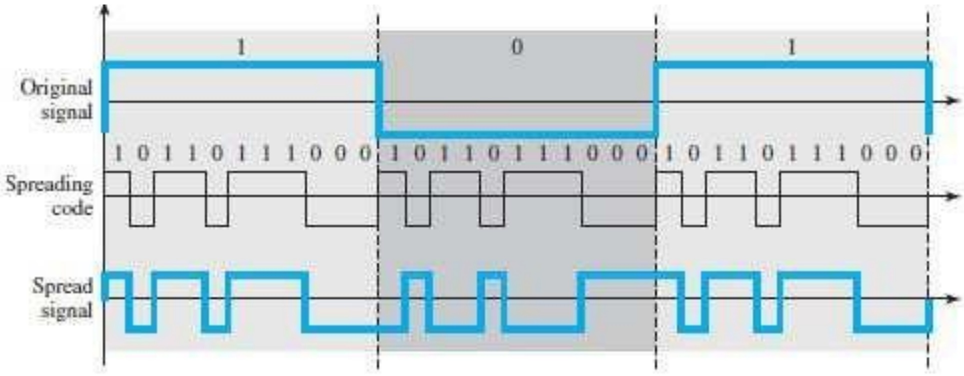


Figure 6.33 DSSS example

- The spread signal can provide privacy if the attacker does not know the code.
- It can also provide immunity against interference if each station uses a different code.

2.5.2.1 Bandwidth Sharing

- Can we share a bandwidth in DSSS?
- The answer is no and yes.

1) If we use a spreading-code that spreads signals that cannot be combined and separated, we cannot share a bandwidth.

 For example:

Some wireless LANs use DSSS and the spread bandwidth cannot be shared.

2) If we use a special spreading-code that spreads signals that can be combined and separated, we can share a bandwidth.

 For example:

Cellular telephony uses DSSS and the spread bandwidth is shared b/w several users.

SWITCHING

2.6 SWITCHING

- A network is a set of connected-devices.
- Problem: Whenever we have multiple-devices, we have the problem of how to connect them to make one-to-one communication possible.
- Solution: Use Switching.
- A switched-network consists of a series of interlinked-nodes, called switches.
- Switches are devices capable of creating temporary connections between two or more devices.
- In a switched-network,

1) Some nodes are connected to the end-systems (For example: PC or TP). 2) Some nodes are used only for routing.

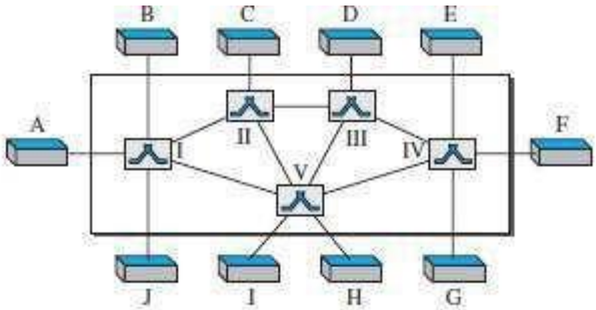


Figure 8.1 Switched network

- As shown in Figure 8.1,
  1) The end-systems are labeled A, B, C, D, and so on.
  2) The switches are labeled I, II, III, IV, and V. Each switch is connected to multiple links.

2.6.1 Three Methods of Switching

- Three methods of Switching are (Figure 8.2):
  1) Circuit Switching 2)
  Packet Switching and
  3) Message Switching.
- The first two are commonly used today.
- The third has been phased out in general communications but still has networking applications.
- Packet switching can further be divided into two subcategories—virtual circuit approach and datagram approach

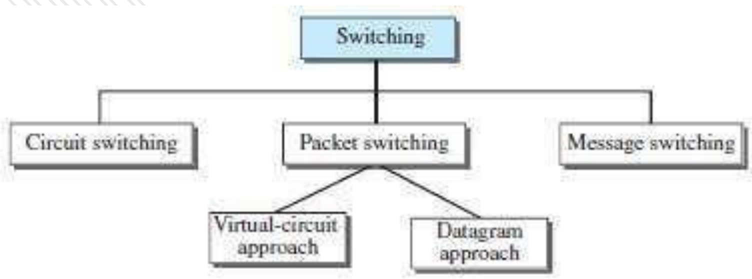


Figure 8.2 *Taxonomy of switched networks*

2.6.2 Switching and TCP/IP Layers

- Switching can happen at several layers of the TCP/IP protocol suite.

1) Switching at Physical Layer

- At the physical layer, we can have only circuit switching.
- There are no packets exchanged at the physical layer.
- The switches at the physical layer allow signals to travel in one path or another.

2) Switching at Data-Link Layer

- At the data-link layer, we can have packet switching.
- However, the term packet in this case means frames or cells.
- Packet switching at the data-link layer is normally done using a virtual-circuit approach.

3) Switching at Network Layer

- At the network layer, we can have packet switching.
- In this case, either a virtual-circuit approach or a datagram approach can be used.
- Currently the Internet uses a datagram approach, but the tendency is to move to a virtualcircuit approach.

4) Switching at Application Layer

- At the application layer, we can have only message switching.
- The communication at the application layer occurs by exchanging messages.
- Conceptually, we can say that communication using e-mail is a kind of message-switched communication, but we do not see any network that actually can be called a message-switched network.

2.7 CIRCUIT SWITCHED NETWORK

- This is similar to telephone system.

- Fixed path (connection) is established between a source and a destination prior to the transfer ofpackets.
- A circuit-switched-network consists of a set of switches connected by physical-links (Figure 8.3).
- A connection between 2 stations is a dedicated-path made of one or more links.
- However, each connection uses only one dedicated-channel on each link.
- Normally, each link is divided into „n" channels by using FDM or TDM.
- The resources need to be reserved during the setup phase.

The resources remain dedicated for the entire duration of data transfer until the teardown phase.

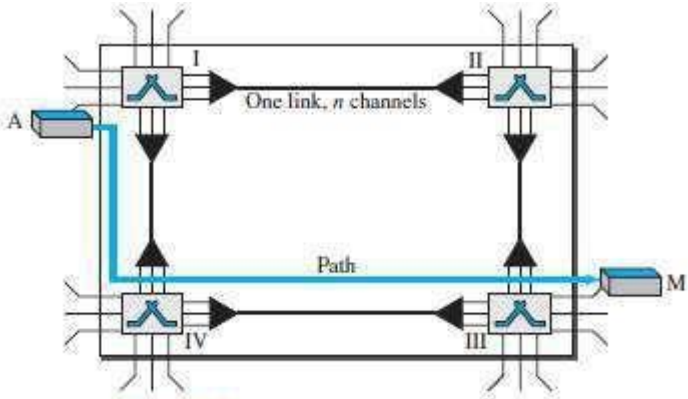


Figure 8.3 A trivial circuit-switched network

- The virtual-circuit setup procedure
  → first determines a path through the network &
  → sets parameters in the switches by exchanging connect-request & connect-confirm messages
- If a switch does not have enough resources to set up a virtual circuit, the switch responds with a connect-reject message and the setup procedure fails (Figure 7.15).
- A connection-release procedure may also be required to terminate the connection.

2.7.1 Three Phases

• The communication requires 3 phases: 1) Connection-setup
2) Data-transfer 3) Connection teardown.

1) Setup Phase

- Before the 2 parties can communicate, a dedicated-circuit needs to be established.
- Normally, the end-systems are connected through dedicated-lines to the switches.
  So, connection-setup means creating dedicated-channels between the switches.
- For ex: Assume system-A needs to connect to system-M. For this, following events occur:
  i) System-A sends a setup-request to switch-I.
  ii) Switch-I finds a channel between itself and switch-IV that can be dedicated for this purpose.
  iii) Switch-I then sends the request to switch-IV, which finds a dedicated-channel between itself and switch-III.
  iv) Switch-III informs system-M of system-A's intention at this time.
  v) Finally, an acknowledgment from system-M needs to be sent in the opposite direction to system-A.
- Only after system A receives this acknowledgment is the connection established.

2) Data Transfer Phase

• After the establishment of the dedicated-circuit (channels), the two parties can transfer data.

3) Teardown Phase

• When one of the parties needs to disconnect, a signal is sent to each switch to release the resources.

2.7.2 Efficiency

• Circuit-switched-networks are inefficient when compared to other two types of networks because 1) Resources are allocated during the entire duration of the connection. 2) These resources are unavailable to other connections.

2.7.3 Delay

• Circuit-switched-networks have minimum delay when compared to other two types of networks

• During data-transfer,

1) The data are not delayed at each switch.
2) The resources are allocated for the duration of the connection.

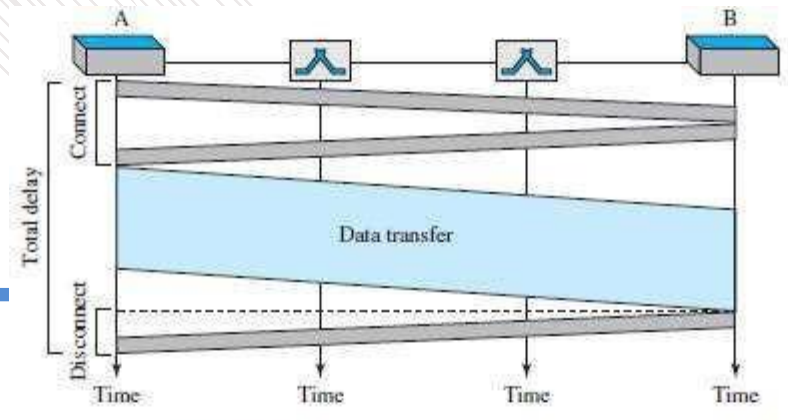


Figure 8.6 *Delay in a circuit-switched network*

• As in the above figure, there is no waiting time at each switch.

• The total delay is the time needed to

1) Create the connection 2) Transfer-data and
3) Disconnect the circuit.

• The delay caused by the setup is the sum of 4 parts:

1) The propagation time of the source-computer request.
2) The request signal transfer time.
3) The propagation time of the acknowledgment from the destination computer.
4) The signal transfer time of the acknowledgment.

• The delay due to data-transfer is the sum of 2 parts:

1) The propagation time.
2) Data-transfer time which can be very long.

2.8 PACKET SWITCHED NETWORK

• The message is divided into packets of fixed or variable size.

• The packet-size is determined by → network and
→ governing protocol.

• There is no resource reservation; resources are allocated on-demand.

2.8.1 Datagram Networks

- This is analogous to postal system.
- Each packet is routed independently through the network.
- Each packet has a header that contains source and destination addresses.
- Each switch examines the header to determine the next hop in the path to the destination.
- If the transmission line is busy then the packet is placed in the queue until the line becomes free.
- Packets are referred to as datagrams.
- Datagram switching is normally done at the network layer.
- In Internet, switching is done by using the datagram switching.
- Advantage:
  1) High utilization of transmission-line can be achieved by sharing among multiple packets.
- Disadvantages:
  1) Packets may arrive out-of-order, and re-sequencing may be required at the destination
  2) Loss of packets may occur when a switch has insufficient buffer

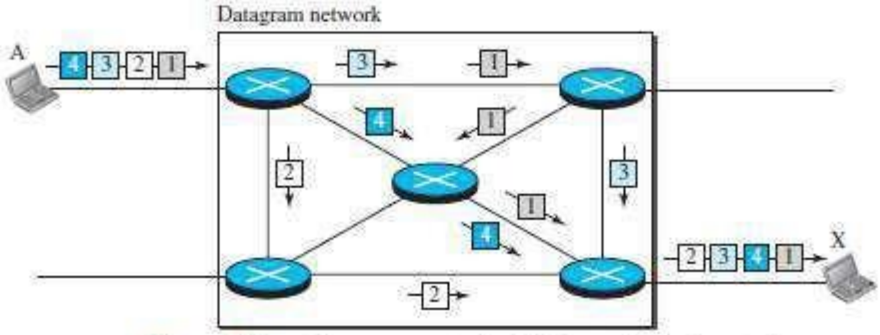


Figure 8.7 *A datagram network with four switches (routers)*

- The Figure 8.7 shows how the 4 packets are transferred from station-A to station-X. The switches are referred to as routers.
- All four packets (or datagrams) belong to the same message, but may travel different paths to reach their destination.
- This is so because the links may be involved in carrying packets from other sources and do not have the necessary bandwidth available to carry all the packets from A to X.
- This approach can cause the datagrams of a transmission to arrive at their destination out-of- order with different delays between the packets.
- Packets may also be lost or dropped because of a lack-of-resources.

It is the responsibility of an upper-layer protocol to
→ reorder the datagrams or →
ask for lost datagrams.

- The datagram-networks are referred to as connectionless networks. This is because 1) The switch does not keep information about the connection state.
  2) There are no setup or teardown phases.
  3) Each packet is treated the same by a switch regardless of its source or destination.

2.8.1.1 Routing Table

- Each switch has a routing-table which is based on the destination-address.
- The routing-tables are dynamic & updated periodically.

- The destination-addresses and the corresponding forwarding output-ports are recorded in the tables.

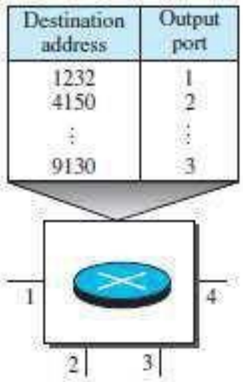


Figure 8.8 *Routing table in a datagram network*

2.8.1.1.1 Destination Address

- Every packet carries a header that contains the destination-address of the packet.
- When the switch receives the packet,

  → This destination-address is examined.

  → The routing-table is consulted to find the corresponding port through which the packet should be forwarded.
- The destination address in the header of a packet in remains the same during the entire journey of the packet.

2.8.1.1.2 Efficiency

- Datagram-networks are more efficient when compared to circuit-switched-network. This is because 1) Resources are allocated only when there are packets to be transferred.

  2) If a source sends a packet and there is a delay of a few minutes before another packet can be sent, the resources can be re-allocated during these minutes for other packets from other sources.

2.8.1.1.3 Delay

- Datagram-networks may have greater delay when compared to circuit-switched-network. This is because
  1) Each packet may experience a wait at a switch before it is forwarded.
  2) Since not all packets in a message necessarily travel through the same switches, the delay is notuniform for the packets of a message.

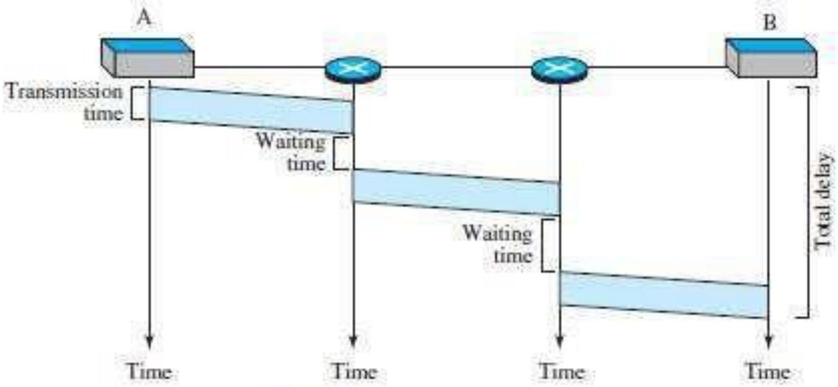


Figure 8.9 *Delay in a datagram network*

The Figure 8.9 gives an example of delay for one single packet. The packet travels through two switches.

There are three transmission times (3T), three propagation delays (slopes 3t of the lines), and two waiting times (W1+ W2).

$$\text{Total delay} = 3T + 3\tau + w_1 + w_2$$

2.8.2 Virtual Circuit Network (VCN) •

This is similar to telephone system.

- A virtual-circuit network is a combination of circuit-switched-network and datagram-network.
- Five characteristics of VCN:
  1) As in a circuit-switched-network, there are setup & teardown phases in addition to the data transfer phase.
  2) As in a circuit-switched-network, resources can be allocated during the setup phase.
     As in a datagram-network, resources can also be allocated on-demand.
  3) As in a datagram-network, data is divided into packets.
     Each packet carries an address in the header.
     However, the address in the header has local jurisdiction, not end-to-end jurisdiction. 4) As in a circuit-switched-network, all packets follow the same path established during the connection.
  5) A virtual-circuit network is implemented in the data link layer.
     A circuit-switched-network is implemented in the physical layer.
     A datagram-network is implemented in the network layer.

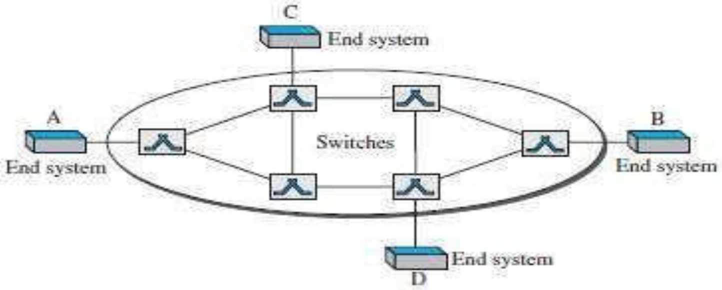


Figure 8.10 *Virtual-circuit network*

- The Figure 8.10 is an example of a virtual-circuit network.
- The network has switches that allow traffic from sources to destinations.
- A source or destination can be a computer, packet switch, bridge, or any other device that connects other networks.

2.8.2.1 Addressing

- Two types of addressing: 1) Global and 2) Local (virtual-circuit identifier).

1) Global Addressing

- A source or a destination needs to have a global address.
- Global address is an address that can be unique in the scope of the network or internationally if the network is part of an international network.

2) Virtual Circuit Identifier

- The identifier used for data-transfer is called the virtual-circuit identifier (VCI). A VCI, unlike a global address, is a small number that has only switch scope. VCI is used by a frame between two switches.
- When a frame arrives at a switch, it has a VCI.
  When the frame leaves, it has a different VCI.

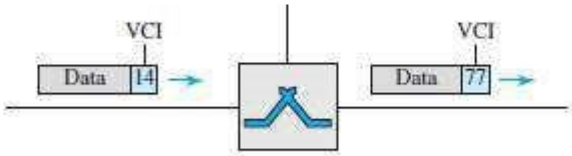


Figure 8.11 *Virtual-circuit identifier*

- Figure 8.11 show how the VCI in a data-frame changes from one switch to another.

2.8.2.2 Three Phases

- A source and destination need to go through 3 phases: setup, data-transfer, and teardown.
  1) In setup phase, the source and destination use their global addresses to help switchesmake table entries for the connection.
  2) In the teardown phase, the source and destination inform the switches to delete thecorresponding entry.
  3) Data-transfer occurs between these 2 phases.

2.8.2.2.1 Data Transfer Phase

- To transfer a frame from a source to its destination, all switches need to have a table-entry for this virtual-circuit.
- The table has four columns.
- The switch holds 4 pieces of information for each virtual-circuit that is already set up.

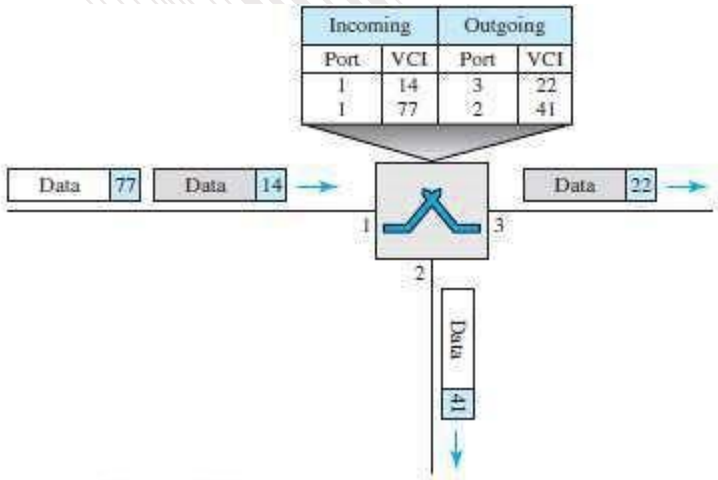


Figure 8.12 *Switch and tables in a virtual-circuit network*

- As shown in Figure 8.12, a frame arrives at port 1 with a VCI of 14.
- When the frame arrives, the switch looks in its table to find port 1 and a VCI of 14.
- When it is found, the switch knows to change the VCI to 22 & send out the frame from port 3.

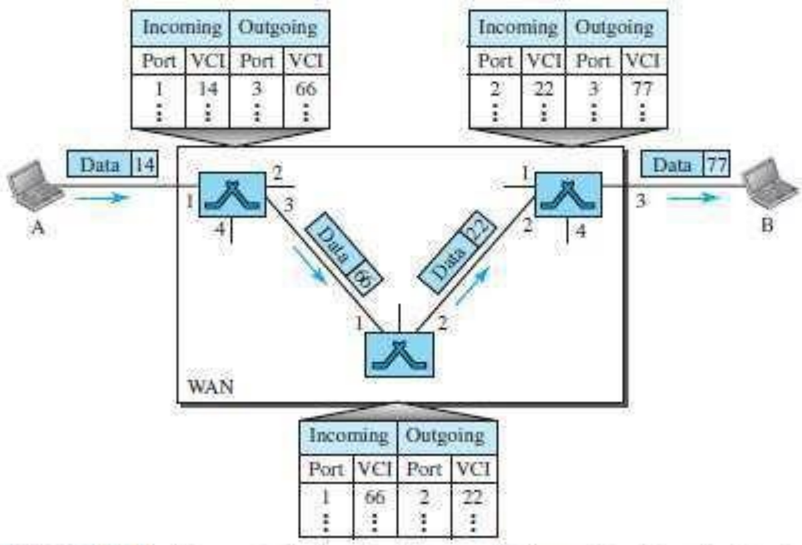


Figure 8.13 *Source-to-destination data transfer in a virtual-circuit network*

- As shown in Figure 8.13, each switch changes the VCI and routes the frame.
- The data-transfer phase is active until the source sends all its frames to the destination.
- The procedure at the switch is the same for each frame of a message.
- The process creates a virtual circuit, not a real circuit, between the source and destination.

2.8.2.2.2 Setup Phase

- A switch creates an entry for a virtual-circuit.
- For example, suppose source A needs to create a virtual-circuit to B. • Two steps are required: 1) Setup-request and 2) Acknowledgment.

1) Setup Request

- A setup-request frame is sent from the source to the destination (Figure 8.14).

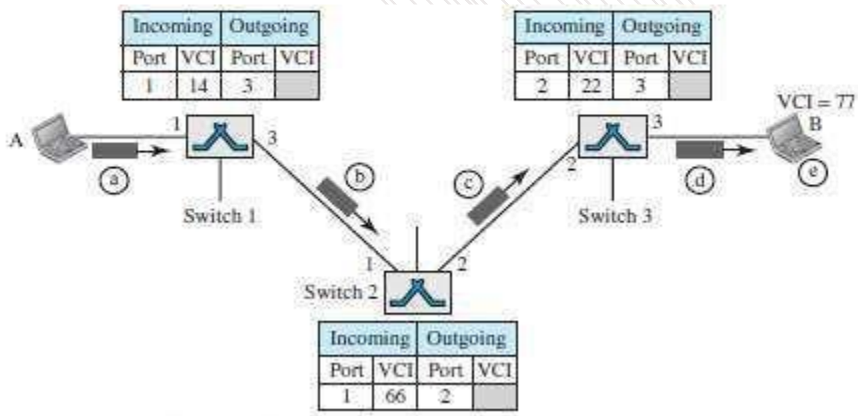


Figure 8.14 *Setup request in a virtual-circuit network*

- Following events occurs:
  - **a)** Source-A sends a setup-frame to switch-1.
  - **b)** Switch-1 receives the setup-frame.

    ¤ Switch-1 knows that a frame going from A to B goes out through port 3. ¤ The switch-1 has a routing table. ¤ The switch

    → creates an entry in its table for this virtual-circuit

    → is only able to fill 3 of the 4 columns.

    ¤ The switch

→ assigns the incoming port (1) and

→ chooses an available incoming-VCI (14) and the outgoing-port (3). → does not yet know the outgoing VCI, which will be found during the acknowledgment step.

¤ The switch then forwards the frame through port-3 to switch-2.

c) Switch-2 receives the setup-request frame.

¤ The same events happen here as at switch-1.

¤ Three columns of the table are completed: In this case, incoming port (1), incoming-VCI (66), and outgoing port (2).

d) Switch-3 receives the setup-request frame.

¤ Again, three columns are completed: incoming port (2), incoming-VCI (22), and outgoing-port (3).

e) Destination-B

→ receives the setup-frame

→ assigns a VCI to the incoming frames that come from A, in this case 77. ¤ This VCI lets the destination know that the frames come from A, and no other sources.

2) Acknowledgment

 A special frame, called the acknowledgment-frame, completes the entries in the switchingtables (Figure 8.15).

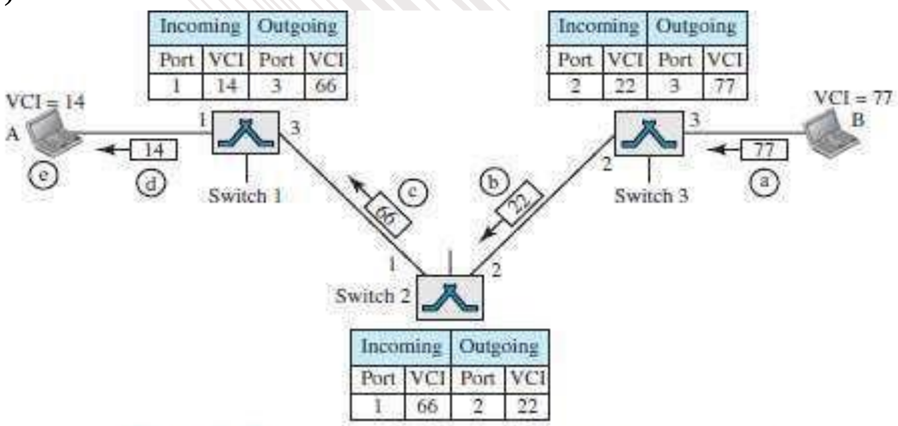


Figure 8.15 *Setup acknowledgment in a virtual-circuit network*

a) The destination sends an acknowledgment to switch-3.

¤ The acknowledgment carries the global source and destination-addresses so the switch knows which entry in the table is to be completed.

¤ The frame also carries VCI 77, chosen by the destination as the incoming-VCI for frames from A.

¤ Switch 3 uses this VCI to complete the outgoing VCI column for this entry.

b) Switch 3 sends an acknowledgment to switch-2 that contains its incoming-VCI in the table, chosen in the previous step.

¤ Switch-2 uses this as the outgoing VCI in the table.

c) Switch-2 sends an acknowledgment to switch-1 that contains its incoming-VCI in the table, chosen in the previous step.

¤ Switch-1 uses this as the outgoing VCI in the table.

d) Finally switch-1 sends an acknowledgment to source-A that contains its incoming-VCI in the table, chosen in the previous step.

e) The source uses this as the outgoing VCI for the data-frames to be sent to destination-B.

2.8.2.3 Teardown Phase

- Source-A, after sending all frames to B, sends a special frame called a teardown request.
- Destination-B responds with a teardown confirmation frame.
- All switches delete the corresponding entry from their tables.

2.8.2.4 Efficiency

- Resource reservation can be made in 2 cases:
  1) During the setup: Here, the delay for each packet is the same.
  2) On demand: Here, each packet may encounter different delays.
- Advantage of on demand resource allocation:

  The source can check the availability of the resources, without actually reserving it.

2.8.2.5 Delay in Virtual Circuit Networks

- There is a one-time delay for setup and a one-time delay for teardown (Figure 8.16).
- If resources are allocated during the setup phase, there is no wait time for individual packets.
- The packet is traveling through two switches (routers).
- There are three transmission times (3T), three propagation times (3τ), data transfer delay, a setup delay and a teardown delay.
- The total delay time is

$$\text{Total delay} + 3T + 3\tau + \text{setup delay} + \text{teardown delay}$$

Circuit Switching	Datagram Packet Switching	Virtual circuit Packet switching
Dedicate transmission path	No dedicate path	No dedicate path
Continuous transmission of data	Transmission of packets	Transmission of packets
Fast enough for interactive	Fast enough for interactive	Fast enough for interactive
Message are not stored	Packets may be stored until delivered	Packets stored until delivered
The path is established for entire conversation	Route established for each packet	Route established for entire conversation
Call setup delay; negligible transmission delay	Packet transmission delay	Call setup delay; Packet transmission delay
Busy signal if called party busy	Sender may be notified if packet not delivered	Sender notified of connection denial
Overload may block call setup; no delay for established calls	Overload increases packet delay	Overload may block call setup; increases packet delay
Electromechanical or computerized switching nodes	Small switching nodes	Small switching nodes
User responsible for message loss protection	Network may be responsible for individual packets	Network may be responsible for packet sequences
Usually no speed or code conversion	Speed and code conversion	Speed and code conversion
Fixed bandwidth	Dynamic use of bandwidth	Dynamic use of bandwidth
No overhead bits after call setup	Overhead bits in each packet	Overhead bits in each packet

Module 2

Digital Transmission

- A computer network is designed to send information from one point to another.
- This information needs to be converted to either a digital signal or an analog signal for transmission.
- In this chapter, we show the **schemes and techniques** that we use to transmit data digitally.
- First, we discuss **digital-to-digital conversion** techniques, methods which convert digital data to digital signals.
- Second, we discuss **analog-to-digital conversion** techniques, methods which change an analog signal to a digital signal.
- Thirdly we discuss **transmission modes**.
- Finally **digital-to-analog conversion** techniques

4-1 DIGITAL-TO-DIGITAL CONVERSION

In this section, we see how 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 may or may not be needed.

Topics discussed in this section:

- **Line Coding**
- **Line Coding Schemes**
- **Block Coding**
- **Scrambling**

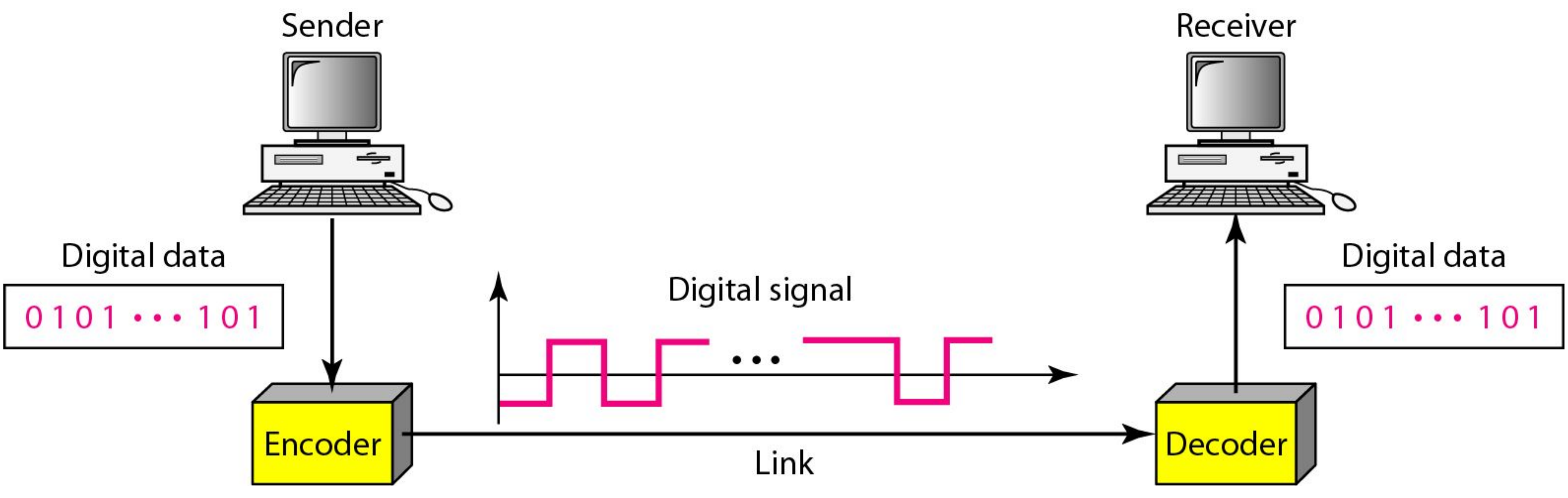
Line Coding

- Converting a string of 1’s and 0’s (digital data) into a sequence of signals that denote the 1’s and 0’s.
- For example a high voltage level (+V) could represent a “1” and a low voltage level (0 or -V) could represent a “0”.

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.

Figure 4.1 *Line coding and decoding*



Mapping Data symbols onto Signal levels

- A data symbol (or element) can consist of a number of data bits:
  - 1 , 0 or
  - 11, 10, 01, ……
- A data symbol can be coded into a single signal element or multiple signal elements
  - 1 -> +V, 0 -> -V
  - 1 -> +V and -V, 0 -> -V and +V
- The ratio ‘r’ is the number of data elements carried by a signal element.

Characteristics

Before discussing different line coding schemes, we address their common characteristics.

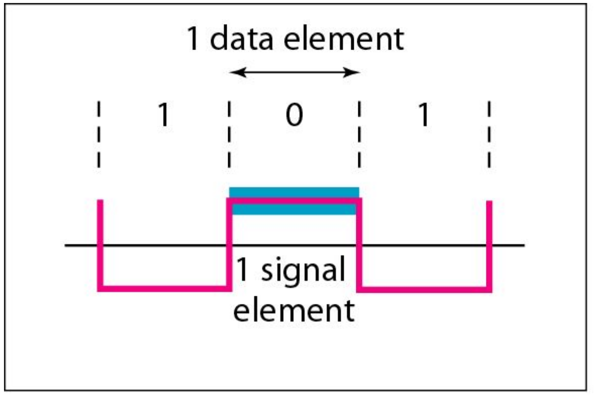
- **Signal Element Versus Data Element**

- Let us distinguish between a data element and a signal element.
- In data communications, our goal is to send data elements. A **data element** is the smallest entity that can represent a piece of information: **this is the bit.**
- In digital data communications, a **signal element** carries data elements.
- In other words, **data elements are what we need to send**; **signal elements are what we can send.**
- Data elements are being **carried**; signal elements are **the carriers**.

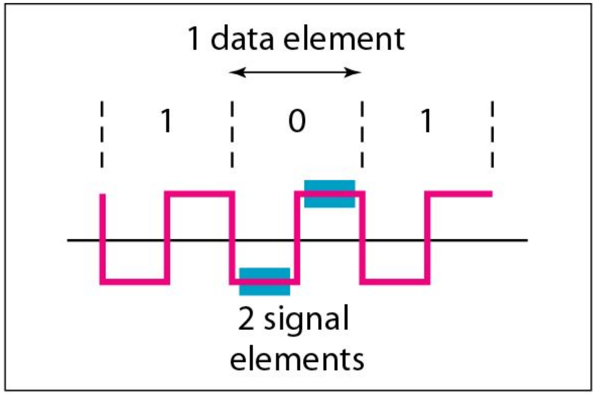
Relationship between data rate and signal rate

- The data rate defines the **number of bits sent per sec** - **bps**. It is often referred to the **bit rate.**
- The signal rate is the number of **signal elements sent in a second** and is measured in **bauds**. It is also referred to as the **modulation rate**.
- Goal is to increase the data rate whilst reducing the baud rate.

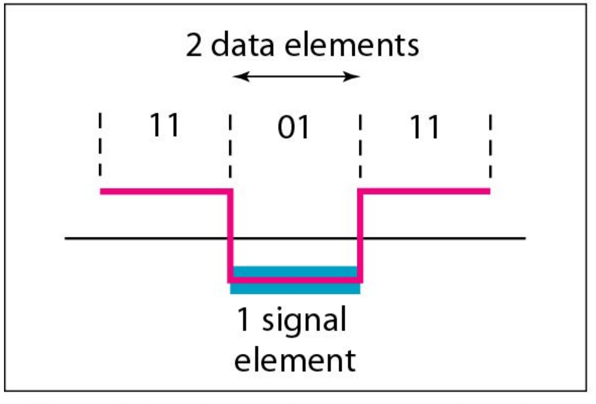
Figure 4.2 *Signal element versus data element*



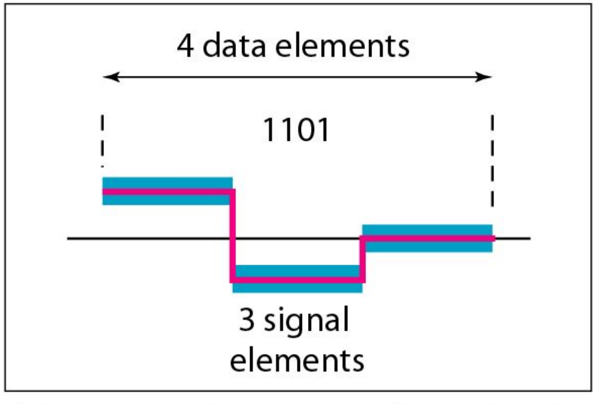
a. One data element per one signal element ($r = 1$)



b. One data element per two signal elements $\left(r = \frac{1}{2}\right)$



c. Two data elements per one signal element ($r = 2$)



d. Four data elements per three signal elements $\left(r = \frac{4}{3}\right)$

1. In part a of the figure, one data element is carried by one signal element *(r = 1)*.
2. In part b of the figure, we need two signal elements (two transitions) to carry each data element *(r =1/2)*
3. In part c of the figure, a signal element carries two data elements *(r = 2)*.
4. Finally, in part d, a group of 4 bits is being carried by a group of three signal elements *(r = 4/3)*.
5. For every line coding scheme we discuss, we will give the value of *r*.

Data rate and Baud rate

- The baud or signal rate can be expressed as:

$$\mathbf{S = c \times N \times 1/r \text{ bauds}}$$

Where N is data rate

c is the case factor (worst, best & avg.)

r is the ratio between data element & signal element

Example 4.1

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 1/2 . The baud rate is then

$$S = c \times N \times \frac{1}{r} = \frac{1}{2} \times 100{,}000 \times \frac{1}{1} = 50{,}000 = 50 \text{ kbaud}$$

Note

Although the actual bandwidth of a digital signal is infinite, the effective bandwidth is finite.

Example 4.2

The maximum data rate of a channel (see Chapter 3) is $N_{max} = 2 \times B \times \log_2 L$ (defined by the Nyquist formula). Does this agree with the previous formula for N_{max}?

Solution

A signal with L levels actually can carry $\log_2 L$ bits per level. If each level corresponds to one signal element and we assume the average case ($c = 1/2$), then we have

$$\boxed{N_{\max} = \frac{1}{c} \times B \times r = 2 \times B \times \log_2 L}$$

Considerations for choosing a good signal element referred to as line encoding

- Baseline wandering - a receiver will evaluate the average power of the received signal (called the baseline) and use that to determine the value of the incoming data elements. If the incoming signal does not vary over a long period of time, the baseline will drift and thus cause errors in detection of incoming data elements.
- A good line encoding scheme will prevent long runs of fixed amplitude.

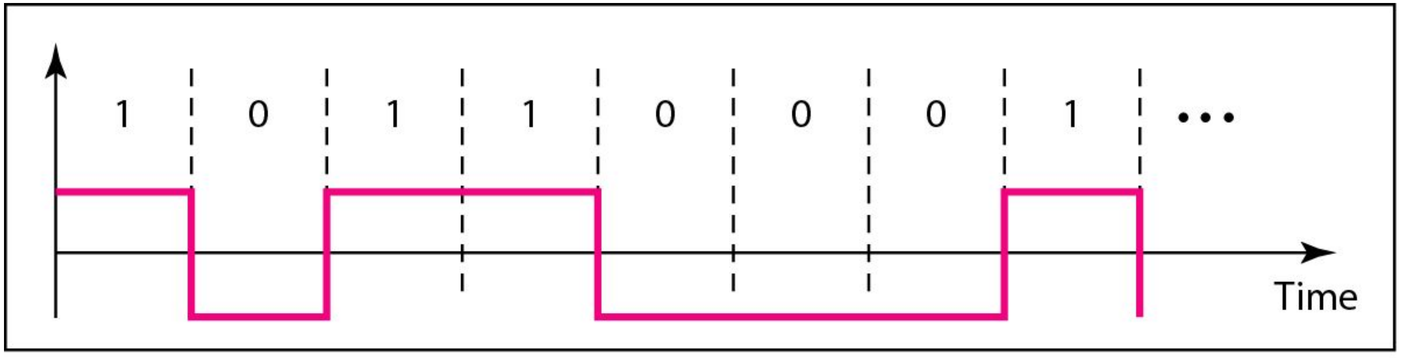
Line encoding

- DC components - when the voltage level remains constant for long periods of time, there is an increase in the low frequencies of the signal. Most channels are bandpass and may not support the low frequencies.
- This will require the removal of the dc component of a transmitted signal.

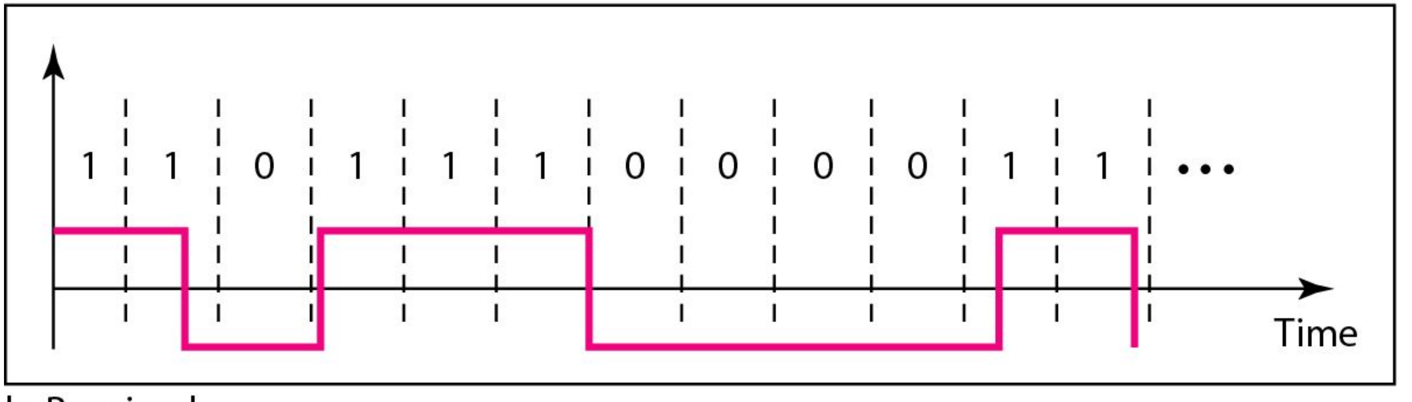
Line encoding

- Self synchronization - the clocks at the sender and the receiver must have the same bit interval.
- If the receiver clock is faster or slower it will misinterpret the incoming bit stream.

Figure 4.3 *Effect of lack of synchronization*



a. Sent



b. Received

Example 4.3

In a digital transmission, the receiver clock is 0.1 percent faster than the sender clock. How many extra bits per second does the receiver receive if the data rate is 1 kbps? How many if the data rate is 1 Mbps?

Solution

At 1 kbps, the receiver receives 1001 bps instead of 1000 bps.

1000 bits sent	1001 bits received	1 extra bps

At 1 Mbps, the receiver receives 1,001,000 bps instead of 1,000,000 bps.

1,000,000 bits sent	1,001,000 bits received	1000 extra bps

Line encoding

- **Error detection** - errors occur during transmission due to line impairments.
- Some codes are constructed such that when an error occurs it can be detected.
- For example: a particular signal transition is not part of the code. When it occurs, the receiver will know that a symbol error has occurred.

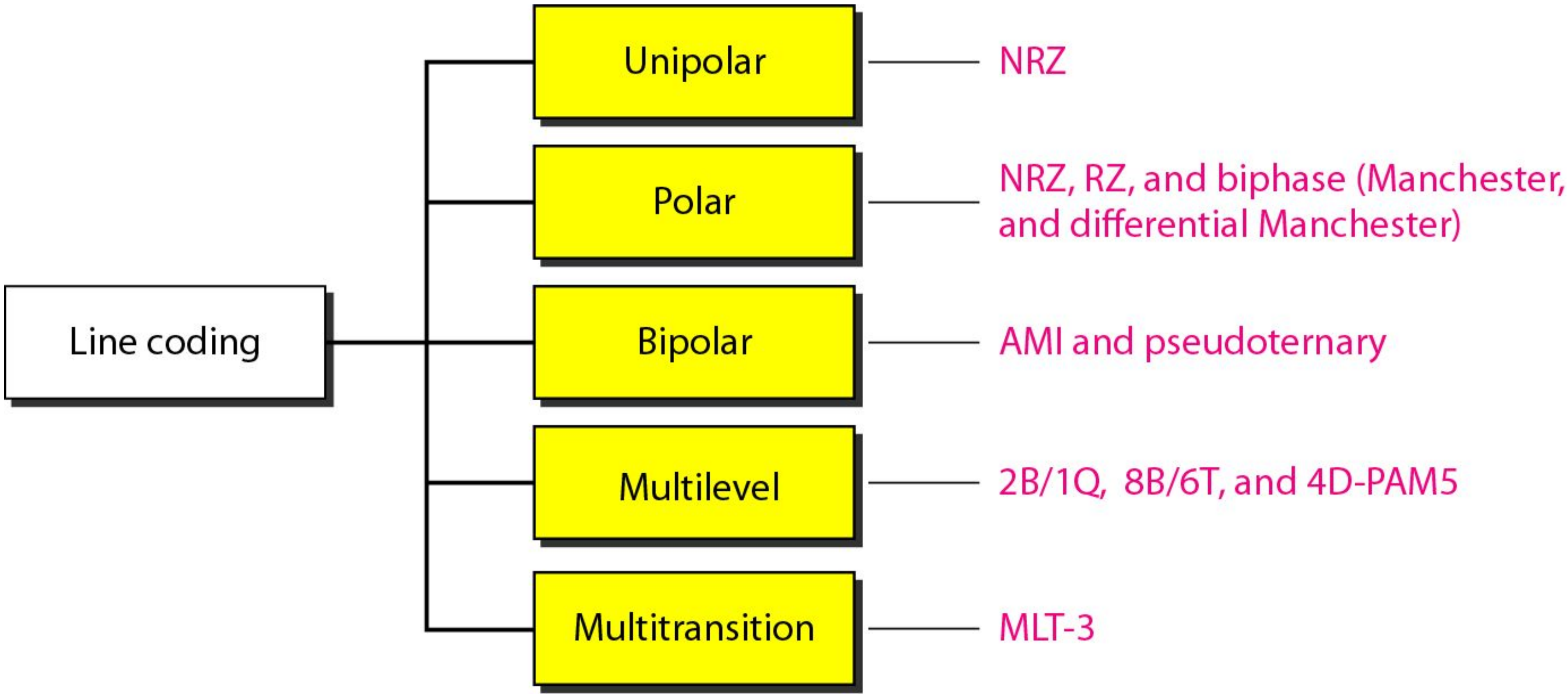
Line encoding

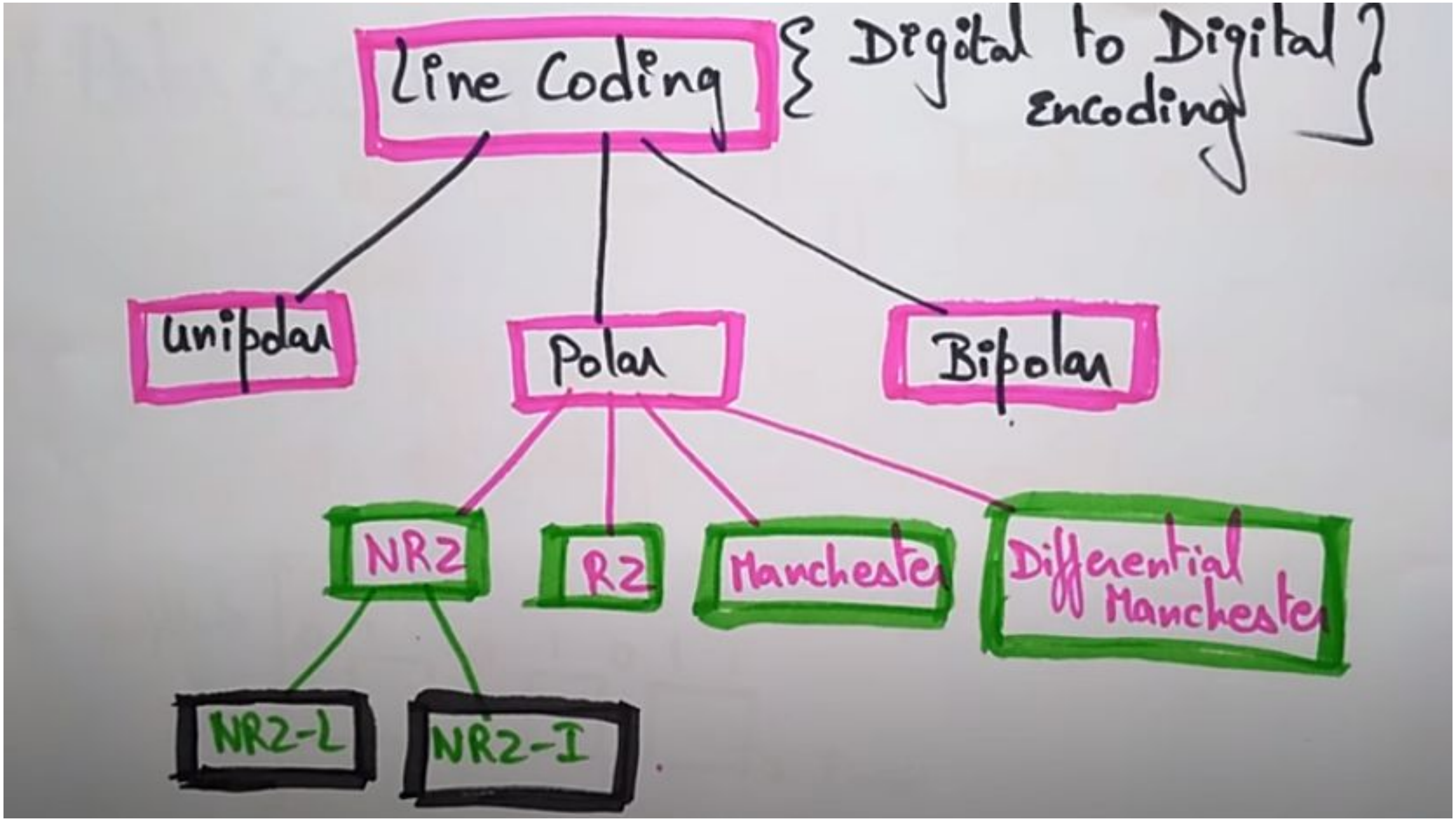
- Noise and interference - there are line encoding techniques that make the transmitted signal “immune” to noise and interference.
- This means that the signal cannot be corrupted, it is stronger than error detection.

Line encoding

- Complexity - the more robust and resilient the code, the more complex it is to implement and the price is often paid in baud rate or required bandwidth.

Figure 4.4 *Line coding schemes*



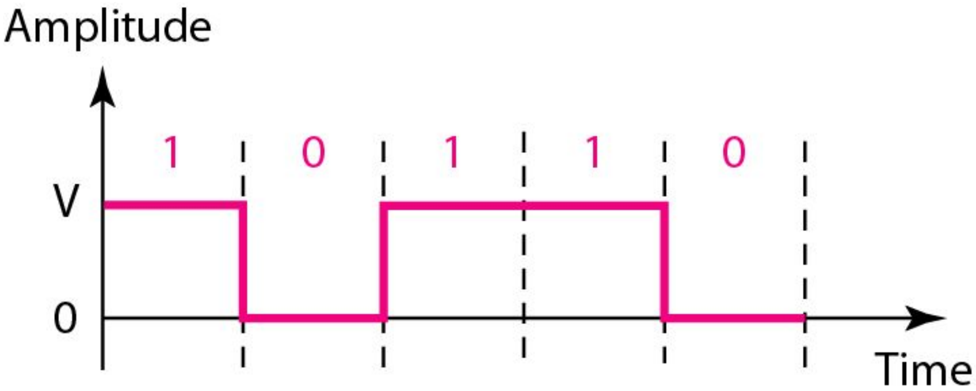


Line Coding
{ Digital to Digital Encoding }
Unipolar
Polar
Bipolar
NRZ
RZ
Manchester
Differential Manchester
NRZ-L
NRZ-I

1. Unipolar

- **All signal levels are on one side of the time axis - either above or below**
- NRZ - Non Return to Zero scheme is an example of this code. The signal level does not return to zero during a symbol transmission.
- Scheme is prone to baseline wandering and DC components. It has no synchronization or any error detection. It is simple but costly in power consumption.

Unipolar NRZ Scheme



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

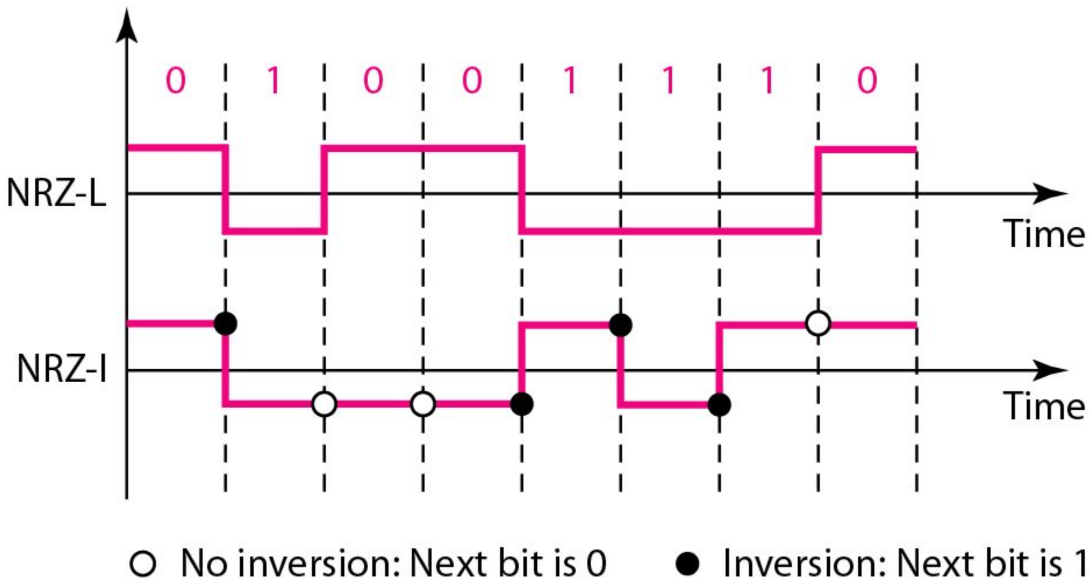
Normalized power

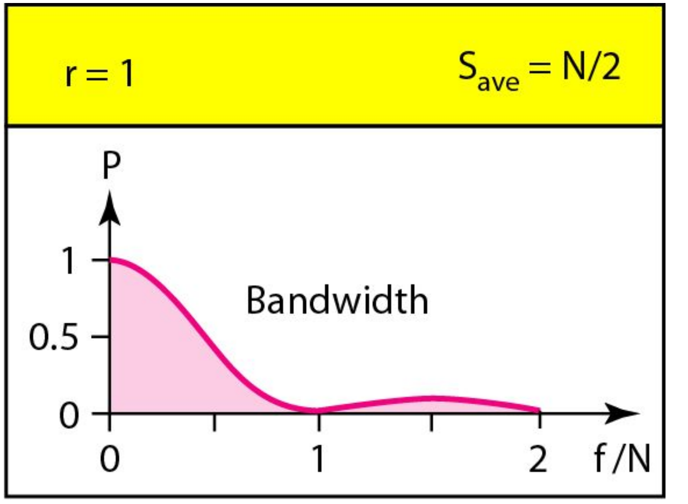
- In a unipolar scheme, all the signal levels are on one side of the time axis, either above or below.
- **NRZ** **(Non-Return-to-Zero):** unipolar scheme was designed as a non-return-to zero (NRZ) scheme in which the positive voltage defines bit 1 and the zero voltage defines bit 0. **It is called NRZ because the signal does not return to zero at the middle of the bit**.

2. Polar - NRZ

- The voltages are on both sides of the time axis.
- Polar NRZ scheme can be implemented with two voltages.
- There are two versions:
  - NZR - Level (NRZ-L) - positive voltage for one symbol and negative for the other
  - **The voltage level for 0 can be positive** and the **voltage level for 1 can be negative**.
  - NRZ - Inversion (NRZ-I) - the change or lack of change in polarity determines the value of a symbol.
  - E.g. a “1” symbol inverts the polarity a “0” does not.

Figure 4.6 *Polar NRZ-L and NRZ-I schemes*





Both schemes have an average signal rate of *N/2 Bd.*

Figure 4.6 *Polar NRZ-L and NRZ-I schemes*

- If there is a long sequence of 0s or 1s in NRZ-L, the average signal power becomes skewed.
- The receiver might have difficulty discriminating the bit value.
- The synchronization problem (sender and receiver clocks are not synchronized) also exists in both schemes. long sequence of 0s can cause a problem in both schemes
- Another problem with NRZ-L occurs when there is a sudden change of polarity in the system. a change in the polarity of the wire results in all 0s interpreted as 1s and all

Note

In NRZ-L the level of the voltage determines the value of the bit. In NRZ-I the inversion or the lack of inversion determines the value of the bit.

Note

NRZ-L and NRZ-I both have an average signal rate of N/2 Bd.

Note

NRZ-L and NRZ-I both have a DC component problem and baseline wandering, it is worse for NRZ-L. Both have no self synchronization &no error detection. Both are relatively simple to implement.

Example 4.4

A system is using NRZ-I to transfer 1-Mbps data. What are the average signal rate and minimum bandwidth?

Solution

The average signal rate is S= c x N x R = 1/2 x N x 1 = 500 kbaud. The minimum bandwidth for this average baud rate is $B_{min} = S = 500$ ***kHz.***

Note c = 1/2 for the avg. case as worst case is 1 and best case is 0

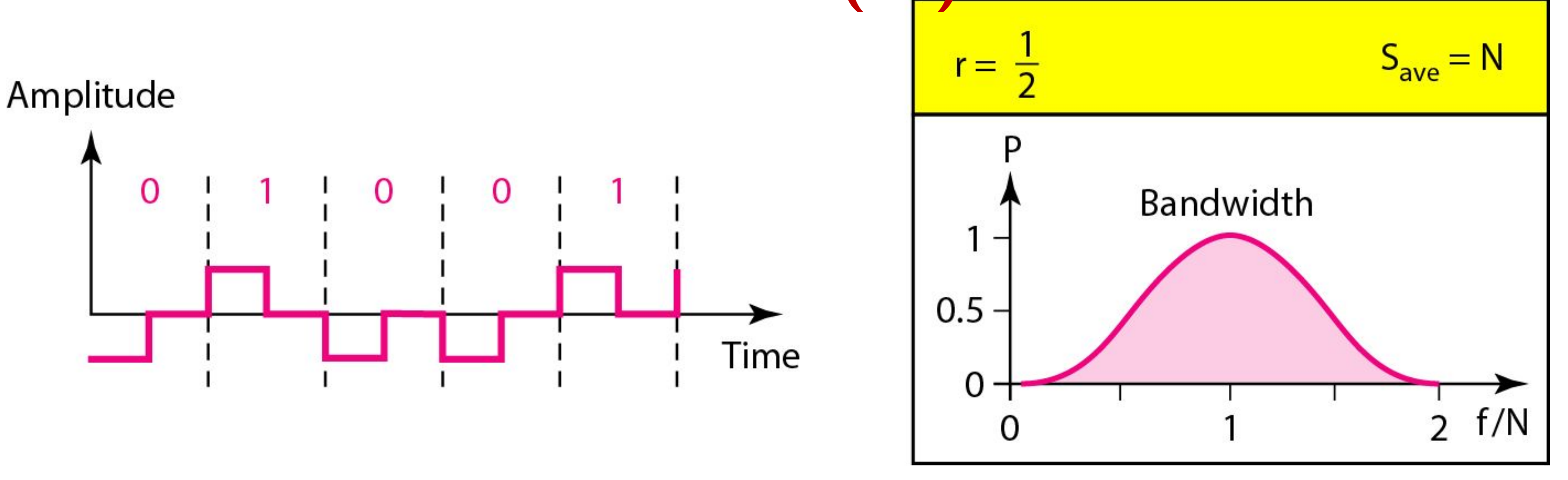
3. Polar - RZ

- The Return to Zero (RZ) scheme uses three voltage values. +, 0, -.
- Each symbol has a transition in the middle. Either from high to zero or from low to zero.
- In RZ, the signal changes not between bits but during the bit.
- The main disadvantage of RZ encoding is that it requires <u>two signal changes to encode a bit</u> and therefore **occupies greater bandwidth.**
- No DC components or baseline wandering.
- Self synchronization - transition indicates symbol value.
- More complex as it uses three voltage level. It has no error detection capability.

Figure 4.7 *Polar RZ scheme*

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**

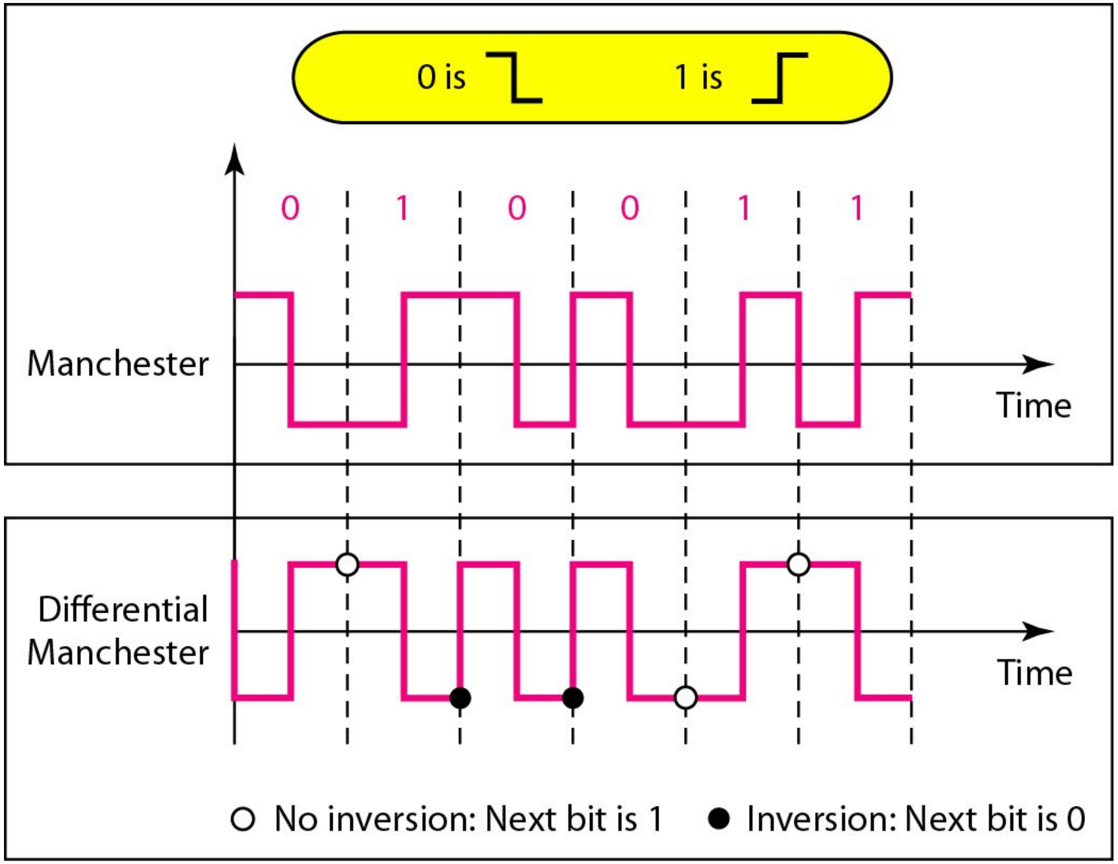


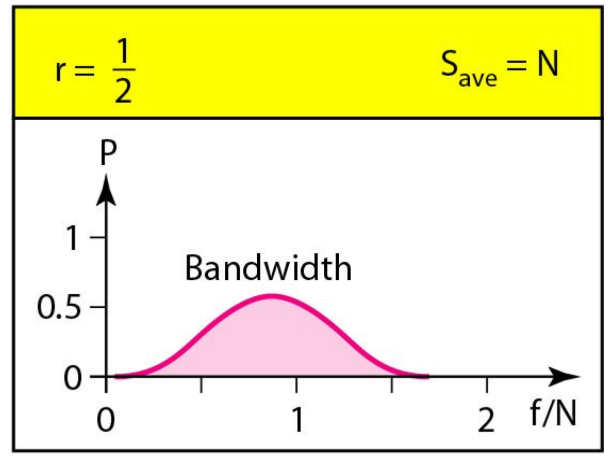
5. Polar - Biphase: Manchester and Differential Manchester

- Manchester coding consists of combining the NRZ-L and RZ schemes.
  - Every symbol has a level transition in the middle: from high to low or low to high. Uses only two voltage levels.
- Differential Manchester coding consists of combining the NRZ-I and RZ schemes.
  - Every symbol has a level transition in the middle. But the level at the beginning of the symbol is determined by the symbol value. One symbol causes a level change the other does not.

- 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.
- **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.

Figure 4.8 *Polar biphase: Manchester and differential Manchester schemes*





❖ The Manchester scheme overcomes several problems associated with NRZ-L, and differential Manchester overcomes several problems associated with NRZ-I.

Note

In Manchester and differential Manchester encoding, the transition at the middle of the bit is used for synchronization.

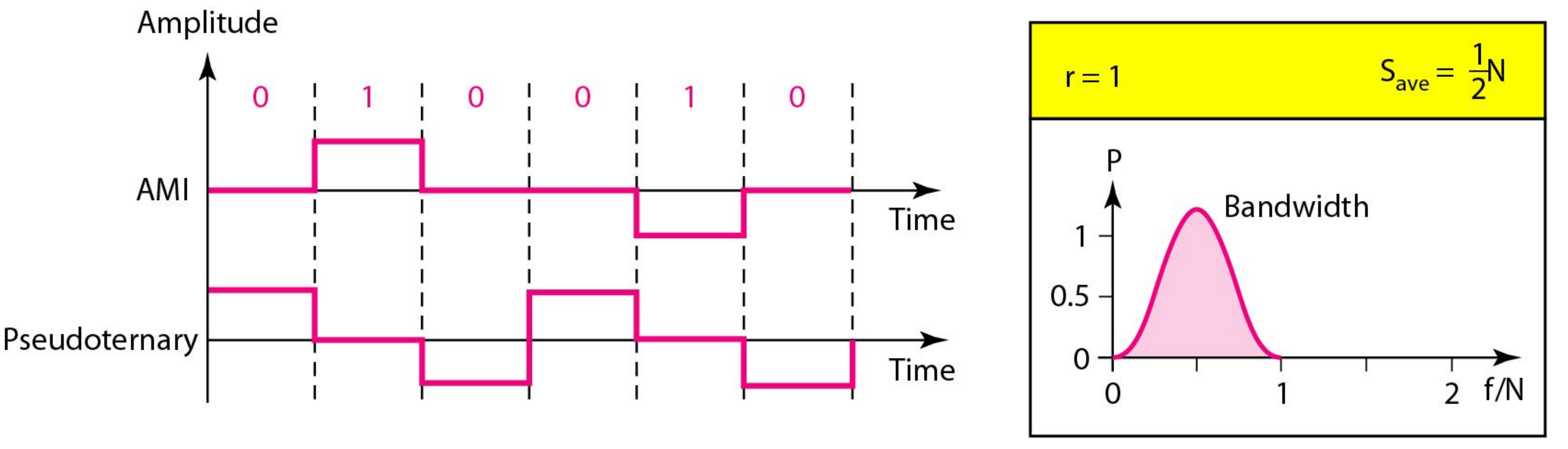
Note

The minimum bandwidth of Manchester and differential Manchester is 2 times that of NRZ. The is no DC component and no baseline wandering. None of these codes has error detection.

4. Bipolar - AMI and Pseudoternary

- Code uses 3 voltage levels: - +, 0, -, to represent the symbols (note not transitions to zero as in RZ).
- Voltage level for one symbol is at "0" and the other alternates between + & -.
- Bipolar **Alternate Mark Inversion (AMI)** - the "0" symbol is represented by zero voltage and the "1" symbol alternates between +V and -V.
- Pseudoternary is the reverse of AMI.

Figure 4.9 *Bipolar schemes: AMI and pseudoternary*



- In bipolar encoding (sometimes called *multilevel binary)*, there are three voltage levels*:* **positive**, **negative**, and **zero**.
- The voltage level for one data element is at zero, while the voltage level for the other element alternates between positive and negative.
- Figure above shows two variations of bipolar encoding: **AMI** and **pseudoternary**.
- A common bipolar encoding scheme is called bipolar **alternate mark inversion** (AMI).
- In the term ***alternate mark inversion***, the word mark comes from telegraphy and means 1. So AMI means alternate 1 inversion.
- A neutral zero voltage **represents binary 0**. Binary 1s are **represented by alternating positive and negative voltages**.
- A variation of AMI encoding is called pseudoternary in which the **1 bit is encoded as a zero voltage** and **the 0 bit is encoded as alternating positive and negative voltages**.

Bipolar

- It is a better alternative to NRZ.
- Has no DC component or baseline wandering.
- Has no self synchronization because long runs of “0”s results in no signal transitions.
- No error detection.

4-2 ANALOG-TO-DIGITAL CONVERSION

A digital signal is superior to an analog signal because it is more robust to noise and can easily be recovered, corrected and amplified. For this reason, the tendency today is to change an analog signal to digital data. In this section we describe two techniques, pulse code modulation and delta modulation.

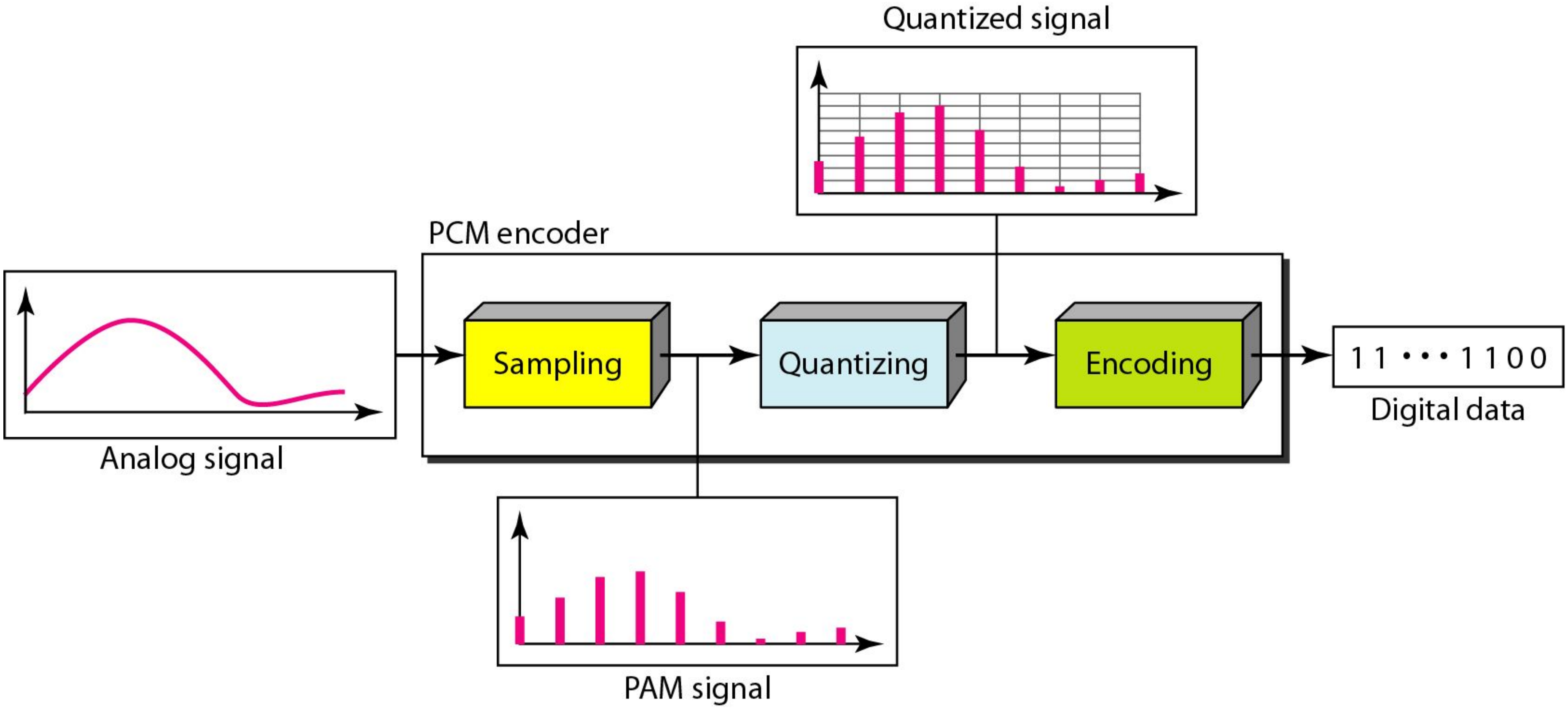
Topics discussed in this section:

- **Pulse Code Modulation (PCM)**
- **Delta Modulation (DM)**

PCM

- PCM consists of three steps to digitize an analog signal:
  1. Sampling
  2. Quantization
  3. Binary encoding
- Before we sample, we have to filter the signal to limit the maximum frequency of the signal as it **affects the sampling rate**.
- Filtering should ensure that we do not distort the signal, ie remove high frequency components that affect the signal shape.

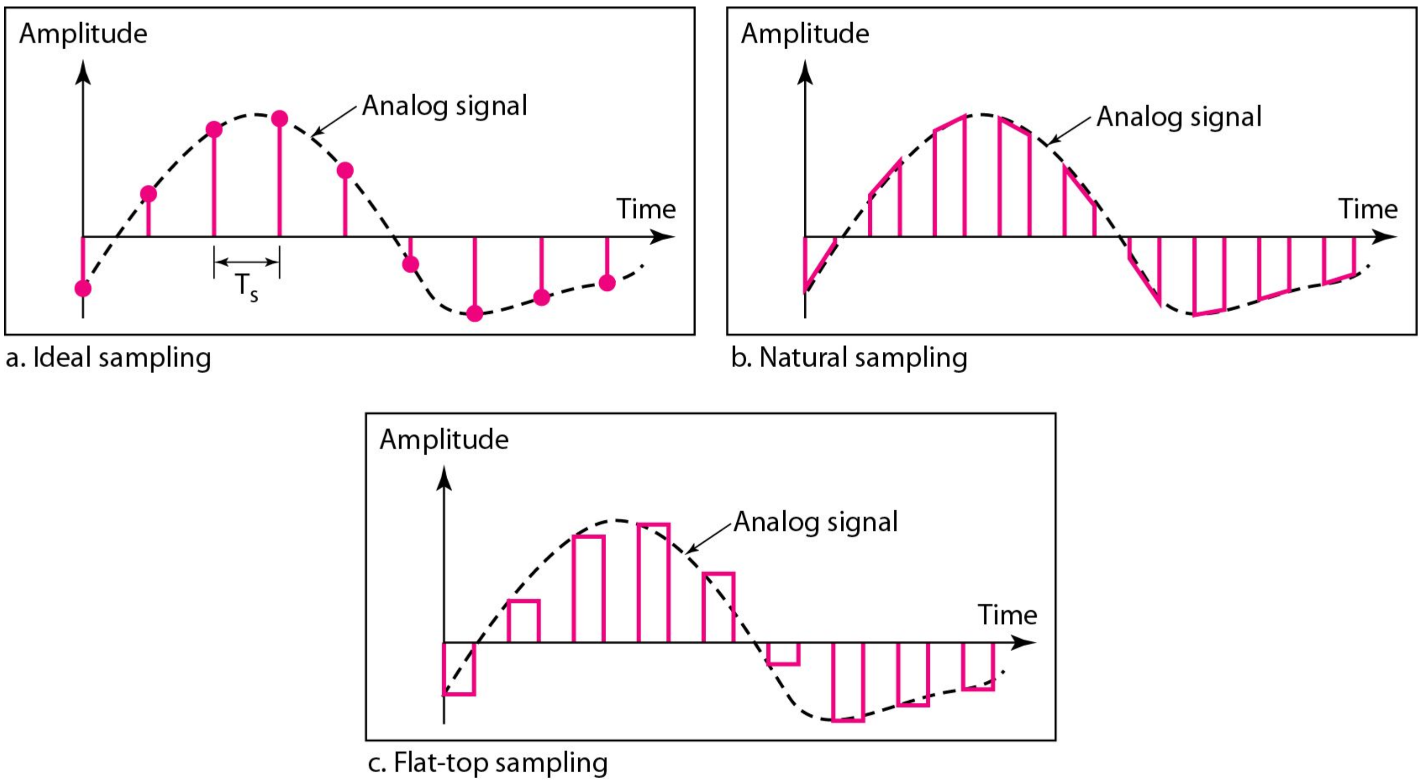
Figure 4.21 *Components of PCM encoder*



Sampling

- Analog signal is sampled every T_S secs.
- T_s is referred to as the sampling interval.
- $f_s = 1/T_s$ is called the sampling rate or sampling frequency.
- There are 3 sampling methods:
  - Ideal - an impulse at each sampling instant
  - Natural - a pulse of short width with varying amplitude
  - Flattop - sample and hold, like natural but with single amplitude value
- The process is referred to as Pulse Amplitude Modulation **PAM** and the outcome is a signal with analog (non integer) values

Figure 4.22 ***Three different sampling methods for PCM***



- The analog signal is sampled every T_s, where T_s is the sample interval or period. The inverse of the sampling interval is called the ***sampling rate*** or ***sampling frequency*** and denoted by f_s where $f_s = 1/T_s$

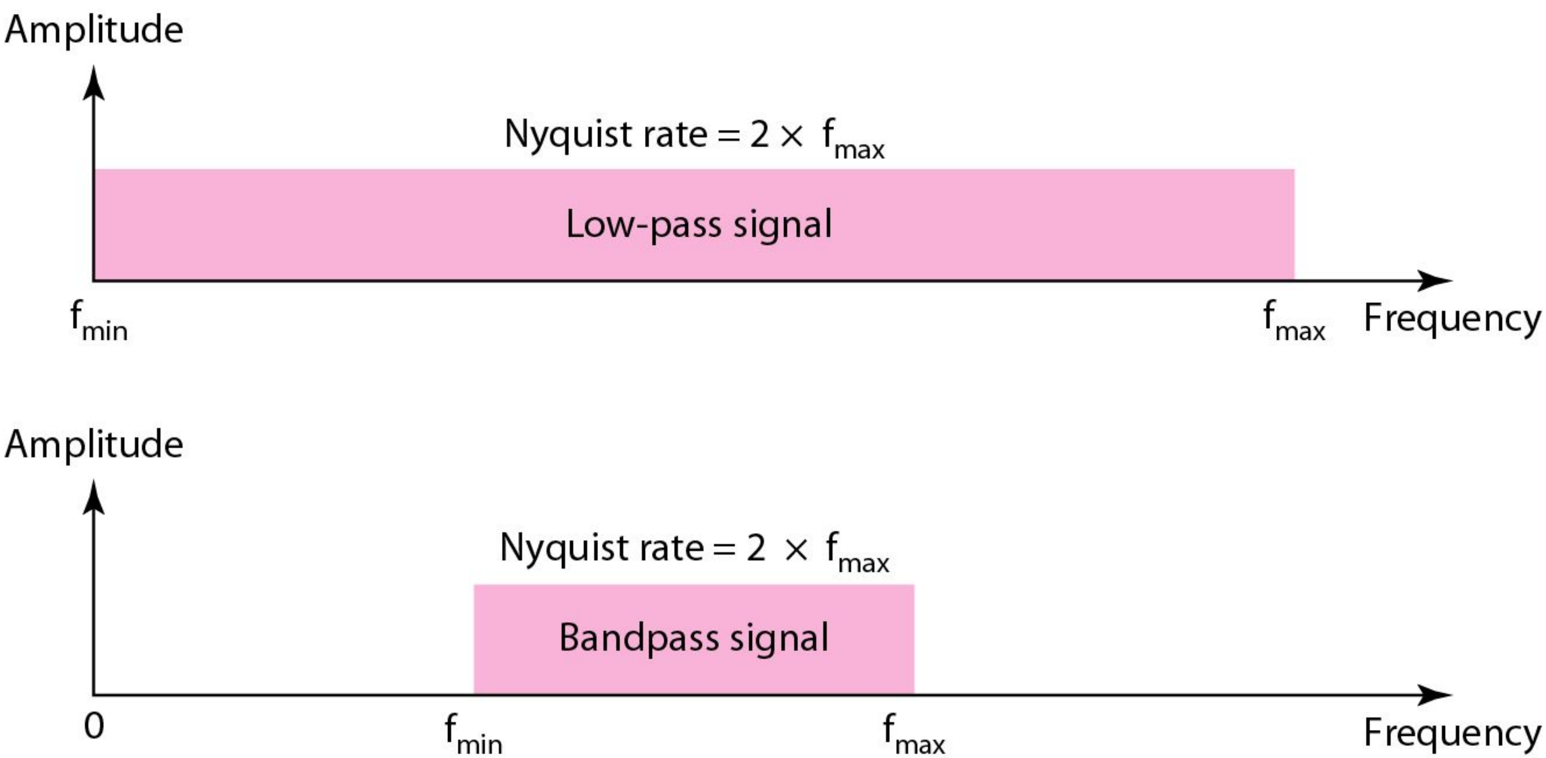
Figure 4.22 ***Three different sampling methods for PCM***

1. In **ideal sampling**, pulses from the analog signal are sampled. This is an ideal sampling method and cannot be easily implemented.
2. In **natural sampling**, a high-speed switch is turned on for only the small period of time when the sampling occurs. The result is a sequence of samples that retains the shape of the analog signal.
3. The most common sampling method, called ***sample and hold,*** however, creates **flat top samples** by using a circuit.

Note

According to the Nyquist theorem, the sampling rate must be at least 2 times the highest frequency contained in the signal.

Figure 4.23 *Nyquist sampling rate for low-pass and bandpass signals*

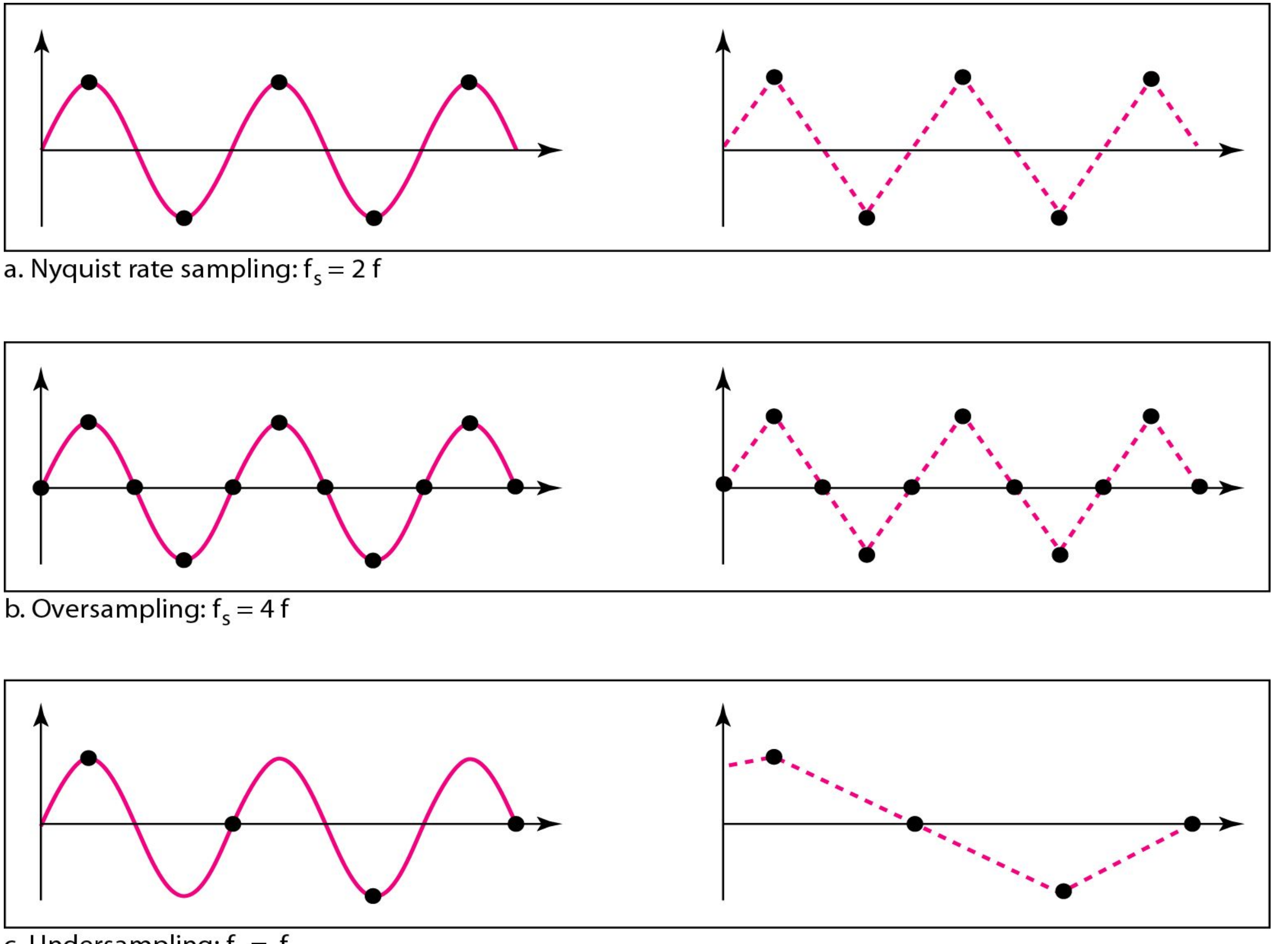


Example 4.6

For an intuitive example of the Nyquist theorem, let us sample a simple sine wave at three sampling rates: $f_s = 4f$ (2 times the Nyquist rate), $f_s = 2f$ (Nyquist rate), and $f_s = f$ (one-half the Nyquist rate). Figure 4.24 shows the sampling and the subsequent recovery of the signal.

It can be seen that sampling at the Nyquist rate can create a good approximation of the original sine wave (part a). Oversampling in part b can also create the same approximation, but it is redundant and unnecessary. Sampling below the Nyquist rate (part c) does not produce a signal that looks like the original sine wave.

Figure 4.24 *Recovery of a sampled sine wave for different sampling rates*



a. Nyquist rate sampling: $f_s = 2f$

b. Oversampling: $f_s = 4f$

c. Undersampling: $f_s = f$

Example 4.9

Telephone companies digitize voice by assuming a maximum frequency of 4000 Hz. The sampling rate therefore is 8000 samples per second.

Example 4.10

A complex low-pass signal has a bandwidth of 200 kHz. What is the minimum sampling rate for this signal?

Solution

The bandwidth of a low-pass signal is between 0 and f, where f is the maximum frequency in the signal. Therefore, we can sample this signal at 2 times the highest frequency (200 kHz). The sampling rate is therefore 400,000 samples per second.

Example 4.11

A complex bandpass signal has a bandwidth of 200 kHz. What is the minimum sampling rate for this signal?

Solution

We cannot find the minimum sampling rate in this case because we do not know where the bandwidth starts or ends. We do not know the maximum frequency in the signal.

Quantization

- **Sampling results** in a series of pulses of varying amplitude values ranging between two limits: **a min and a max.**
- The amplitude values are infinite between the two limits.
- We need to map the *infinite* amplitude values onto a finite set of known values.
- This is achieved by dividing the distance between min and max into L zones, each of height Δ.

$$\Delta = (\max - \min)/L$$

PCM

O/P ⇒ Sampling → Quantization → Encoded

Eg: $V_{min} = -127$

$V_{max} = +127$

L = zones
↳ levels
L = 2, 4, 8, 10

$\Delta = 25$ $\Delta = (V_{max} - V_{min})/L$
$= (127+127)/10 = 25.4 \approx 25$

① Normalized PAM value (Actual value)

$\frac{24}{25} = 0.96$

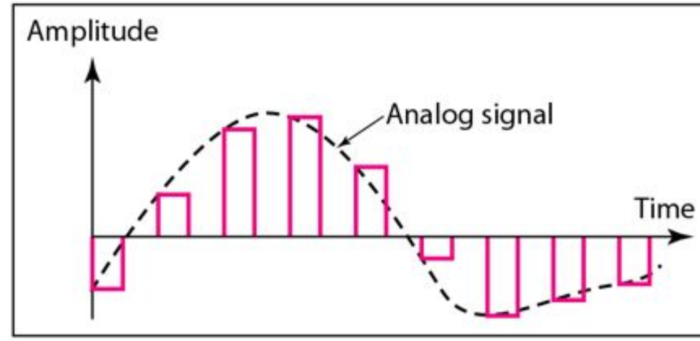
② Normalized Quantized value $0.96 \approx 1$

③ Normalized error   +0.04 (difference)

④ Quantization code 6 (0101) (Level value from 0 to 9)

⑤ Encoded word = 0110

$n_b = \log_2 L$
$= \log_2 10 \approx 4$



c. Flat-top sampling

Quantization Levels

- The midpoint of each zone is assigned a value from 0 to L-1 (resulting in L values)
- Each sample falling in a zone is then approximated to the value of the midpoint.
- The quantization code is assigned for each sample based on the quantization levels

Quantization Zones

- Assume we have a voltage signal with amplitutes V_{min}=-20V and V_{max}=+20V.
- We want to use L=8 quantization levels.
- Zone width $\Delta = (20 - -20)/8 = 5$
- The 8 zones are: -20 to -15, -15 to -10, -10 to -5, -5 to 0, 0 to +5, +5 to +10, +10 to +15, +15 to +20
- The midpoints are: -17.5, -12.5, -7.5, -2.5, 2.5, 7.5, 12.5, 17.5

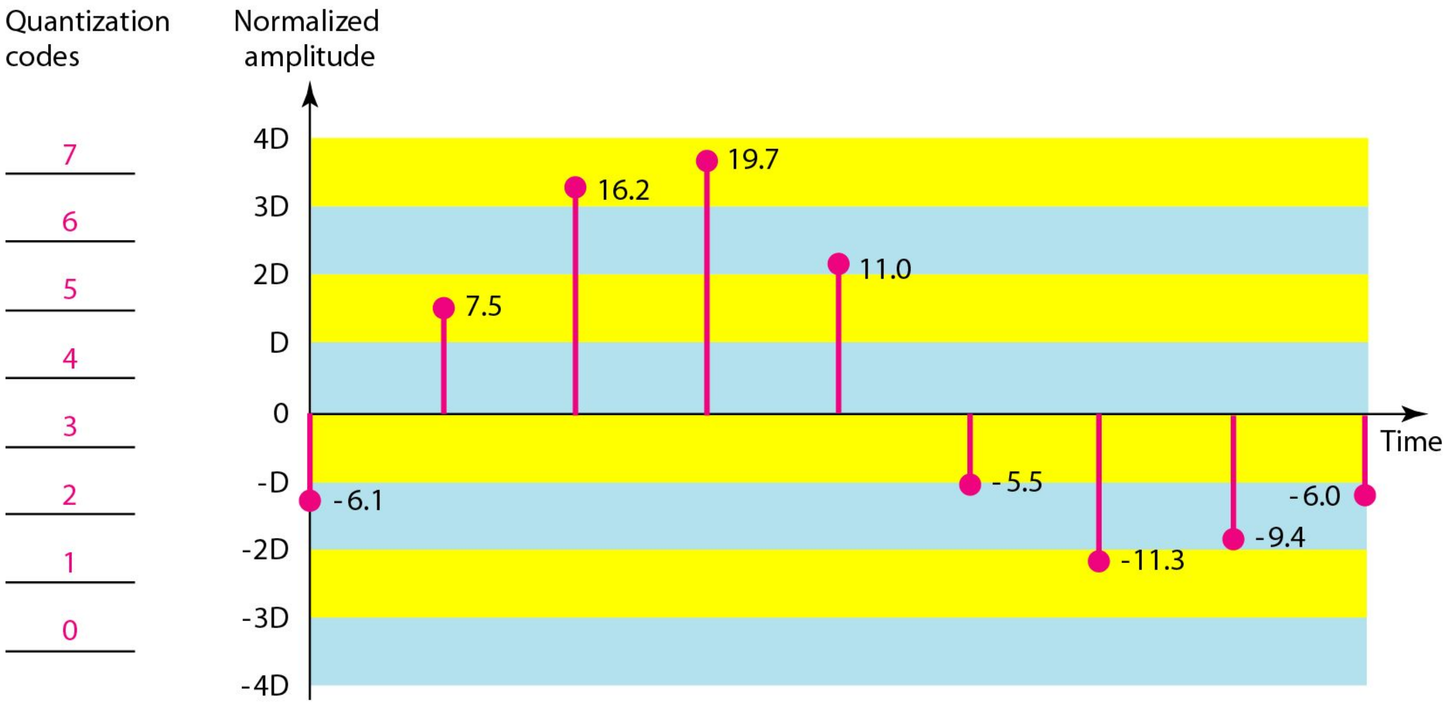
Assigning Codes to Zones

- Each zone is then assigned a binary code.
- The number of bits required to encode the zones, or the number of bits per sample as it is commonly referred to, is obtained as follows:

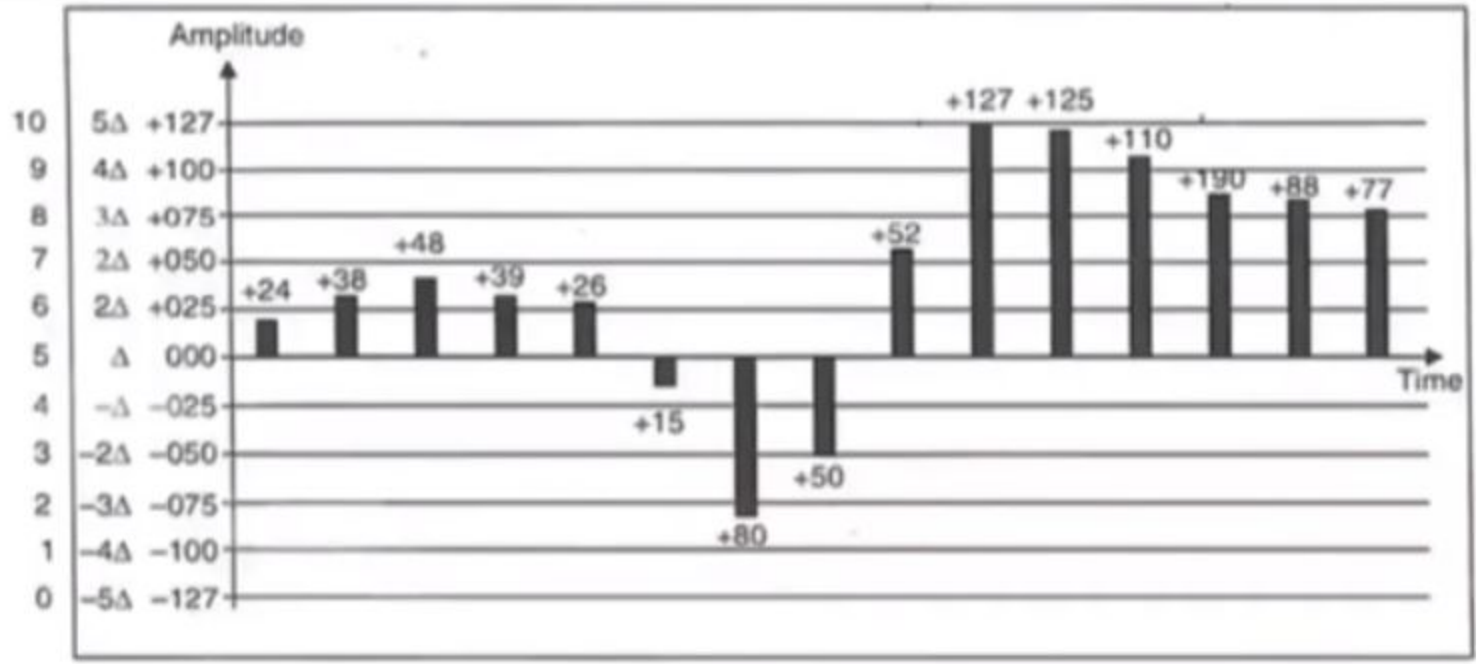
$$n_b = \log_2 L$$

- Given our example, $n_b = 3$
- The 8 zone (or level) codes are therefore: 000, 001, 010, 011, 100, 101, 110, and 111
- Assigning codes to zones:
  - 000 will refer to zone -20 to -15
  - 001 to zone -15 to -10, etc.

Figure 4.26 *Quantization and encoding of a sampled signal*



Normalized PAM values	-1.22	1.50	3.24	3.94	2.20	-1.10	-2.26	-1.88	-1.20
Normalized quantized values	-1.50	1.50	3.50	3.50	2.50	-1.50	-2.50	-1.50	-1.50
Normalized error	-0.38	0	+0.26	-0.44	+0.30	-0.40	-0.24	+0.38	-0.30
Quantization code	2	5	7	7	6	2	1	2	2
Encoded words	010	101	111	111	110	010	001	010	010



Row 1	Normalized PAM values (actual values)	0.96	1.52	1.92	1.56	1.04	-0.6	-3.2	-2	2.08	5.08	5	4.4	3.6	3.52	3.08
Row 2	Normalized quantized values	1	1.5	2	1.5	1	−0.5	−3.5	−2	2	5	5	4.5	3.5	3.5	3
Row 3	Normalized error	+0.04	−0.02	+0.08	−0.06	−0.04	+0.1	−0.3	0	−0.08	−0.08	0	+0.1	−0.1	−0.02	−0.08
Row 4	Quantization code	5	6	6	6	5	4	1	3	7	10	10	9	8	8	8
Row 5	Encoded word	0101	0110	0110	0110	0101	0100	0001	0011	0111	1010	1010	1001	1000	1000	1000

PCM

o/p ⇒ Sampling → Quantization → Encoded

Eg: $V_{min} = -127$

$V_{max} = +127$

L = zones

↳ levels

L = 2, 4, 8, 10

$\Delta = 25$ $\quad \Delta = (V_{max} - V_{min})/L$

$= (127 + 127)/10 = 25.4 \approx 25$

① Normalized PAM value (Actual value)

$\frac{24}{25} = 0.96$

② Normalized Quantized value $0.96 \sim 1$

③ Normalized error $+0.04$ (difference)

④ Quantization code 6 (level value from 0 to 9) 0101

⑤ Encoded word = 0110

$n_b = \log_2 L$

$= \log_2 10 \approx 4$

Quantization Error

- When a signal is quantized, we introduce an error - the coded signal is an approximation of the actual amplitude value.
- The difference between actual and coded value (midpoint) is referred to as the quantization error.
- The more zones, the smaller Δ which results in smaller errors.
- BUT, the more zones the more bits required to encode the samples -> higher bit rate

Quantization Error and SN_QR

- Signals with lower amplitude values will suffer more from quantization error as the error range: **Δ/2, is fixed for all signal levels**.
- Non linear quantization is used to alleviate this problem. Goal is to keep <u>SN_QR fixed for all sample values.</u>
- Two approaches:
  - The quantization levels follow a logarithmic curve. Smaller Δ's at lower amplitudes and larger Δ's at higher amplitudes.
  - Companding: The sample values are compressed at the sender into logarithmic zones, and then expanded at the receiver. **The zones are fixed in height**.

Bit rate and bandwidth requirements of PCM

- The bit rate of a PCM signal can be calculated form the **number of bits per sample x the sampling rate**

  Bit rate = n_b x f_s

- The bandwidth required to transmit this signal depends on the **type of line encoding used.** Refer to previous section for discussion and formulas.
- A digitized signal will always need more bandwidth than the original analog signal. Price we pay for robustness and other features of digital transmission.

Example 4.14

We want to digitize the human voice. What is the bit rate, assuming 8 bits per sample?

Solution

The human voice normally contains frequencies from 0 to 4000 Hz. So the sampling rate and bit rate are calculated as follows:

$$\text{Sampling rate} = 4000 \times 2 = 8000 \text{ samples/s}$$
$$\text{Bit rate} = 8000 \times 8 = 64{,}000 \text{ bps} = 64 \text{ kbps}$$

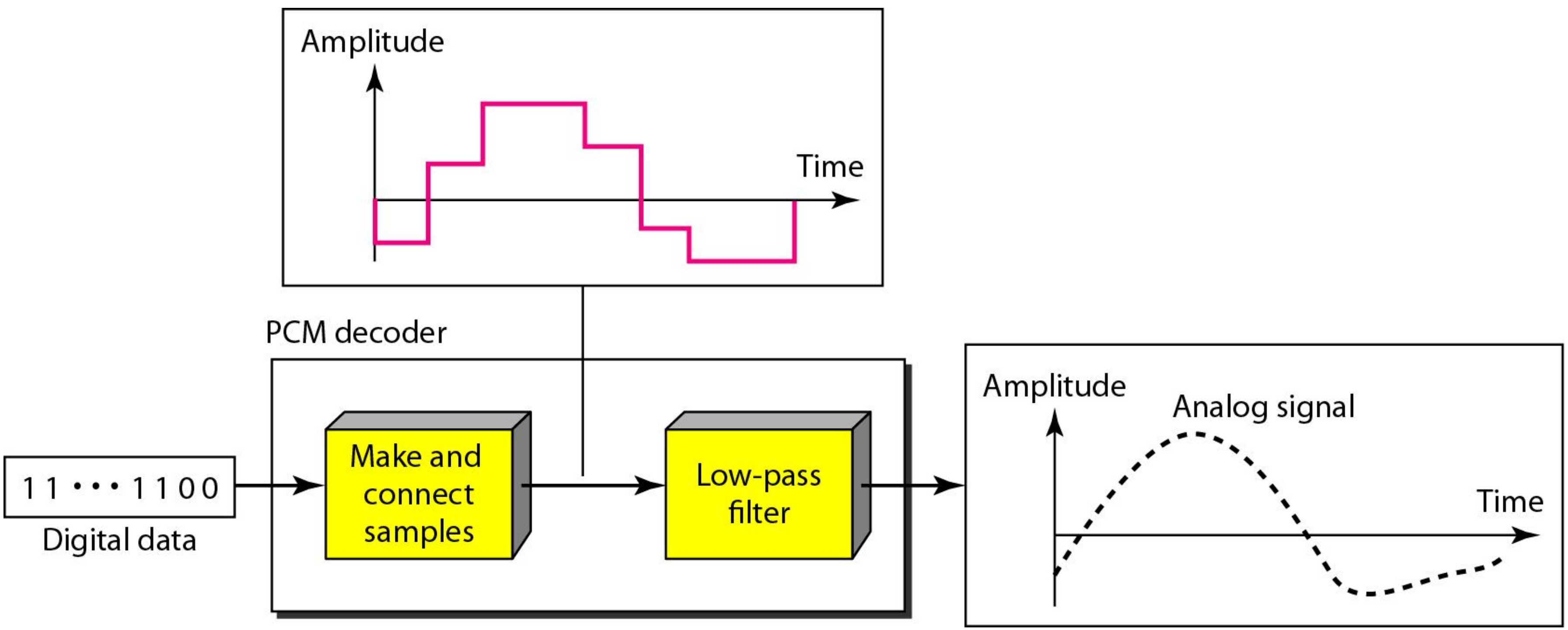
PCM Decoder

- To recover an analog signal from a digitized signal we follow the following steps:
  - We use a hold circuit that holds the amplitude value of a pulse till the next pulse arrives.
  - We pass this signal through a low pass filter with a cutoff frequency that is equal to the highest frequency in the pre-sampled signal.
- The higher the value of L, the less distorted a signal is recovered.

PCM Decoder

- The recovery of the original signal requires the **PCM decoder**.
- The decoder first uses circuitry to **convert the code words** into a pulse that holds the amplitude until the next pulse.
- After the staircase signal is completed, it is passed through a **low-pass filter** to smooth the staircase signal into an analog signal.

Figure 4.27 *Components of a PCM decoder*



Example 4.15

We have a low-pass analog signal of 4 kHz. If we send the analog signal, we need a channel with a minimum bandwidth of 4 kHz. If we digitize the signal and send 8 bits per sample, we need a channel with a minimum bandwidth of 8 × 4 kHz = 32 kHz.

Delta Modulation

- This scheme sends only the **difference between pulses**, if the pulse at time t_{n+1} is higher in amplitude value than the pulse at time t_n, then a single bit, say a "1", is used to indicate the positive value.
- If the pulse is lower in value, resulting in a negative value, a "0" is used.
- This scheme works well for small changes in signal values between samples.
- **If changes in amplitude are large, this will result in large errors.**

Figure 4.28 *The process of delta modulation*

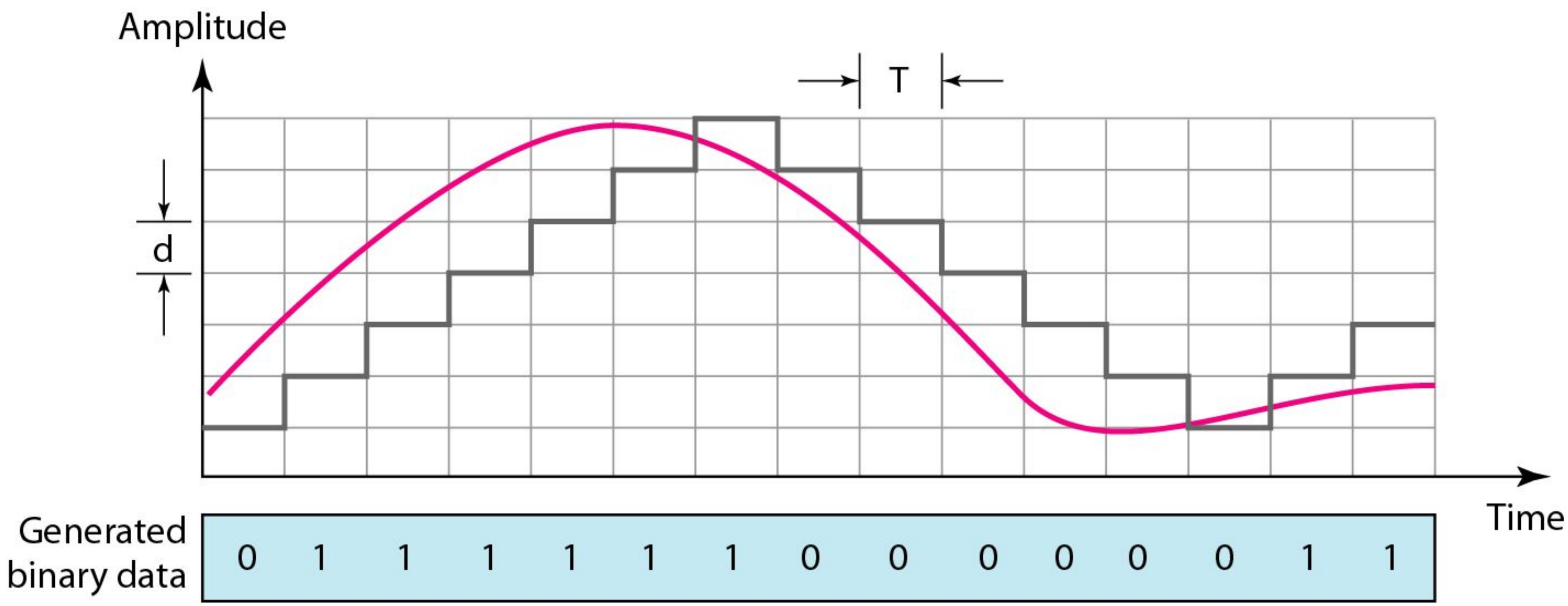


Figure 4.29 *Delta modulation components*

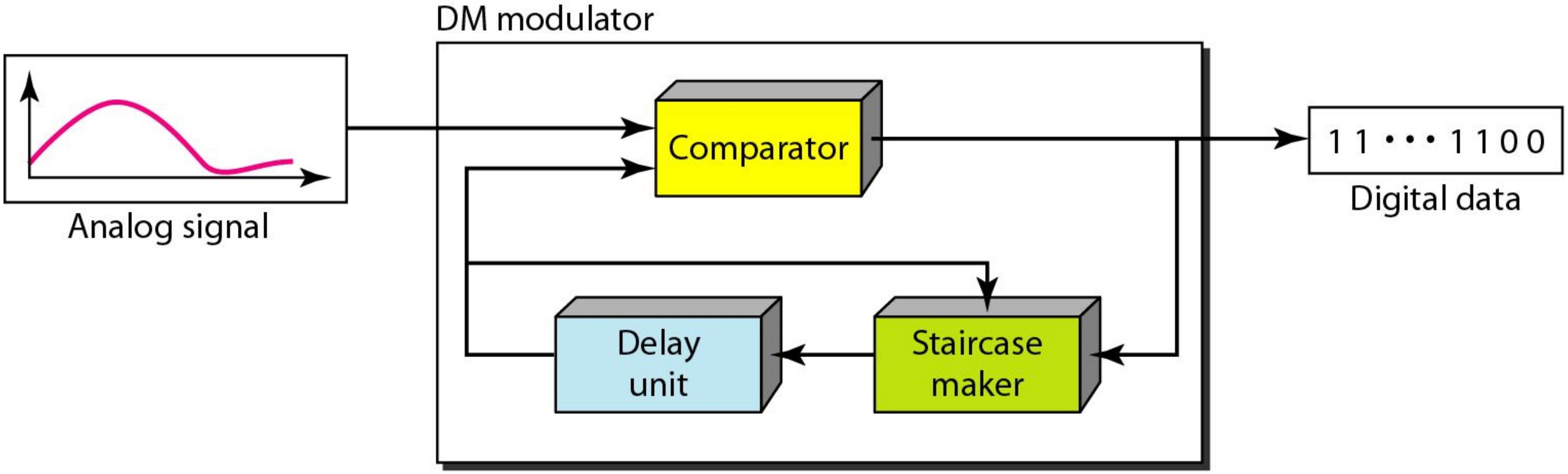
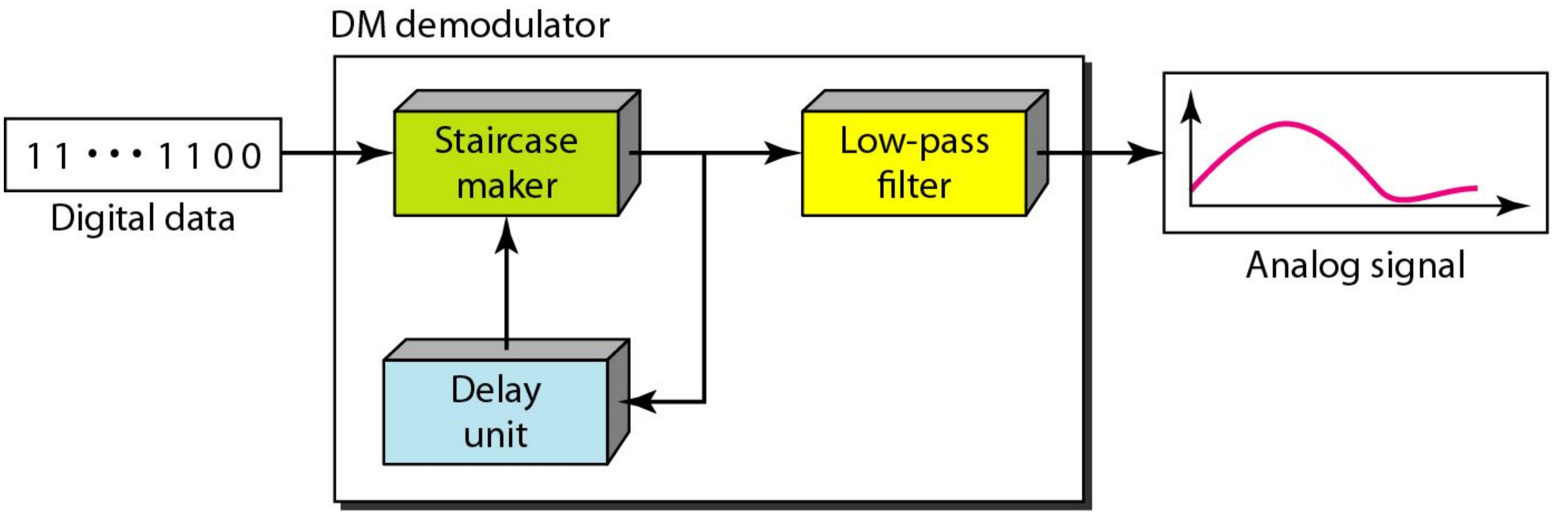


Figure 4.30 *Delta demodulation components*



Delta PCM (DPCM)

- Instead of using one bit to indicate positive and negative differences, we can use more bits -> quantization of the difference.
- Each bit code is used to represent the value of the difference.
- The more bits the more levels -> the higher the accuracy.

4-3 TRANSMISSION MODES

The transmission of binary data across a link can be accomplished in either parallel or 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 there is only one way to send parallel data, there are three subclasses of serial transmission: asynchronous, synchronous, and isochronous.

Topics discussed in this section:

- **Parallel Transmission**
- **Serial Transmission**

Figure 4.31 *Data transmission and modes*

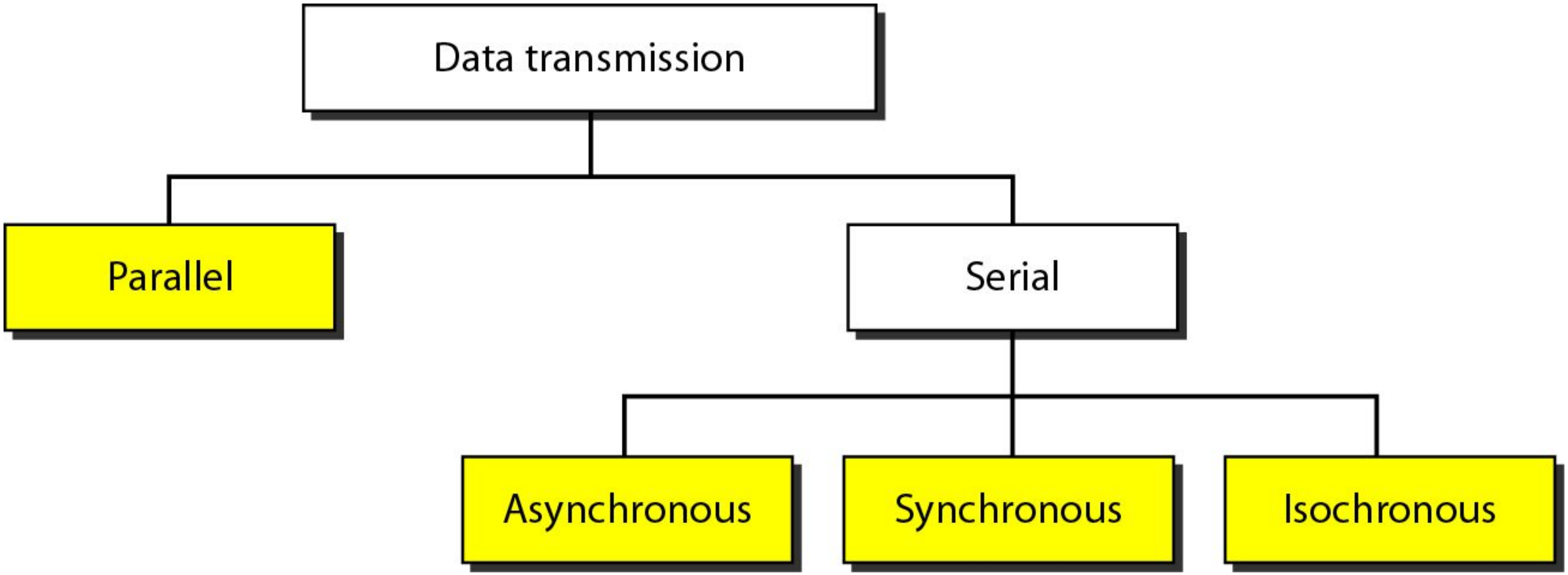
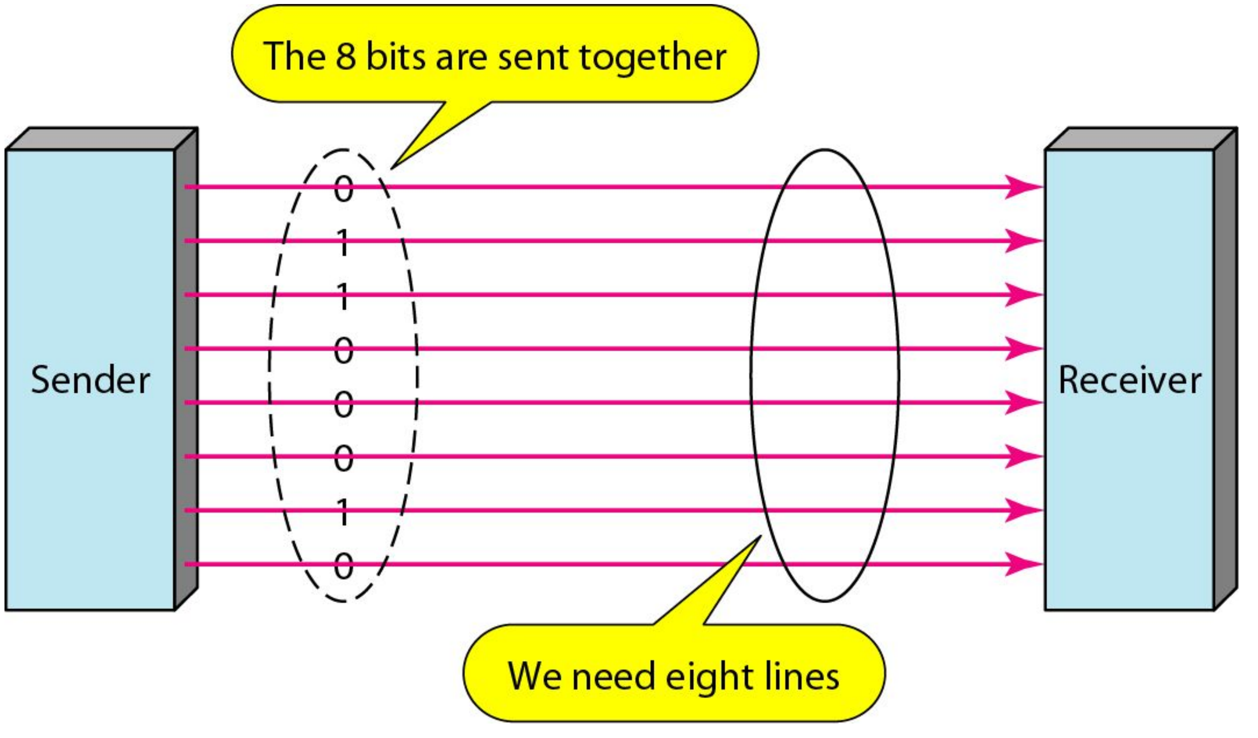


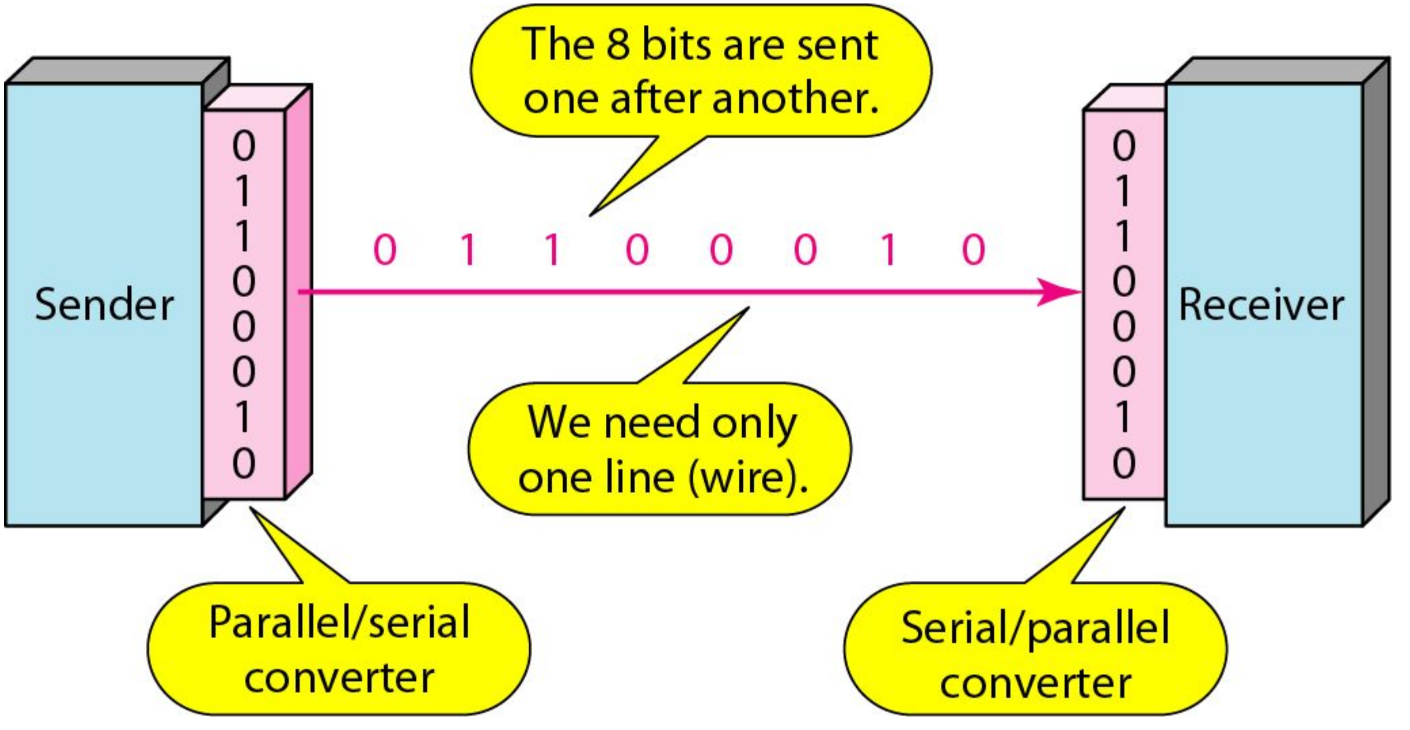
Figure 4.32 *Parallel transmission*



- Binary data, consisting of 1s and 0s, may be organized into **groups of *n bits*** *each.*
- The mechanism for parallel transmission is a conceptually simple one: **Use *n wires* to send *n bits at one time*.** That way each bit has its own wire, and all n bits *of one* group can be transmitted with each clock tick from one device to another .

- The advantage of parallel transmission is **speed**.
- But there is a significant disadvantage: **cost**. Parallel transmission requires
- ***n-communication* lines** (wires) just to transmit the data stream. Because this is expensive, parallel transmission is usually limited to short distances.

Figure 4.33 *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.

- 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**.

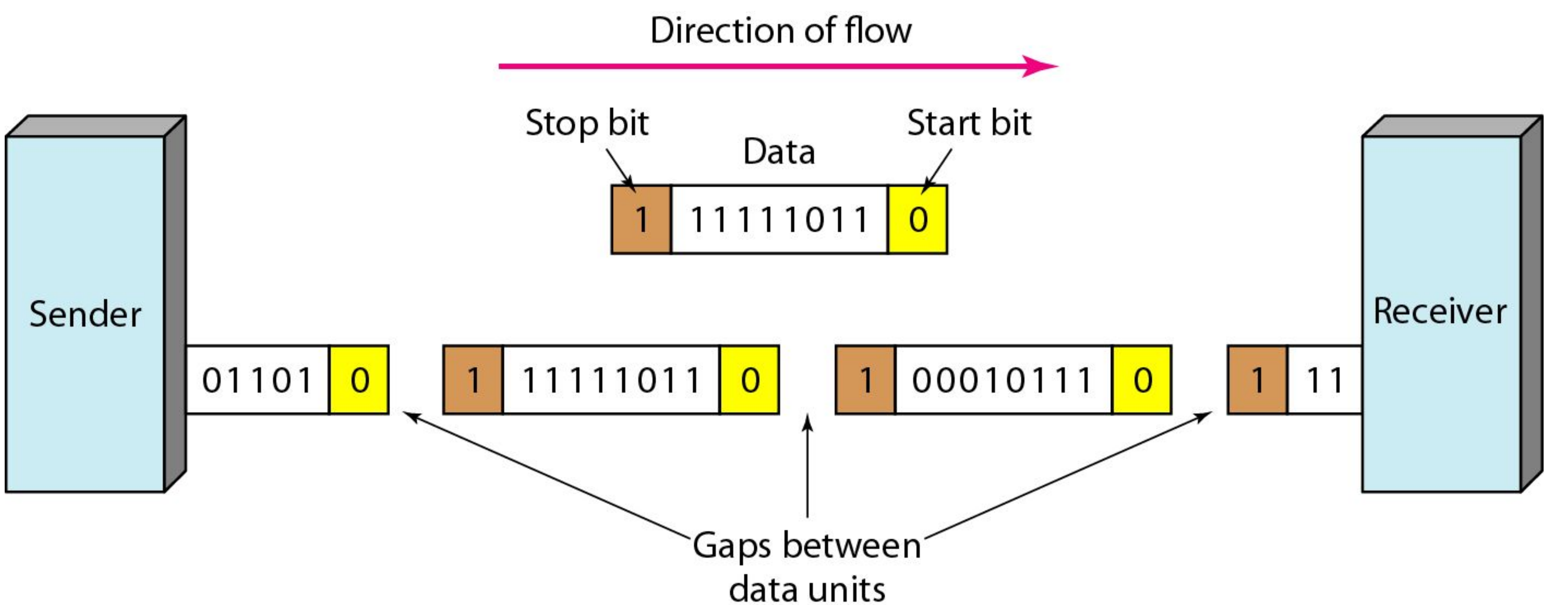
Note

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

Note

Asynchronous here means "asynchronous at the byte level," but the bits are still synchronized; their durations are the same.

Figure 4.34 *Asynchronous transmission*

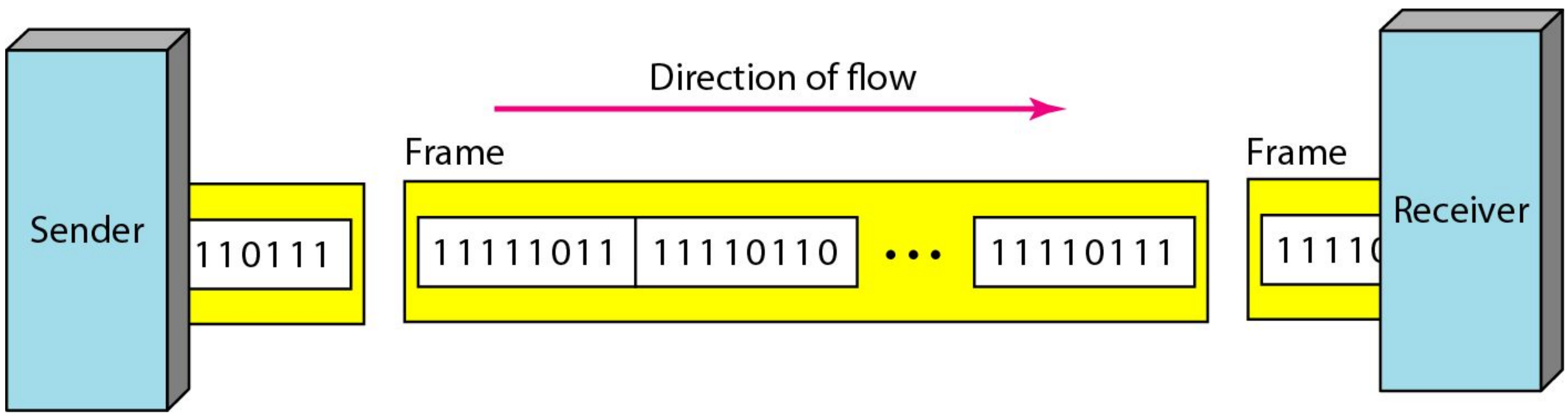


- In **asynchronous transmission**, information is received and translated by agreed upon patterns. As long as those patterns are followed, the receiving device can retrieve the information without regard to the rhythm in which it is sent.
- In **asynchronous transmission**, we send **1 start bit** (0) at the beginning and **1 or more stop bits** (1s) at the end of each byte. There may be a **gap** between each byte.

Note

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 bits are usually sent as bytes and many bytes are grouped in a frame. A frame is identified with a start and an end byte.

Figure 4.35 *Synchronous transmission*



- In **synchronous transmission,** the bit stream is combined into longer "frames," which may contain multiple bytes. Each byte, however, is introduced onto the transmission link without a gap between it and the next one.
- It is left to the receiver to separate the bit stream into bytes for decoding purposes.
- 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.

Isochronous

- In isochronous transmission we cannot have uneven gaps between frames.
- Transmission of bits is fixed with equal gaps.
- In **real-time** audio and video, in which uneven delays between frames are not acceptable, synchronous transmission fails, the entire stream of bits must be **synchronized**. The isochronous transmission guarantees that the **data arrive at a fixed rate**.

5-1 DIGITAL-TO-ANALOG CONVERSION

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

Topics discussed in this section:

- **Aspects of Digital-to-Analog Conversion**
- **Amplitude Shift Keying**
- **Frequency Shift Keying**
- **Phase Shift Keying**
- **Quadrature Amplitude Modulation**

Digital to Analog Conversion

- Digital data needs to be carried on an analog signal.
- A **carrier** signal (frequency f_c) performs the function of transporting the digital data in an analog waveform.
- The analog carrier signal is manipulated to uniquely identify the digital data being carried.

Figure 5.1 ***Digital-to-analog conversion***

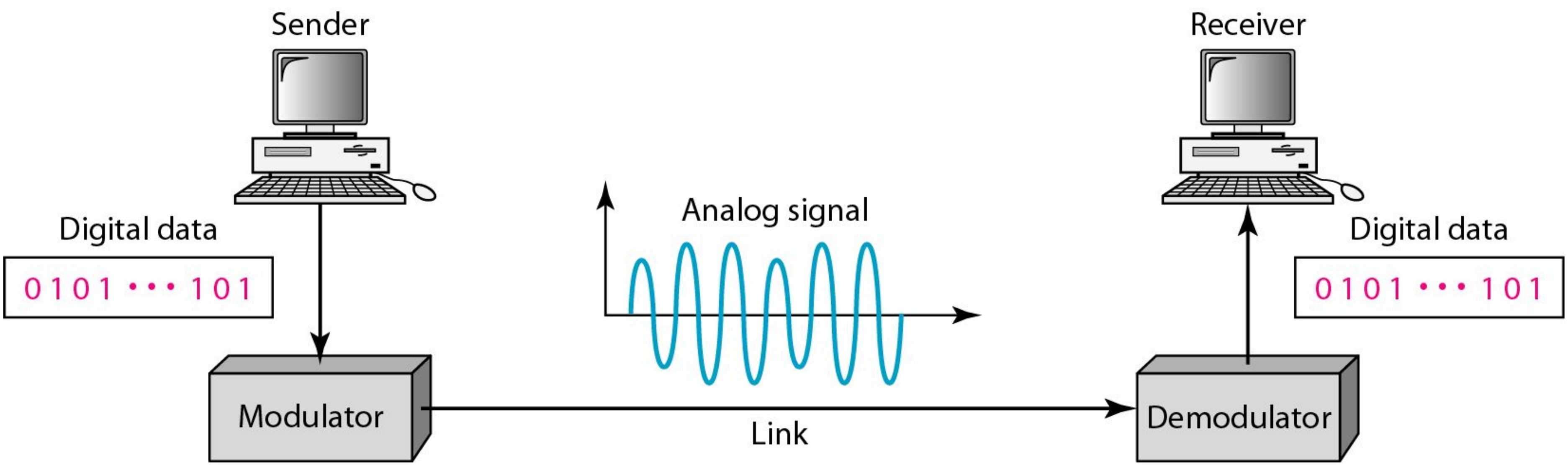
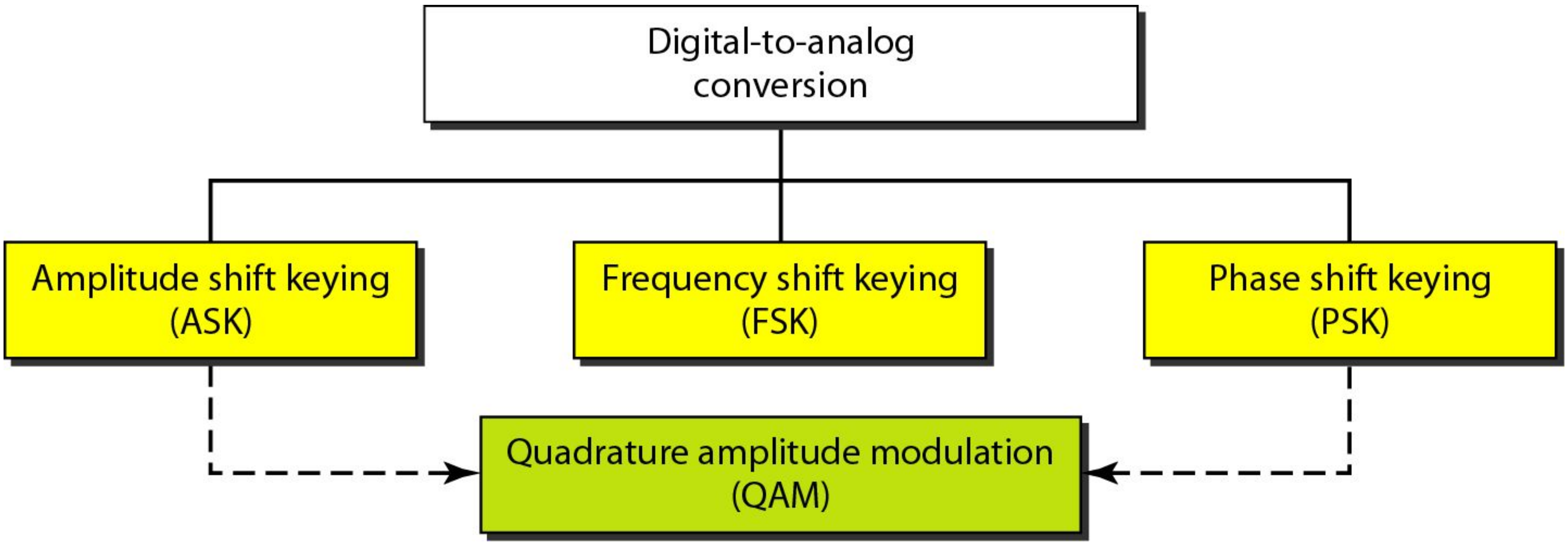


Figure 5.2 *Types of digital-to-analog conversion*



Note

Bit rate, N, is the number of bits per second (bps). Baud rate is the number of signal elements per second (bauds). In the analog transmission of digital data, the signal or baud rate is less than or equal to the bit rate.

S=Nx1/r bauds

Where r is the number of data bits per signal element.

Example 5.1

An analog signal carries 4 bits per signal element. If 1000 signal elements are sent per second, find the bit rate.

Solution

In this case, r = 4, S = 1000, and N is unknown. We can find the value of N from

$$S = N \times \frac{1}{r} \qquad \text{or} \qquad N = S \times r = 1000 \times 4 = 4000 \text{ bps}$$

Example 5.2

An analog signal has a bit rate of 8000 bps and a baud rate of 1000 baud. How many data elements are carried by each signal element? How many signal elements do we need?

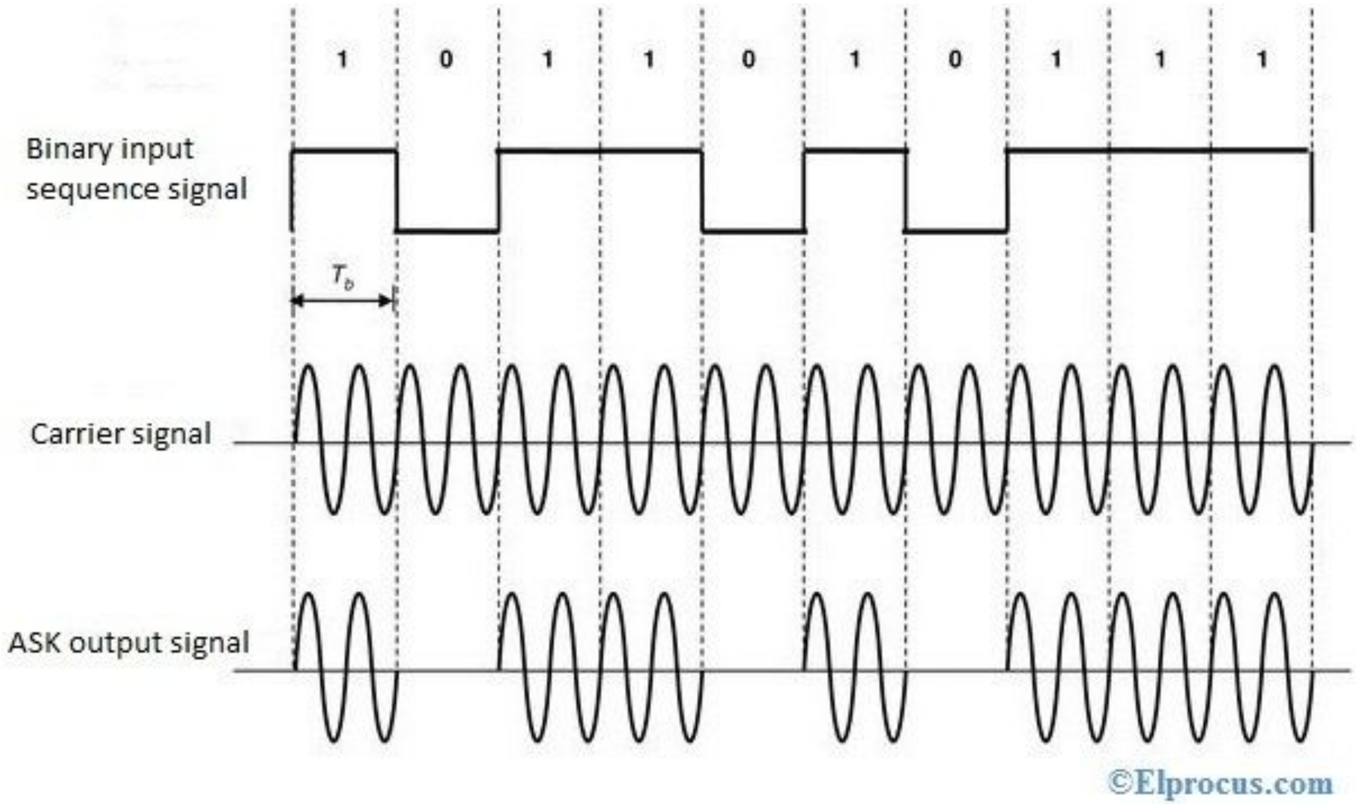
Solution

In this example, S = 1000, N = 8000, and r and L are unknown. We find first the value of r and then the value of L.

$$S = N \times \frac{1}{r} \rightarrow r = \frac{N}{S} = \frac{8000}{1000} = 8 \text{ bits/baud}$$

$$r = \log_2 L \rightarrow L = 2^r = 2^8 = 256$$

1. Amplitude Shift Keying (ASK)



Amplitude Shift Keying (ASK)

- ASK is implemented by changing the amplitude of a carrier signal to reflect amplitude levels in the digital signal.
- For example: a digital "1" could not affect the signal, whereas a digital "0" would, by making it zero.
- The line encoding will determine the values of the analog waveform to reflect the digital data being carried.

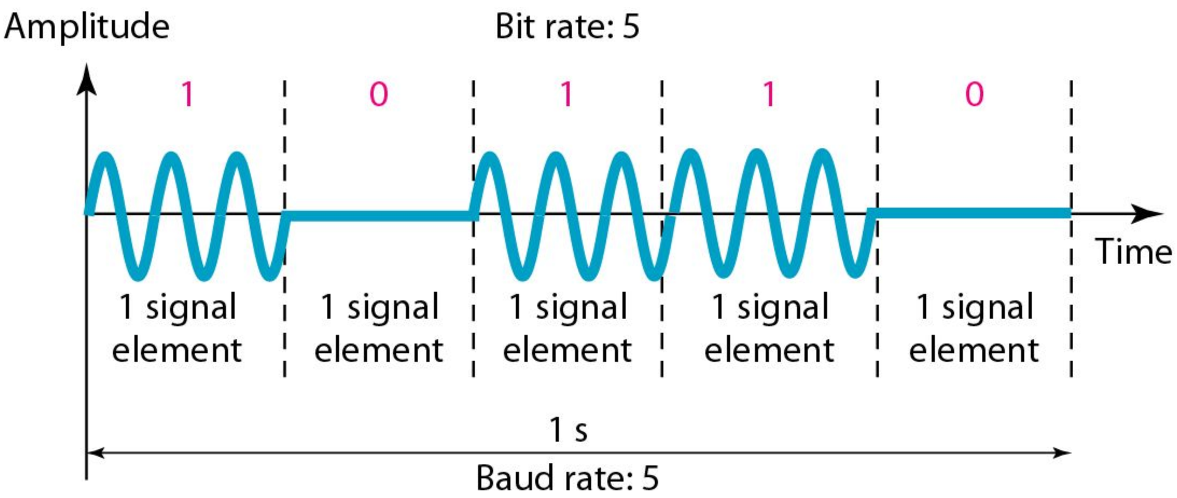
Bandwidth of ASK

- The bandwidth B of ASK is proportional to the signal rate S.

$$\mathbf{B = (1+d)S}$$

- "d" is due to modulation and filtering, lies between 0 and 1.

Figure 5.3 *Binary amplitude shift keying*



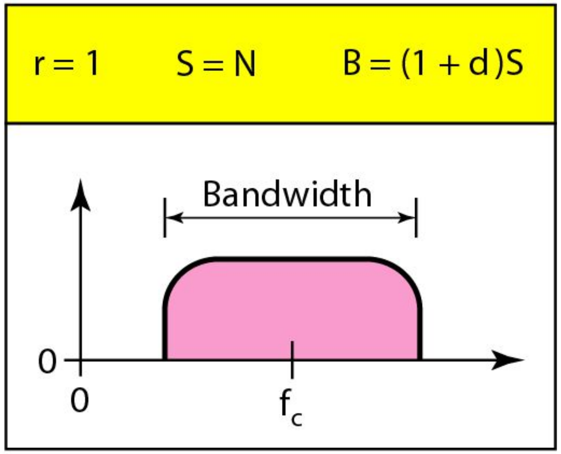
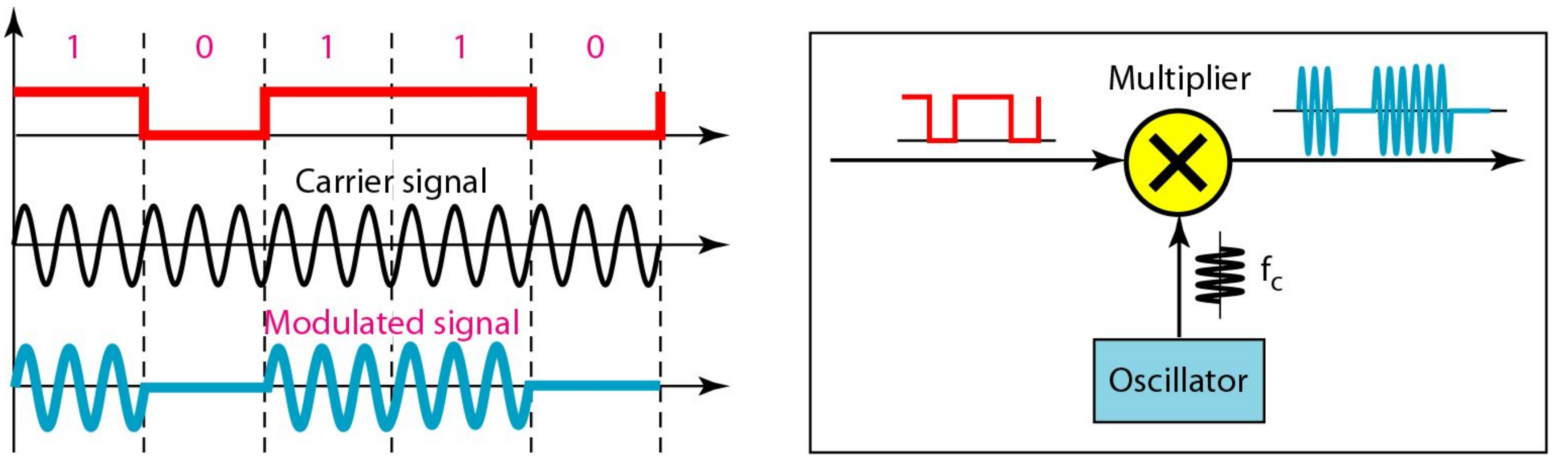


Figure 5.4 *Implementation of binary ASK*



Example 5.3

We have an available bandwidth of 100 kHz which spans from 200 to 300 kHz. What are the carrier frequency and the bit rate if we modulated our data by using ASK with d = 1?

Solution

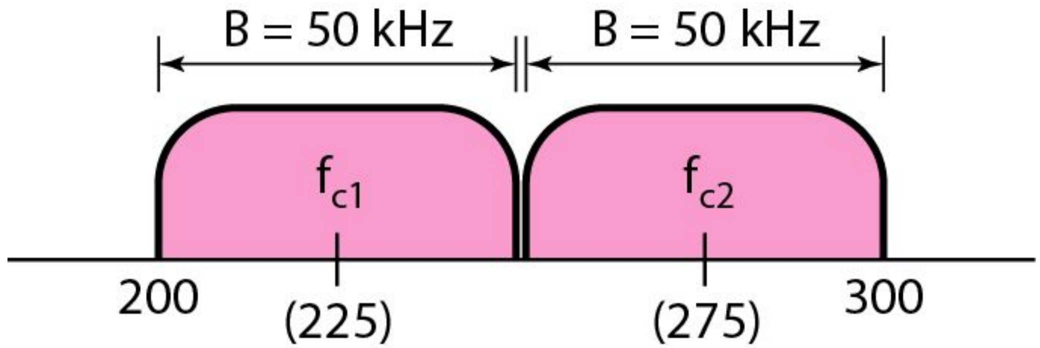
The middle of the bandwidth is located at 250 kHz. This means that our carrier frequency can be at f_c = 250 kHz. We can use the formula for bandwidth to find the bit rate (with d = 1 and r = 1).

$$B = (1 + d) \times S = 2 \times N \times \frac{1}{r} = 2 \times N = 100 \text{ kHz} \quad \Rightarrow \quad N = 50 \text{ kbps}$$

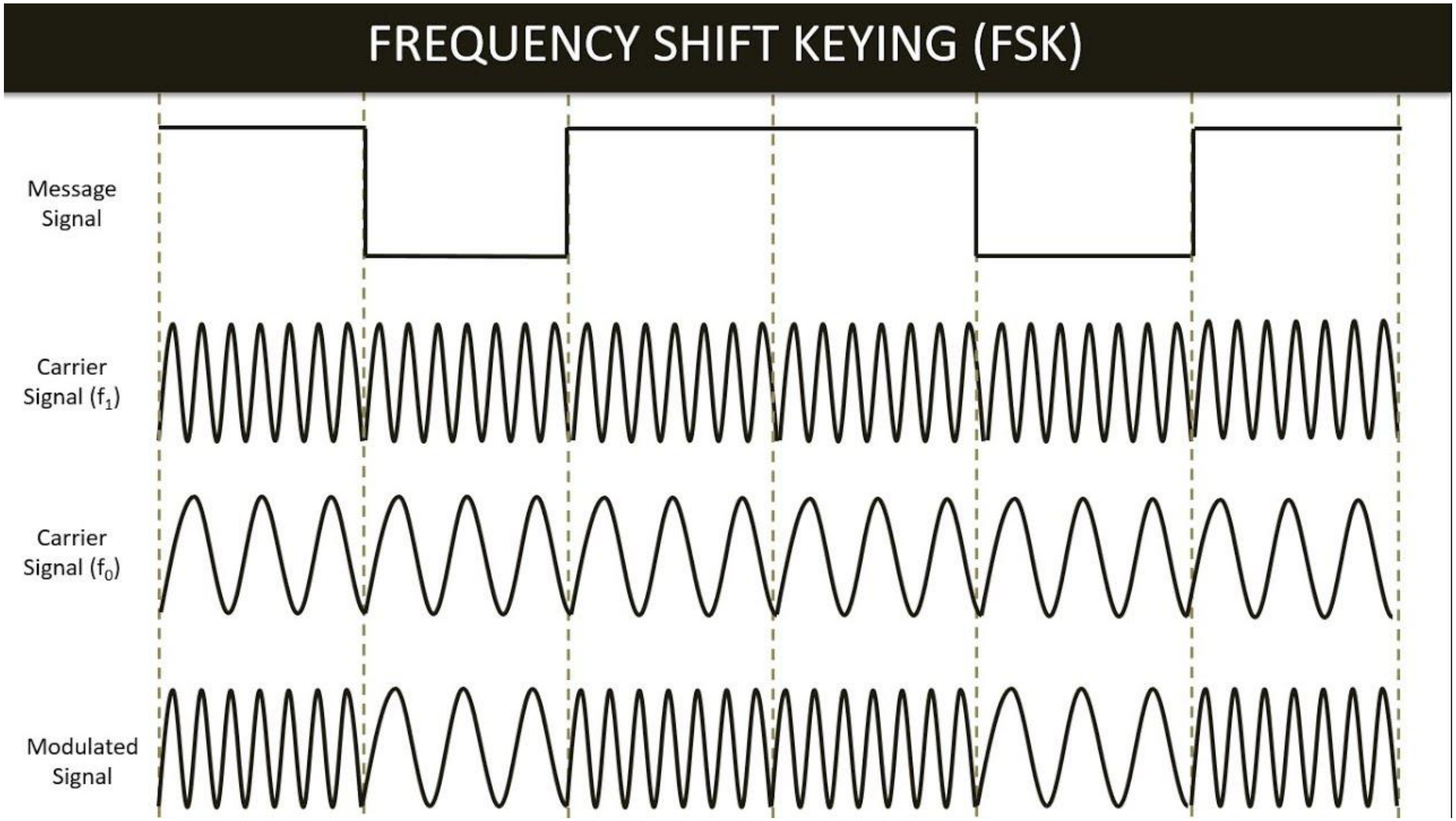
Example 5.4

In data communications, we normally use full-duplex links with communication in both directions. We need to divide the bandwidth into two with two carrier frequencies, as shown in Figure 5.5. The figure shows the positions of two carrier frequencies and the bandwidths. The available bandwidth for each direction is now 50 kHz, which leaves us with a data rate of 25 kbps in each direction.

Figure 5.5 *Bandwidth of full-duplex ASK used in Example 5.4*



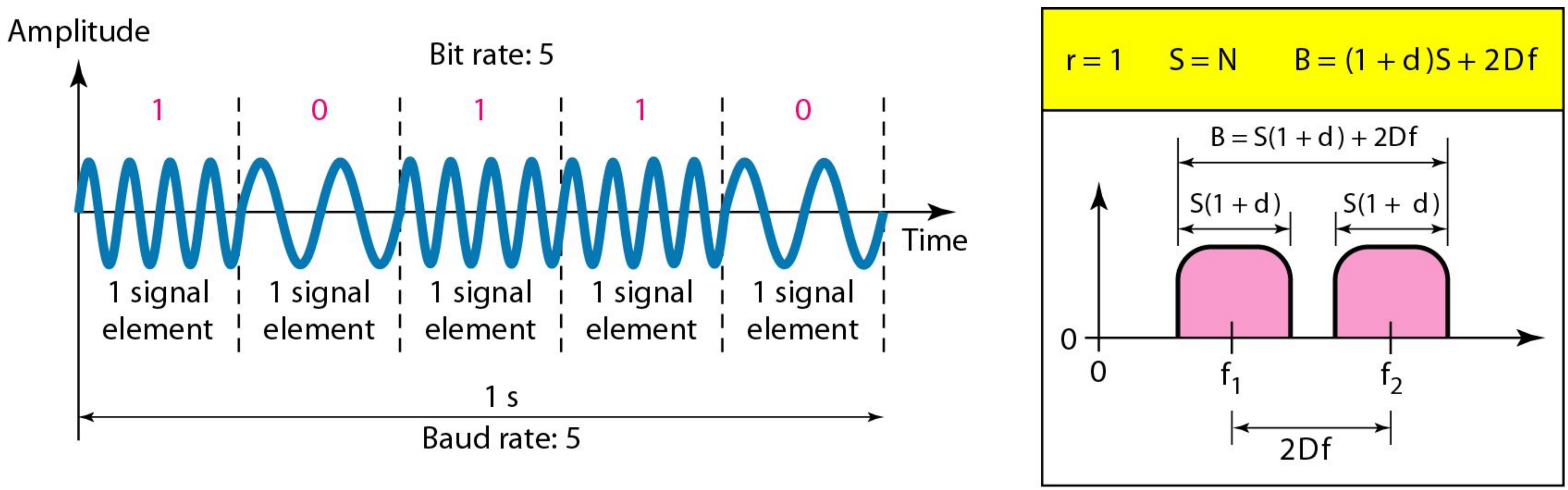
2. Frequency Shift Keying(FSK)



Frequency Shift Keying(FSK)

- The digital data stream changes the frequency of the carrier signal, f_c.
- For example, a "1" could be represented by $f_1=f_c+\Delta f$, and a "0" could be represented by $f_2=f_c-\Delta f$.

Figure 5.6 *Binary frequency shift keying*



Bandwidth of FSK

- If the difference between the two frequencies (f_1 and f_2) is $2\Delta f$, then the required BW B will be:

$$\mathbf{B = (1+d) \times S + 2\Delta f}$$

Example 5.5

We have an available bandwidth of 100 kHz which spans from 200 to 300 kHz. What should be the carrier frequency and the bit rate if we modulated our data by using FSK with d = 1?

Solution

This problem is similar to Example 5.3, but we are modulating by using FSK. The midpoint of the band is at 250 kHz. We choose 2Δf to be 50 kHz; this means

$$B = (1 + d) \times S + 2\Delta f = 100 \quad \longrightarrow \quad 2S = 50 \text{ kHz} \quad S = 25 \text{ kbaud} \quad N = 25 \text{ kbps}$$

Coherent and Non Coherent

- In a non-coherent FSK scheme, when we change from one frequency to the other, we do not adhere to the current phase of the signal.
- In coherent FSK, the switch from one frequency signal to the other only occurs at the same phase in the signal.

Multi level FSK

- Similarly to ASK, FSK can use **multiple bits per signal element.**
- That means we need to provision for multiple frequencies, each one to represent a group of data bits.
- The bandwidth for FSK can be higher

$$B = (1+d)xS + (L-1)/2\Delta f = LxS$$

Figure 5.7 ***Bandwidth of MFSK used in Example 5.6***

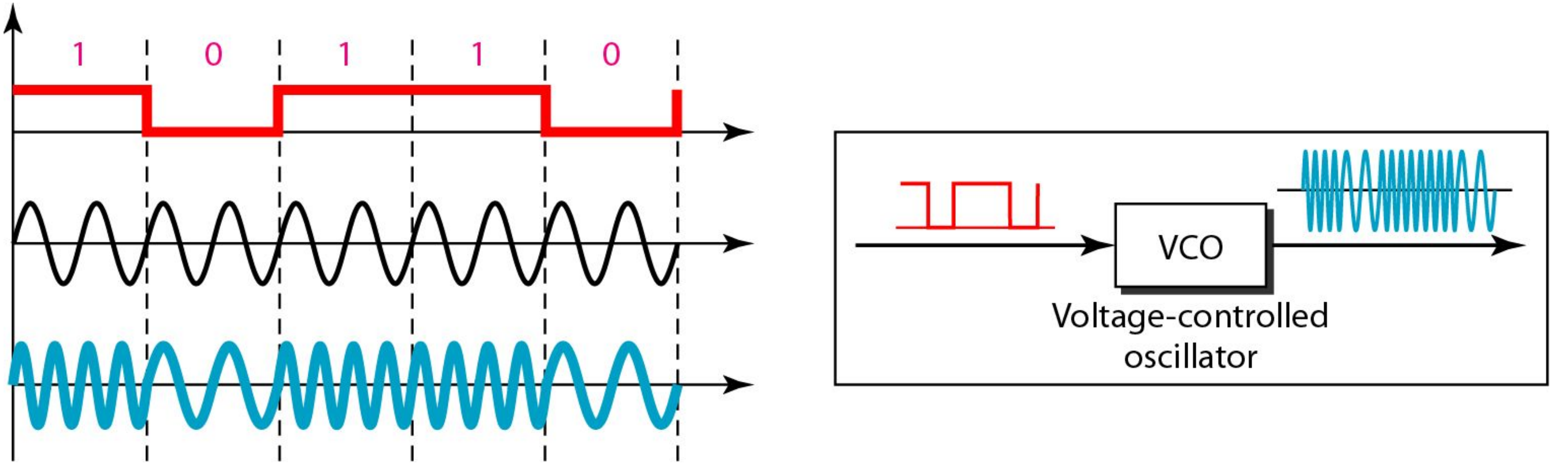
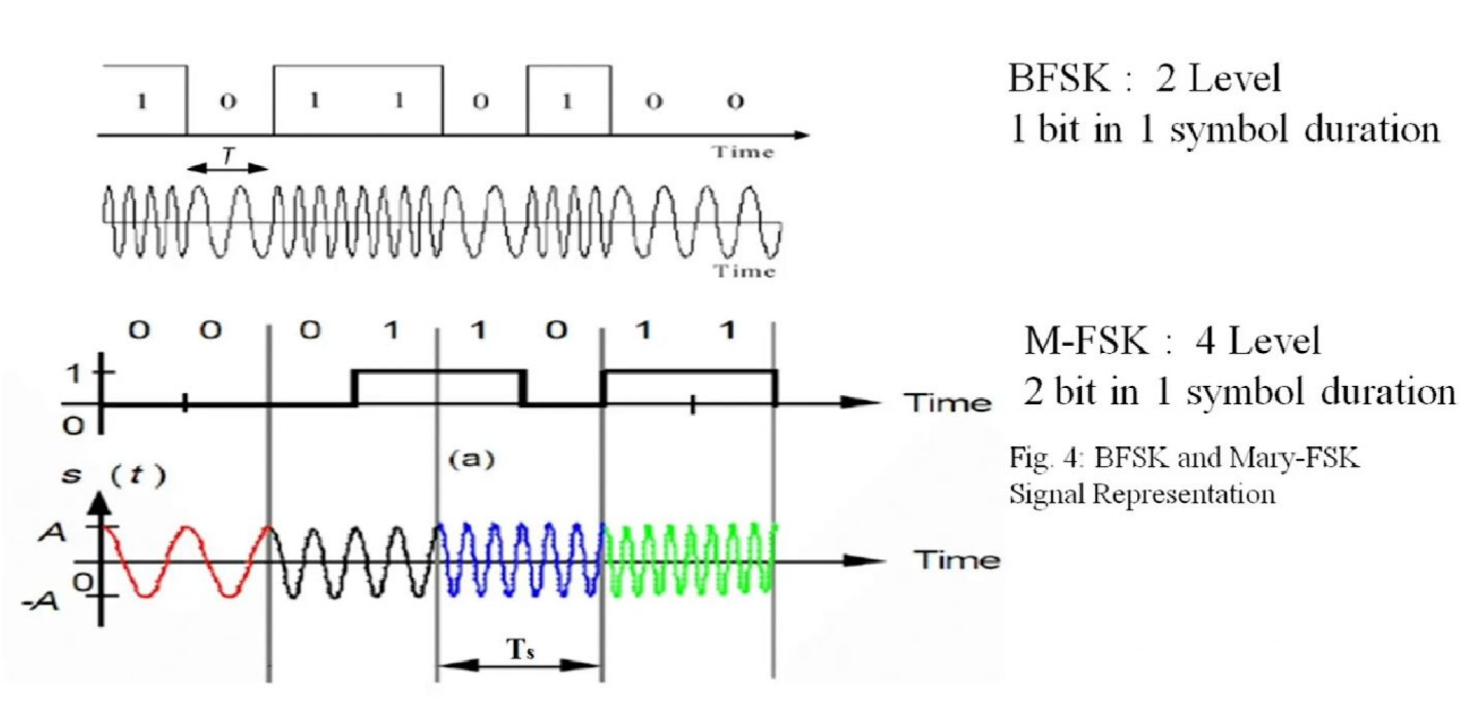


Figure 5.7 ***Bandwidth of MFSK used in Example 5.6***



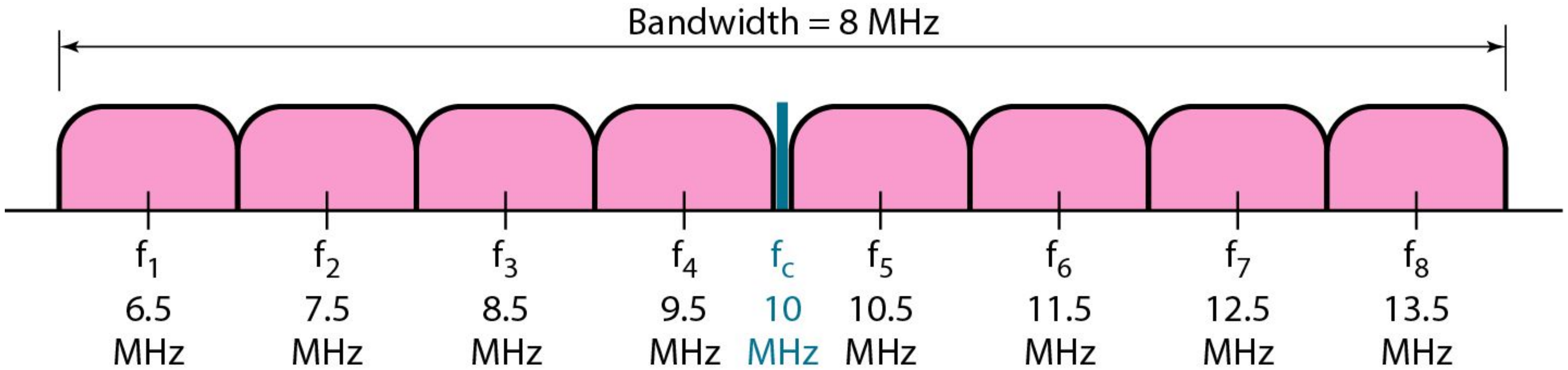
Example 5.6

We need to send data 3 bits at a time at a bit rate of 3 Mbps. The carrier frequency is 10 MHz. Calculate the number of levels (different frequencies), the baud rate, and the bandwidth.

Solution

We can have $L = 2^3 = 8$. The baud rate is $S = 3$ Mbps/3 = 1 Mbaud. This means that the carrier frequencies must be 1 MHz apart ($2\Delta f = 1$ MHz). The bandwidth is $B = 8 \times 1M = 8M$. Figure 5.8 shows the allocation of frequencies and bandwidth.

Figure 5.8 ***Bandwidth of MFSK used in Example 5.6***



3. Phase Shift Keyeing

- We vary the phase shift of the carrier signal to represent digital data.
- Both peak amplitude and frequency remain constant as the phase changes.
- PSK is more common than ASK or FSK
- The simplest PSK is binary PSK

3. Phase Shift Keyeing

- We vary the phase shift of the carrier signal to represent digital data.
- The bandwidth requirement, B is:

$$\mathbf{B = (1+d)xS}$$

- PSK is much more robust than ASK as it is not that vulnerable to noise, which changes amplitude of the signal. Noise can change the amplitude easier than it can change the phase

Figure 5.9 *Binary phase shift keying*

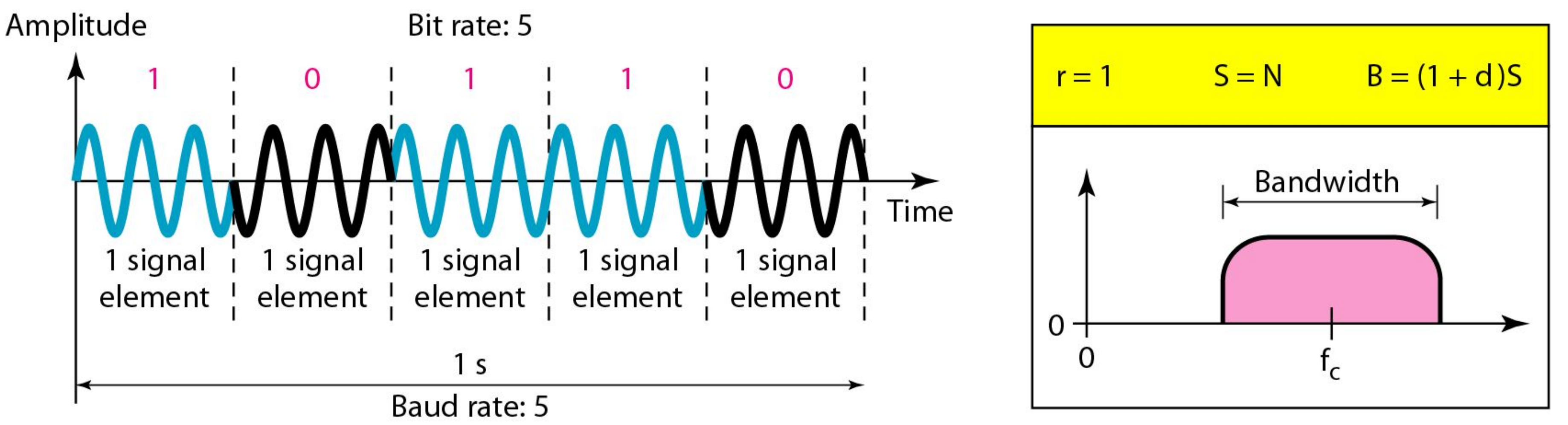
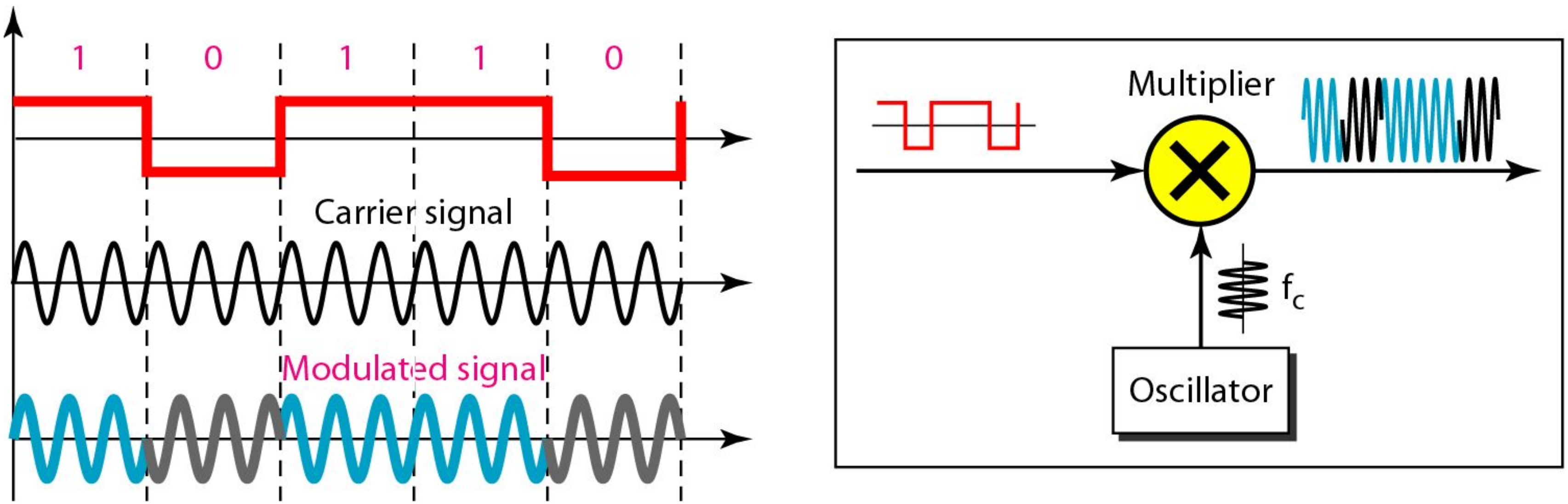


Figure 5.10 *Implementation of BASK*



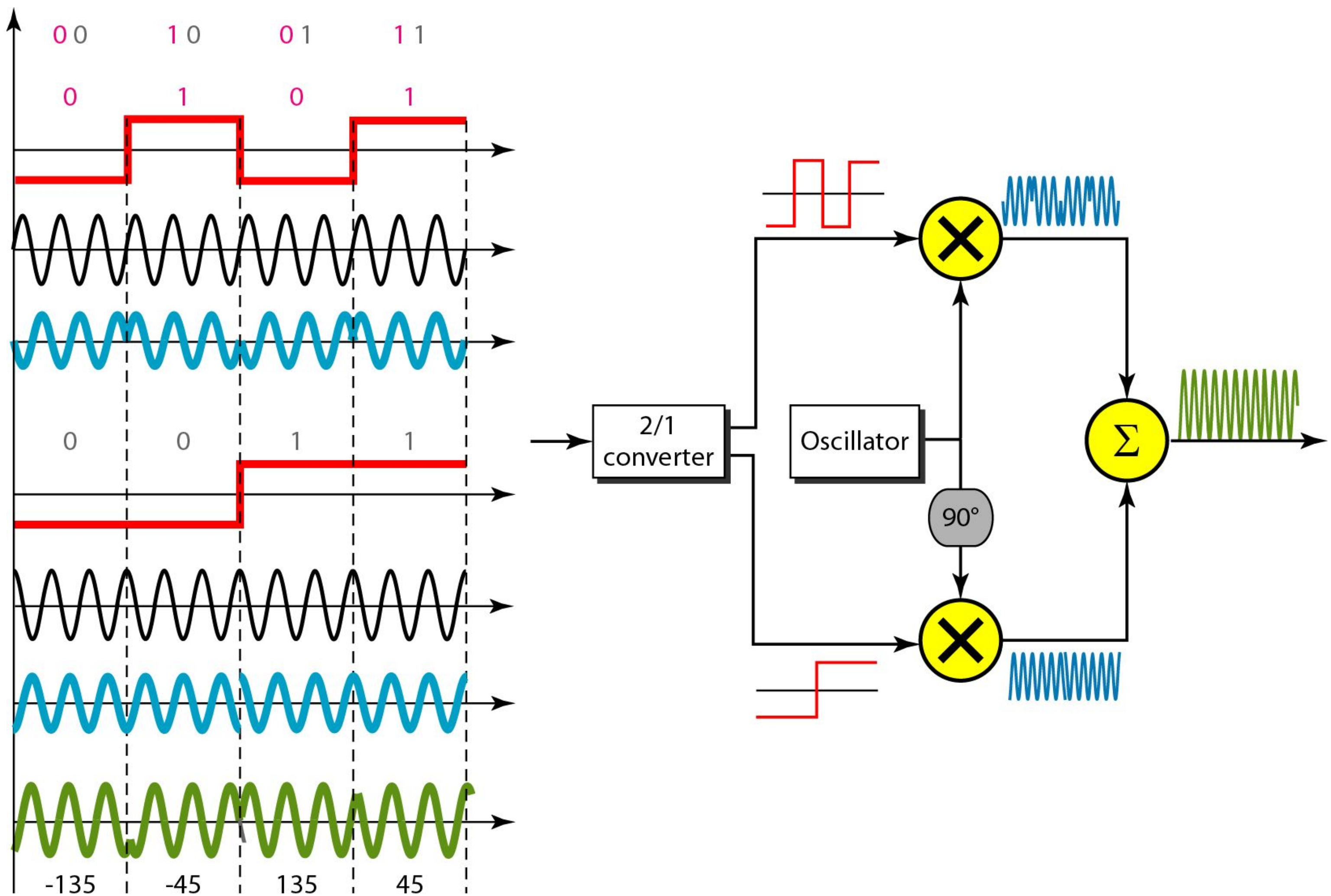
Quadrature PSK

- To increase the bit rate, we can code 2 or more bits onto one signal element.
- In QPSK, we parallelize the bit stream so that every two incoming bits are split up and PSK a carrier frequency. One carrier frequency is phase shifted 90° from the other - in quadrature.
- The two PSKed signals are then added to produce one of 4 signal elements. L = 4 here.

Quadrature PSK

- *QPSK because it uses two separate BPSK modulations; one* is in-phase, the other quadrature (out-of-phase).
  1. The incoming bits are first passed through a serial-to-parallel conversion that sends one bit to one modulator and the next bit to the other modulator.
  2. The two composite signals created by each multiplier are sine waves with the same frequency, but different phases.
  3. We can send 2 bits per signal element (r = 2).

Figure 5.11 *QPSK and its implementation*



Example 5.7

Find the bandwidth for a signal transmitting at 12 Mbps for QPSK. The value of $d = 0$.

Solution

For QPSK, 2 bits is carried by one signal element. This means that $r = 2$. So the signal rate (baud rate) is $S = N \times (1/r) = 6$ Mbaud. With a value of $d = 0$, we have $B = S = 6$ MHz.

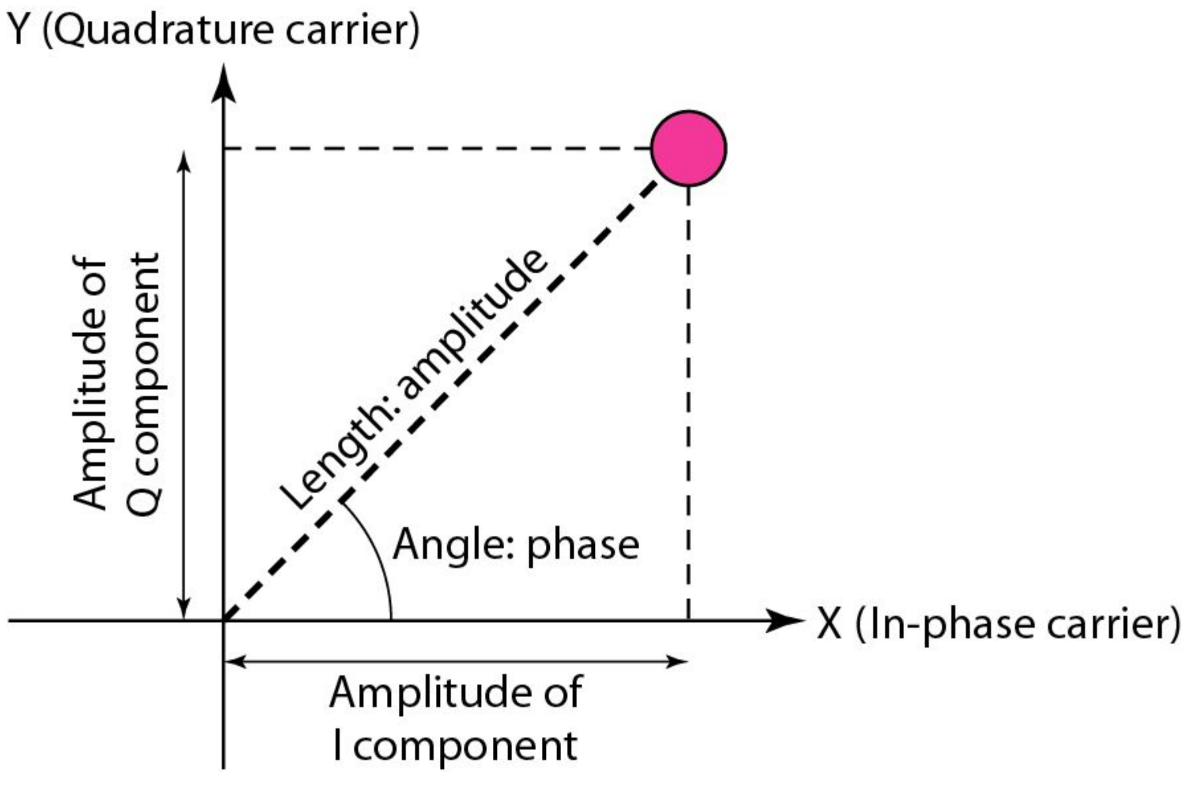
Constellation Diagrams

- A **constellation diagram** helps us to define the amplitude and phase of a signal when we are using two carriers, one in quadrature of the other.
- The X-axis represents the in-phase carrier and the Y-axis represents quadrature carrier.

Constellation Diagrams

- Diagram is useful when we are dealing with multilevel ASK, PSK, or QAM
- In a constellation diagram, a signal element type is represented as a dot

Figure 5.12 *Concept of a constellation diagram*



Constellation Diagrams

- The projection of the point on the *X axis* defines the peak amplitude of the in-phase component
- The projection of the point on the *Y axis defines the peak amplitude of the quadrature component.*

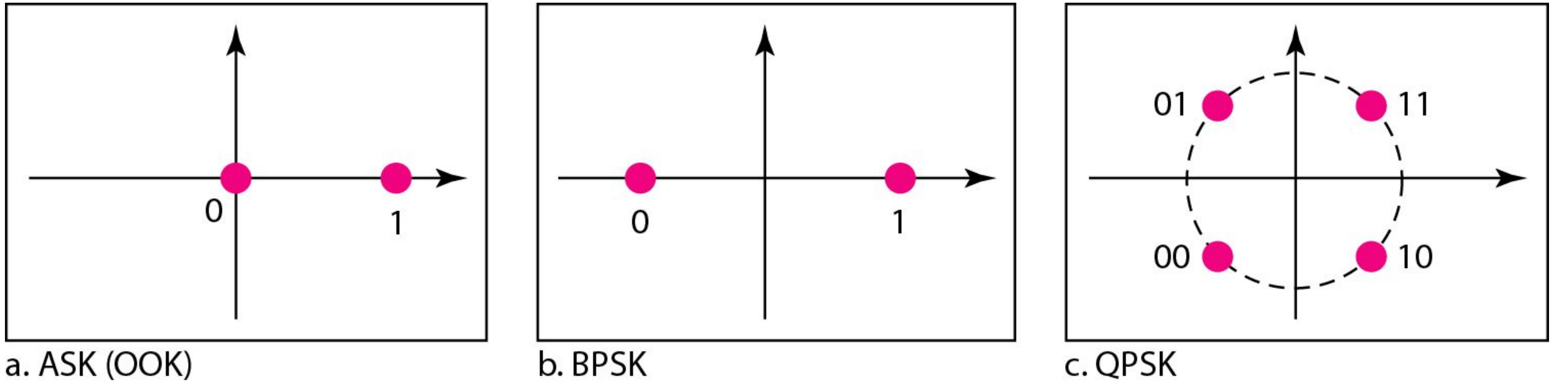
Example 5.8

Show the constellation diagrams for an ASK (OOK), BPSK, and QPSK signals.

Solution

Figure 5.13 shows the three constellation diagrams.

Figure 5.13 *Three constellation diagrams*



1. **Figure 5.13** ***Three constellation diagrams***

1. For ASK, we are using only an in-phase carrier. Therefore, the two points should be on the *X axis,* Binary 0 has an amplitude of 0 V; binary 1 has an amplitude of 1 V (for example) The points are located at the origin and at 1 unit.
2. BPSK also uses only an in-phase carrier. However, we use a polar NRZ signal for modulation It creates two types of signal elements, one with amplitude 1 and the other with amplitude -1. BPSK creates two different signal elements, one with amplitude 1 V and in phase and the other with amplitude 1 V and 180° out of phase.
3. QPSK uses two carriers, one in-phase and the other quadrature. The point representing 11 is made of two combined signal elements, both with an amplitude of 1 V. One element is represented by an in-phase carrier, the other

Note

Quadrature amplitude modulation is a combination of ASK and PSK.

Figure 5.14 *Constellation diagrams for some QAMs*

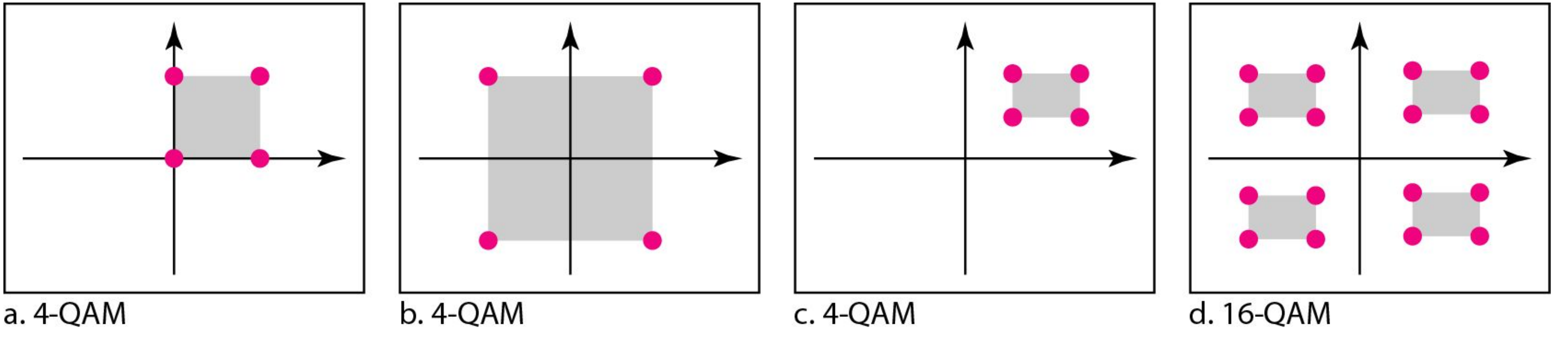


Figure 5.14 *Constellation diagrams for some QAMs*

The possible variations of QAM are numerous

1. Simplest 4-QAM scheme (four different signal element types) using a unipolar NRZ signal to modulate each carrier.
2. 4-QAM using polar NRZ-exactly the same as QPSK
3. QAM-4 in which we used a signal with two positive levels to modulate each of the two carriers
4. 16-QAM constellation of a signal with eight levels, four positive and four negative.

Module – 2

DIGITAL TRANSMISSION

Digital to Digital Conversion

• Data can be analog or digital, so can be the signal that represents it.

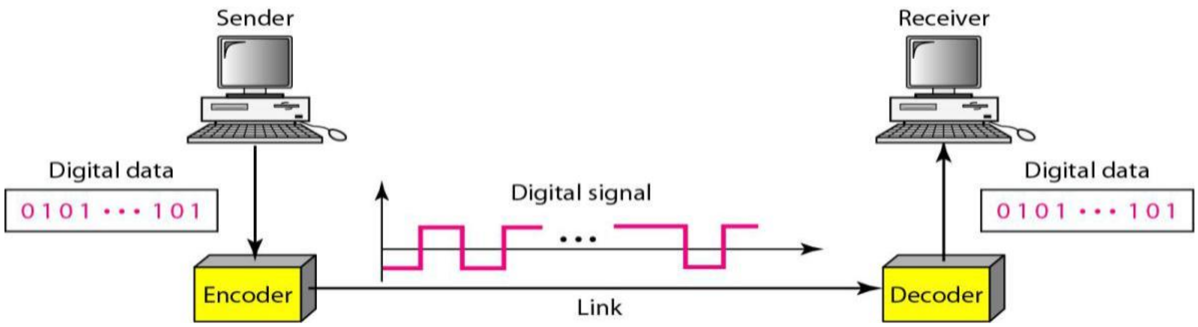
• Signal encoding is the conversion from analog/digital data to analog/digital signal.

• The possible encodings are: 1) Digital data to digital signal 2) Digital data to analog signal 3) Analog data to digital signal 4) Analog data to analog signal

Line Coding

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

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.



Characteristics

Different characteristics of digital signal are

1) Signal Element Vs Data Element
2) Data Rate Vs Signal Rate
3) Bandwidth
4) Baseline Wandering
5) DC Components
6) Built-in Error Detection
7) Self-synchronization
8) Immunity to Noise and Interference
9) Complexity

Anjan kumar K N, Devaraju B M, Medha Gourayya, Dept. of CSE, RNSIT

1) Data Element vs. Signal Element

Data Element	Signal Element
A data-element is the smallest entity that can represent a piece of information (Figure 4.2).	A signal-element is shortest unit (timewise) of a digital-signal.
A data-element is the bit.	A signal-element carries data-elements.
Data-elements are being carried.	Signal-elements are the carriers.

- Ratio r is defined as number of data-elements carried by each signal-element.

2) Data Rate vs. Signal Rate

Data Rate	Signal Rate
The data-rate defines the number of data-elements (bits) sent in 1 sec.	The signal-rate is the number of signal-elements sent in 1 sec.
The unit is bits per second (bps).	The unit is the baud.
The data-rate is sometimes called the bit-rate.	The signal-rate is sometimes called the pulse rate, the modulation rate, or the baud rate
Goal in data-communications: increase the data-rate.	Goal in data-communications: decrease the signal-rate.
Increasing the data-rate increases the speed of transmission.	Decreasing the signal-rate decreases the bandwidth requirement.

- The relationship between data-rate and signal-rate is given by

$$S_{ave} = c \times N \times (1/r) \quad \text{baud}$$

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

- This relationship depends on
  → value of r.
  → data pattern.

(If we have a data pattern of all 1s or all 0s, the signal-rate may be different from a data pattern of alternating 0s and 1s).

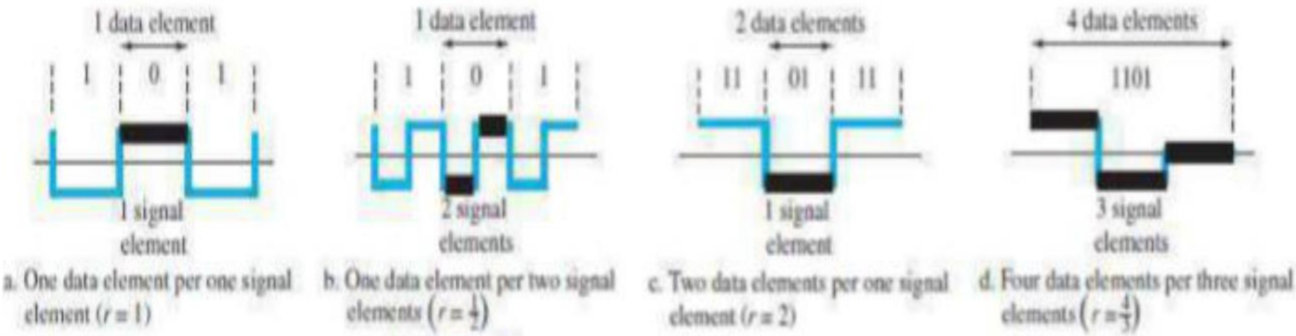


Figure 4.2 *Signal element versus data element*

Anjan kumar K N, Devaraju B M, Medha Gourayya, Dept. of CSE, RNSIT

3) **Bandwidth**

Digital signal that carries information is non-periodic. The bandwidth of a non-periodic signal is continuous with an infinite range. However, most digital-signals we encounter in real life have a bandwidth with finite values. The effective bandwidth is finite.The baud rate, not the bit-rate, determines the required bandwidth for a digital-signal.More changes in the signal mean injecting more frequencies into the signal. (Frequency means change and change means frequency.) The bandwidth refers to range of frequencies used for transmitting a signal. Relationship b/w baud rate (signal-rate) and the bandwidth (range of frequencies) is given as $B_{min} = c \times N \times (1/r)$

where N = data-rate (in bps)
c = case factor, which varies for each case
r = previously defined factor
Bmin = minimum bandwidth

4) **Baseline Wandering**

While decoding, the receiver calculates a running-average of the received signal-power. This average is called the baseline.The incoming signal-power is estimated against this baseline to determine the value of the data-element.A long string of 0s or 1s can cause a drift in the baseline (baseline wandering). Thus, make it difficult for the receiver to decode correctly.A good line-coding scheme needs to prevent baseline wandering.

5) DC Components.

When the voltage-level in a digital-signal is constant for a while, the spectrum creates very low frequencies. These frequencies around zero are called DC (direct-current) components. DC components present problems for a system that cannot pass low frequencies. For example: Telephone line cannot pass frequencies below 200 Hz. For Telephone systems, we need a scheme with no DC component.

6) **Built-in Error Detection**

Built-in error-detecting capability has to be provided to detect the errors that occurred during transmission.

7) Self Synchronization

To correctly interpret the signals received from the sender, the receiver's bit intervals must correspond exactly to the sender's bit intervals.If the receiver clock is faster or slower, the bit intervals are not matched and the receiver might misinterpret the signals.

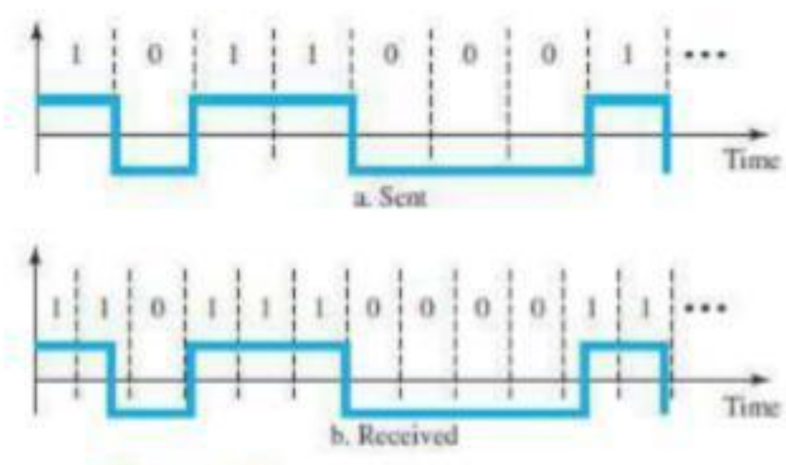


Figure 4.3 *Effect of lack of synchronization*

As shown in figure 4.3, we have a situation where the receiver has shorter bit duration. The sender sends 10110001, while the receiver receives 110111000011.

A self-synchronizing digital-signal includes timing-information in the data being transmitted.

- This can be achieved if there are transitions in the signal that alert the receiver to the beginning, middle, or end of the pulse.

- If the receiver's clock is out-of-synchronization, these points can reset the clock.

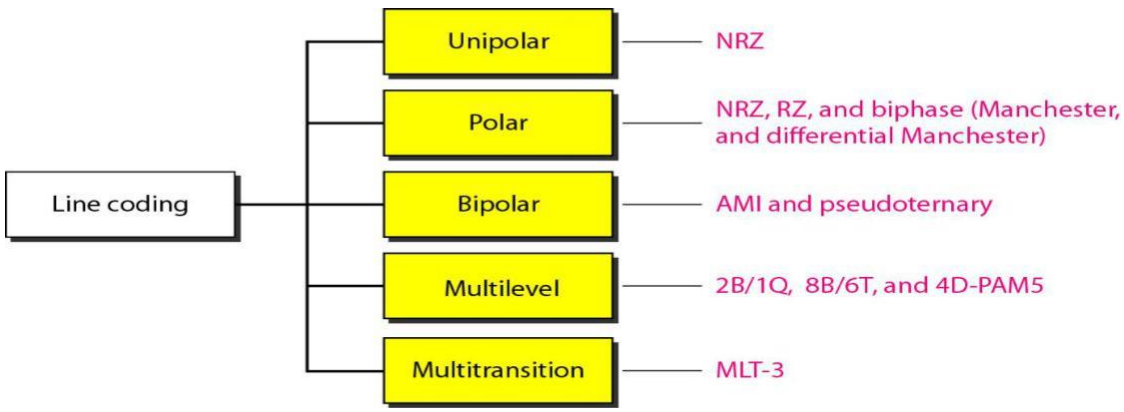
8) Immunity to Noise & Interference

The code should be immune to noise and other interferences.

9) Complexity

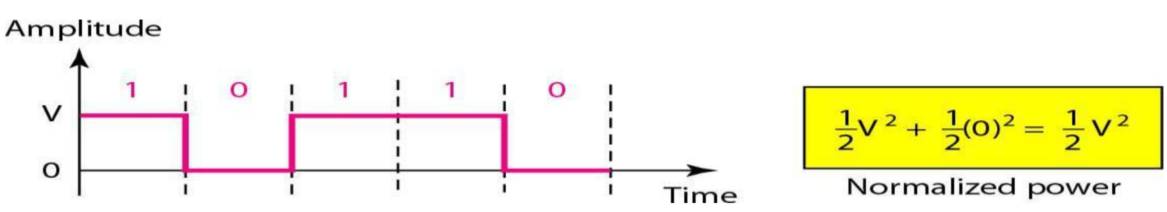
A complex scheme is more costly to implement than a simple one.For ex: A scheme that uses 4 signal-levels is more difficult to interpret than one that uses only 2 levels.

Line Coding Schemes



Unipolar Scheme

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



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 I can be negative.

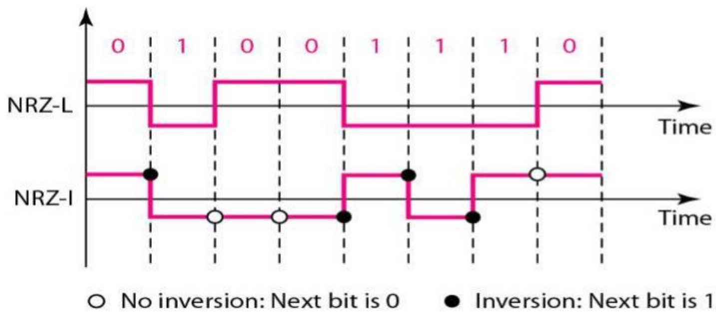
Non-Return-to-Zero (NRZ): In polar NRZ encoding, we use two levels of voltage

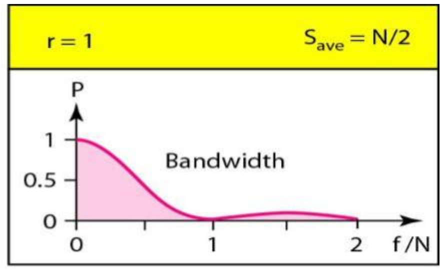
Anjan kumar K N, Devaraju B M, Medha Gourayya, Dept. of CSE, RNSIT

amplitude. We can have two versions of polar NRZ: NRZ-Land NRZ-I

In **NRZ-L** the level of the voltage determines the value of the bit.

In **NRZ-I** the inversion or the lack of inversion determines the value of the bit.



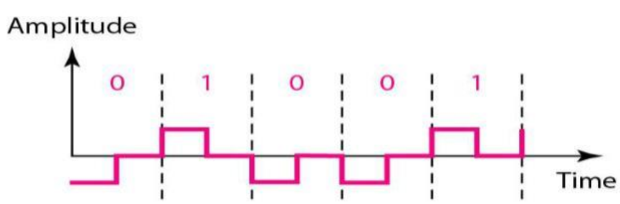


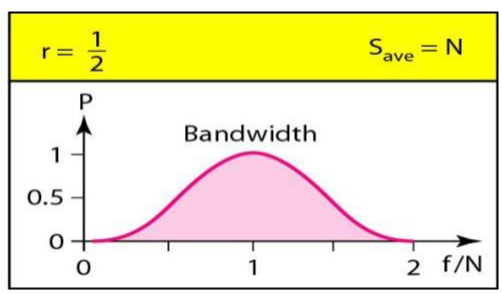
Return to Zero (RZ):

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. In the below Figure we see that the signal goes to 0 in the middle of each bit. It remains there until the beginning of the next bit.

The main disadvantage of RZ encoding is that it requires two signal changes to encode a bit and therefore occupies greater bandwidth.



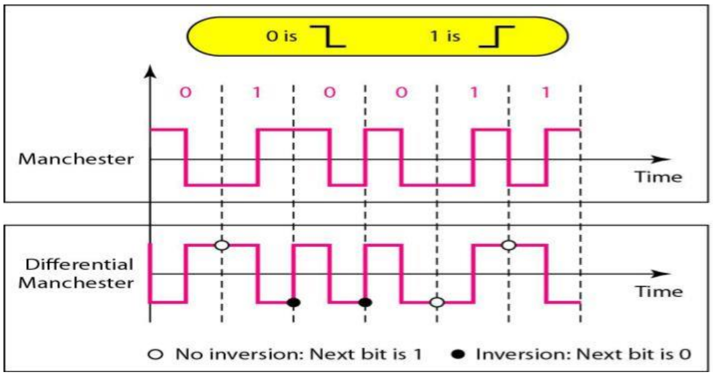


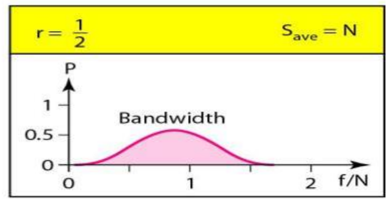
Manchester and Differential Manchester:

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.

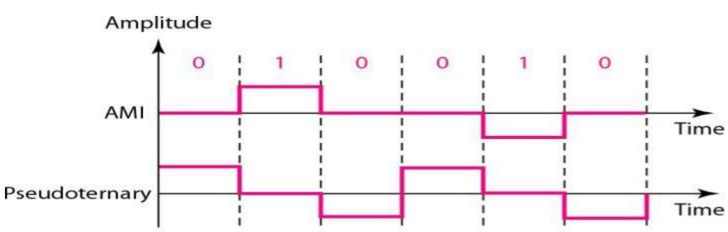


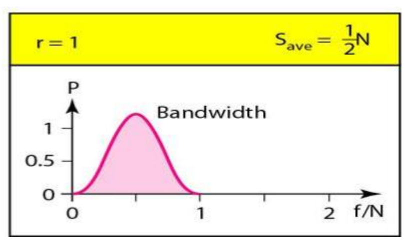


Bipolar Schemes

In bipolar encoding (sometimes called *multilevel binary),* there are three voltage levels: positive, negative, and zero. The voltage level for one data element is at zero, while the voltage level for the other element alternates between positive and negative.

AMI and Pseudoternary: A common bipolar encoding scheme is called bipolar alternate mark inversion (AMI). In the term alternate mark inversion, the word mark comes from telegraphy and means 1. So AMI means alternate I inversion. A neutral zero voltage represents binary O. Binary Is are represented by alternating positive and negative voltages. A variation of AMI encoding is called pseudoternary in which the 1 bit is encoded as a zero voltage and the 0 bit is encoded as alternating positive and negative voltages.





Anjan kumar K N, Devaraju B M, Medha Gourayya, Dept. of CSE, RNSIT

Physical Layer- 2

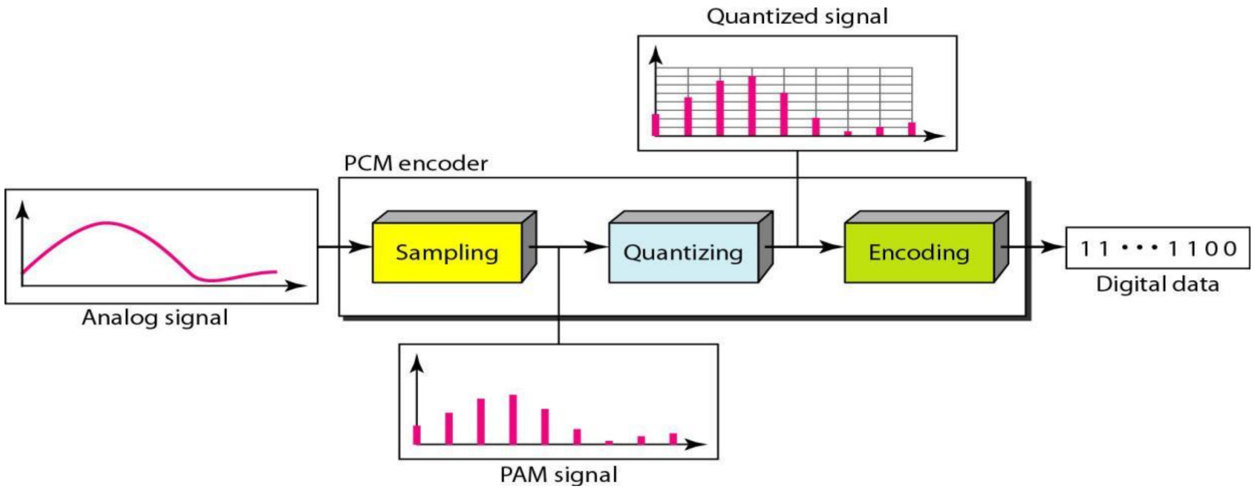
Analog-To-Digital Conversion

- The tendency today is to change analog signal to digital data.The two techniques are PCM(Pulse Code Modulation) and DM(Delta Modulation)

Pulse Code Modulation (PCM)

The most common technique to change an analog signal to digital data (digitization) is called pulse code modulation (PCM). A PCM encoder has three processes.

1. The analog signal is sampled.
2. The sampled signal is quantized.
3. The quantized values are encoded as streams of bits.

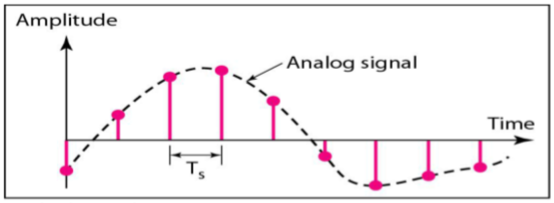


Sampling:

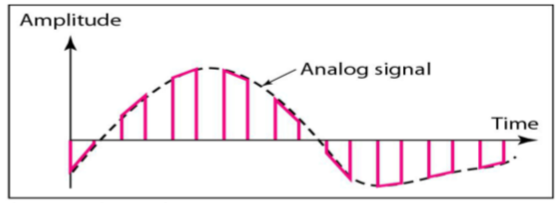
The first step in PCM is sampling. The analog signal is sampled every *Ts* s, where *Ts* is the sample interval or period.

The inverse of the sampling interval is called the sampling rate or sampling frequency and denoted by *fs*, where $fs = 1/Ts$.

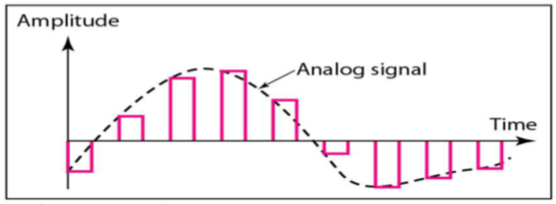
There are three sampling methods-ideal, natural, and flat-top.



a. Ideal sampling



b. Natural sampling



c. Flat-top sampling

In ideal sampling, pulses from the analog signal are sampled.

In natural sampling, a high-speed switch is turned on for only the small period of time when the sampling occurs. The result is a sequence of samples that retains the shape of the analog signal.

The most common sampling method, called **sample and hold**, however, creates flat-top samples by using a circuit.

The sampling process is sometimes referred to as **pulse amplitude modulation** (PAM).

Sampling Rate: According to the Nyquist theorem, to reproduce the original analog signal, one necessary condition is that the sampling rate be at least twice the highest frequency in the original signal.

Quantization

The result of sampling is a series of pulses with amplitude values between the maximum and minimum amplitudes of the signal.

The set of amplitudes can be infinite with nonintegral values between the two limits.

These values cannot be used in the encoding process.

The following are the steps in quantization:

1. We assume that the original analog signal has instantaneous amplitudes between *V*min and *V*max.

2. We divide the range into *L* zones, each of height (delta).

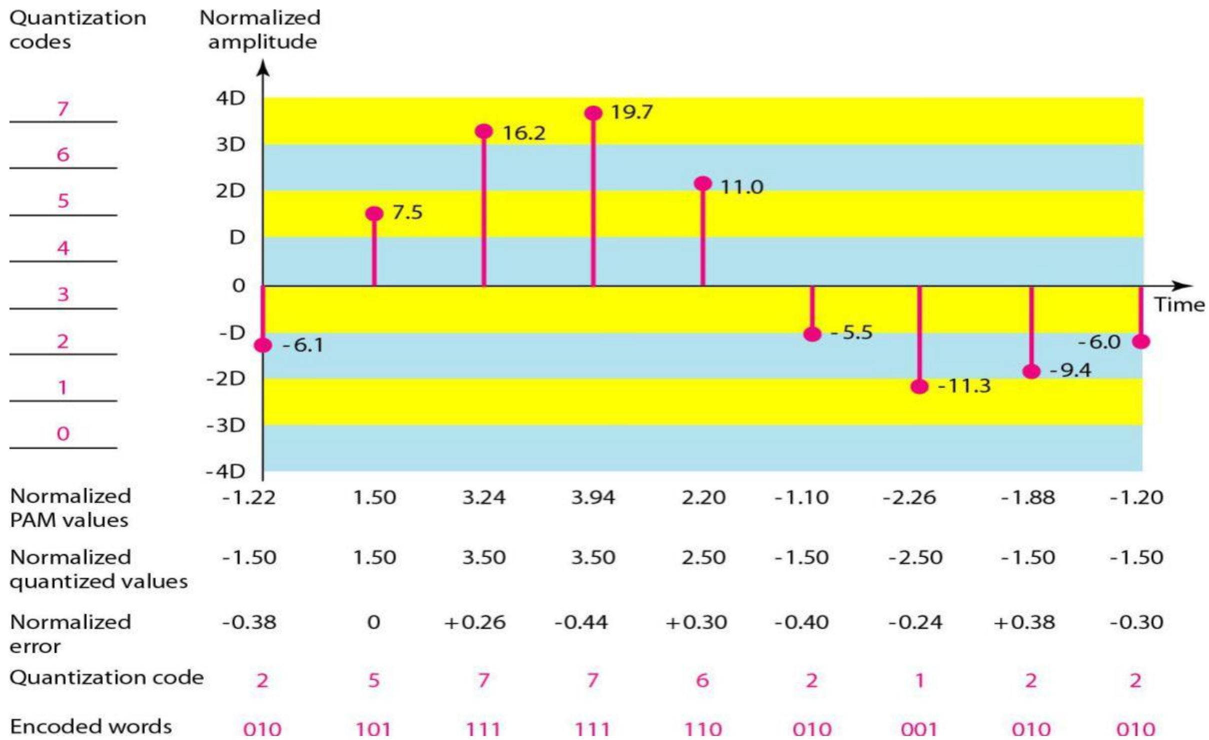
= (Vmax-Vmin)/L

3. We assign quantized values of 0 to *L* - 1 to the midpoint of each zone.

Anjan kumar K N, Devaraju B M, Medha Gourayya, Dept. of CSE, RNSIT

4. We approximate the value of the sample amplitude to the quantized values.

As a simple example, assume that we have a sampled signal and the sample amplitudes are between -20 and +20 V. We decide to have eight levels (L = 8). This means that =5 V.



Quantization Levels: In the previous example, we showed eight quantization levels. The choice of L, the number of levels, depends on the range of the amplitudes of the analog signal and how accurately we need to recover the signal. If the amplitude of a signal fluctuates between two values only, we need only two levels; if the signal, like voice, has many amplitude values, we need more quantization levels. In audio digitizing, L is normally chosen to be 256; in video it is normally thousands. Choosing lower values of L increases the quantization error if there is a lot of fluctuation in the signal.

Quantization Error: One important issue is the error created in the quantization process. Quantization is an approximation process. The input values to the quantizer are the real values; the output values are the approximated values. The output values are chosen to be the middle value in the zone. If the input value is also at the middle of the zone, there is no quantization error; otherwise, there is an error.

The quantization error changes the signal-to-noise ratio of the signal, which in turn reduces the upper limit capacity according to Shannon.

It can be proven that the contribution of the quantization error to the SNR_{dB} of the signal depends on the number of quantization levels L, or the bits per sample nb' as shown in the following formula:

$$SNR_{dB} = 6.02 n_b + 1.76 \text{ dB}$$

Encoding:

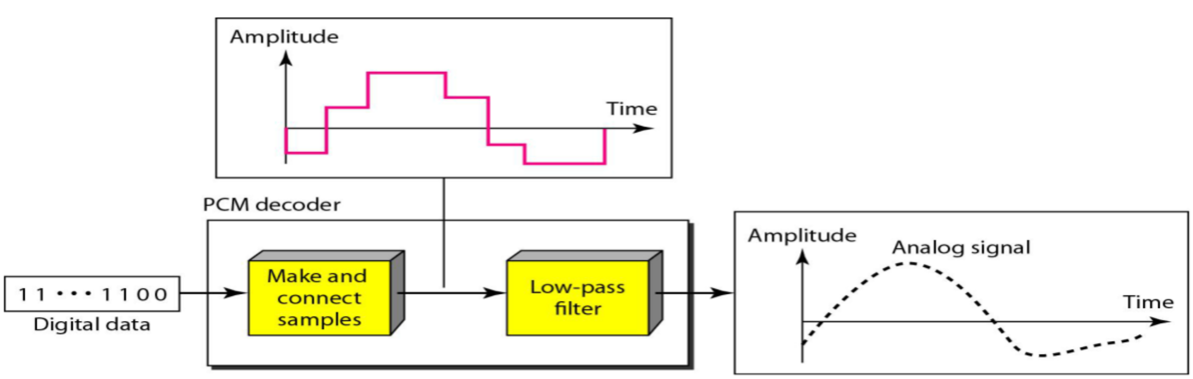
The last step in PCM is encoding. After each sample is quantized and the number of bits per sample is decided, each sample can be changed to an n_b-bit code word.

A quantization code of 2 is encoded as 010; 5 is encoded as 101; and so on. Note that the number of bits for each sample is determined from the number of quantization levels. If the number of quantization levels is L, the number of bits is $n_b = \log_2 L$. In our example L is 8 and n_b is therefore 3. The bit rate can be found from the formula

$$\text{Bit rate} = \text{sampling rate} \times \text{number of bits per sample} = f_s \times n_b$$

Original SIgnal Recovery

- It requires PCM decoder.Decoder first usescircuitry to code words into pulse that holds the amplitude untill the next pulse.
- After the stair case signal is completed,it is passed through a low pass filterto smooth the stair case signal into analog signal.The below figure shows the components of PCM decoder.

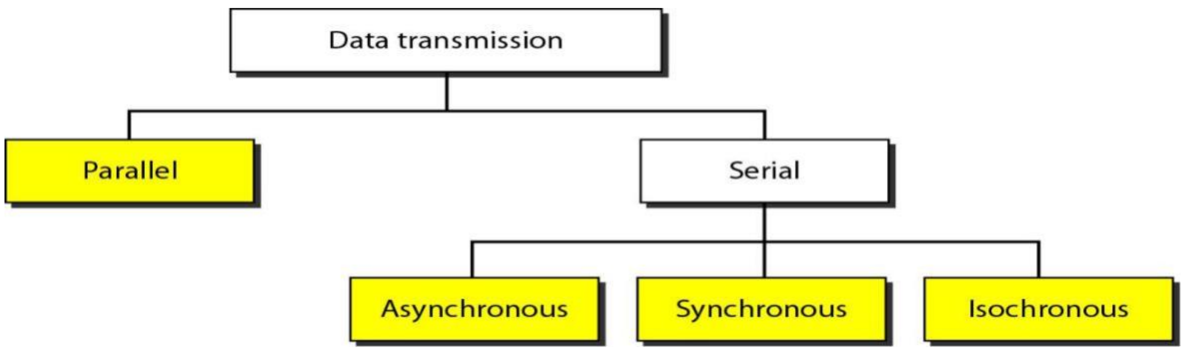


Transmission Modes

The transmission of binary data across a link can be accomplished in either parallel or serial mode.

In parallel mode, multiple bits are sent with each clock tick.

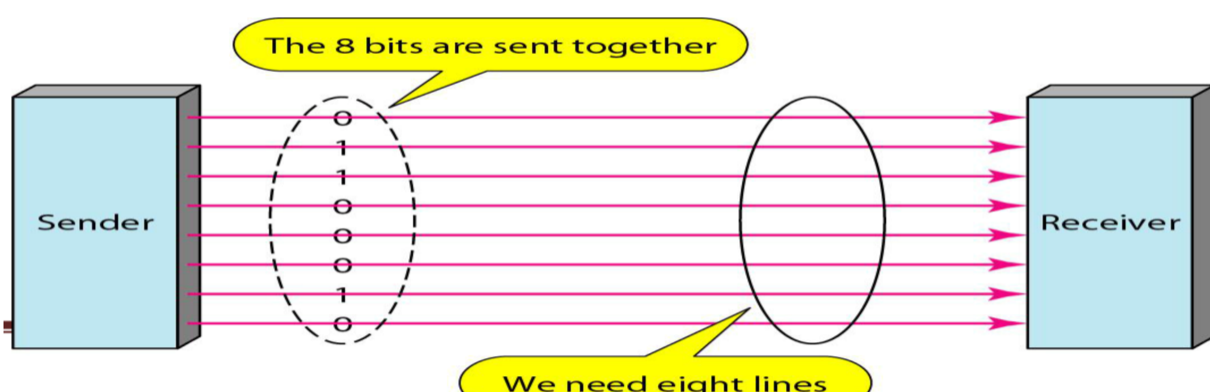
In serial mode, 1 bit is sent with each clock tick. While there is only one way to send parallel data, there are three subclasses of serial transmission: asynchronous, synchronous, and isochronous.



Parallel Transmission

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

The mechanism for parallel transmission is a conceptually simple one: Use *n* wires to send *n* bits at one time. That way each bit has its own wire, and all *n* bits of one group can be transmitted with each clock tick from one device to another. Figure shows how parallel transmission works for *n* =8. Typically, the eight wires are bundled in a cable with a connector at each end.



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

But there is a significant disadvantage: cost. Parallel transmission requires n communication lines (wires in the example) just to transmit the data stream. Because this is expensive, parallel transmission is usually limited to short distances.

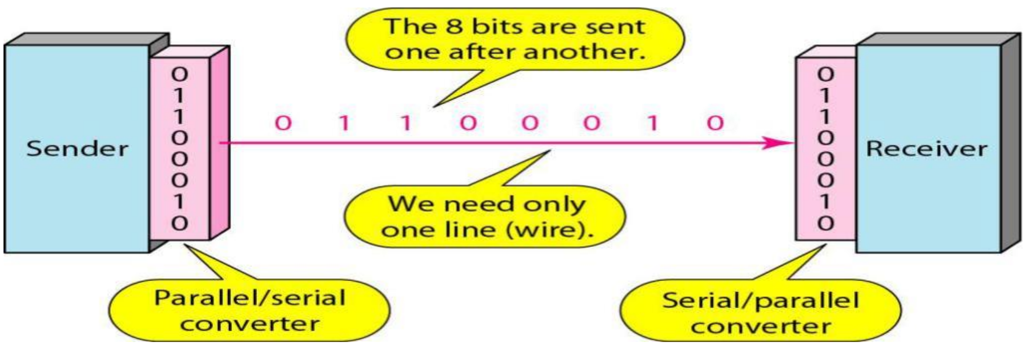
Serial Transmission

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.



Asynchronous Transmission

Asynchronous transmission is so named because the timing of a signal is unimportant.

Instead, information is received and translated by agreed upon patterns.

As long as those patterns are followed, the receiving device can retrieve the information without regard to the rhythm in which it is sent.

Patterns are based on grouping the bit stream into bytes. Each group, usually 8 bits, is sent along the link as a unit. The sending system handles each group independently, relaying it to the link whenever ready, without regard to a timer.

Without synchronization, the receiver cannot use timing to predict when the next group will arrive.

Anjan kumar K N, Devaraju B M, Medha Gourayya, Dept. of CSE, RNSIT

To alert the receiver to the arrival of a new group, therefore, an extra bit is added to the beginning of each byte. This bit, usually a 0, is called the start bit.

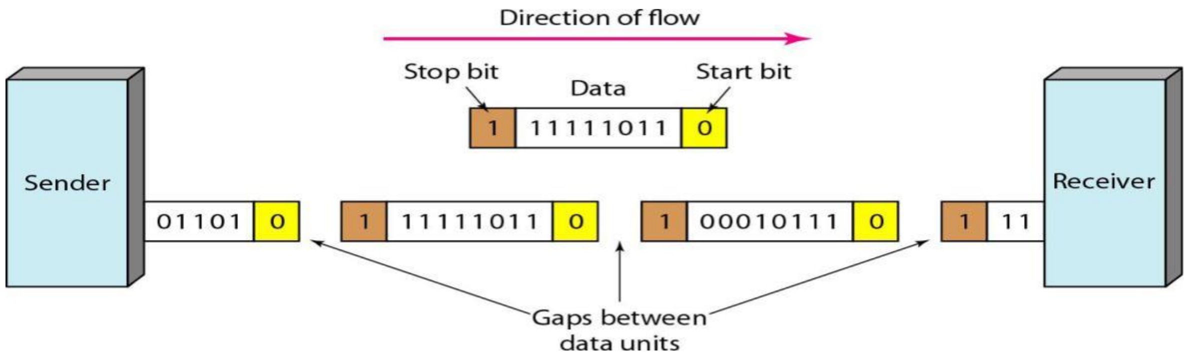
To let the receiver know that the byte is finished, 1 or more additional bits are appended to the end of the byte. These bits, usually 1 s, are called stop bits.

By this method, each byte is increased in size to at least 10 bits, of which 8 bits is information and 2 bits or more are signals to the receiver.

In addition, the transmission of each byte may then be followed by a gap of varying duration. This gap can be represented either by an idle channel or by a stream of additional stop bits.

This mechanism is called *asynchronous* because, at the byte level, the sender and receiver do not have to be synchronized. But within each byte, the receiver must still be synchronized with the incoming bit stream.

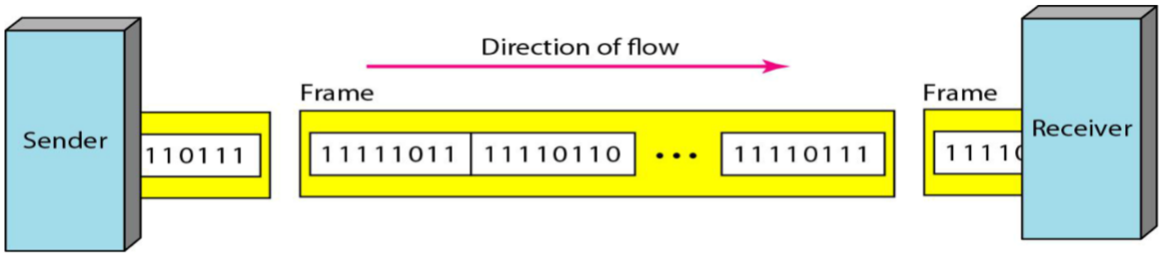
The addition of stop and start bits and the insertion of gaps into the bit stream make asynchronous transmission slower than forms of transmission that can operate without the addition of control information. But it is cheap and effective.



Synchronous Transmission

In synchronous transmission, the bit stream is combined into longer "frames," which may contain multiple bytes.

Each byte, however, is introduced onto the transmission link without a gap between it and the next one. It is left to the receiver to separate the bit stream into bytes for decoding purposes.



Anjan kumar K N, Devaraju B M, Medha Gourayya, Dept. of CSE, RNSIT

The advantage of synchronous transmission is speed. With no extra bits or gaps to introduce at the sending end and remove at the receiving end, and, by extension, with fewer bits to move across the link, synchronous transmission is faster than asynchronous transmission. Although there is no gap between characters in synchronous serial transmission, there may be uneven gaps between frames.

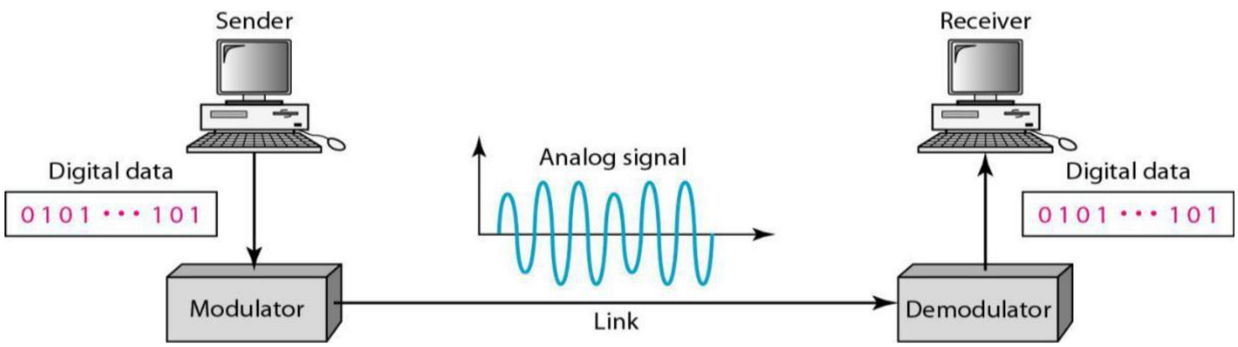
Isochronous

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 frames, there should be no delays between frames. For this type of application, synchronization between characters is not enough; the entire stream of bits must be synchronized.

The isochronous transmission guarantees that the data arrive at a fixed rate.

Digital-To-Analog Conversion

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

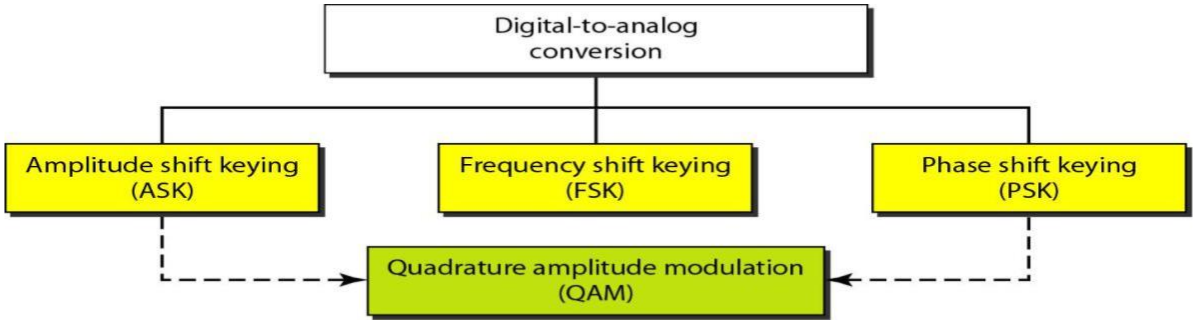


Sine wave is defined by three characteristics: amplitude, frequency, and phase.

When we vary anyone of these characteristics, we create a different version of that wave. So, by changing one characteristic of a simple electric signal, we can use it to represent digital data.

Any of the three characteristics can be altered in this way, giving us at least three mechanisms for modulating digital data into an analog signal: amplitude shift keying (ASK), frequency shift keying (FSK), and phase shift keying (PSK).

In addition, there is a fourth (and better) mechanism that combines changing both the amplitude and phase, called quadrature amplitude modulation (QAM). QAM is the most efficient of these options and is the mechanism commonly used today.



Aspects of Digital-to-Analog Conversion

Data Element versus Signal Element: Data element is the smallest piece of information to be exchanged, that is the bit. Signal element is the smallest unit of a signal that is constant.

Data Rate Versus Signal Rate: The relationship between them is

$$S = N \times 1/r \text{ baud}$$

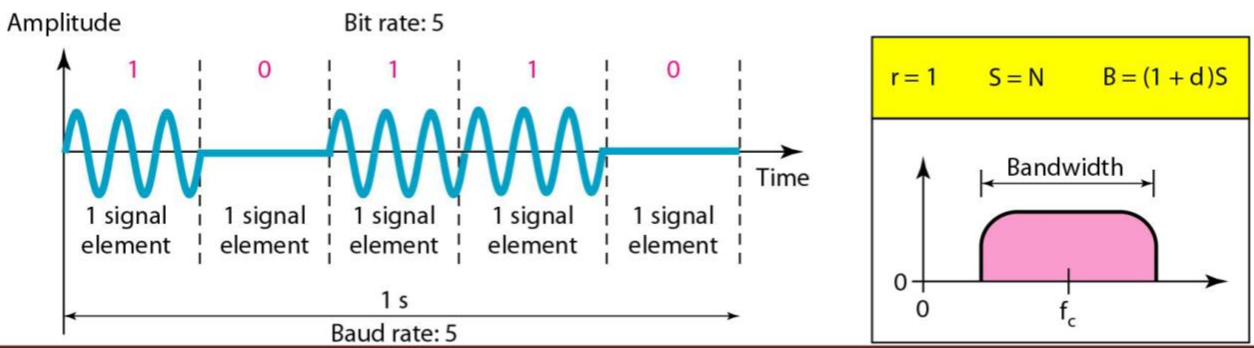
Where, N is the data rate (bps), r is the number of data elements carried in one signal element. $r = \log_2 L$, where L is the type of signal element.

Amplitude Shift Keying

In amplitude shift keying, the amplitude of the carrier signal is varied to create signal elements. Both frequency and phase remain constant while the amplitude changes.

Binary ASK (BASK)

Although we can have several levels (kinds) of signal elements, each with a different amplitude, ASK is normally implemented using only two levels. This is referred to as binary amplitude shift keying or on-off keying (OOK). The peak amplitude of one signal level is 0; the other is the same as the amplitude of the carrier frequency.



Anjan kumar K N, Devaraju B M, Medha Gourayya, Dept. of CSE, RNSIT

Bandwidth of ASK is

$$B = (1 + d) \times S$$

Where S is the signal rate and the B is the bandwidth. The d depends on the modulation and filtering process. The value of d is between 0 and 1.

Multilevel ASK

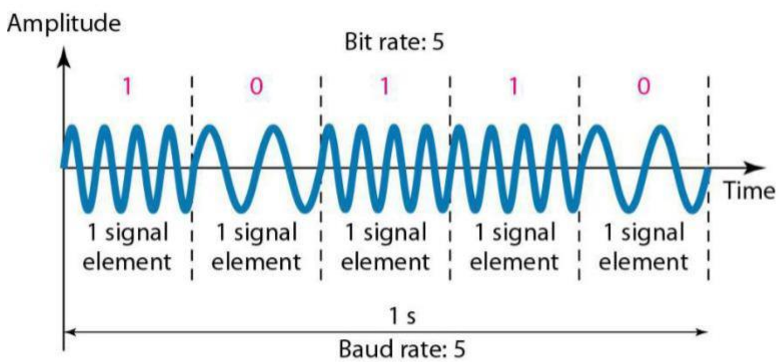
We can have multilevel ASK in which there are more than two levels. We can use 4,8, 16, or more different amplitudes for the signal and modulate the data using 2, 3, 4, or more bits at a time.

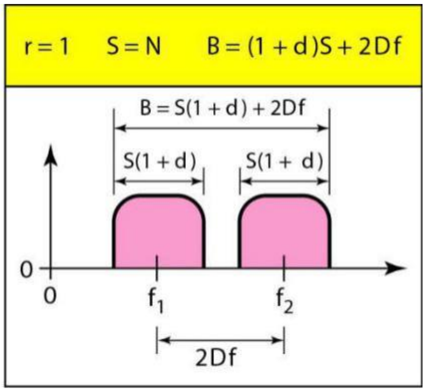
Frequency Shift Keying

In frequency shift keying, the frequency of the carrier signal is varied to represent data. The frequency of the modulated signal is constant for the duration of one signal element, but changes for the next signal element if the data element changes. Both peak amplitude and phase remain constant for all signal elements.

Binary FSK (BFSK)

One way to think about binary FSK (or BFSK) is to consider two carrier frequencies. In Figure, we have selected two carrier frequencies,f} and12. We use the first carrier if the data element is 0; we use the second if the data element is 1. However, note that this is an unrealistic example used only for demonstration purposes. Normally the carrier frequencies are very high, and the difference between them is very small.





Anjan kumar K N, Devaraju B M, Medha Gourayya, Dept. of CSE, RNSIT

Bandwidth for BFSK:

We can think of FSK as two ASK signals, each with its own carrier frequency (*f1 or f2*) If the difference between the two frequencies is $2\Delta f$, then the required bandwidth is

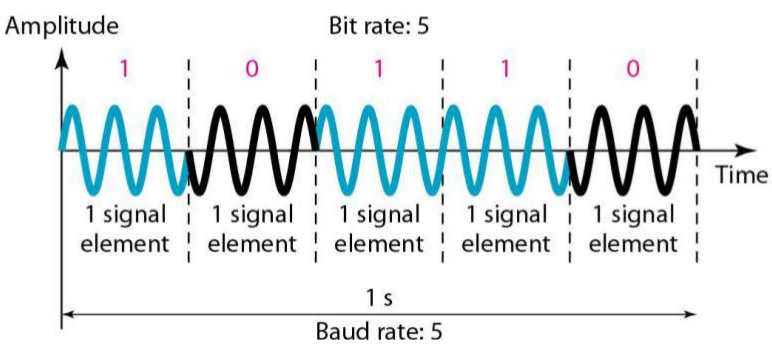
$$B=(1+d) \times S+2\Delta f$$

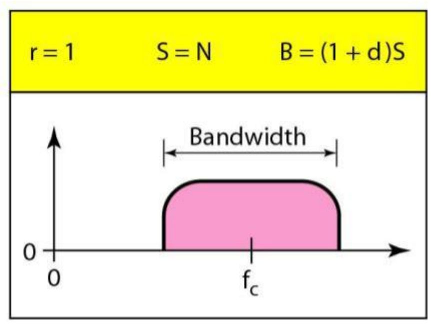
Phase Shift Keying

In phase shift keying, the phase of the carrier is varied to represent two or more different signal elements. Both peak amplitude and frequency remain constant as the phase changes.

Binary PSK (BPSK)

The simplest PSK is binary PSK, in which we have only two signal elements, one with a phase of 0°, and the other with a phase of 180°. Below Figure gives a conceptual view of PSK. Binary PSK is as simple as binary ASK with one big advantage-it is less susceptible to noise. In ASK, the criterion for bit detection is the amplitude of the signal; in PSK, it is the phase. Noise can change the amplitude easier than it can change the phase. In other words, PSK is less susceptible to noise than ASK. PSK is superior to FSK because we do not need two carrier signals.





Bandwidth:

$$B = (1 + d) \times S$$

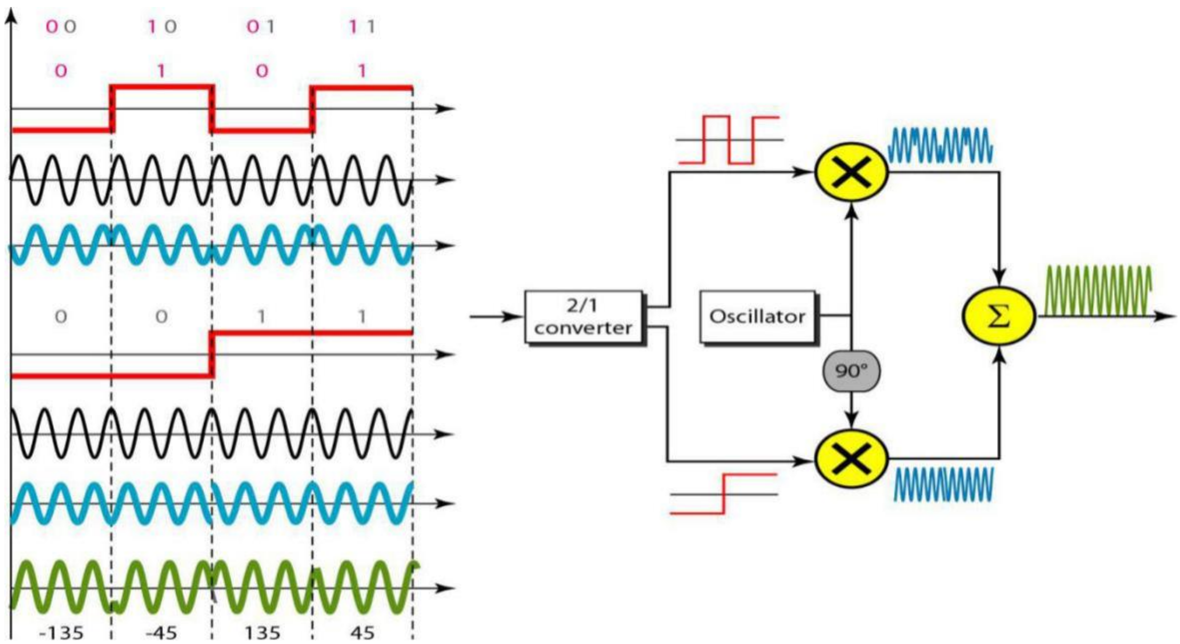
Where S is the signal rate and the *B* is the bandwidth. The d depends on the modulation and filtering process. The value of d is between 0 and 1.

Quadrature PSK (QPSK):

The simplicity of BPSK enticed designers to use 2 bits at a time in each signal element, thereby decreasing the baud rate and eventually the required bandwidth. The scheme is called quadrature PSK or QPSK because it uses two separate BPSK modulations; one is in-phase, the other

Anjan kumar K N, Devaraju B M, Medha Gourayya, Dept. of CSE, RNSIT

quadrature (out-of-phase). The incoming bits are first passed through a serial-to-parallel conversion that sends one bit to one modulator and the next bit to the other modulator. If the duration of each bit in the incoming signal is T, the duration of each bit sent to the corresponding BPSK signal is 2T. This means that the bit to each BPSK signal has one-half the frequency of the original signal.



Quadrature Amplitude Modulation

Quadrature amplitude modulation is a combination of ASK and PSK. The idea of using two carriers, one in-phase and the other quadrature, with different amplitude levels for each carrier is the concept behind quadrature amplitude modulation (QAM).

Anjan kumar K N, Devaraju B M, Medha Gourayya, Dept. of CSE, RNSIT

Module-3

BANDWIDTH UTILIZATION

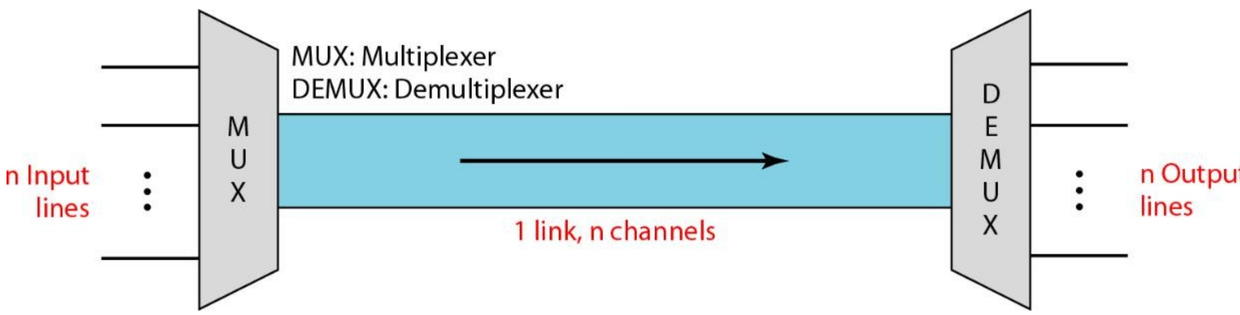
Bandwidth utilization is the wise use of available bandwidth to achieve specific goals. Efficiency can be achieved by multiplexing; privacy and anti-jamming can be achieved by spreading.

Multiplexing

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

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

Below figure shows the basic format of a multiplexed system.

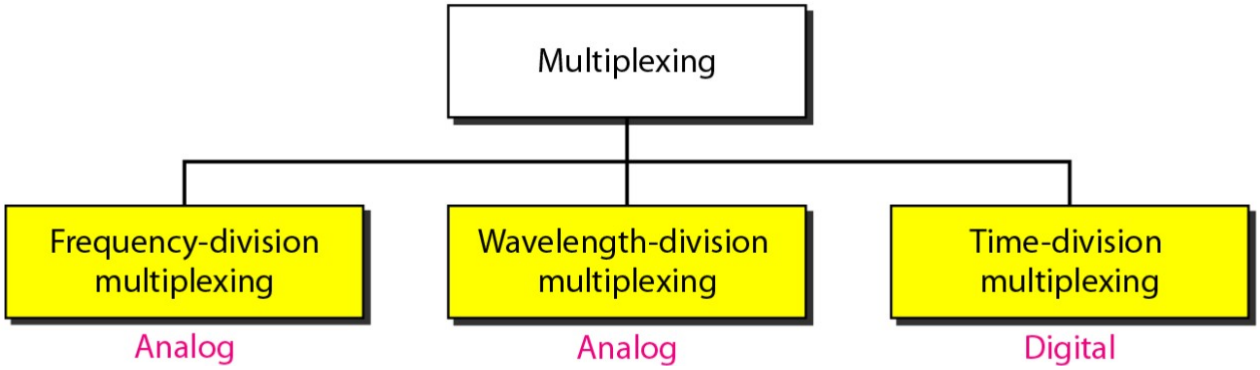


The lines on the left direct their transmission streams to a multiplexer (MUX), which combines them into a single stream (many-to one).

At the receiving end, that stream is fed into a demultiplexer (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, wavelength-division multiplexing, and time-division multiplexing.



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.

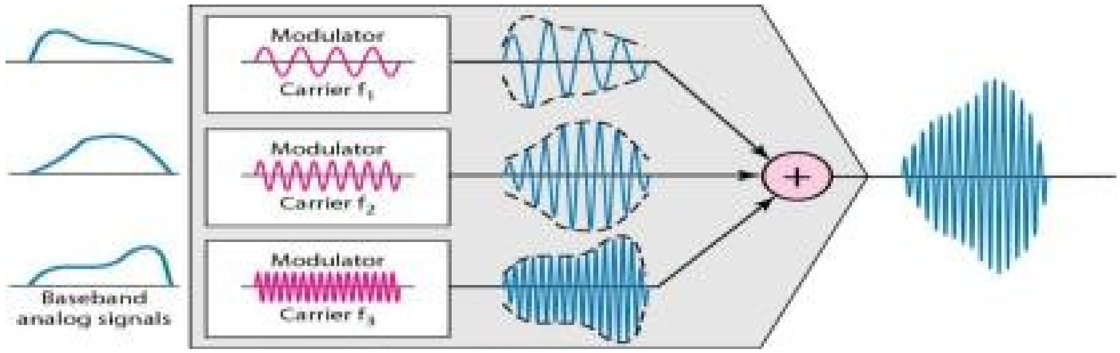
These modulated signals are then combined into a single composite signal that can be transported by the link.

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.

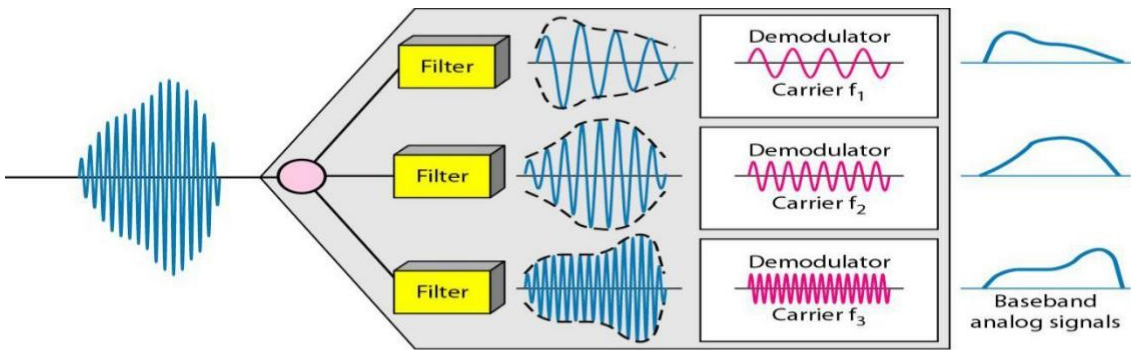
Multiplexing Process

Each source generates a signal of a similar frequency range. Inside the multiplexer, these similar signals modulates 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 demultiplexer 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.

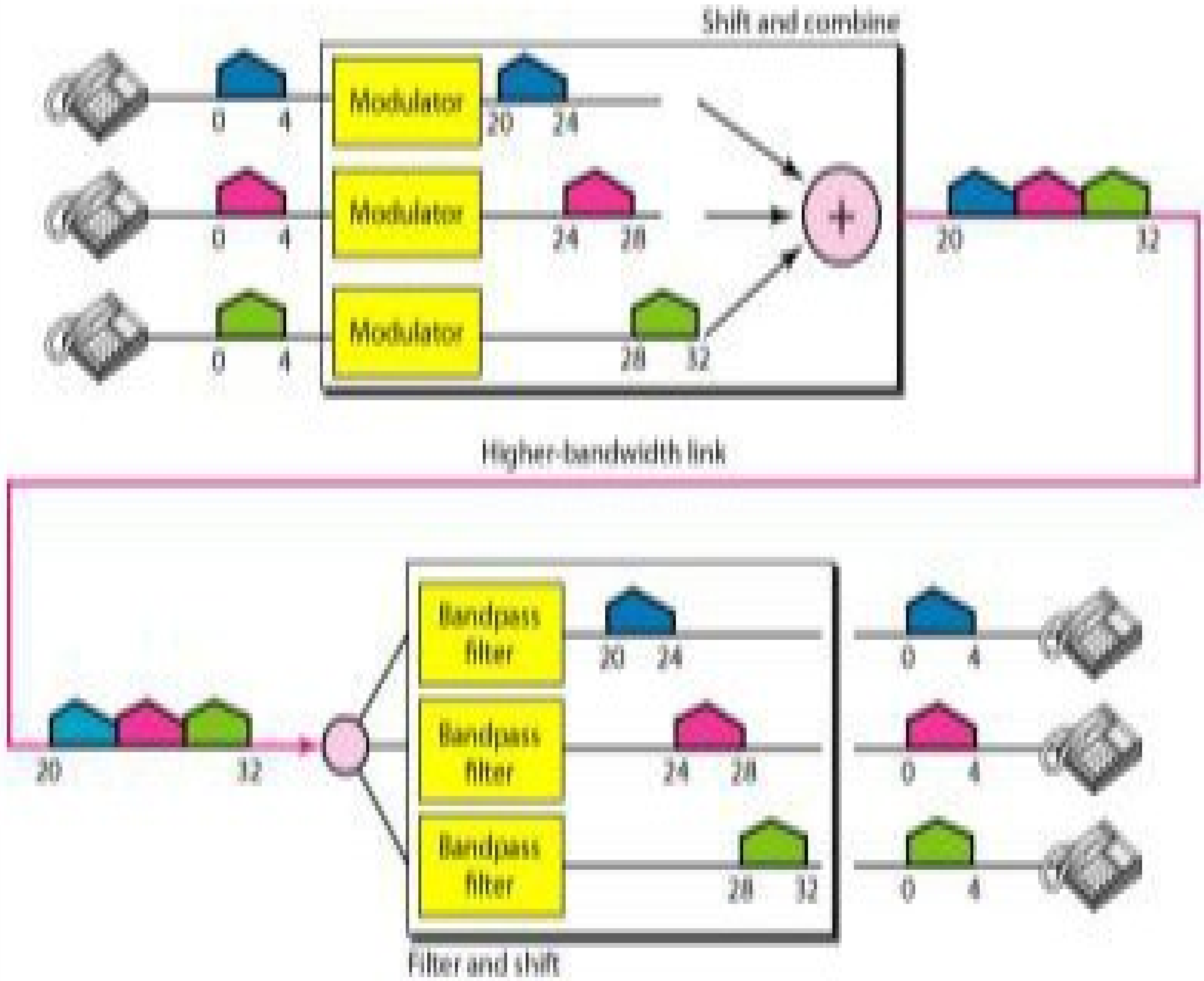


Examples:

1. Assume that a voice channel occupies a bandwidth of 4 kHz. We need to combine three voice channels into a link with a bandwidth of 12 kHz, from 20 to 32 kHz. Show the configuration, using the frequency domain. Assume there are no guard bands.

Solution

We shift (modulate) each of the three voice channels to a different bandwidth, as shown in Figure

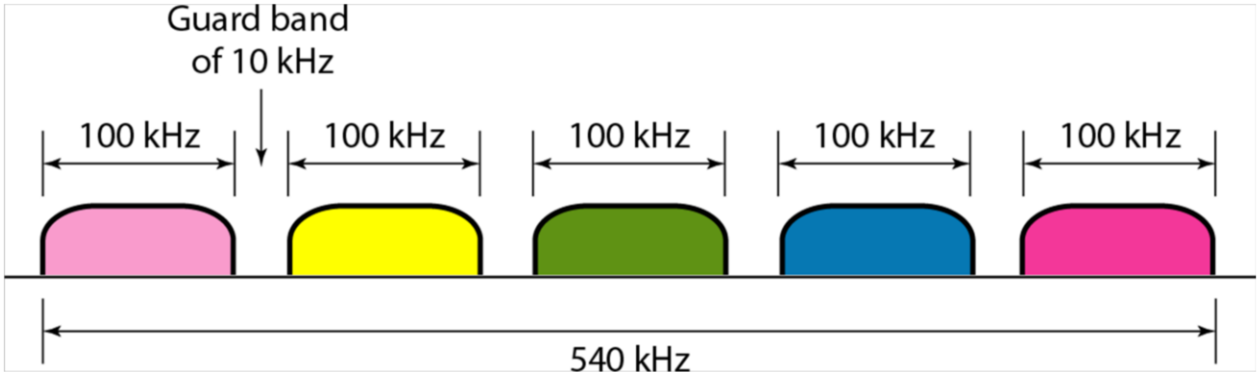


We use the 20- to 24-kHz bandwidth for the first channel, the 24- to 28-kHz bandwidth for the second channel, and the 28- to 32-kHz bandwidth for the third one. Then we combine them. At the receiver, each channel receives the entire signal, using a filter to separate out its own signal. The first channel uses a filter that passes frequencies between 20 and 24 kHz and filters out (discards) any other frequencies. The second channel uses a filter that passes frequencies between 24 and 28 kHz, and the third channel uses a filter that passes frequencies between 28 and 32 kHz. Each channel then shifts the frequency to start from zero.

2. Five channels, each with a 100-kHz bandwidth, are to be multiplexed together. What is the minimum bandwidth of the link if there is a need for a guard band of 10kHz between the channels to prevent interference?

Solution:

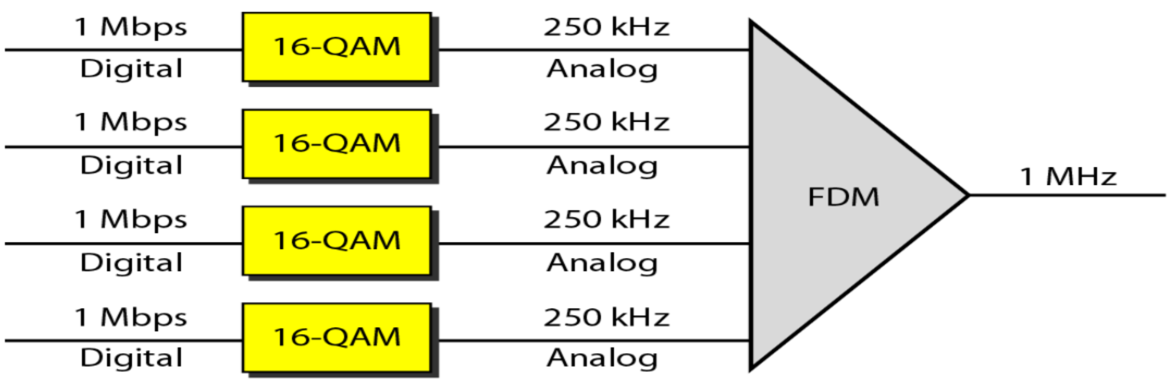
For five channels, we need at least four guard bands. This means that the required bandwidth is at least 5 x 100 + 4 x 10 =540 kHz, as shown in Figure



3) Four data channels (digital), each transmitting at I Mbps, use a satellite channel of I MHz. Design an appropriate configuration, using FDM.

Solution

The satellite channel is analog. We divide it into four channels, each channel having a 250-kHz bandwidth. Each digital channel of 1 Mbps is modulated such that each 4 bits is modulated to 1 Hz. One solution is 16-QAM modulation. Figure shows one possible configuration.



Application:

The Analog Carrier System

To maximize the efficiency of infrastructure, telephone companies have traditionally multiplexed signals from lower-bandwidth lines onto higher-bandwidth lines. In this way, many switched or leased lines can be combined into fewer but bigger channels. For analog lines, FDM is used.

One of these hierarchical systems used by AT&T is made up of groups, supergroups, master groups, and jumbo groups.

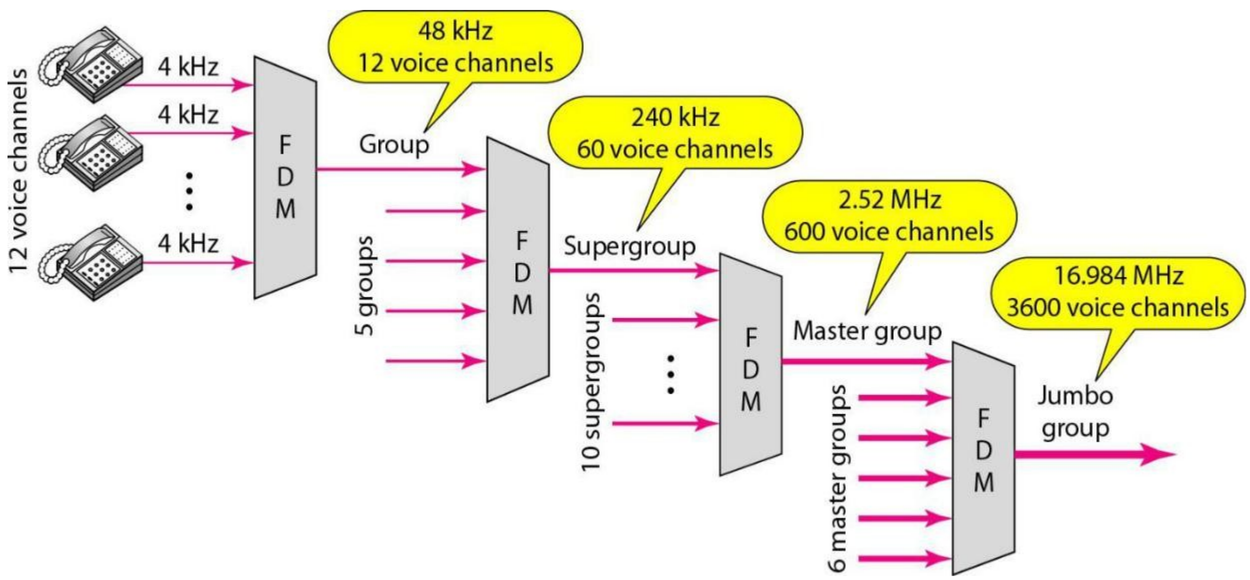
In this analog hierarchy, 12 voice channels are multiplexed onto a higher-bandwidth line to create a group. A group has 48 kHz of bandwidth and supports 12 voice channels.

At the next level, up to five groups can be multiplexed to create a composite signal called a supergroup. A supergroup has a bandwidth of 240 kHz and supports up to 60 voice channels. Supergroups can be made up of either five groups or 60 independent voice channels.

At the next level, 10 supergroups are multiplexed to create a master group. A master group must have 2.40 MHz of bandwidth, but the need for guard bands between the supergroups increases the necessary bandwidth to 2.52 MHz. Master groups support up to 600 voice channels.

Finally, six master groups can be combined into a jumbo group. A jumbo group must have

15.12 MHz (6 x 2.52 MHz) but is augmented to 16.984 MHz to allow for guard bands between the master groups.



➢ ***Other Applications of FDM***

AM and FM radio broadcasting.

Radio uses the air as the transmission medium. A special band from 530 to 1700 kHz is assigned to AM radio. All radio stations need to share this band. Each AM station needs 10 kHz of bandwidth. Each station uses a different carrier frequency, which means it is shifting its signal and multiplexing. The signal that goes to the air is a combination of signals. A receiver receives all these signals, but filters (by tuning) only the one which is desired. Without multiplexing, only one AM station could broadcast to the common link, the air.

However, we need to know that there is physical multiplexer or demultiplexer here. Multiplexing is done at the data link layer.

The situation is similar in FM broadcasting. However, FM has a wider band of 88 to 108 MHz because each station needs a bandwidth of 200 kHz.

➢ ***Television broadcasting.***

Each TV channel has its own bandwidth of 6 MHz.

➢ ***The first generation of cellular telephones.***

Each user is assigned two 30-kHz channels, one for sending voice and the other for receiving. The voice signal, which has a bandwidth of 3 kHz (from 300 to 3300 Hz), is modulated by using FM.

Example

The Advanced Mobile Phone System (AMPS) uses two bands. The first band of 824 to 849 MHz is used for sending, and 869 to 894 MHz is used for receiving. Each user has a bandwidth of 30 kHz in each direction. The 3-kHz voice is modulated using FM, creating 30 kHz of modulated signal. How many people can use their cellular phones simultaneously?

Solution:

Each band is 25 MHz. If we divide 25 MHz by 30 kHz, we get 833.33. In reality, the band is divided into 832 channels. Of these, 42 channels are used for control, which means only 790 channels are available for cellular phone users.

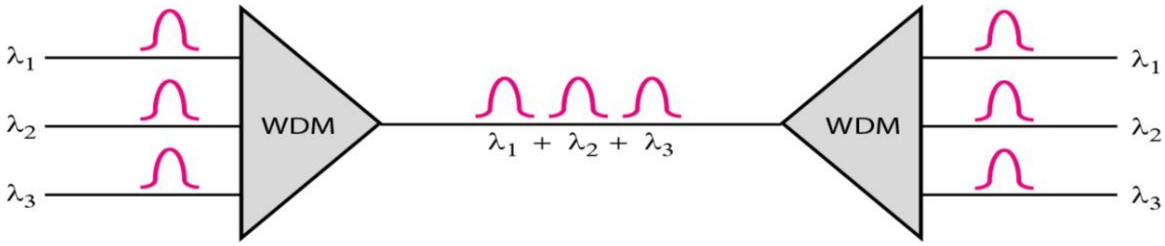
Wavelength-Division Multiplexing

Wavelength-division multiplexing (WDM) is designed to use the high-data-rate capability of fiber-optic cable.

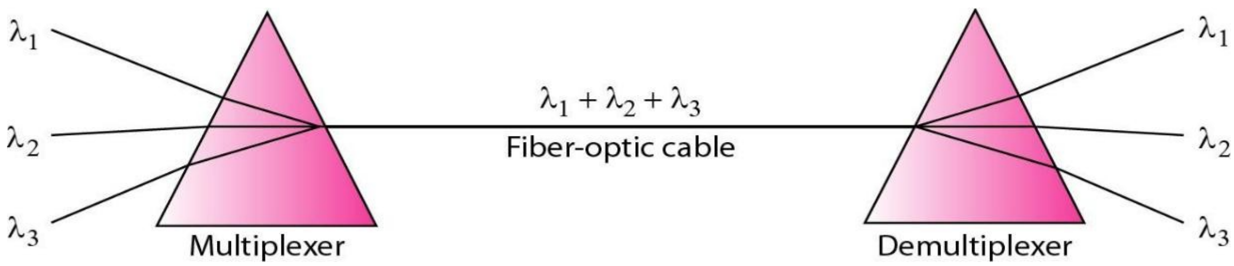
The optical fiber data rate is higher than the data rate of metallic transmission cable. Using a fiber-optic cable for one single line wastes the available bandwidth. Multiplexing allows us to combine several lines into one.

WDM is conceptually the same as FDM, except that the multiplexing and demultiplexing involve optical signals transmitted through fiber-optic channels

Very narrow bands of light from different sources are combined to make a wider band of light. At the receiver, the signals are separated by the demultiplexer.



The combining and splitting of light sources are easily handled by a prism. Prism bends a beam of light based on the angle of incidence and the frequency. Using this technique, a multiplexer can be made to combine several input beams of light, each containing a narrow band of frequencies, into one output beam of a wider band of frequencies. A demultiplexer can also be made to reverse the process.



One application of WDM is the SONET network in which multiple optical fiber lines are multiplexed and demultiplexed.

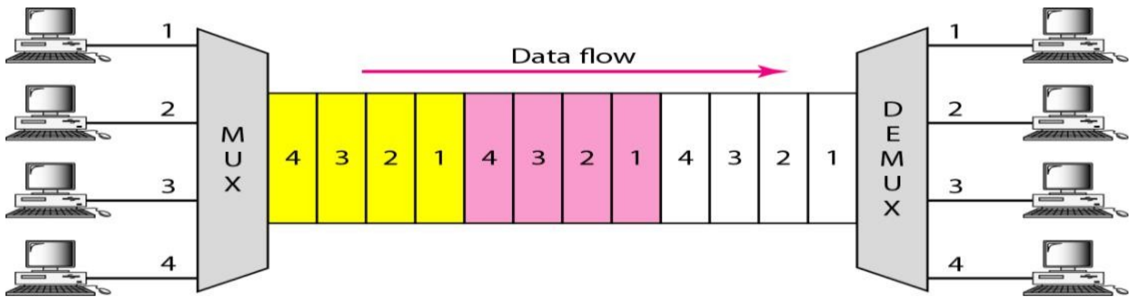
A new method, called dense WDM (DWDM), can multiplex a very large number of channels by spacing channels very close to one another. It achieves even greater efficiency.

Synchronous 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.

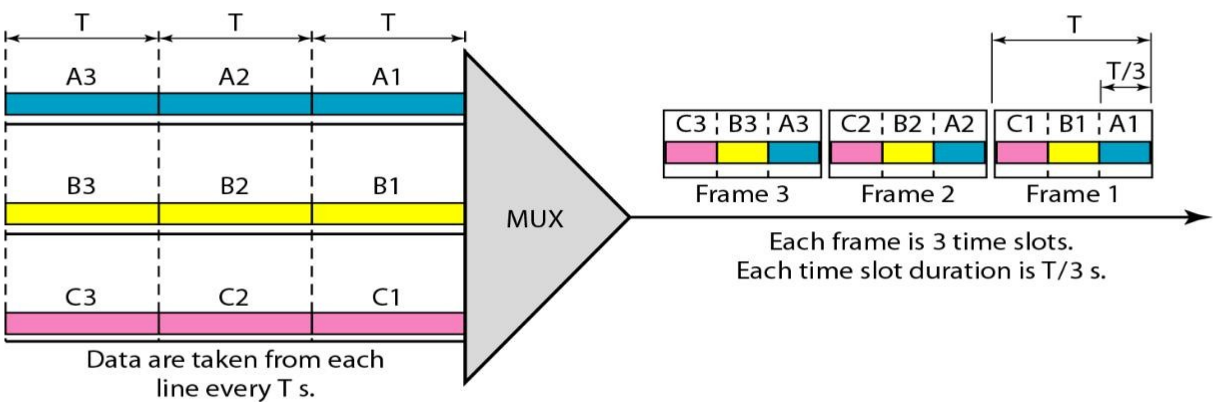
TDM is a digital multiplexing technique for combining several low-rate channels into one high-rate one.



TDM is divided into two different schemes: synchronous and statistical

➢ ***Time Slots and Frames***

In synchronous TDM, the data flow of each input connection is divided into units, where each input occupies one input time slot. A unit can be 1 bit, one character, or one block of data. Each input unit becomes one output unit and occupies one output time slot. However, the duration of an output time slot is *n* times shorter than the duration of an input time slot. If an input time slot is *T* s, the output time slot is *Tn* s, where *n* is the number of connections.



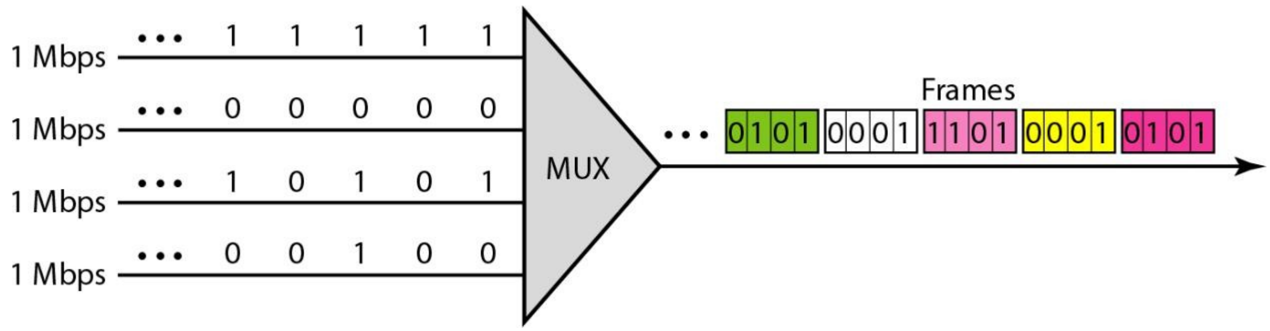
In synchronous TDM, a round of data units from each input connection is collected into a frame. If we have n connections, a frame is divided into n time slots and one slot is allocated for each unit, one for each input line. If the duration of the input unit is T, the duration of each slot is Tin and the duration of each frame is T.

In synchronous TDM, the data rate of the link is n times faster, and the unit duration is n times shorter.

Time slots are grouped into frames. A frame consists of one complete cycle of time slots, with one slot dedicated to each sending device. In a system with *n* input lines, each frame has *n* slots, with each slot allocated to carrying data from a specific input line.

Example:

Figure shows synchronous TOM with a data stream for each input and one data stream for the output. The unit of data is 1 bit. Find (a) the input bit duration, (b) the output bit duration, (c) the output bit rate, and (d) the output frame rate.

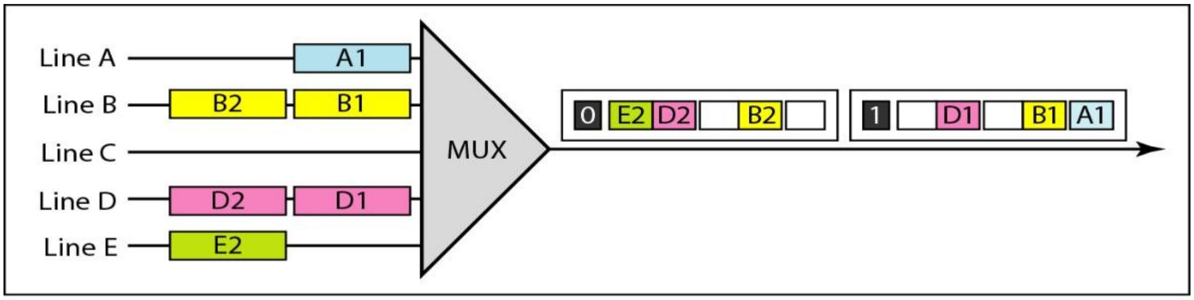


Solution

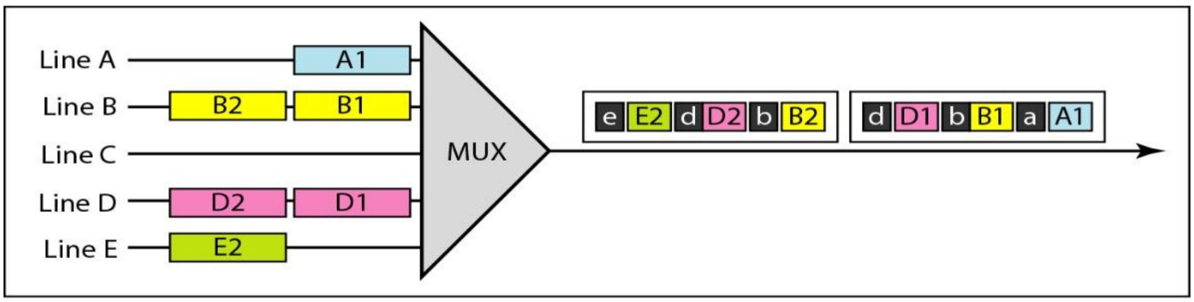
- The input bit duration is the inverse of the bit rate: 1/1 Mbps = 1 µs.
- The output bit duration is one-fourth of the input bit duration, or 1/4µs.
- The output bit rate is the inverse of the output bit duration or 1/4 µs, or 4 Mbps. This can also be deduced from the fact that the output rate is 4 times as fast as any input rate; so the output rate =4 x 1 Mbps =4 Mbps.
- The frame rate is always the same as any input rate. So the frame rate is 1,000,000 frames per second. Because we are sending 4 bits in each frame, we can verify the result of the previous question by multiplying the frame rate by the number of bits per frame.

Statistical Time-Division Multiplexing

In statistical time-division multiplexing, slots are dynamically allocated to improve bandwidth efficiency. Only when an input line has a slot's worth of data to send is it given a slot in the output frame. In statistical multiplexing, the number of slots in each frame is less than the number of input lines. The multiplexer checks each input line in round robin fashion; it allocates a slot for an input line if the line has data to send; otherwise, it skips the line and checks the next line.



a. Synchronous TDM



b. Statistical TDM

Spread Spectrum

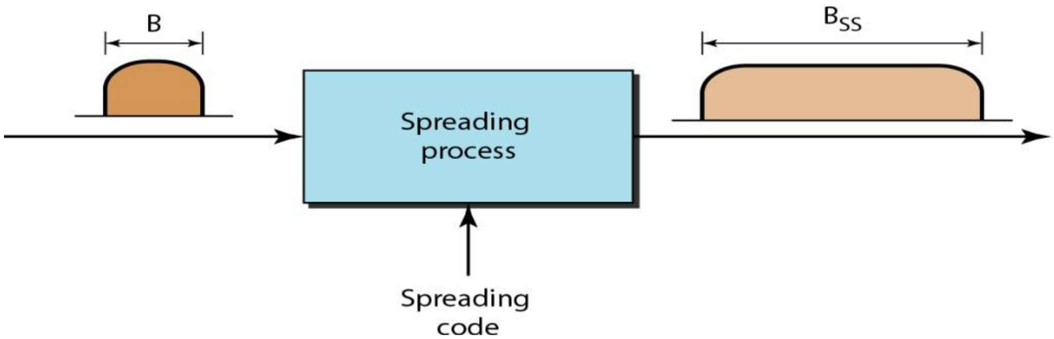
Multiplexing combines signals from several sources to achieve bandwidth efficiency; the available bandwidth of a link is divided between the sources.

In spread spectrum, we also combine signals from different sources to fit into a larger bandwidth.

Spread spectrum is designed to be used in wireless applications (LANs and WANs).

In wireless applications, all stations use air (or a vacuum) as the medium for communication. Stations must be able to share this medium without interception by an eavesdropper and without being subject to jamming from a malicious intruder (in military operations, for example).

If the required bandwidth for each station is B, spread spectrum expands it to Bss such that Bss » B. The expanded bandwidth allows the source to wrap its message in a protective envelope for a more secure transmission.



Spread spectrum achieves its goals through two principles:

1. The bandwidth allocated to each station needs to be, by far, larger than what is needed. This allows redundancy.
2. The expanding of the original bandwidth *B* to the bandwidth Bss must be done by a process that is independent of the original signal. In other words, the spreading process occurs after the signal is created by the source.

After the signal is created by the source, the spreading process uses a spreading code and spreads the bandwidth. The figure shows the original bandwidth B and the spreaded bandwidth Bss. The spreading code is a series of numbers that look random, but are actually a pattern.

There are two techniques to spread the bandwidth:

- Frequency hopping spread spectrum (FHSS)
- Direct sequence spread spectrum (DSSS)

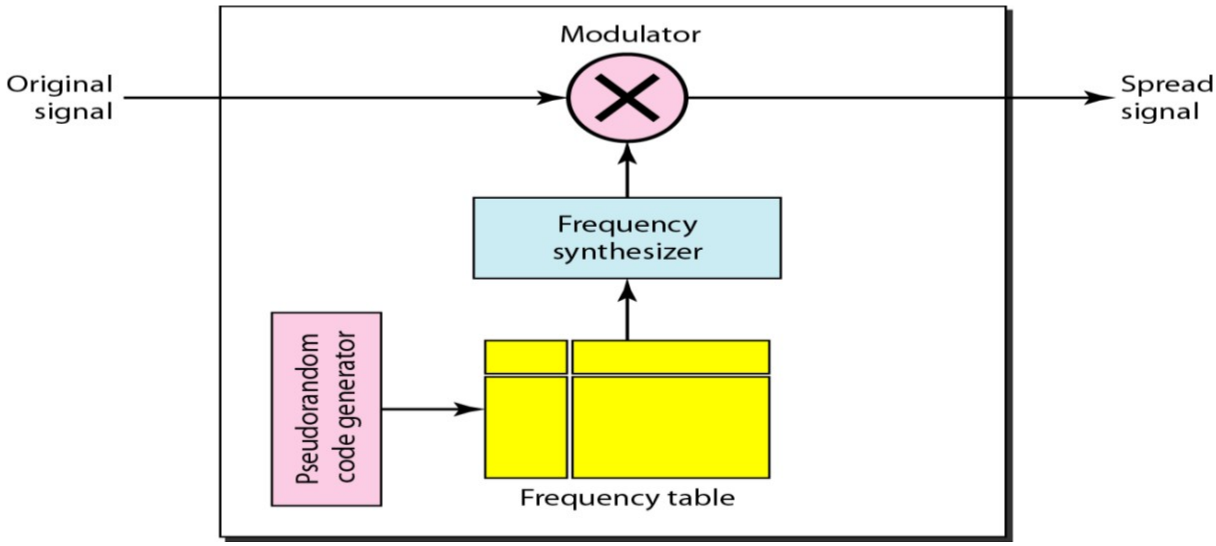
Frequency Hopping Spread Spectrum (FHSS)

The frequency hopping spread spectrum (FHSS) technique uses *M* different carrier frequencies that are modulated by the source signal.

At one moment, the signal modulates one carrier frequency; at the next moment, the signal modulates another carrier frequency. Although the modulation is done using one carrier frequency at a time, *M* frequencies are used in the long run.

The bandwidth occupied by a source after spreading is *BpHSS »B.*

The general layout for FHSS is shown below:

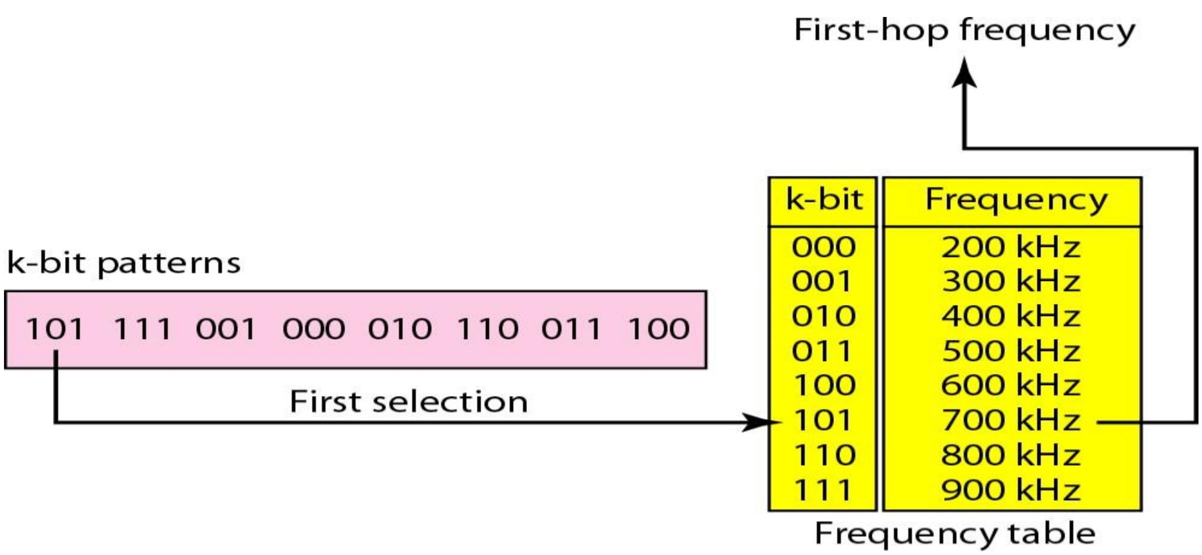


A pseudorandom code generator, called pseudorandom noise (PN), creates a k-bit pattern for every hopping period *Th.*

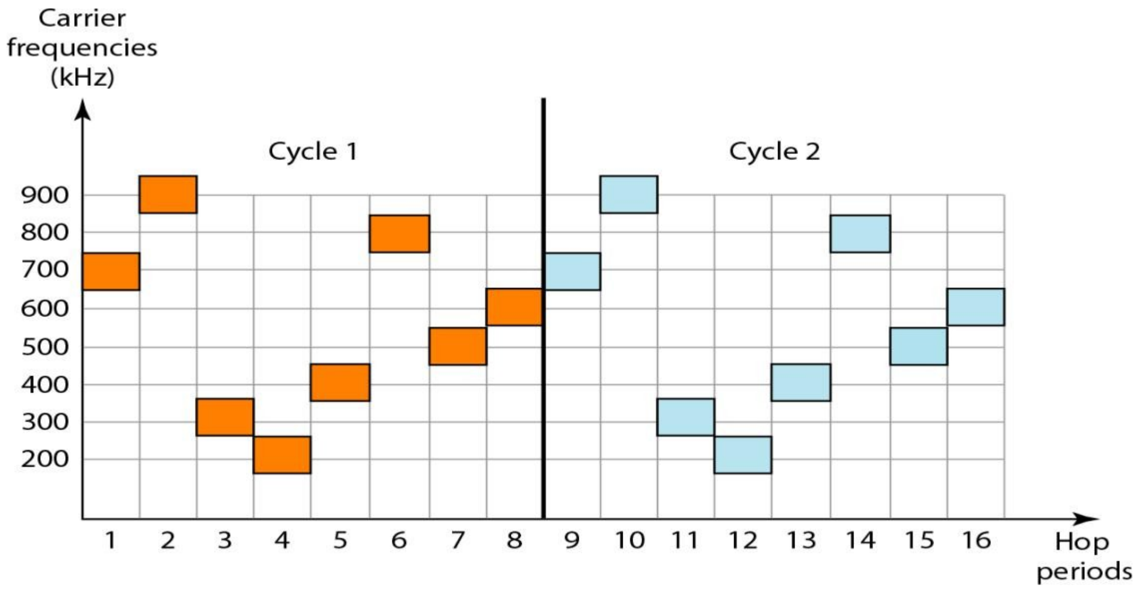
The frequency table uses the pattern to find the frequency to be used for this hopping period and passes it to the frequency synthesizer.

The frequency synthesizer creates a carrier signal of that frequency, and the source signal modulates the carrier signal.

Suppose we have decided to have eight hopping frequencies. This is extremely low for real applications and is just for illustration. In this case, M is 8 and k is 3. The pseudorandom code generator will create eight different 3-bit patterns. These are mapped to eight different frequencies in the frequency table.

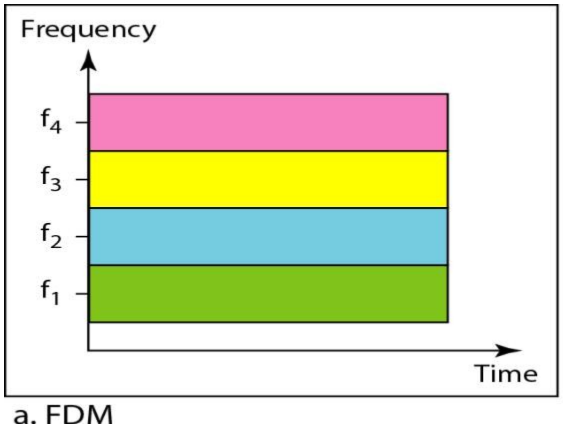


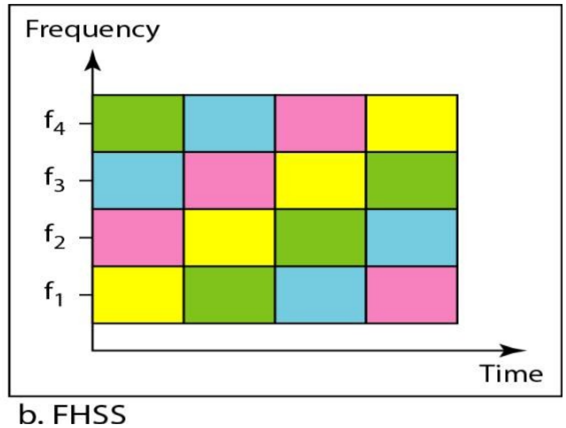
The pattern for this station is 101, 111, 001, 000, 010, all, 100. Note that the pattern is pseudorandom it is repeated after eight hoppings. This means that at hopping period 1, the pattern is 101. The frequency selected is 700 kHz; the source signal modulates this carrier frequency. The second k-bit pattern selected is 111, which selects the 900-kHz carrier; the eighth pattern is 100, the frequency is 600 kHz. After eight hoppings, the pattern repeats, starting from 101 again. Figure shows how the signal hops around from carrier to carrier. We assume the required bandwidth of the original signal is 100 kHz.



Bandwidth Sharing

If the number of hopping frequencies is *M,* we can multiplex *M* channels into one by using the same Bss bandwidth. This is possible because a station uses just one frequency in each hopping period; *M* - 1 other frequencies can be used by other *M* - 1 stations. In other words, *M* different stations can use the same Bss if an appropriate modulation technique such as multiple FSK (MFSK) is used.



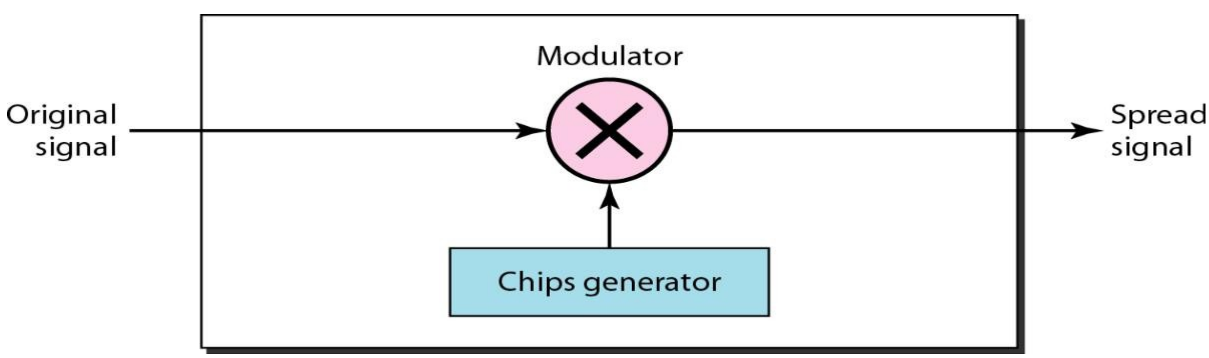


Above Figure shows an example of four channels using FDM and four channels using FHSS. In FDM, each station uses 11M of the bandwidth, but the allocation is fixed; in FHSS, each station uses 11M of the bandwidth, but the allocation changes hop to hop.

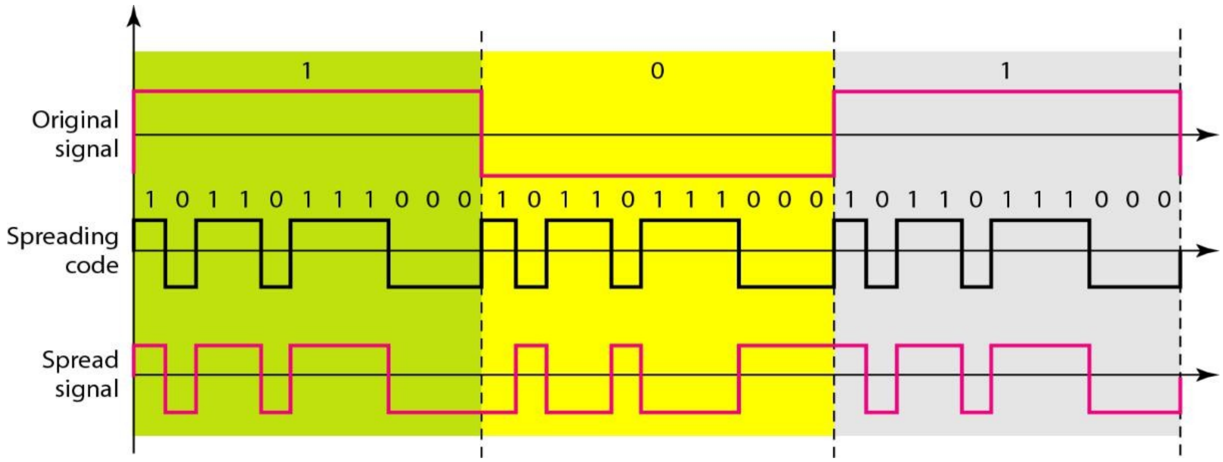
Direct Sequence Spread Spectrum

The direct sequence spread spectrum (DSSS) technique also expands the bandwidth of the original signal, but the process is different.

In DSSS, we replace each data bit with 11 bits using a spreading code. In other words, each bit is assigned a code of 11 bits, called **chips**, where the chip rate is 11 times that of the data bit.



As an example, let us consider the sequence used in a wireless LAN, the famous Barker sequence where 11 is 11. We assume that the original signal and the chips in the chip generator use polar NRZ encoding. Below Figure shows the chips and the result of multiplying the original data by the chips to get the spread signal. In Figure, the spreading code is 11 chips having the pattern 10110111000 (in this case). If the original signal rate is N, the rate of the spread signal is l1N. This means that the required bandwidth for the spread signal is 11 times larger than the bandwidth of the original signal. The spread signal can provide privacy if the intruder does not know the code. It can also provide immunity against interference if each station uses a different code.

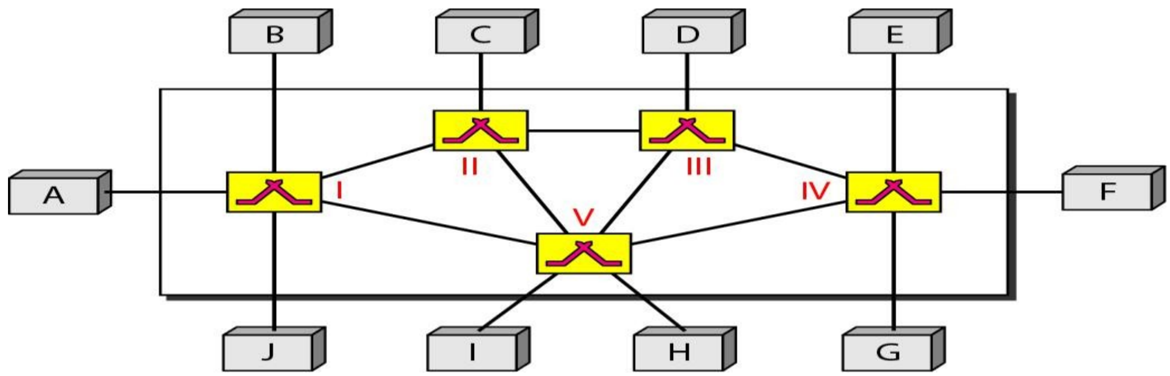


Switching

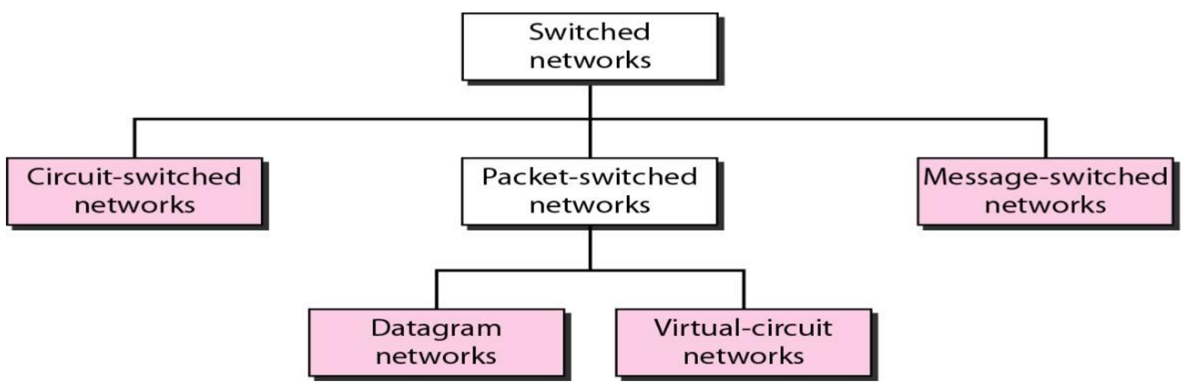
A switched network consists of a series of interlinked nodes, called switches.

Switches are devices capable of creating temporary connections between two or more devices linked to the switch.

In a switched network, some of these nodes are connected to the end systems. Others are used only for routing.



Switched networks can be divided into three broad categories: circuit-switched networks, packet-switched networks, and message-switched. Packet-switched networks can further be divided into two subcategories-virtual-circuit networks and datagram networks.



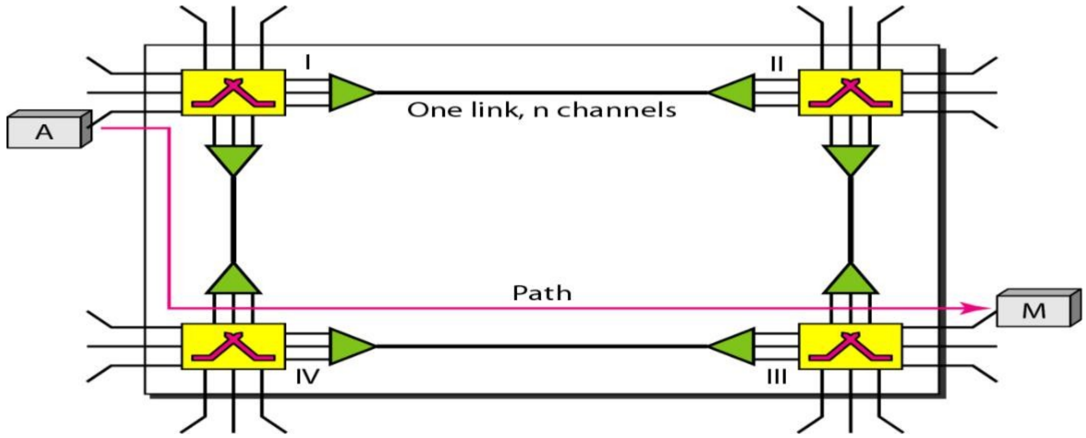
• Circuit Switched Networks

A circuit-switched network consists of a set of switches connected by physical links.

A connection between two stations is a dedicated path made of one or more links. However, each connection uses only one dedicated channel on each link.

Each link is normally divided into *n* channels by using FDM or TDM.

Below Figure shows a trivial circuit-switched network with four switches and four links. Each link is divided into n (n is 3 in the figure) channels by using FDM or TDM.



Circuit switching takes place at the physical layer.

Before starting communication, the stations must make a reservation for the resources to be used during the communication. These resources, such as channels (bandwidth in FDM and time slots in TDM), switch buffers, switch processing time, and switch input/output ports, must remain dedicated during the entire duration of data transfer until the teardown phase.

1) Data transferred between the two stations are not packetized. The data are a continuous flow sent by the source station and received by the destination station, although there may be periods of silence.

2) There is no addressing involved during data transfer. The switches route the data based on their occupied band (FDM) or time slot (TDM).

Three Phases

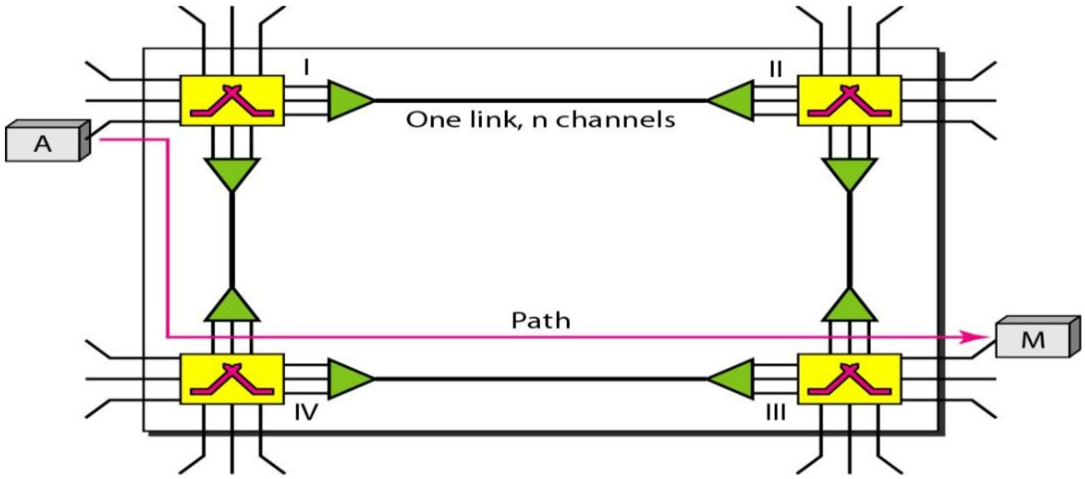
The actual communication in a circuit-switched network requires three phases: connection setup, data transfer, and connection teardown.

1. Setup Phase

Before the two parties can communicate, a dedicated circuit needs to be established. The end systems are normally connected through dedicated lines to the switches, so connection setup means creating dedicated channels between the switches.

In the next step to making a connection, an acknowledgment from system M needs to be sent in the opposite direction to system A. Only after system A receives this acknowledgment is the connection established.

2. Data Transfer Phase



After the establishment of the dedicated circuit (channels), the two parties can transfer data.

3. Teardown Phase

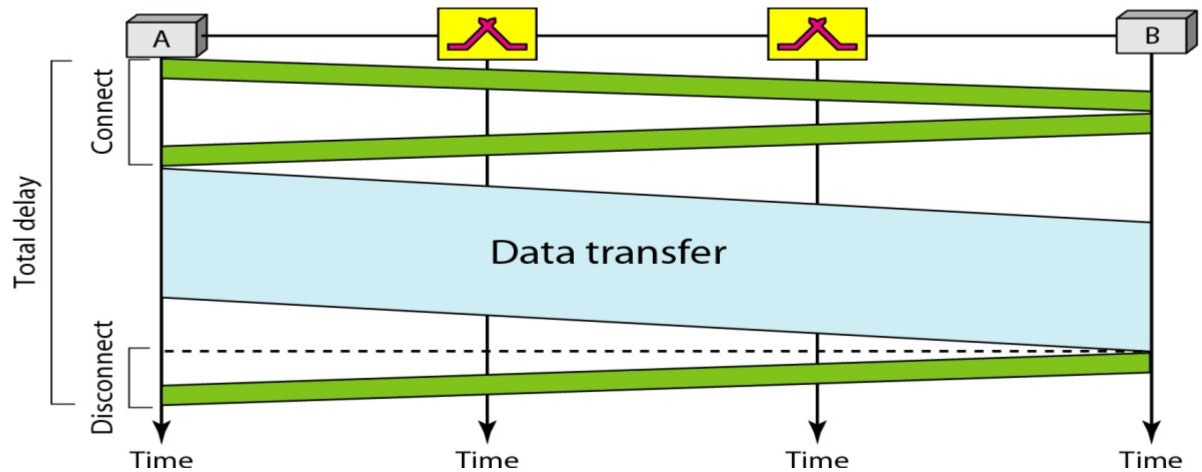
When one of the parties needs to disconnect, a signal is sent to each switch to release the resources.

Efficiency

Circuit-switched networks are not as efficient as the other two types of networks because resources are allocated during the entire duration of the connection. These resources are unavailable to other connections.

Delay

Although a circuit-switched network normally has low efficiency, the delay in this type of network is minimal. During data transfer the data are not delayed at each switch; the resources are allocated for the duration of the connection.



As Figure shows, there is no waiting time at each switch. The total delay is due to the time needed to create the connection, transfer data, and disconnect the circuit.

The delay caused by the setup is the sum of four parts: the propagation time of the source computer request, the request signal transfer time, the propagation time of the acknowledgment from the destination computer, and the signal transfer time of the acknowledgment.

The delay due to data transfer is the sum of two parts: the propagation time and data transfer time, which can be very long.

The third box shows the time needed to tear down the circuit.

Datagram Networks

In data communications, we need to send messages from one end system to another.

If the message is going to pass through a packet-switched network, it needs to be divided into packets of fixed or variable size.

The size of the packet is determined by the network and the governing protocol.

In packet switching, there is no resource allocation for a packet. This means that there is no reserved bandwidth on the links, and there is no scheduled processing time for each packet.

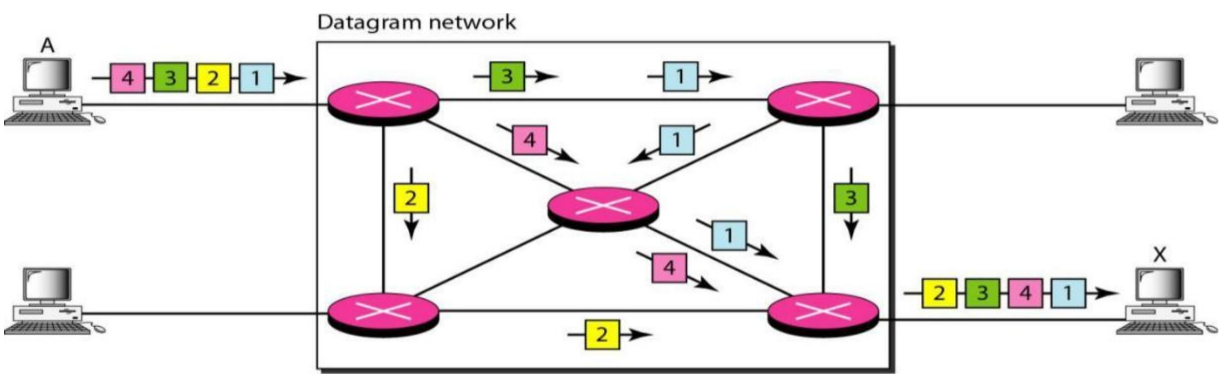
Resources are allocated on demand. The allocation is done on a first come, first-served basis. When a switch receives a packet, no matter what is the source or destination, the packet must wait if there are other packets being processed.

In a datagram network, each packet is treated independently of all others. Even if a packet is part of a multipacket transmission, the network treats it as though it existed alone. Packets in this approach are referred to as datagrams.

Datagram switching is normally done at the network layer.

The datagram networks are sometimes referred to as connectionless networks. The term *connectionless* here means that the switch (packet switch) does not keep information about the connection state. There are no setup or teardown phases. Each packet is treated the same by a switch regardless of its source or destination.

Example: Below Figure shows how the datagram approach is used to deliver four packets from station A to station X. The switches in a datagram network are traditionally referred to as routers.

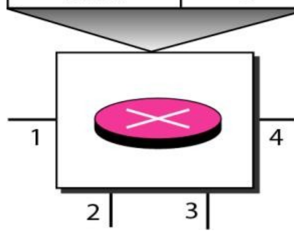


In this example, all four packets (or datagrams) belong to the same message, but may travel different paths to reach their destination. This is so because the links may be involved in carrying packets from other sources and do not have the necessary bandwidth available to carry all the packets from A to X. This approach can cause the datagrams of a transmission to arrive at their destination out of order with different delays between the packets. Packets may also be lost or dropped because of a lack of resources.

Routing Table

Each switch (or packet switch) has a routing table which is based on the destination address. The routing tables are dynamic and are updated periodically. The destination addresses and the corresponding forwarding output ports are recorded in the tables.

Destination address	Output port
1232	1
4150	2
⋮	⋮
9130	3



Destination Address

Every packet in a datagram network carries a header that contains the destination address of the packet.

When the switch receives the packet, this destination address is examined; the routing table is consulted to find the corresponding port through which the packet should be forwarded.

The destination address in the header of a packet in a datagram network remains the same during the entire journey of the packet.

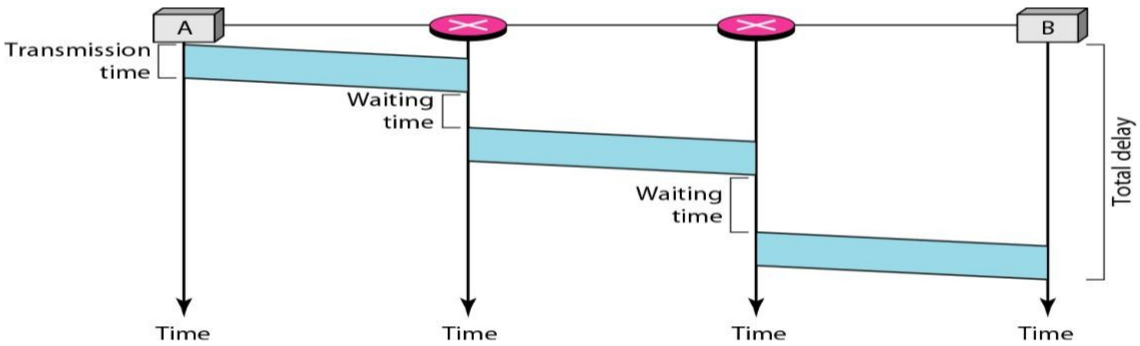
Efficiency

The efficiency of a datagram network is better than that of a circuit-switched network; resources are allocated only when there are packets to be transferred. If a source sends a packet and there is a delay of a few minutes before another packet can be sent, the resources can be reallocated during these minutes for other packets from other sources.

Delay

There may be greater delay in a datagram network than in a virtual-circuit network. Although there are no setup and teardown phases, each packet may experience a wait at a switch before it is forwarded. In addition, since not all packets in a message necessarily travel through the same switches, the delay is not uniform for the packets of a message.

Below figure shows an example of delay in a datagram network for one single packet.



The packet travels through two switches. There are three transmission times (3T), three propagation delays (slopes 3τ of the lines), and two waiting times (W1 + W2)' We ignore the processing time in each switch. The total delay is

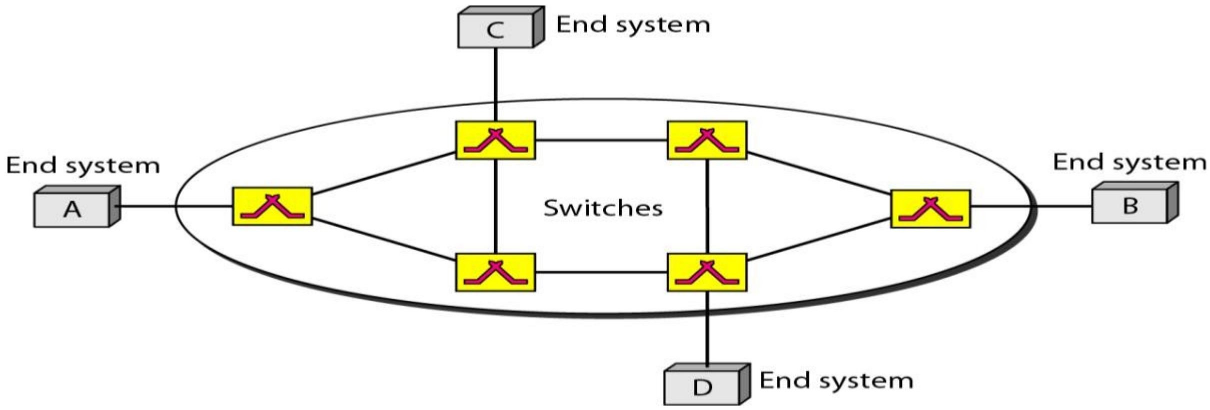
$$\text{Total delay} = 3T + 3\tau + W1 + W2$$

Virtual-Circuit Networks

A virtual-circuit network is a cross between a circuit-switched network and a datagram network. It has some characteristics of both.

- As in a circuit-switched network, there are setup and teardown phases in addition to the data transfer phase.
- Resources can be allocated during the setup phase, as in a circuit-switched network, or on demand, as in a datagram network.
- As in a datagram network, data are packetized and each packet carries an address in the header. However, the address in the header has local jurisdiction, not end-to-end jurisdiction.
- As in a circuit-switched network, all packets follow the same path established during the connection.
- A virtual-circuit network is normally implemented in the data link layer, while a circuit-switched network is implemented in the physical layer and a datagram network in the network layer.

Below Figure is an example of a virtual-circuit network. The network has switches that allow traffic from sources to destinations. A source or destination can be a computer, packet switch, bridge, or any other device that connects other networks.



Addressing

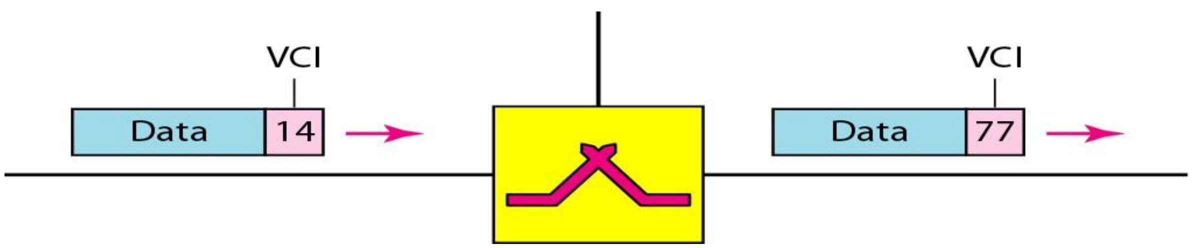
In a virtual-circuit network, two types of addressing are involved: global and local (virtual-circuit identifier).

Global Addressing

A source or a destination needs to have a global address-an address that can be unique in the scope of the network or internationally if the network is part of an international network.

Virtual-Circuit Identifier

The identifier that is actually used for data transfer is called the virtual-circuit identifier (VCI). A VCI is a small number that has only switch scope; it is used by a frame between two switches. When a frame arrives at a switch, it has a VCI; when it leaves, it has a different VCI. Below Figure shows how the VCI in a data frame changes from one switch to another.



Three Phases

Data Transfer Phase

To transfer a frame from a source to its destination, all switches need to have a table entry for this virtual circuit. The table, in its simplest form, has four columns. This means that the switch holds four pieces of information for each virtual circuit that is already set up. Below Figure shows such a switch and its corresponding table.

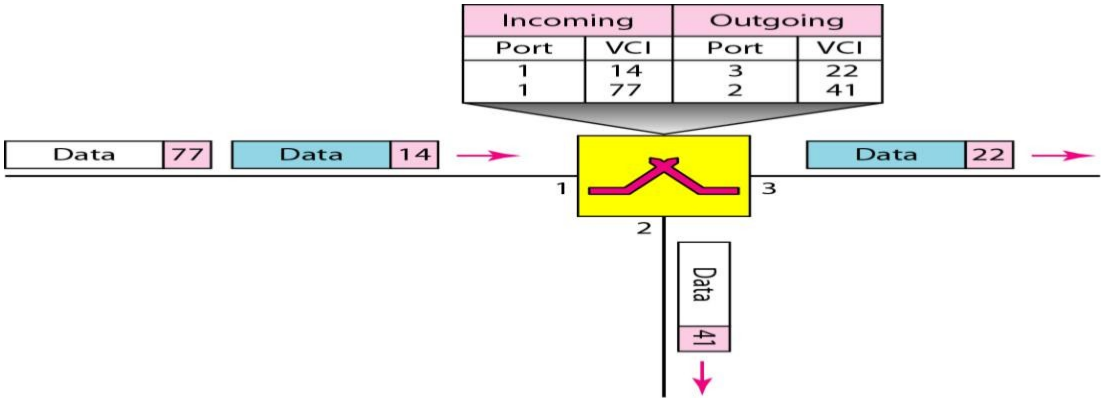
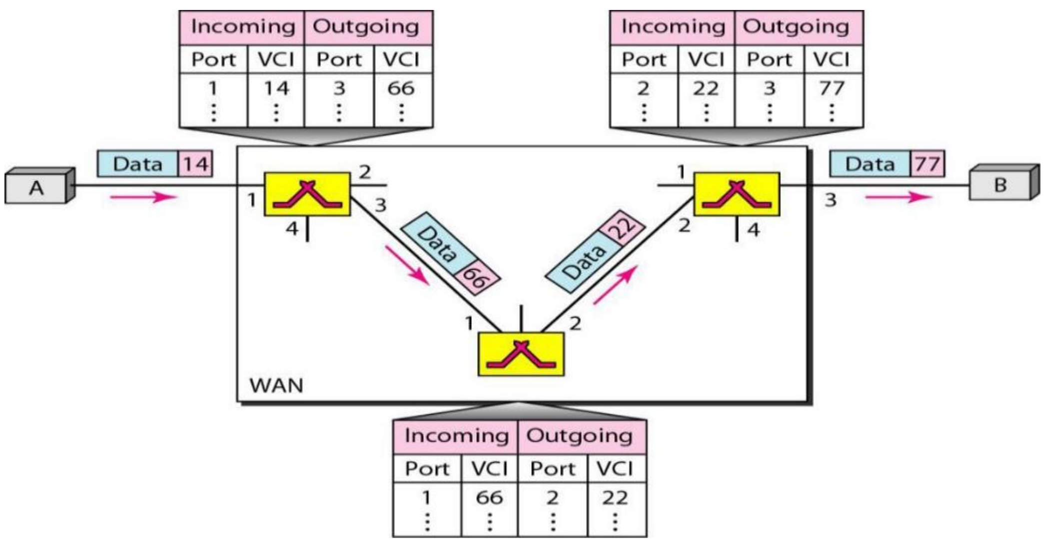


Figure shows a frame arriving at port 1 with a VCI of 14. When the frame arrives, the switch looks in its table to find port 1 and a VCI of 14. When it is found, the switch knows to change the VCI to 22 and send out the frame from port 3.

Below Figure shows how a frame from source A reaches destination B and how its VCI changes during the trip. Each switch changes the VCI and routes the frame.



The data transfer phase is active until the source sends all its frames to the destination.

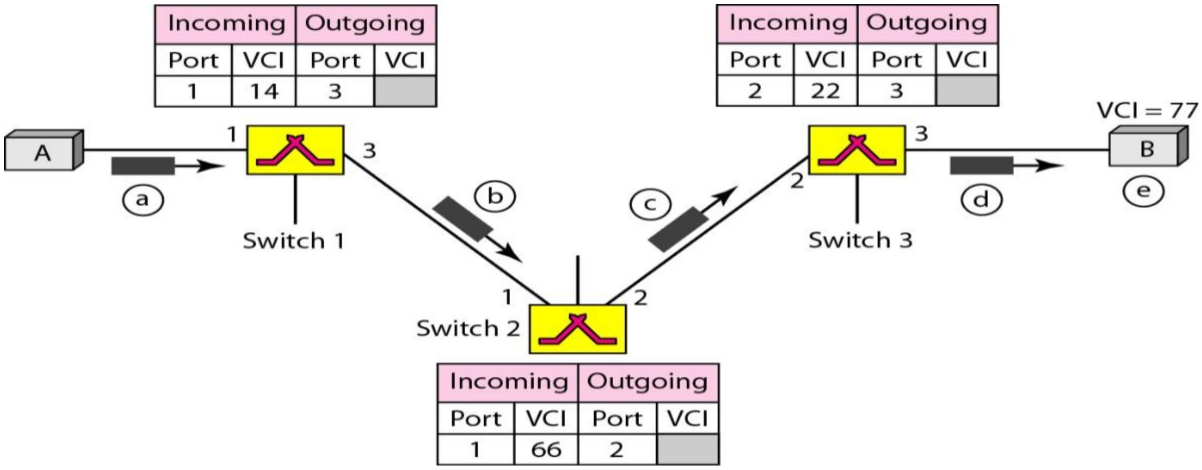
The procedure at the switch is the same for each frame of a message. The process creates a virtual circuit, not a real circuit, between the source and destination.

Setup Phase

In the setup phase, a switch creates an entry for a virtual circuit.

For example, suppose source A needs to create a virtual circuit to B. Two steps are required: the setup request and the acknowledgment.

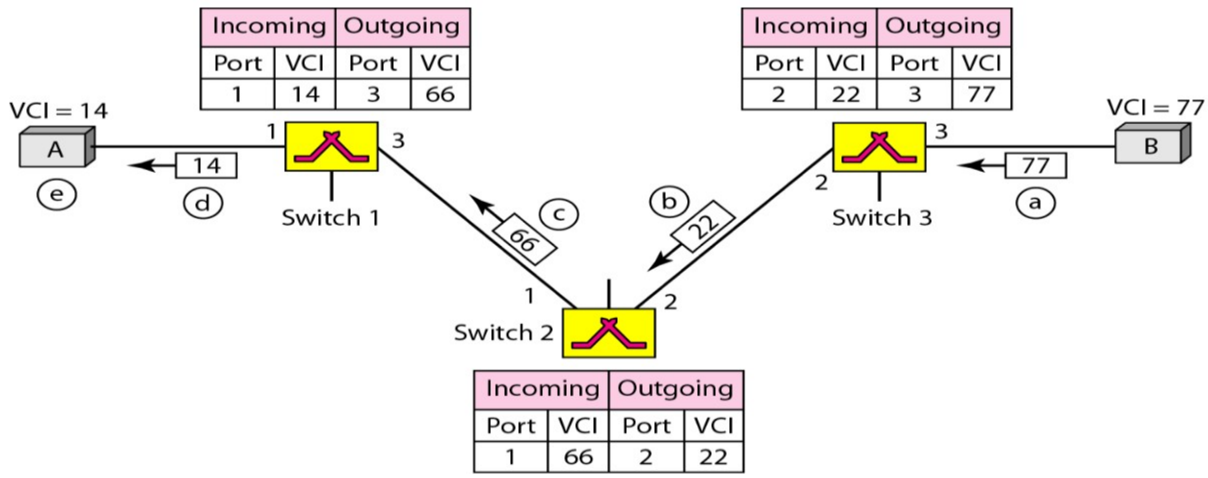
Setup Request: A setup request frame is sent from the source to the destination.



Setup request in a virtual-circuit network

1. Source A sends a setup frame to switch 1.
2. Switch 1 receives the setup request frame. It knows that a frame going from A to B goes out through port 3. The switch, in the setup phase, acts as a packet switch; it has a routing table which is different from the switching table. For the moment, assume that it knows the output port. The switch creates an entry in its table for this virtual circuit, but it is only able to fill three of the four columns. The switch assigns the incoming port (1) and chooses an available incoming VCI (14) and the outgoing port (3). It does not yet know the outgoing VCI, which will be found during the acknowledgment step. The switch then forwards the frame through port 3 to switch 2.
3. Switch 2 receives the setup request frame. The same events happen here as at switch 1; three columns of the table are completed: in this case, incoming port (l), incoming VCI (66), and outgoing port (2).
4. Switch 3 receives the setup request frame. Again, three columns are completed: incoming port (2), incoming VCI (22), and outgoing port (3).
5. Destination B receives the setup frame, and if it is ready to receive frames from A, it assigns a VCI to the incoming frames that come from A, in this case 77. This VCI lets the destination know that the frames come from A, and not other sources.

Acknowledgment: A special frame, called the acknowledgment frame, completes the entries in the switching tables.



Setup acknowledgment in a virtual-circuit network

1. The destination sends an acknowledgment to switch 3. The acknowledgment carries the global source and destination addresses so the switch knows which entry in the table is to be completed. The frame also carries VCI 77, chosen by the destination as the incoming VCI for frames from A. Switch 3 uses this VCI to complete the outgoing VCI column for this entry. Note that 77 is the incoming VCI for destination B, but the outgoing VCI for switch 3.
2. Switch 3 sends an acknowledgment to switch 2 that contains its incoming VCI in the table, chosen in the previous step. Switch 2 uses this as the outgoing VCI in the table.
3. Switch 2 sends an acknowledgment to switch 1 that contains its incoming VCI in the table, chosen in the previous step. Switch 1 uses this as the outgoing VCI in the table.
4. Finally switch 1 sends an acknowledgment to source A that contains its incoming VCI in the table, chosen in the previous step.
5. The source uses this as the outgoing VCI for the data frames to be sent to destination B.

Teardown Phase

In this phase, source A, after sending all frames to B, sends a special frame called a *teardown request.* Destination B responds with a teardown confirmation frame. All switches delete the corresponding entry from their tables.

Efficiency

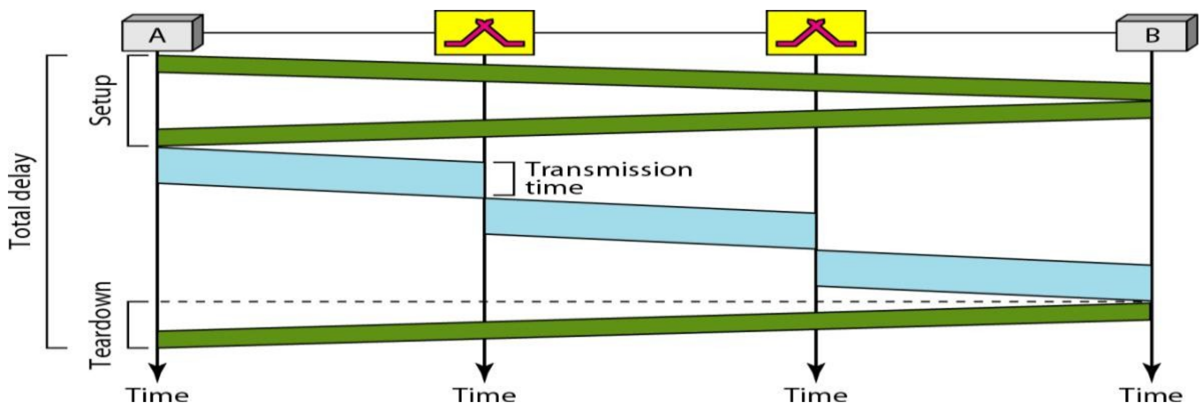
Resource reservation in a virtual-circuit network can be made during the setup or can be on demand during the data transfer phase.

In the first case, the delay for each packet is the same; in the second case, each packet may encounter different delays. There is one big advantage in a virtual-circuit network even if resource allocation is on demand. The source can check the availability of the resources, without actually reserving it.

Delay in Virtual-Circuit Networks

In a virtual-circuit network, there is a one-time delay for setup and a one-time delay for teardown.

If resources are allocated during the setup phase, there is no wait time for individual packets. Below Figure shows the delay for a packet traveling through two switches in a virtual-circuit network.



The packet is traveling through two switches (routers). There are three transmission times (3T), three propagation times (3τ), data transfer depicted by the sloping lines, a setup delay (which includes transmission and propagation in two directions), and a teardown delay (which includes transmission and propagation in one direction). We ignore the processing time in each switch. The total delay time is

$$\text{Total delay} = 3T + 3\tau + \text{setup delay} + \text{teardown delay}$$

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

Error Dection and Correction

Introduction

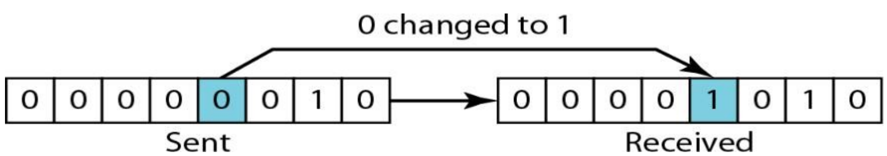
For many applications system must guarentee that the data received are same to the data transmitted.During transmission data may be corrupted because of many factors.Hence there shold be some mechanism to detect and correct errors.

Types of Errors

There are two types of error: Single bit error and Burst error.

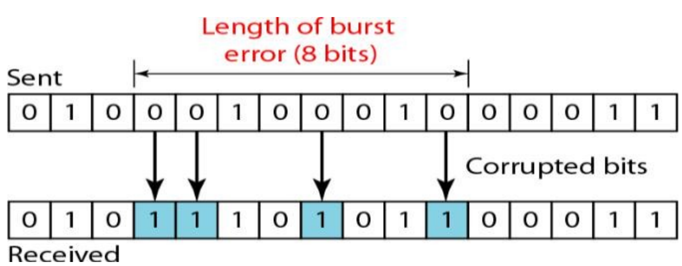
Single-Bit Error

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. Single-bit errors are the least likely type of error in serial data transmission.



Burst Error

The term burst error means that 2 or more bits in the data unit have changed from 1 to 0 or from 0 to 1. In the below figure, 0100010001000011 was sent, but 0101110101100011 was received.



Note that a burst error does not necessarily mean that the errors occur in consecutive bits. The length of the burst is measured from the first corrupted bit to the last corrupted bit. Some bits in between may not have been corrupted.

Redundancy

To detect or correct errors some extra bits are sent with 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.
- 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.

  If we need to correct one single error in an 8-bit data unit, we need to consider eight possible error locations; if we need to correct two errors in a data unit of the same size, we need to consider 28 possibilities.

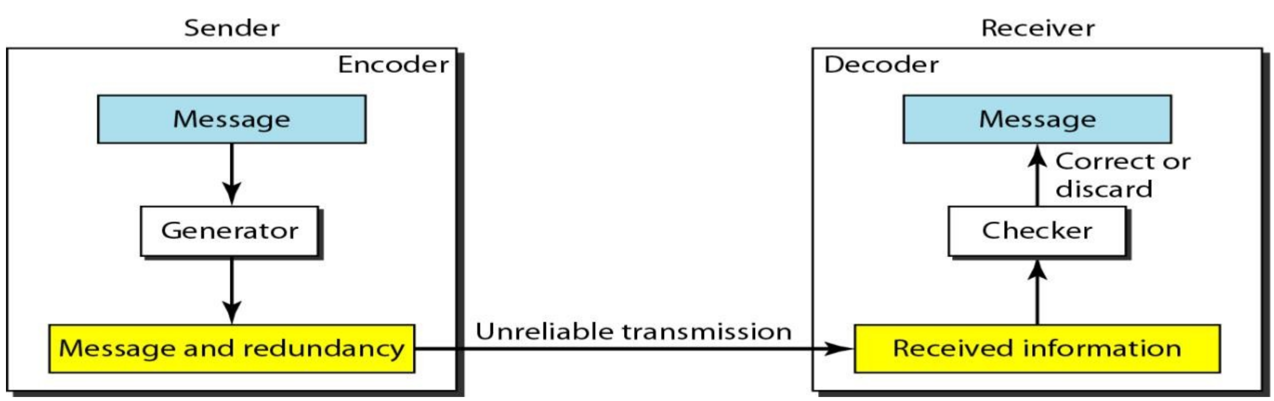
Forward Error Correction versus Retransmission

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, 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 until a message arrives that the receiver believes is error-free.

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.
- Coding schemes can be divided into two broad categories: block coding and convolution coding.



Modular Arithmetic

In modular arithmetic only integers in the range 0 to N-1 is used. This is known as *modulo-N* arithmetic.
For example, if the modulus is 12, we use only the integers 0 to 11, inclusive.

Modulo-2 Arithmetic

In this arithmetic, the modulus *N* is 2. We can use only 0 and 1. Operations in this arithmetic are very simple. The following shows how we can add or subtract 2 bits. Adding:

0+0=0 0+1=1 1+0=1 1+1=0

Subtracting:

0-0=0 0-1=1 1-0=1 1-1=0

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 I if two bits are different.

$0 \oplus 0 = 0 \qquad 1 \oplus 1 = 0$

a. Two bits are the same, the result is 0.

$0 \oplus 1 = 1 \qquad 1 \oplus 0 = 1$

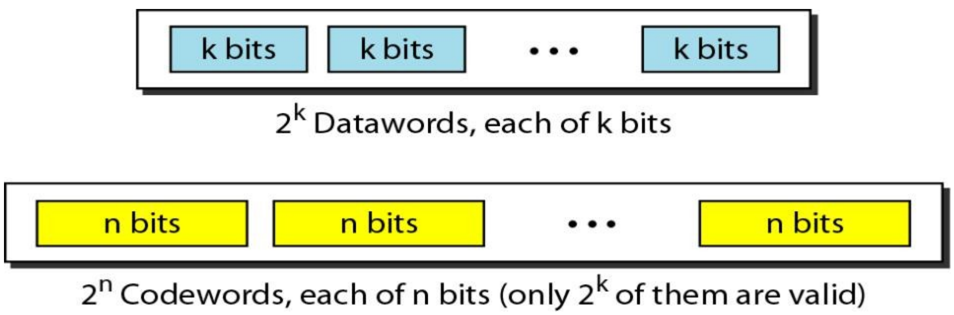
b. Two bits are different, the result is 1.

$$\begin{array}{cccccc} & 1 & 0 & 1 & 1 & 0 \\ \oplus & 1 & 1 & 1 & 0 & 0 \\ \hline & 0 & 1 & 0 & 1 & 0 \end{array}$$

c. Result of XORing two patterns

3.1 Block Coding

- In block coding message is divided into k bits blocks called **datawords**. Then *r* redundant bits are added to each block to make the length $n = k + r$. The resulting *n-bit* blocks are called **codewords**.
- With k bits, we can create a combination of 2k datawords; with n bits, we can create a combination of 2n codewords.
- Since n > k, the number of possible codewords is larger than the number of possible datawords.
- The block coding process is one-to-one; the same dataword is always encoded as the same codeword. This means that we have 2n - 2k codewords that are not used. We call these codewords invalid or illegal.

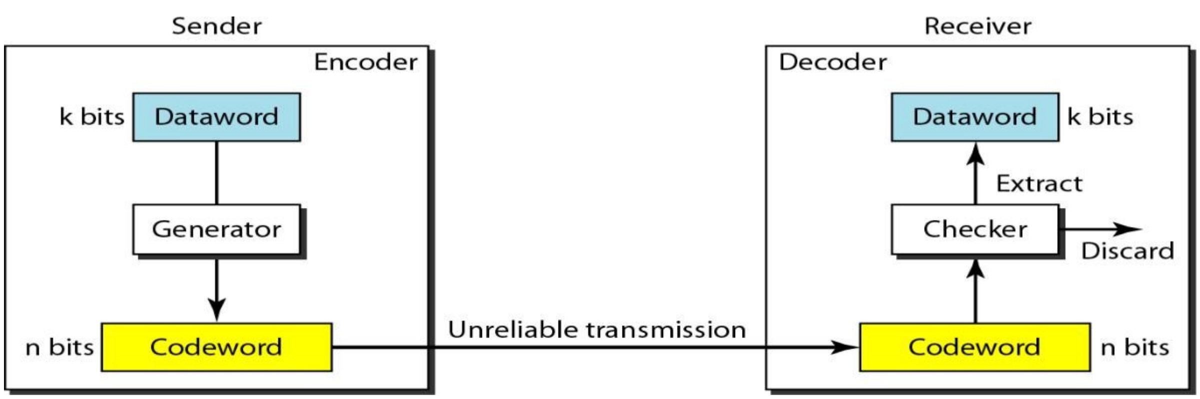


Error Detection

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

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

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



Process of error detection in block coding

- The sender creates codewords out of datawords 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 codewords, the word is accepted; the corresponding dataword 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.

Example:

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

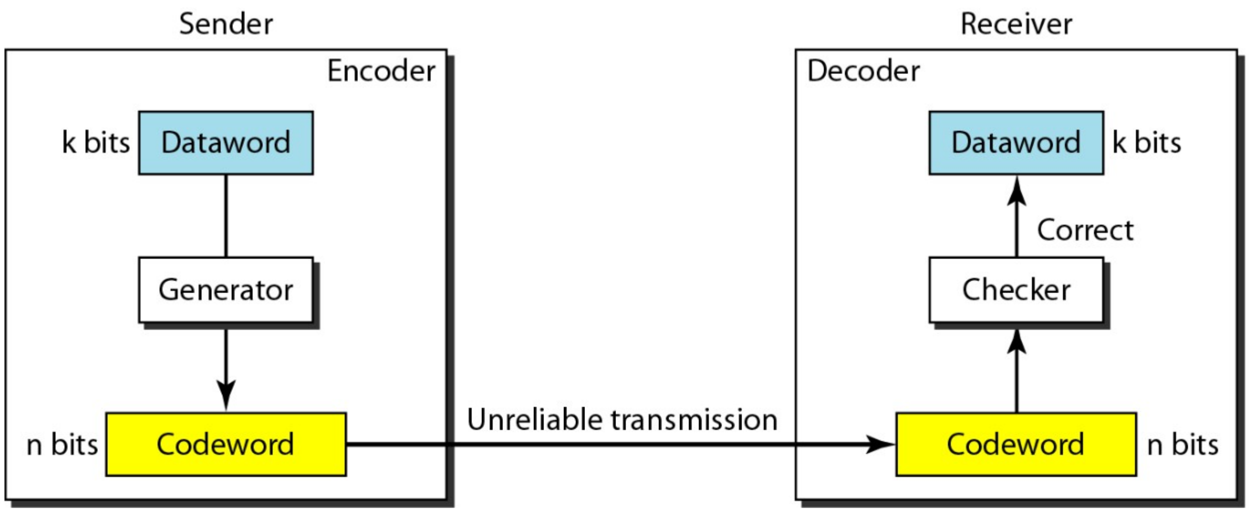
Dataword	Codeword
00	000
01	011
10	101
11	110

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

3. The receiver receives 011. It is a valid codeword. The receiver extracts the dataword 01 from it.
4. The codeword is corrupted during transmission, and 111 is received (the leftmost bit is corrupted).
   This is not a valid codeword and is discarded.
5. 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.

Error Correction

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.



Below Table shows the datawords and codewords.

Dataword	Codeword
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.

3. 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.
4. By the same reasoning, the original codeword cannot be the third or fourth one in the table.
5. 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. Hamming distance between two words x and y is represented as d(x, y).The Hamming distance can be found by applying the XOR operation on the two words and counting the number of 1s in the result.

Example:

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

Minimum Hamming Distance: The minimum Hamming distance is the smallest Hamming distance between all possible pairs in a set of words. It is represented as d_{min}.

Example 1

Find the minimum Hamming distance of the coding scheme in below table:

Dataword	Codeword
00	000
01	011
10	101
11	110

Solution

d(000,011) = 2, d(000,101) = 2, d(000,110) = 2, d(011,101) = 2, d(011,110) = 2, d(101,110) = 2 The $\mathbf{d_{min}}$ in this case is 2.

Example 2

Find the minimum Hamming distance of the coding scheme in below table:

Dataword	Codeword
00	00000
01	01011
10	10101
11	11110

Solution

d(00000,01011) = 3, d(00000,10101) = 3, d(00000,11110) = 4

d(01011,10101) = 4, d(01011,11110) = 3, d(10101,11110) = 3 The $\mathbf{d_{min}}$ in this case is 3.

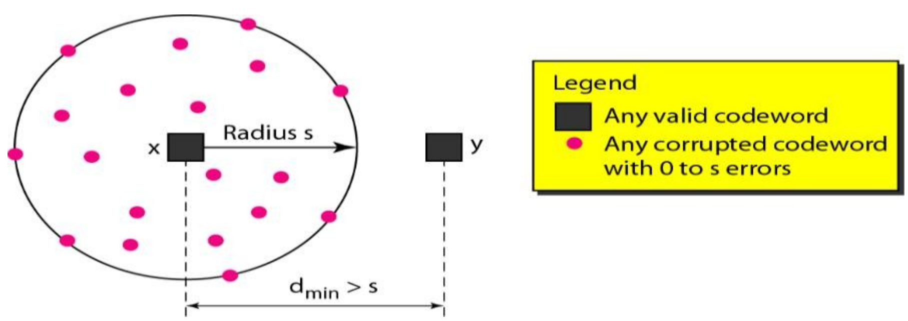
Coding scheme needs to have at least three parameters: the codeword size *n,* the dataword size *k,* and the minimum Hamming distance *dmin.* A coding scheme C is written as *C(n, k)* with a separate expression for *dmin.*
Ex: *C(5,* 2) with dmin = 3.

Hamming Distance and Error

- When a codeword is corrupted during transmission, the Hamming distance between the sent and received codewords is the number of bits affected by the error.
- The Hamming distance between the received codeword and the sent codeword is the number of bits that are corrupted during transmission.
- For example, if the codeword 00000 is sent and 01101 is received, 3 bits are in error and the Hamming distance between the two is d(00000, 01101) =3.

Minimum Distance for Error Detection

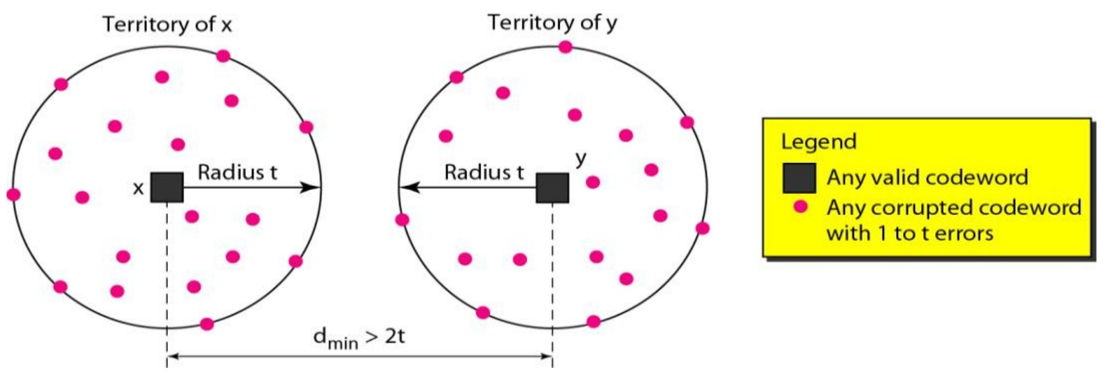
- If S errors occur during transmission, the Hamming distance between the sent codeword and received codeword is S.
- To guarantee the detection of up to S errors in all cases, the minimum Hamming distance in a block code must be dmin =*S* + 1.
- Let us assume that the sent codeword *x* is at the center of a circle with radius *S.* All other received codewords that are created by 1 to *S* errors are points inside the circle or on the perimeter of the circle. All other valid codewords must be outside the circle.



Minimum Distance for Error Correction

- When a received codeword is not a valid codeword, the receiver needs to decide which valid codeword was actually sent. The decision is based on the concept of territory, an exclusive area surrounding the codeword. Each valid codeword has its own territory.

- We use a geometric approach to define each territory. We assume that each valid codeword has a circular territory with a radius of *t* and that the valid codeword is at the center.
- For example, suppose a codeword *x* is corrupted by *t* bits or less. Then this corrupted codeword is located either inside or on the perimeter of this circle. If the receiver receives a codeword that belongs to this territory, it decides that the original codeword is the one at the center.
- To guarantee correction of up to t errors in all cases, the minimum Hamming distance in a block code must be dmin == 2t + 1.



3.2 Linear Block Codes

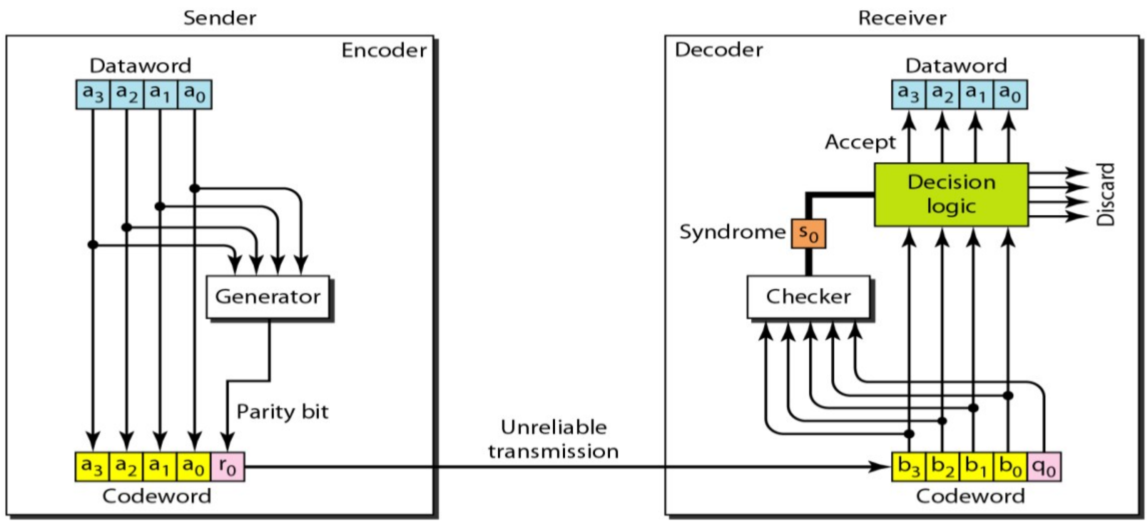
Linear block code is a code in which the exclusive OR of two valid codewords creates another valid codeword.

Minimum Distance for Linear Block Codes: The minimum Hamming distance is the number of 1s in the nonzero valid codeword with the smallest number of 1s.

Some Linear Block Codes

1) <u>Simple Parity-Check Code</u>

- In this code, a *k-bit* dataword is changed to an n-bit codeword where $n = k + 1$. The extra bit, called the parity bit, is selected to make the total number of 1s in the codeword even.
- A simple parity-check code is a single-bit error-detecting code in which $n = k + 1$ with dmin =2.



- The encoder uses a generator that takes a copy of a 4-bit dataword (a_0, a_1, a_2, and a_3) and generates a parity bit r_0.
- The dataword bits and the parity bit create the 5-bit codeword. The parity bit that is added makes the number of 1s in the codeword even.

  Example: Simple parity-check code C(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

- This is normally done by adding the 4 bits of the dataword (modulo-2); the result is the parity bit. In other words,

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

- If the number of 1s is even, the result is 0; if the number of 1s is odd, the result is 1. In both cases, the total number of 1s in the codeword is even.
- The sender sends the codeword which may be corrupted during transmission.
- The receiver receives a 5-bit word.
- The checker at the receiver does the same thing as the generator in the 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.

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

- 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.

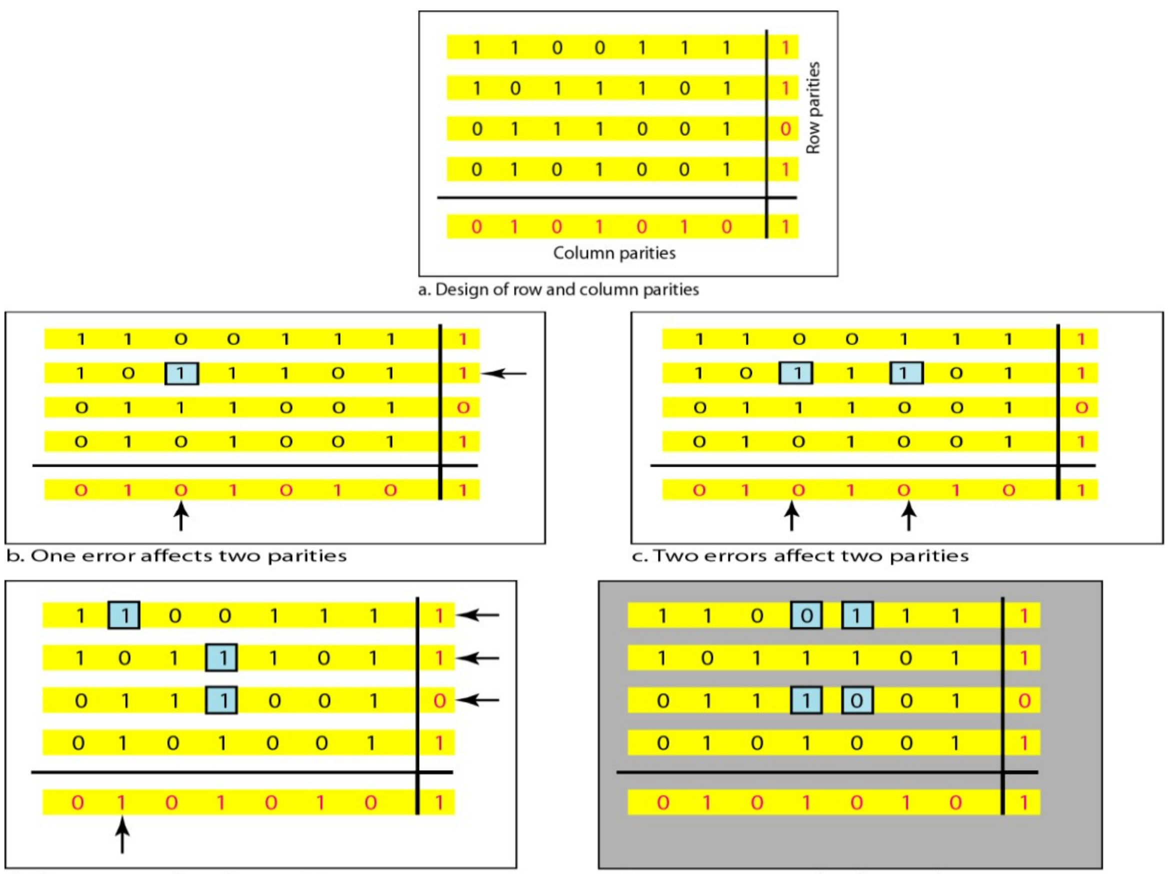
Assume the sender sends the dataword 1011. The codeword created from this dataword is 10111, which is sent to the receiver.

6. No error occurs; the received codeword is 10111. The syndrome is O. The dataword 1011 is created.
7. One single-bit error changes a_1 The received codeword is 10011. The syndrome is 1. No dataword is created.
8. One single-bit error changes r_0 The received codeword is 10110. The syndrome is 1. No dataword is created. Note that although none of the dataword bits are corrupted, no dataword is created because the code is not sophisticated enough to show the position of the corrupted bit.
9. An error changes *ro* 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.

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

- A better approach is the two-dimensional parity check. In this method, the dataword is organized in a table.
- The data to be sent, five 7-bit bytes, are put in separate rows.
- For each row and each column, 1 parity-check bit is calculated.
- The whole table is then sent to the receiver, which finds the syndrome for each row and each column.
- The two-dimensional parity check can detect up to three errors that occur anywhere in the table. However, errors affecting 4 bits may not be detected.



a. Design of row and column parities

b. One error affects two parities

c. Two errors affect two parities

d. Three errors affect four parities

e. Four errors cannot be detected

Hamming Codes

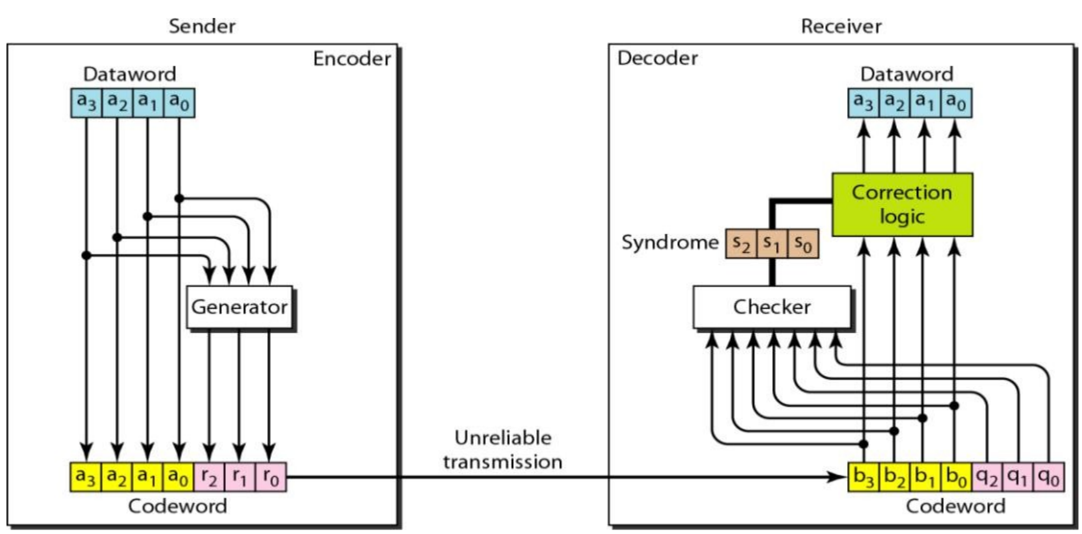
These codes were originally designed with *dmin* = 3, which means that they can detect up to two errors or correct one single error.

In hamming code we need to choose an integer m, say *m* >= 3. The values of n and *k* are then calculated from m as *n* = *2m* – 1 and *k* = *n* - *m*. The number of check bits *r* =*m*. Eg: if m = 3, n=7, k = 4

Hamming code C(7, 4) - n=7, k = 4:

Datawords	Codewords	Datawords	Codewords
0000	0000000	1000	1000110
0001	0001101	1001	1001011
0010	0010111	1010	1010001
0011	0011010	1011	1011100
0100	0100011	1100	1100101
0101	0101110	1101	1101000
0110	0110100	1110	1110010
0111	0111001	1111	1111111

Below figure shows the structure of the encoder and decoder:



A copy of a 4-bit dataword is fed into the generator that creates three parity checks.

$\mathbf{r_0 = a_2 + a_1 + a_0}$ modulo-2

$\mathbf{r_1 = a_3 + a_2 + a_1}$ modulo-2

$\mathbf{r_2 = a_1 + a_0 + a_3}$ modulo-2

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:

$\mathbf{s_0 = b_2 + b_1 + b_0}$ modulo-2

$\mathbf{s_1 = b_3 + b_2 + b_1}$ modulo-2

$\mathbf{s_2 = b_1 + b_0 + b_3}$ modulo-2

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.

Syndrome	000	001	010	011	100	101	110	111
Error	None	q_0	q_1	b_2	q_2	b_0	b_3	b_1

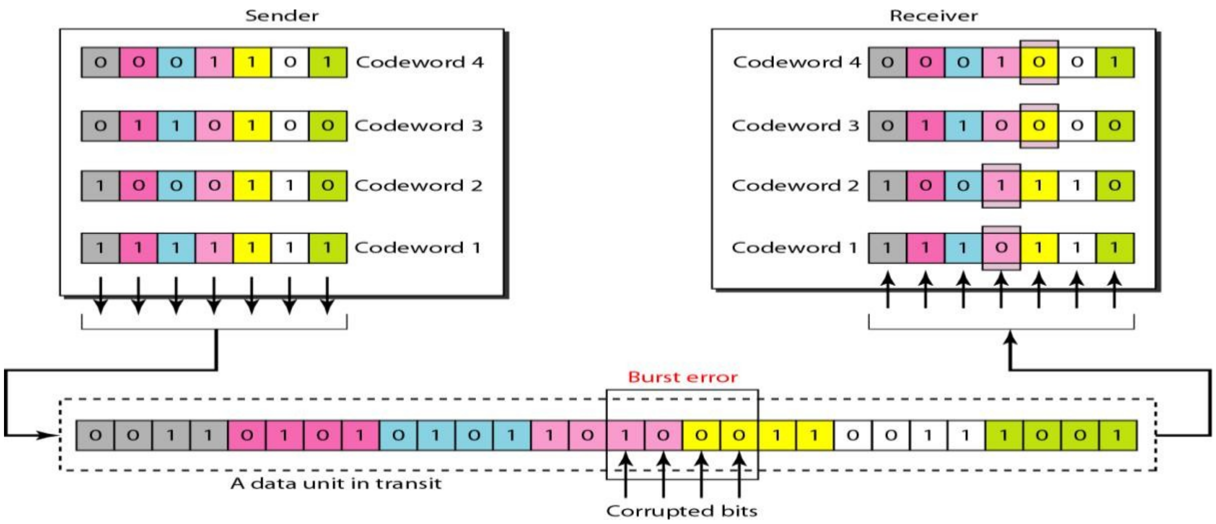
For example, if q_0 is in error, S_0 is the only bit affected; the syndrome, therefore, is 001. If b_2 is in error, S_0 and S_1 are the bits affected; the syndrome therefore is 01l. Similarly, if b_1 is in error, all 3 syndrome bits are affected and the syndrome is 111.

Example:

1. The dataword 0100 becomes the codeword 0100011. The codeword 0100011 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. Therefore b_2 is in error. After flipping b_2 (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.

Performance

A Hamming code can only correct a single error or detect a double error. However, there is a way to make it detect a burst error.



The key is to split a burst error between several codewords, one error for each codeword.

To make the Hamming code respond to a burst error of size *N,* we need to make *N* codewords out of our frame. Then, instead of sending one codeword at a time, we arrange the codewords in a table and send the bits in the table a column at a time.

In the above Figure, the bits are sent column by column (from the left). In each column, the bits are sent from the bottom to the top. In this way, a frame is made out of the four codewords and sent to the receiver. It is shown in the figure that when a burst error of size 4 corrupts the frame, only 1 bit from each codeword is corrupted. The corrupted bit in each codeword can then easily be corrected at the receiver.

3.3 Cyclic Codes

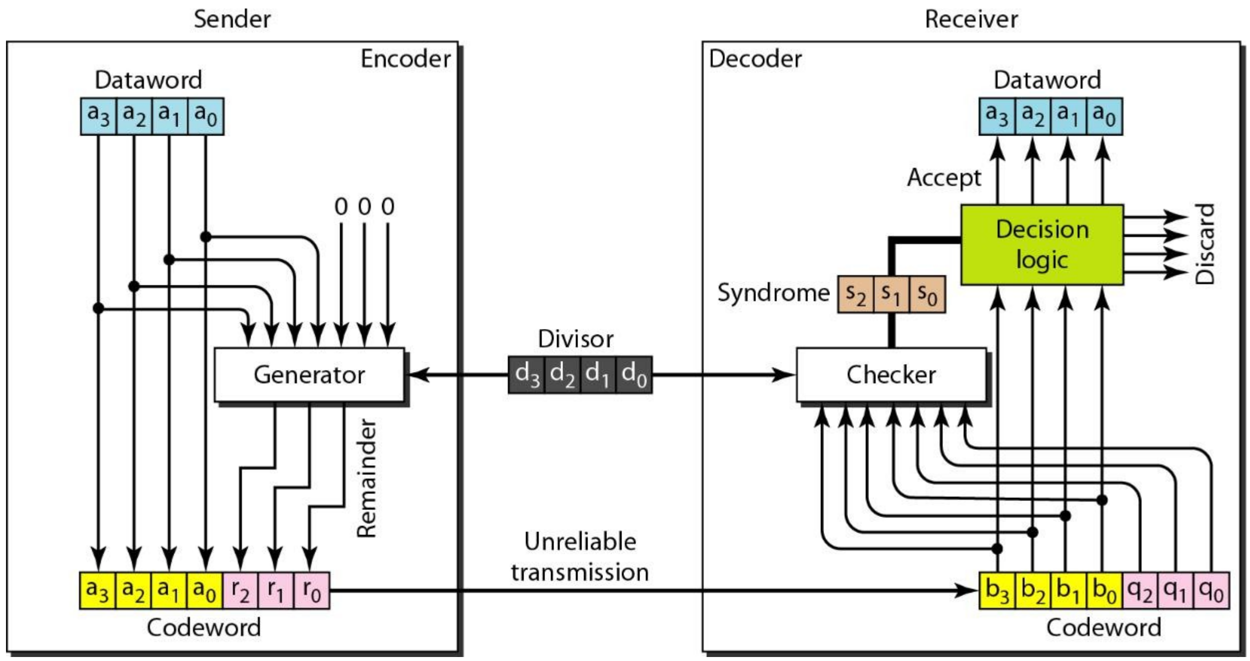
Cyclic codes are special linear block codes in which, if a codeword is cyclically shifted (rotated), the result is another codeword.

For example, if 1011000 is a codeword and we cyclically left-shift, then 0110001 is also a codeword.

In this case, 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



Below Table shows an example of a CRC code.

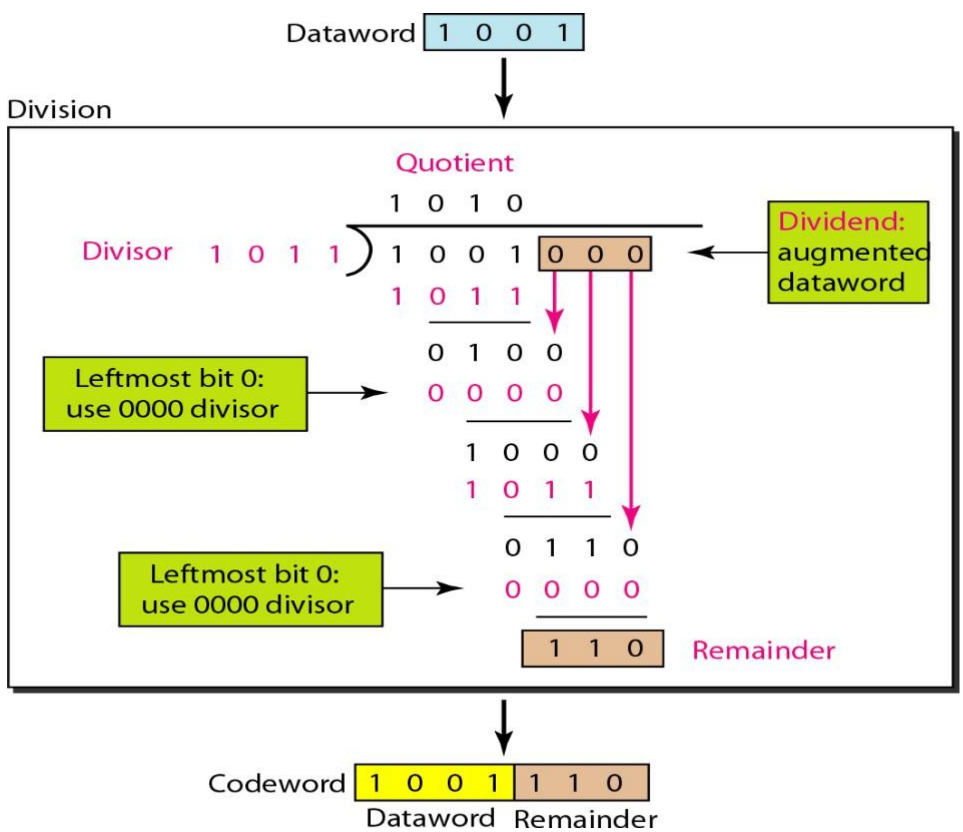
Dataword	Codeword	Dataword	Codeword
0000	0000000	1000	1000101
0001	0001011	1001	1001110
0010	0010110	1010	1010011
0011	0011101	1011	1011000
0100	0100111	1100	1100010
0101	0101100	1101	1101001
0110	0110001	1110	1110100
0111	0111010	1111	1111111

- In the encoder, the dataword has k bits (4 here); the codeword has n bits (7 here). The size of the dataword is augmented by adding n - k (3 here) 0s to the right-hand side of the word. The n-bit result is fed into the generator.

- The generator uses a divisor of size n - k + 1 (4 here), predefined and agreed upon. The generator divides the augmented dataword by the divisor (modulo-2 division).
- The quotient of the division is discarded; the remainder is appended to the dataword to create the codeword.
- The decoder receives the possibly corrupted codeword. A copy of all n bits is fed to the checker which is a replica of the generator.
- The remainder produced by the checker is a syndrome of n - k (3 here) bits, which is fed to the decision logic analyzer.
- The analyzer has a simple function. If the syndrome bits are all as, the 4 leftmost bits of the codeword are accepted as the dataword (interpreted as no error); otherwise, the 4 bits are discarded (error).

Encoder

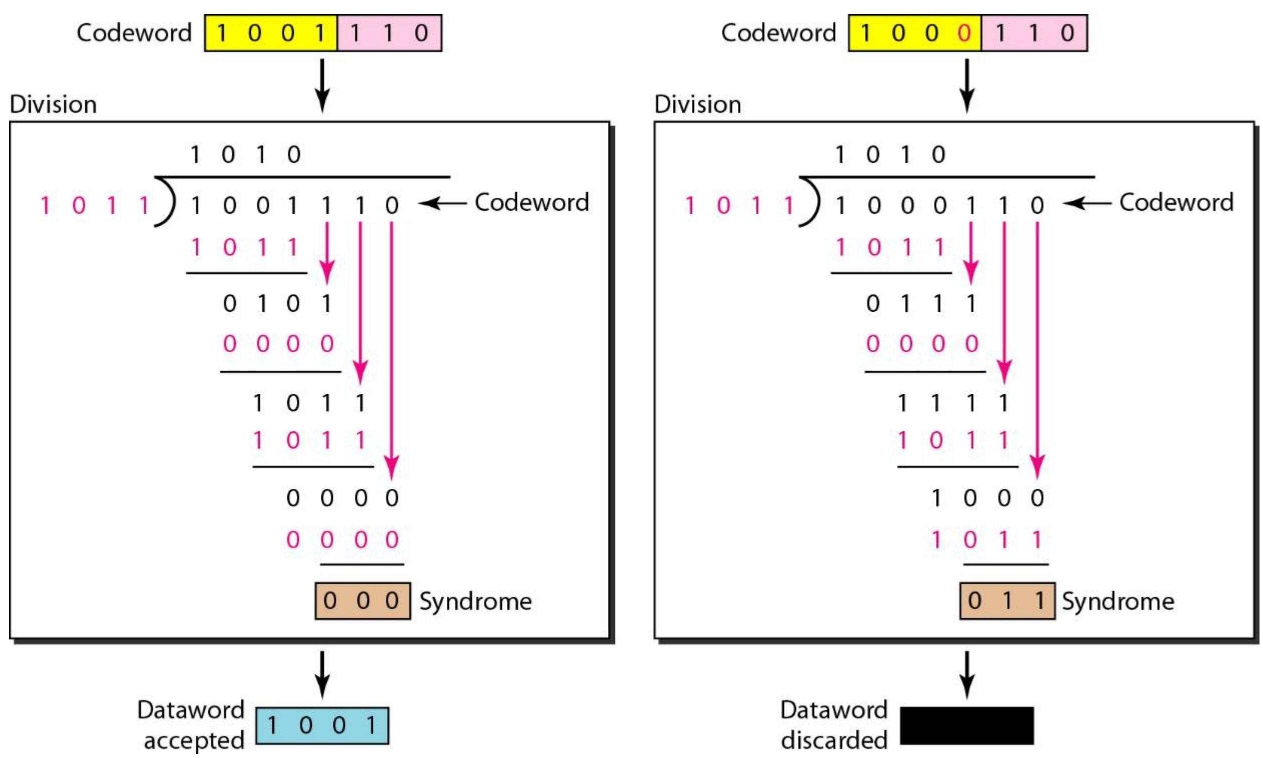
The encoder takes the dataword and augments it with n - k number of 0s. It then divides the augmented dataword by the divisor.



Decoder

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

The left hand figure shows the value of syndrome when no error has occurred; the syndrome is 0. The right-hand part of the figure shows the case in which there is one single error. The syndrome is not all 0s



Hardware Implementation

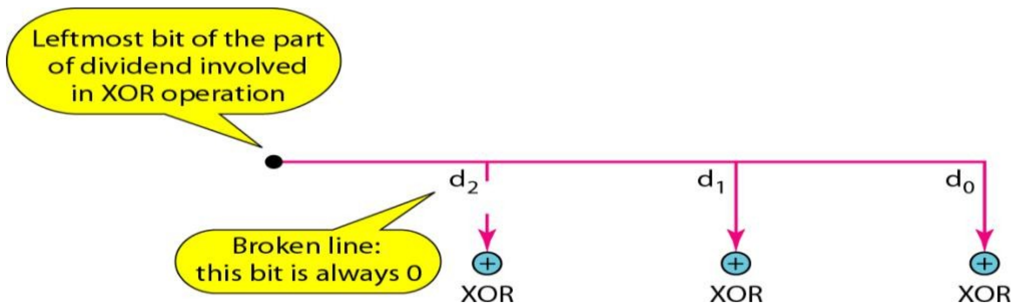
One of the advantages of a cyclic code is that the encoder and decoder can easily and cheaply be implemented in hardware by using a handful of electronic devices. Also, a hardware implementation increases the rate of check bit and syndrome bit calculation.

Divisor:

1. The divisor is repeatedly XORed with part of the dividend.
2. The divisor has $n - k + 1$ bits which either are predefined or are all Os. In other words, the bits do not change from one dataword to another. In previous example, the divisor bits were

either 1011 or 0000. The choice was based on the leftmost bit of the part of the augmented data bits that are active in the XOR operation.

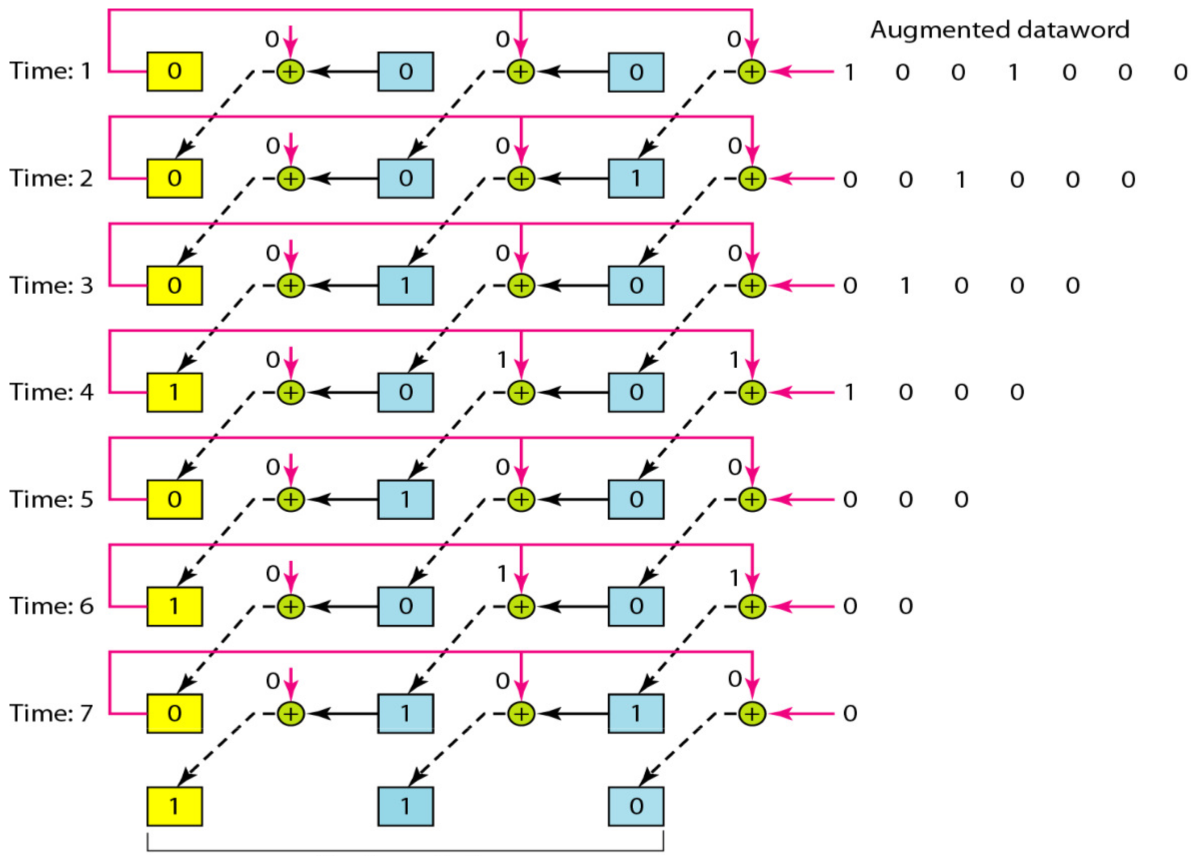
3. A close look shows that only *n* - *k* bits of the divisor is needed in the XOR operation. The leftmost bit is not needed because the result of the operation is always 0, no matter what the value of this bit. The reason is that the inputs to this XOR operation are either both 0s or both 1s.



Steps:

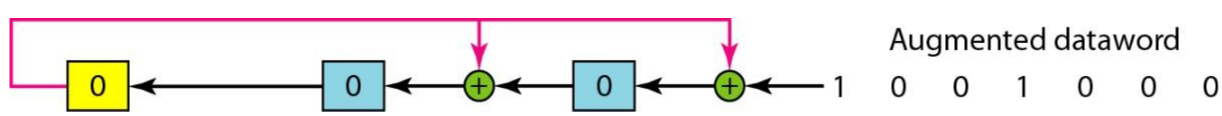
1. Assume that the remainder is originally all Os (000 in our example).
2. At each time click (arrival of 1 bit from an augmented dataword), repeat the following two actions:
   a. Use the leftmost bit to make a decision about the divisor (011 or 000).
   b. The other 2 bits of the remainder and the next bit from the augmented dataword (total of 3 bits) are XORed with the 3-bit divisor to create the next remainder.

Below Figure shows this simulator, but note that this is not the final design; there will be more improvements.

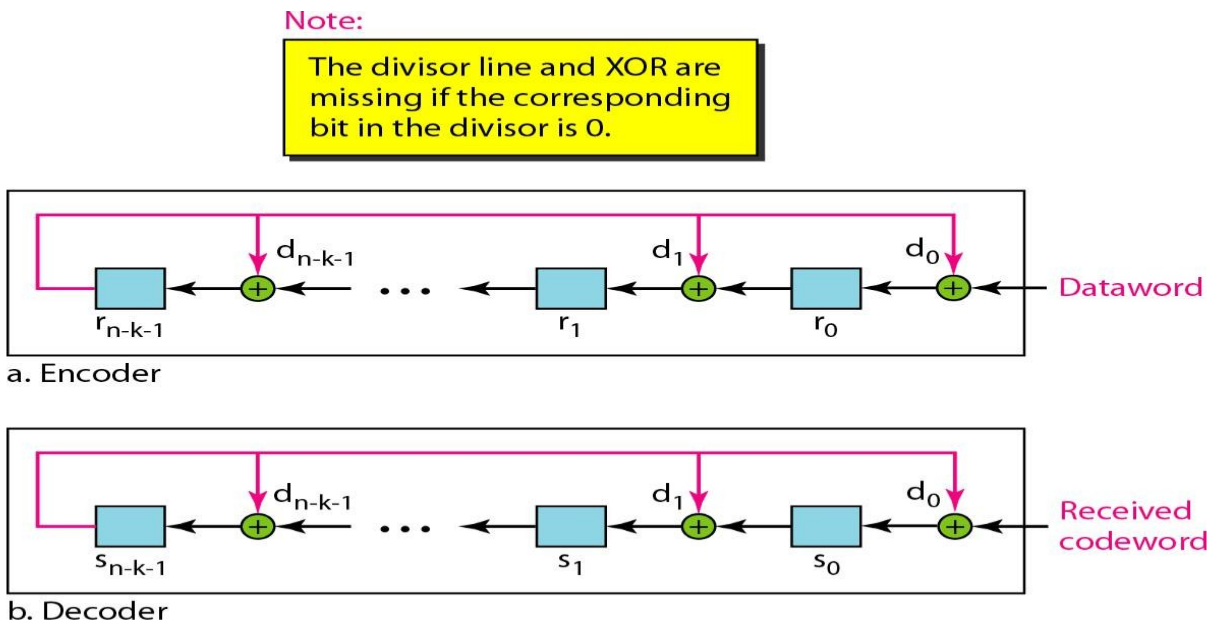


At each clock tick, shown as different times, one of the bits from the augmented dataword is used in the XOR process.

The above design is for demonstration purposes only. It needs simplification to be practical. First, we do not need to keep the intermediate values of the remainder bits; we need only the final bits. We therefore need only 3 registers instead of 24. After the XOR operations, we do not need the bit values of the previous remainder. Also, we do not need 21 XOR devices; two are enough because the output of an XOR operation in which one of the bits is 0 is simply the value of the other bit. This other bit can be used as the output. With these two modifications, the design becomes tremendously simpler and less expensive, as shown below

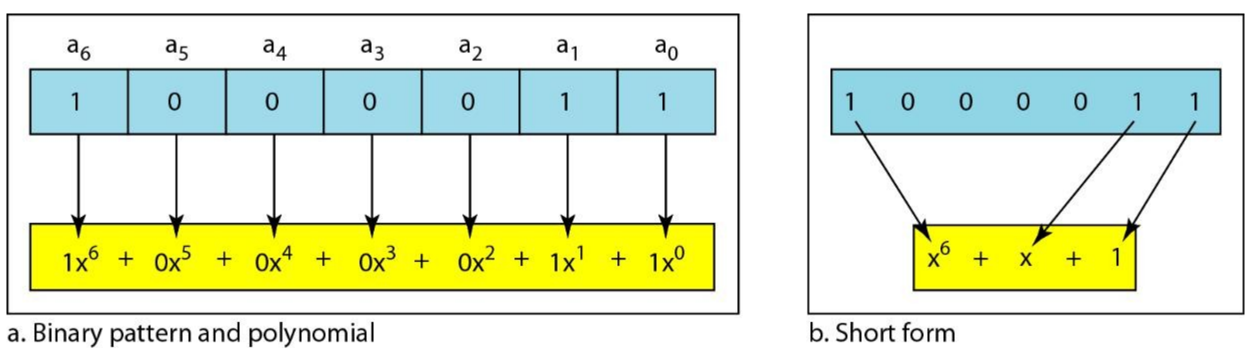


General Design



<u>Polynomials</u>

A pattern of Os 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 value of the bit. Figure shows a binary pattern and its polynomial representation.



Degree of a Polynomial

The degree of a polynomial is the highest power in the polynomial.

For example, the degree of the polynomial $x6 + x + 1$ is 6. Note that the degree of a polynomial is 1 less that the number of bits in the pattern. The bit pattern in this case has 7 bits.

Adding and Subtracting Polynomials

Adding and subtracting polynomials in mathematics are done by adding or subtracting the coefficients of terms with the same power. In our case, the coefficients are only 0 and 1, and adding is in modulo-2. This has two consequences. First, addition and subtraction are the same. Second, adding or subtracting is done by combining terms and deleting pairs of identical terms. For example, adding *x5 + x4 + x2* and *x6 + x4 + x2* gives just *x6 + x5.* The terms *x4* and *x2* are deleted. However, note that if we add, for example, three polynomials and we get *x2* three times, we delete a pair of them and keep the third.

Multiplying or Dividing Terms

In this arithmetic, multiplying a term by another term is very simple; we just add the powers.
For example, x3 x x4 is x7 , For dividing, we just subtract the power of the second term from the power of the first.

Multiplying Two Polynomials

Multiplying a polynomial by another is done term by term. Each term of the first polynomial must be multiplied by all terms of the second. The result, of course, is then simplified, and pairs of equal terms are deleted. The following is an example:

$$\begin{aligned}
&(x^5 + x^3 + x^2 + x)(x^2 + x + 1) \\
&\quad = x^7 + x^6 + x^5 + x^5 + x^4 + x^3 + x^4 + x^3 + x^2 + x^3 + x^2 + x \\
&\quad = x^7 + x^6 + x^3 + x
\end{aligned}$$

Dividing One Polynomial by Another

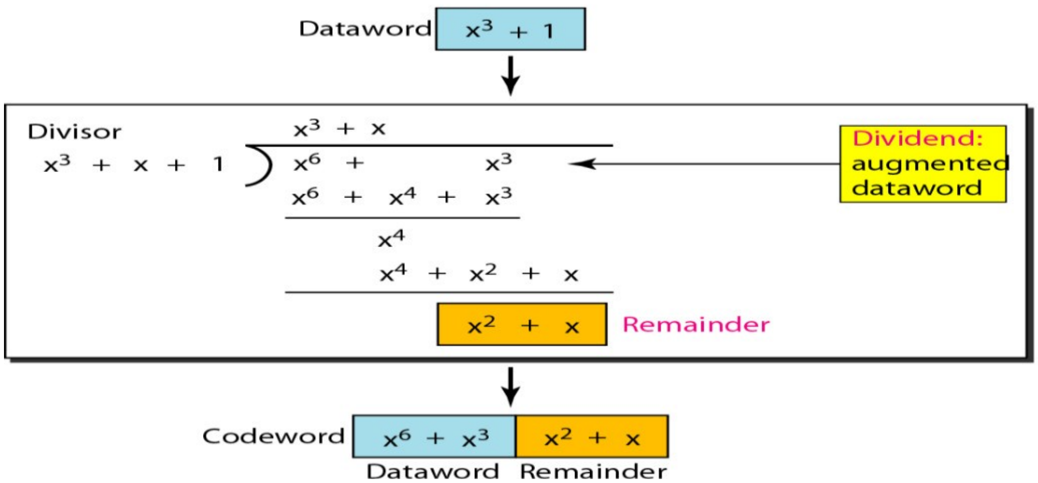
We divide the first term of the dividend by the first term of the divisor to get the first term of the quotient. We multiply the term in the quotient by the divisor and subtract the result from the dividend. We repeat the process until the dividend degree is less than the divisor degree.

Shifting

A binary pattern is often shifted a number of bits to the right or left. Shifting to the left means adding extra Os as rightmost bits; shifting to the right means deleting some rightmost bits. Shifting to the left is accomplished by multiplying each term of the polynomial by xn, where m is the number of shifted bits; shifting to the right is accomplished by dividing each term of the polynomial by xn. The following shows shifting to the left and to the right. Note that we do not have negative powers in the polynomial representation.

Shifting left 3 bits: 10011 becomes 10011000 $x^4 + x + 1$ becomes $x^7 + x^4 + x^3$

Shifting right 3 bits: 10011 becomes 10 $x^4 + x + 1$ becomes x



Cyclic Code Analysis

Following notations can be used in the cyclic codes:

Dataword: *d(x)* Syndrome: *s(x)* Codeword: *c(x)*

Error: *e(x)* Generator: *g(x)*

In a cyclic code,

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

The received codeword is the sum of the sent codeword and the error.

Received codeword =*c(x) + e(x)*

The receiver divides the received codeword by g(x) to get the syndrome.

$$\frac{\text{Received codeword}}{g(x)} = \frac{c(x)}{g(x)} + \frac{e(x)}{g(x)}$$

The Right hand side of above equation is called as syndrome.

If Syndrome does not have a remainder (syndrome =0), either *e(x)* is 0 or *e(x)* is divisible by *g(x)*.

In a cyclic code, those *e(x)* errors that are divisible by *g(x)* are not caught.

Single-Bit Error

A single-bit error is $e(x) = x^i$, where *i* is the position of the bit. If a single-bit error is caught, then x^i is not divisible by *g(x)*.
If the generator has more than one term and the coefficient of x^0 is 1, all single errors can be caught.

Example:

Which of the following *g(x)* values guarantees that a single-bit error is caught? For each case, what is the error that cannot be caught?

a. $x + 1$

b. x^3

c. 1

Solution:

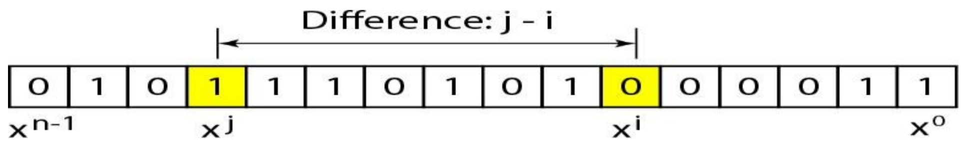
a. No x^i can be divisible by $x + 1$. In other words, $x^i/(x + 1)$ always has a remainder. So the syndrome is nonzero. Any single-bit error can be caught.

b. If i is equal to or greater than 3, x^i is divisible by *g(x)*. The remainder of x^i/x^3 is zero, and the receiver is fooled into believing that there is no error, although there might be one. All single-bit errors in positions 1 to 3 are caught.

c. All values of *i* make *j* divisible by *g(x)*. No single-bit error can be caught

Two Isolated Single-Bit Errors

Two isolated single bit errors can be represented as $e(x) = x^j + x^i$. The values of i and j define the positions of the errors, and the difference j - i defines the distance between the two errors.



We can write $e(x) = x^i (x^{j-i} + 1) = x^i (x^t + 1)$.
t

If a generator cannot divide x

be detected.

+ 1 (t between 0 and n - 1), then all isolated double errors can

Example:

Find the status of the following generators related to two isolated, single-bit errors.

a. $x + 1$

b. $x^4 + I$

c. $x^7 + x^6 + 1$

d. $x \quad + x^{15}$

14 + 1

Solution:

a. This is a very poor choice for a generator. Any two errors next to each other cannot be detected.

b. This generator cannot detect two errors that are four positions apart. The two errors can be anywhere, but if their distance is 4, they remain undetected.

c. This is a good choice for this purpose.

d. This polynomial cannot divide any error of *type* $x^t + 1$ if t is less than 32,768. This means that a codeword with two isolated errors that are next to each other or up to 32,768 bits apart can be detected by this generator.

Odd Numbers of Errors

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

Burst Errors

If L is the burst size and r is the degree of generator polynomial.

- All burst errors with L <= r will be detected.
- All burst errors with L = r + 1 will be detected with probability $1 - (1/2)^{r-1}$
- All burst errors with L > r + 1 will be detected with probability $1 - (1/2)^{r}$

Example:

Find the suitability of the following generators in relation to burst errors of different lengths.

a. x6 + 1

b. x18+ x7 + x + 1

c. x32+x23+x7+1

Solution:

a. This generator can detect all burst errors with a length less than or equal to 6 bits; 3 out of 100 burst errors with length 7 will slip by; 16 out of 1000 burst errors of length 8 or more will slip by.

b. This generator can detect all burst errors with a length less than or equal to 18 bits; 8 out of 1 million burst errors with length 19 will slip by; 4 out of I million burst errors of length 20 or more will slip by.

c. This generator can detect all burst errors with a length less than or equal to 32 bits; 5 out of 10 billion burst errors with length 33 will slip by; 3 out of 10 billion burst errors of length 34 or more will slip by.

A good polynomial generator needs to have the following characteristics:

1. It should have at least two terms.
2. The coefficient of the term x0 should be 1.
3. It should not divide $x^t + 1$, for t between 2 and n - 1.
4. It should have the factor x + 1.

Standard Polynomials

Name	*Polynomial*	*Application*
CRC-8	$x^8 + x^2 + x + 1$	ATM header
CRC-10	$x^{10} + x^9 + x^5 + x^4 + x^2 + 1$	ATM AAL
CRC-16	$x^{16} + x^{12} + x^5 + 1$	HDLC
CRC-32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	LANs

Advantages of Cyclic Codes

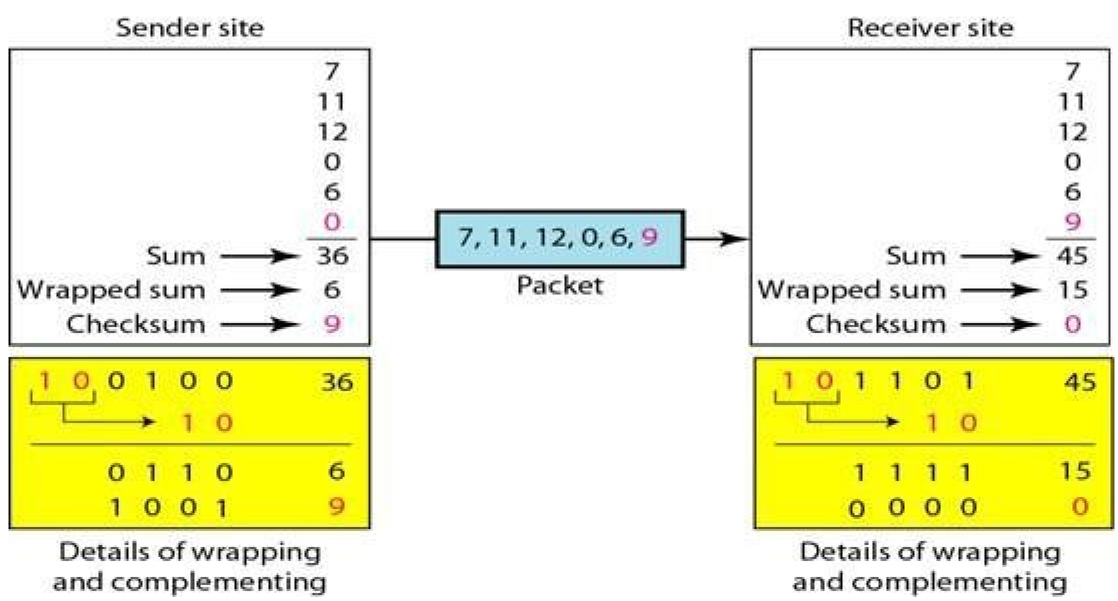
- Cyclic codes have a very good performance in detecting single-bit errors, double errors, an odd number of errors, and burst errors.

- They can easily be implemented in hardware and software.
- They are especially fast when implemented in hardware.

3.4 Checksum

The checksum is used in the Internet by several protocols. The checksum is based on the concept of redundancy.

Below Figure shows the process at the sender and at the receiver. The sender initializes the checksum to 0 and adds all data items and the checksum (the checksum is considered as one data item and is shown in color). The result is 36. However, 36 cannot be expressed in 4 bits. The extra two bits are wrapped and added with the sum to create the wrapped sum value 6. In the figure, we have shown the details in binary. The sum is then complemented, resulting in the checksum value 9 (15 - 6 = 9). The sender now sends six data items to the receiver including the checksum 9. The receiver follows the same procedure as the sender. It adds all data items (including the checksum); the result is 45. The sum is wrapped and becomes 15. The wrapped sum is complemented and becomes 0. Since the value of the checksum is 0, this means that the data is not corrupted. The receiver drops the checksum and keeps the other data items. If the checksum is not zero, the entire packet is dropped.



<u>Internet Checksum</u>

Traditionally, the Internet has been using a 16-bit checksum.

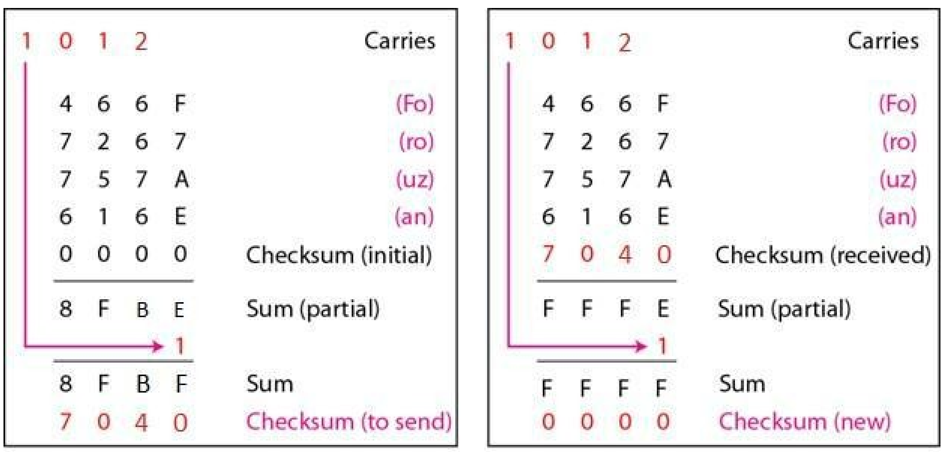
Sender site:

1. The message is divided into 16-bit words.
2. The value of the checksum word is set to 0.
3. All words including the checksum are added using one's complement addition.
4. The sum is complemented and becomes the checksum.
5. The checksum is sent with the data.

Receiver site:

1. The message (including checksum) is divided into 16-bit words.
2. All words are added using one's complement addition.
3. The sum is complemented and becomes the new checksum.
4. If the value of checksum is 0, the message is accepted; otherwise, it is rejected.

Example:



a. Checksum at the sender site   a. Checksum at the receiver site

Performance

The traditional checksum uses a small number of bits (16) to detect errors in a message of any size (sometimes thousands of bits). However, it is not as strong as the CRC in error-checking capability. For example, if the value of one word is incremented and the value of another word is decremented by the same amount, the two errors cannot be detected because the sum and checksum remain the same. Also if the values of several words are incremented but the total change is a multiple of 65535, the sum and the checksum does not change, which means the errors are not detected.

Module 3

Bandwidth Utilization: Multiplexing and Spreading Spectrum

Note

Bandwidth utilization is the wise use of available bandwidth to achieve specific goals.

Efficiency can be achieved by multiplexing; privacy and anti-jamming can be achieved by spreading.

6-1 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.

Topics discussed in this section:

Frequency-Division Multiplexing
Wavelength-Division Multiplexing
Synchronous Time-Division Multiplexing
Statistical Time-Division Multiplexing

Figure 6.1 *Dividing a link into channels*

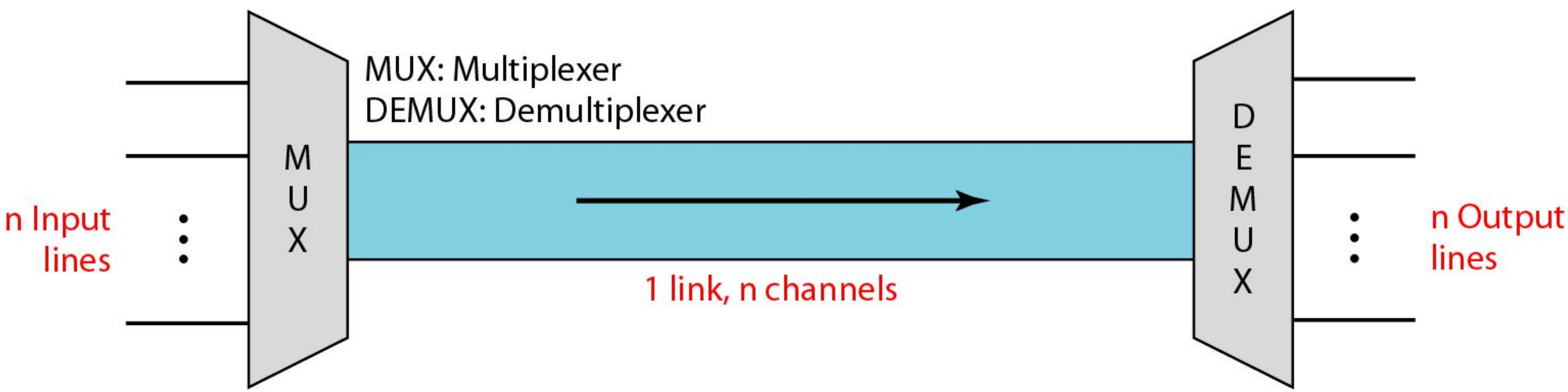


Figure 6.2 *Categories of multiplexing*

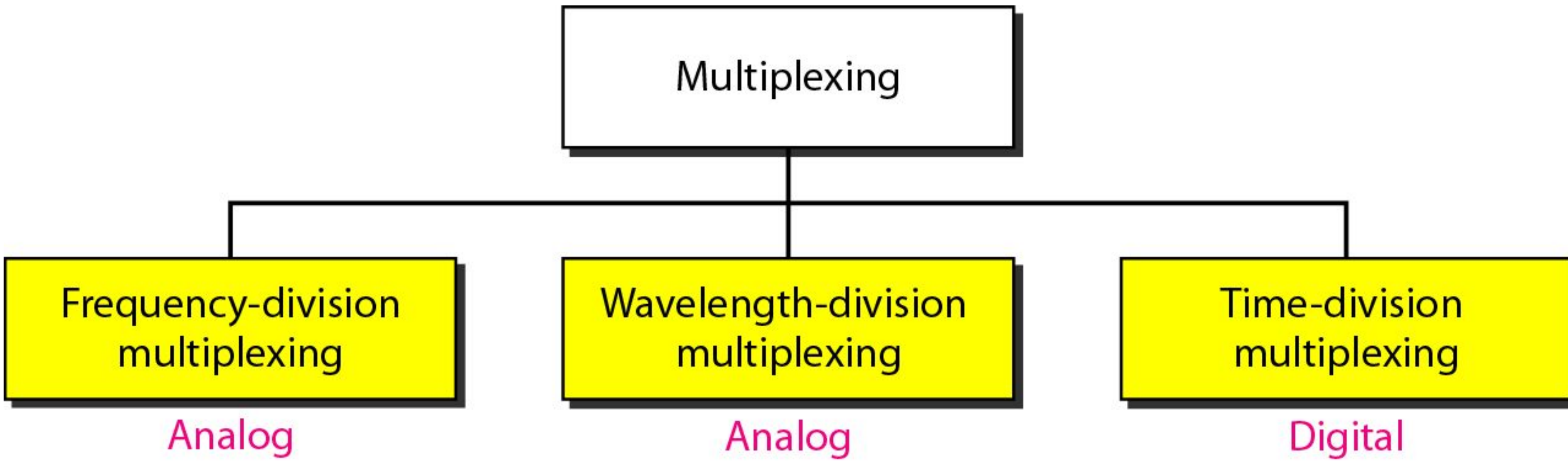


Figure 6.3 *Frequency-division multiplexing*

Note

FDM is an analog multiplexing technique that combines analog signals.

Figure 6.4 *FDM process*

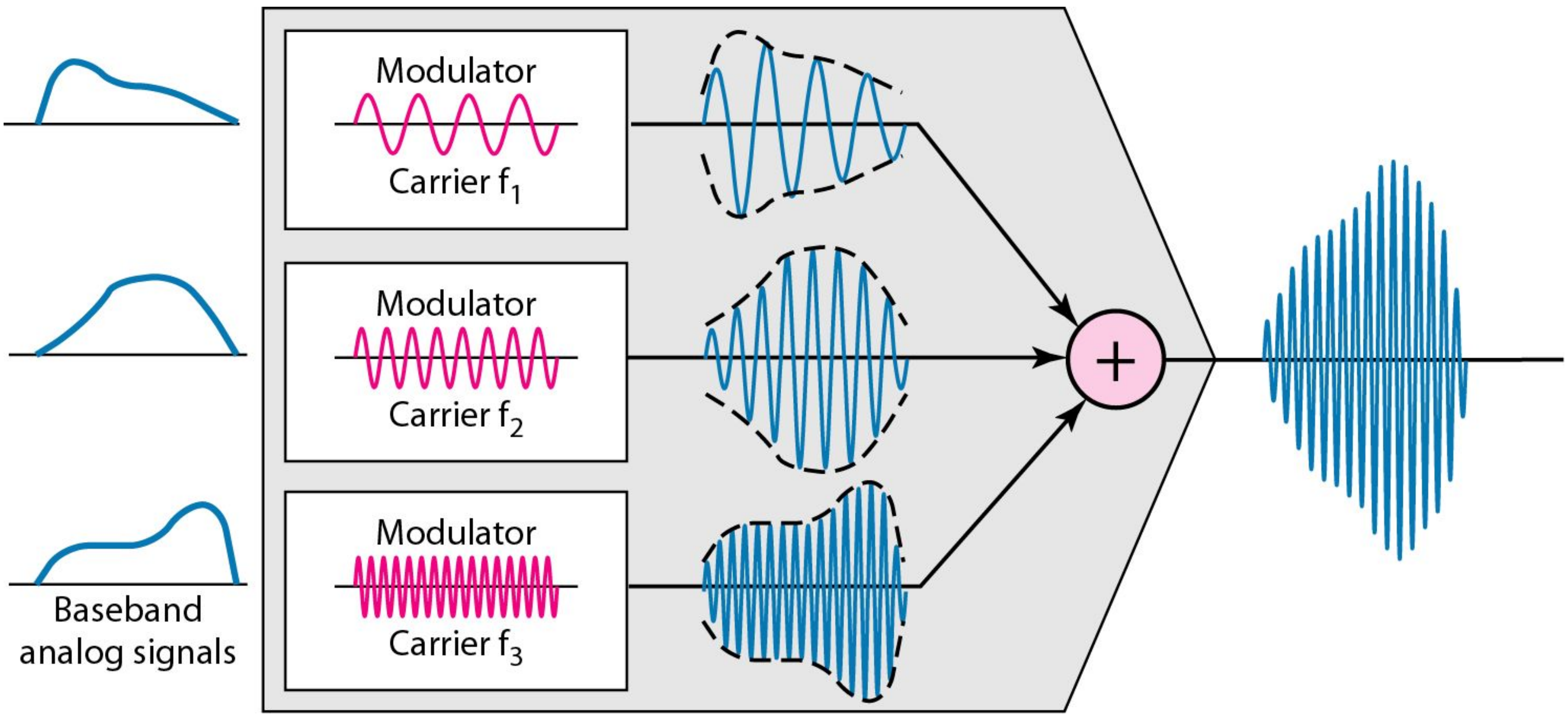
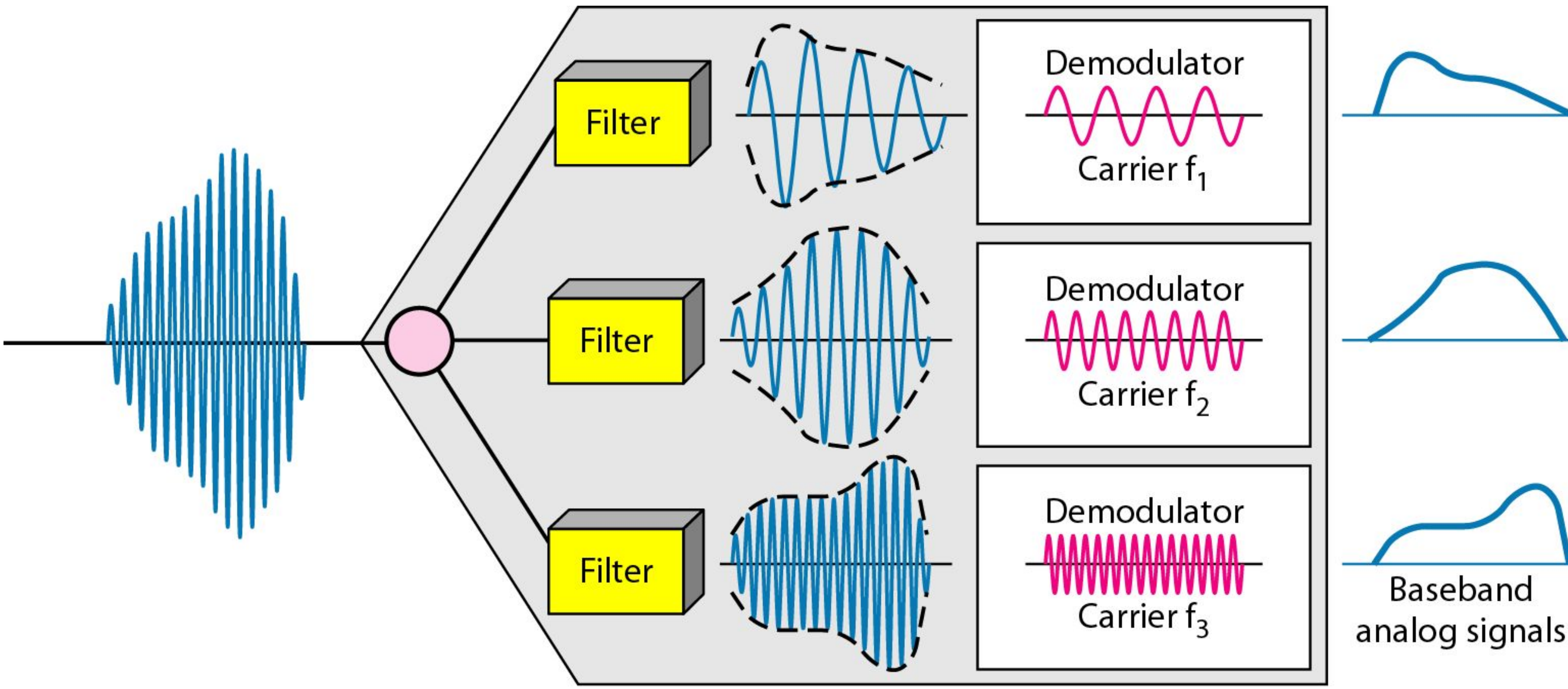


Figure 6.5 *FDM demultiplexing example*



Example 6.1

Assume that a voice channel occupies a bandwidth of 4 kHz. We need to combine three voice channels into a link with a bandwidth of 12 kHz, from 20 to 32 kHz. Show the configuration, using the frequency domain. Assume there are no guard bands.

Solution

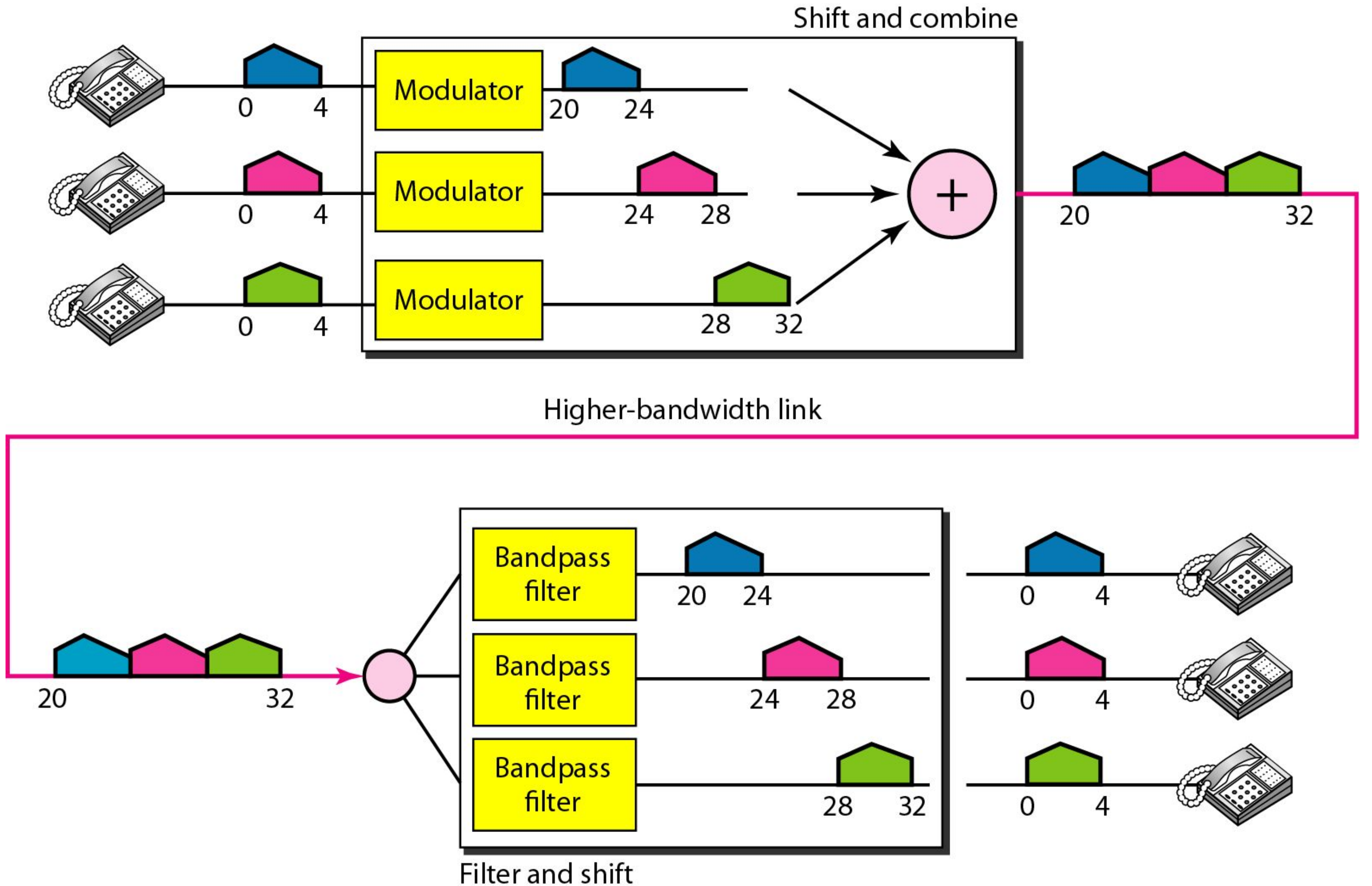
Example 6.1

Assume that a voice channel occupies a bandwidth of 4 kHz. We need to combine three voice channels into a link with a bandwidth of 12 kHz, from 20 to 32 kHz. Show the configuration, using the frequency domain. Assume there are no guard bands.

Solution

We shift (modulate) each of the three voice channels to a different bandwidth, as shown in Figure 6.6. We use the 20- to 24-kHz bandwidth for the first channel, the 24- to 28-kHz bandwidth for the second channel, and the 28- to 32-kHz bandwidth for the third one. Then we combine them as shown in Figure 6.6.

Figure 6.6 *Example 6.1*



Example 6.2

Five channels, each with a 100-kHz bandwidth, are to be multiplexed together. What is the minimum bandwidth of the link if there is a need for a guard band of 10 kHz between the channels to prevent interference?

Solution :

Example 6.2

Five channels, each with a 100-kHz bandwidth, are to be multiplexed together. What is the minimum bandwidth of the link if there is a need for a guard band of 10 kHz between the channels to prevent interference?

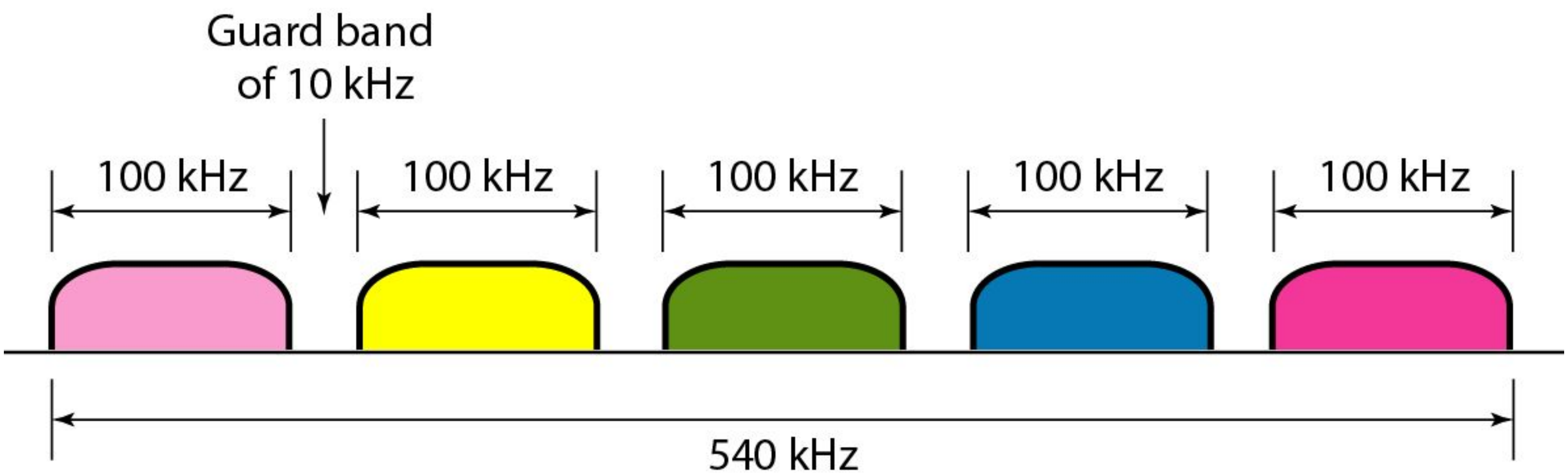
Solution

For five channels, we need at least four guard bands. This means that the required bandwidth is at least

$$5 \times 100 + 4 \times 10 = 540 \text{ kHz},$$

as shown in Figure 6.7.

Figure 6.7 *Example 6.2*



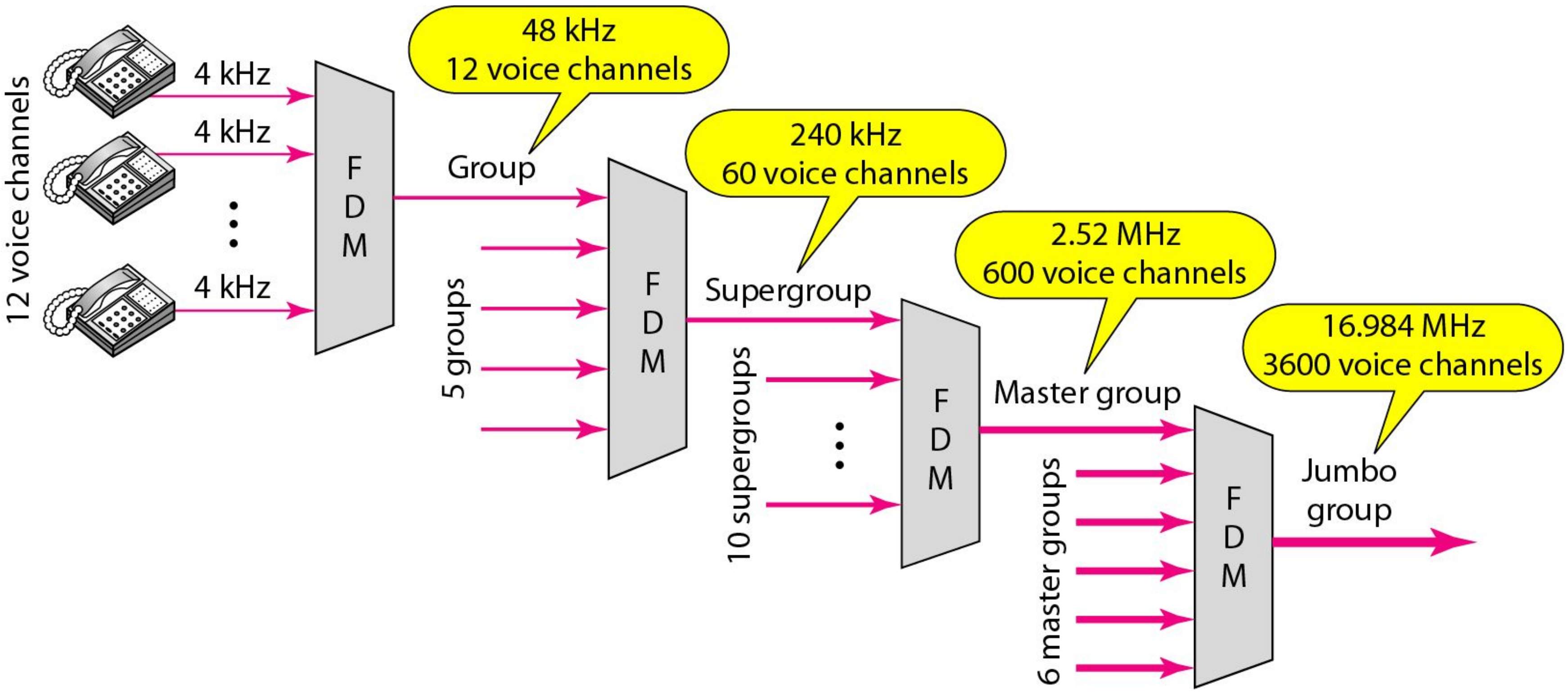
The Analog Carrier System

- To maximize the efficiency of their infrastructure, telephone companies have traditionally multiplexed signals from lower-bandwidth lines onto higher-bandwidth lines.
- In this way, many switched or leased lines can be combined into fewer but bigger channels.
- For analog lines, FDM is used.
- One of these hierarchical systems used by telephone companies is made up of groups, supergroups, master groups, and jumbo groups (see Figure 6.9)

The Analog Carrier System

- **Analog hierarchy,** 12 voice channels are multiplexed onto a higher-bandwidth line to create a **group.**
- A group has 48 kHz of bandwidth and supports 12 voice channels.
- At the next level, up to five **groups** can be multiplexed to create a composite signal called a **supergroup.**
- A **supergroup** has a bandwidth of 240 kHz and supports up to 60 voice channels.
- **Supergroups** can be made up of either five groups or 60 independent voice channels.
- At the next level, 10 supergroups are multiplexed to create a **master group.**
- A master group must have 2.40 MHz of bandwidth, but the need for guard bands between the supergroups increases the necessary bandwidth to 2.52 MHz.
- Master groups support up to 600 voice channels.
- Finally, six master groups can be combined into a **jumbo group.** A jumbo group must have 15.12 MHz (6 × 2.52 MHz) but is augmented to 16.984 MHz to allow for guard bands between the master groups.

Figure 6.9 *Analog hierarchy*



Wavelength-Division Multiplexing (WDM)

Note

WDM is an analog multiplexing technique to combine optical signals.

Figure 6.10 ***Wavelength-division multiplexing***

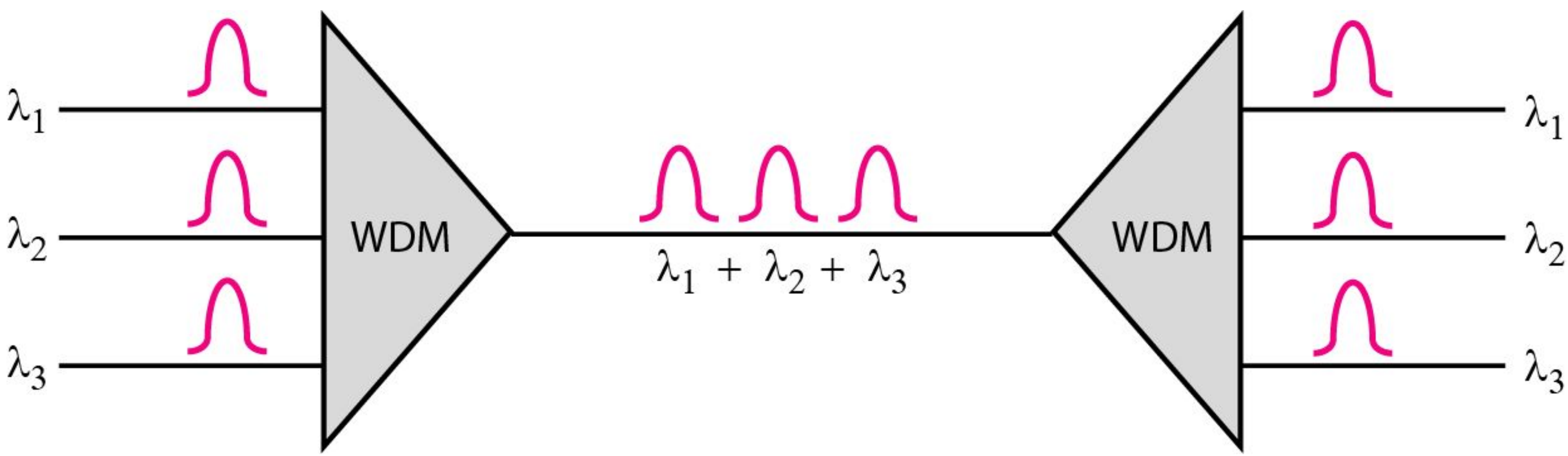


Figure 6.11 *Prisms in wavelength-division multiplexing and demultiplexing*

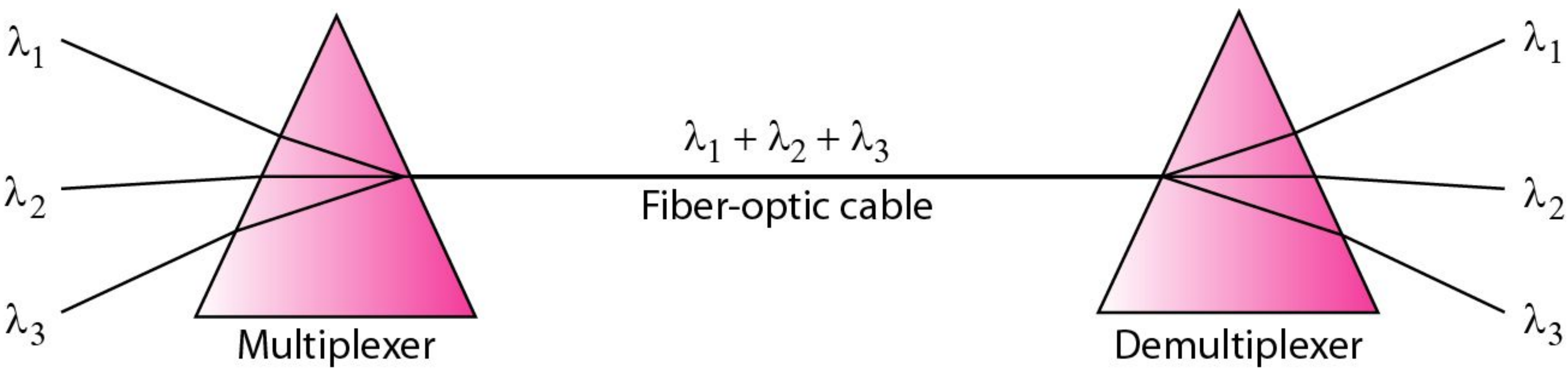
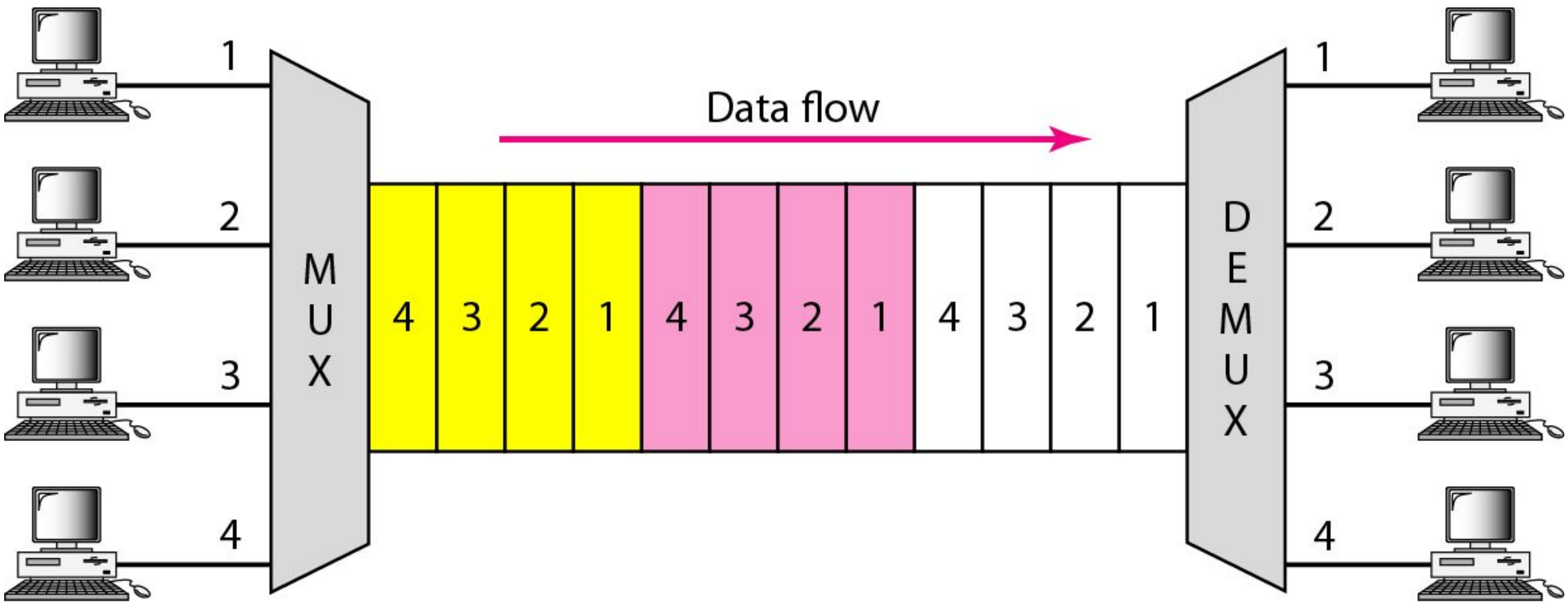


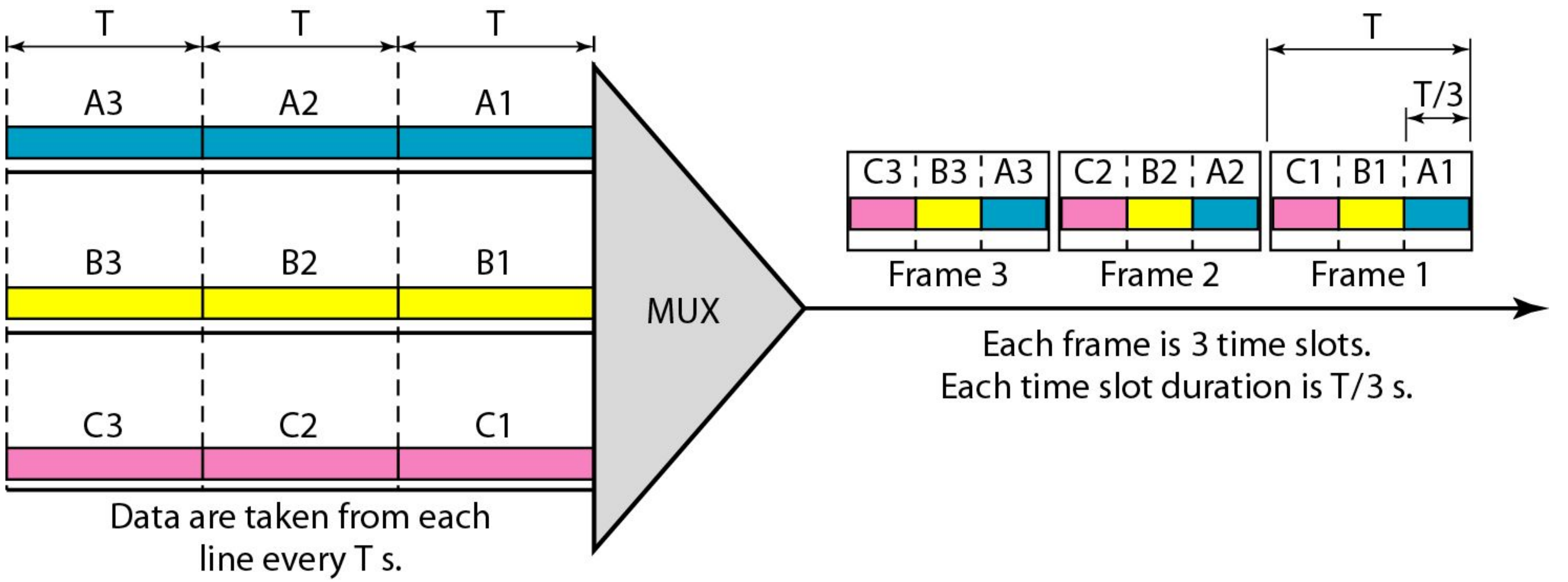
Figure 6.12 Time-Division Multiplexing (*TDM*)



Note

TDM is a digital multiplexing technique for combining several low-rate channels into one high-rate one.

Figure 6.13 *Synchronous time-division multiplexing*



Note

In synchronous TDM, the data rate of the link is *n* times faster, and the unit duration is *n* times shorter.

Example 6.5

In Figure 6.13, the data rate for each input connection is 1 kbps. If 1 bit at a time is multiplexed (a unit is 1 bit), what is the duration of (a) each input slot, (b) each output slot, and (c) each frame?

Solution

We can answer the questions as follows:

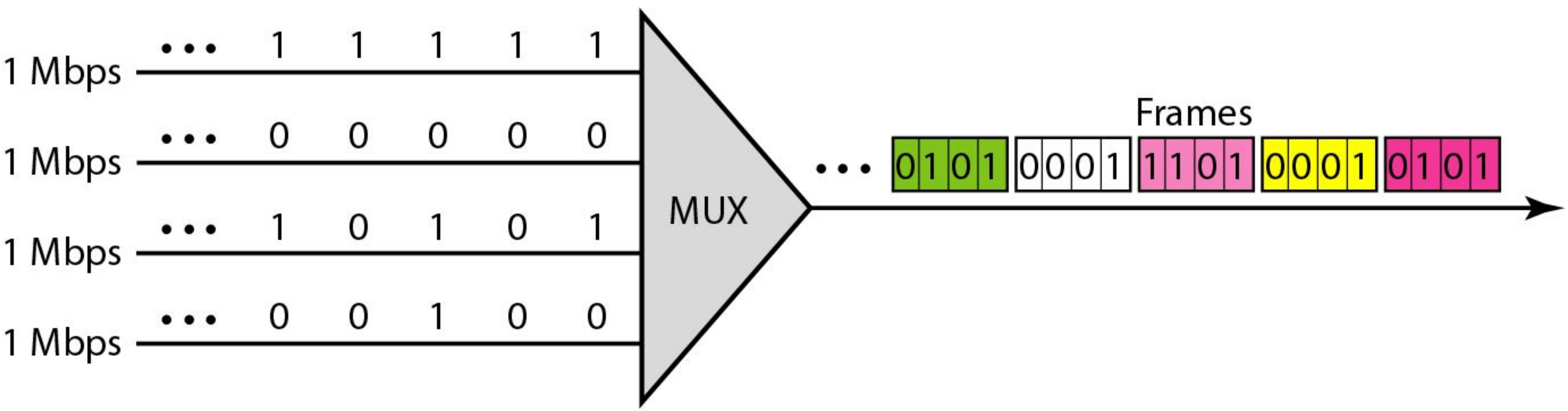
a. The data rate of each input connection is 1 kbps. This means that the bit duration is 1/1000 s or 1 ms. The duration of the input time slot is 1 ms (same as bit duration).

Example 6.5 (continued)

b. The duration of each output time slot is one-third of the input time slot. This means that the duration of the output time slot is 1/3 ms.

c. Each frame carries three output time slots. So the duration of a frame is 3 × 1/3 ms, or 1 ms. The duration of a frame is the same as the duration of an input unit.

Figure 6.14 *Example 6.6*



Example 6.6

Figure 6.14 shows synchronous TDM with a data stream for each input and one data stream for the output. The unit of data is 1 bit. Find (a) the input bit duration, (b) the output bit duration, (c) the output bit rate, and (d) the output frame rate.

Solution

We can answer the questions as follows:

a. The input bit duration is the inverse of the bit rate: 1/1 Mbps = 1 μs.

b. The output bit duration is one-fourth of the input bit duration, or ¼ μs.

Example 6.6 (continued)

c. ***The output bit rate is the inverse of the output bit duration or 1/(4μs) or 4 Mbps. This can also be deduced from the fact that the output rate is 4 times as fast as any input rate; so the output rate = 4 × 1 Mbps = 4 Mbps.***

d. ***The frame rate is always the same as any input rate. So the frame rate is 1,000,000 frames per second. Because we are sending 4 bits in each frame, we can verify the result of the previous question by multiplying the frame rate by the number of bits per frame.***

Example 6.7

Four 1-kbps connections are multiplexed together. A unit is 1 bit. Find (a) the duration of 1 bit before multiplexing, (b) the transmission rate of the link, (c) the duration of a time slot, and (d) the duration of a frame.

Solution

We can answer the questions as follows:

a. The duration of 1 bit before multiplexing is 1 / 1 kbps, or 0.001 s (1 ms).

b. The rate of the link is 4 times the rate of a connection, or 4 kbps.

Example 6.7 (continued)

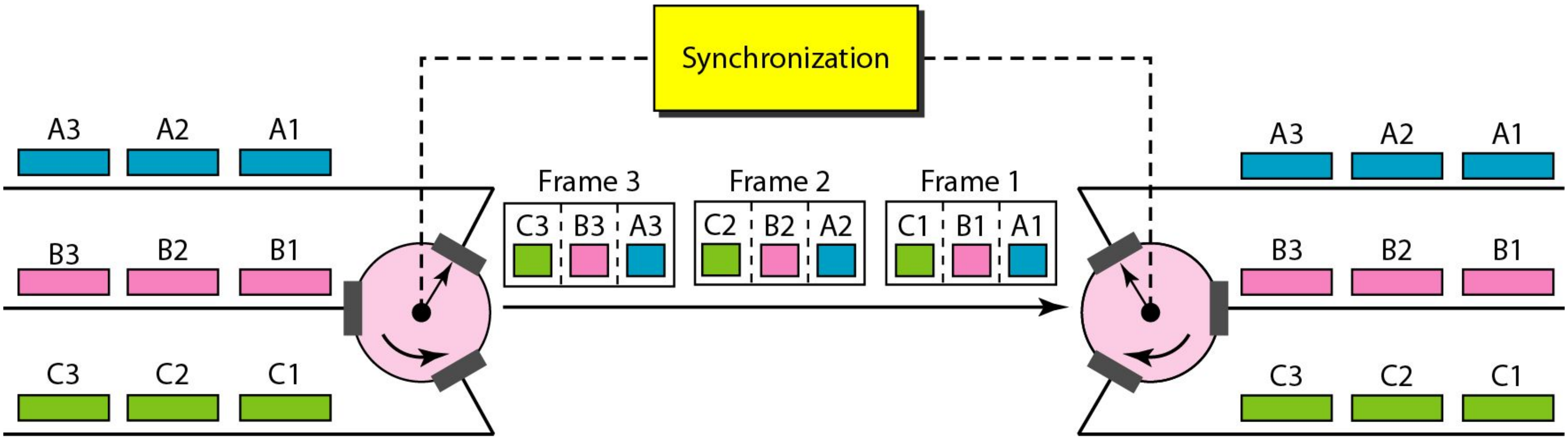
c. ***The duration of each time slot is one-fourth of the duration of each bit before multiplexing, or 1/4 ms or 250 μs. Note that we can also calculate this from the data rate of the link, 4 kbps. The bit duration is the inverse of the data rate, or 1/4 kbps or 250 μs.***

d. ***The duration of a frame is always the same as the duration of a unit before multiplexing, or 1 ms. We can also calculate this in another way. Each frame in this case has four time slots. So the duration of a frame is 4 times 250 μs, or 1 ms.***

Interleaving

- TDM can be visualized as two **fast-rotating switches**, one on the multiplexing side and the other on the demultiplexing side.
- The **switches** are synchronized and **rotate at the same speed**, but in opposite directions.
- On the multiplexing side, as the switch opens in front of a connection, that connection has the opportunity to send a unit onto the path. This process is called **interleaving.**
- On the demultiplexing side, as the switch opens in front of a connection, that connection has the opportunity to receive a unit from the path.

Figure 6.15 *Interleaving*



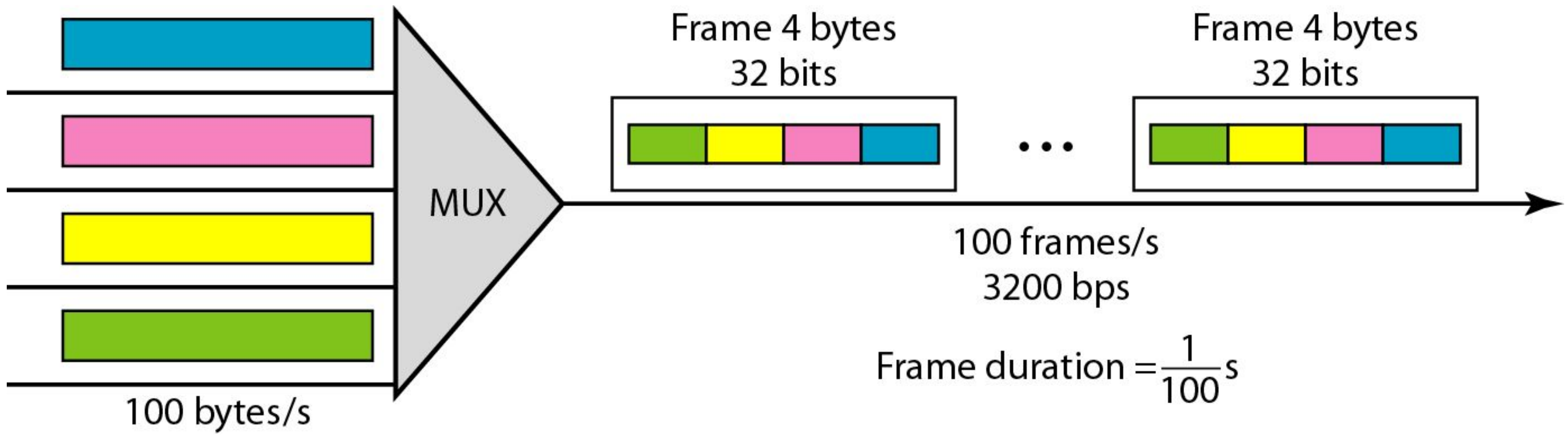
Example 6.8

Four channels are multiplexed using TDM. If each channel sends 100 bytes /s and we multiplex 1 byte per channel, show the frame traveling on the link, the size of the frame, the duration of a frame, the frame rate, and the bit rate for the link.

Solution

The multiplexer is shown in Figure 6.16. Each frame carries 1 byte from each channel; the size of each frame, therefore, is 4 bytes, or 32 bits. Because each channel is sending 100 bytes/s and a frame carries 1 byte from each channel, the frame rate must be 100 frames per second. The bit rate is 100 × 32, or 3200 bps.

Figure 6.16 *Example 6.8*



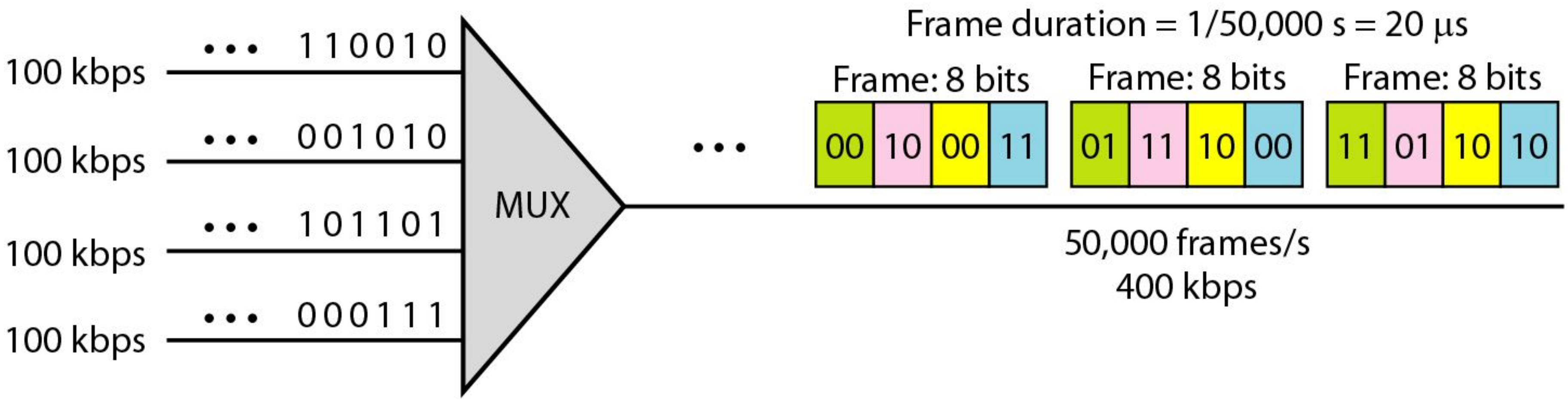
Example 6.9

A multiplexer combines four 100-kbps channels using a time slot of 2 bits. Show the output with four arbitrary inputs. What is the frame rate? What is the frame duration? What is the bit rate? What is the bit duration?

Solution

Figure 6.17 shows the output for four arbitrary inputs. The link carries 50,000 frames per second. The frame duration is therefore 1/50,000 s or 20 μs. The frame rate is 50,000 frames per second, and each frame carries 8 bits; the bit rate is 50,000 × 8 = 400,000 bits or 400 kbps. The bit duration is 1/400,000 s, or 2.5 μs.

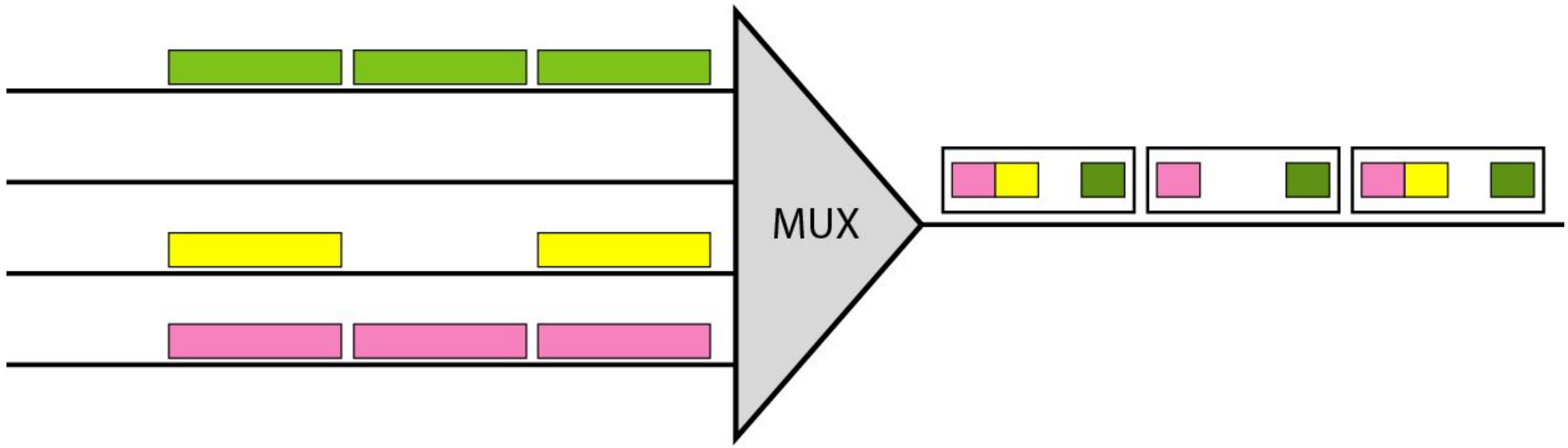
Figure 6.17 *Example 6.9*



Empty Slots & Statistical TDM

- Synchronous TDM is not as efficient as it could be.
- If a source does not have data to send, the corresponding slot in the output frame is empty.
- See next figure, the first output frame has three slots filled, the second frame has two slots filled, and the third frame has three slots filled.
- No frame is full.
- Problem.
- Solution?
- Statistical TDM can improve the efficiency by removing the empty slots from the frame.

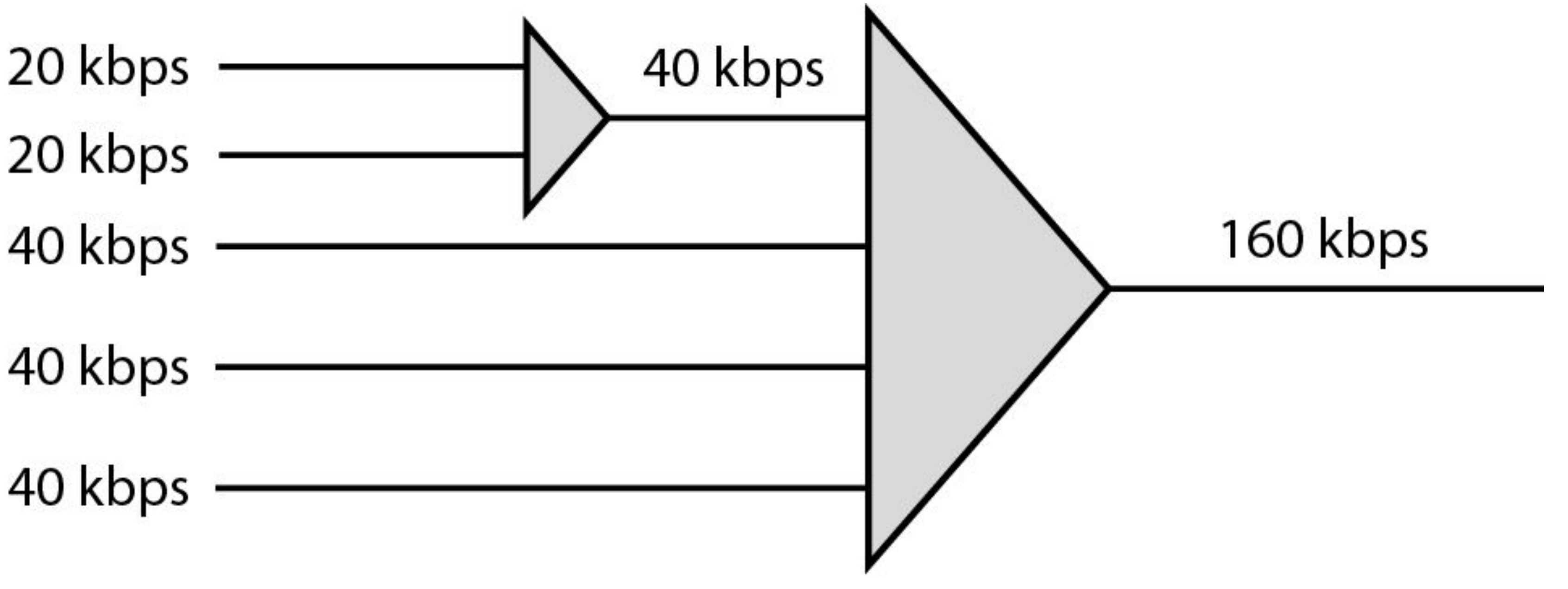
Figure 6.18 *Empty slots*



Multilevel Multiplexing

- Multilevel multiplexing is a technique used when the data rate of an input line is a multiple of others.
- For example, in Figure 6.19, we have two inputs of 20 kbps and three inputs of 40 kbps.
- The first two input lines can be multiplexed together to provide a data rate equal to the last three.
- A second level of multiplexing can create an output of 160 kbps.

Figure 6.19 *Multilevel multiplexing*



Multiple-Slot Allocation

- Sometimes it is more efficient to allot more than one slot in a frame to a single input line.
- For example, we might have an input line that has a data rate that is a multiple of another input.
- We insert a **demultiplexer** in the line to make two inputs out of one.

Figure 6.20 *Multiple-slot multiplexing*

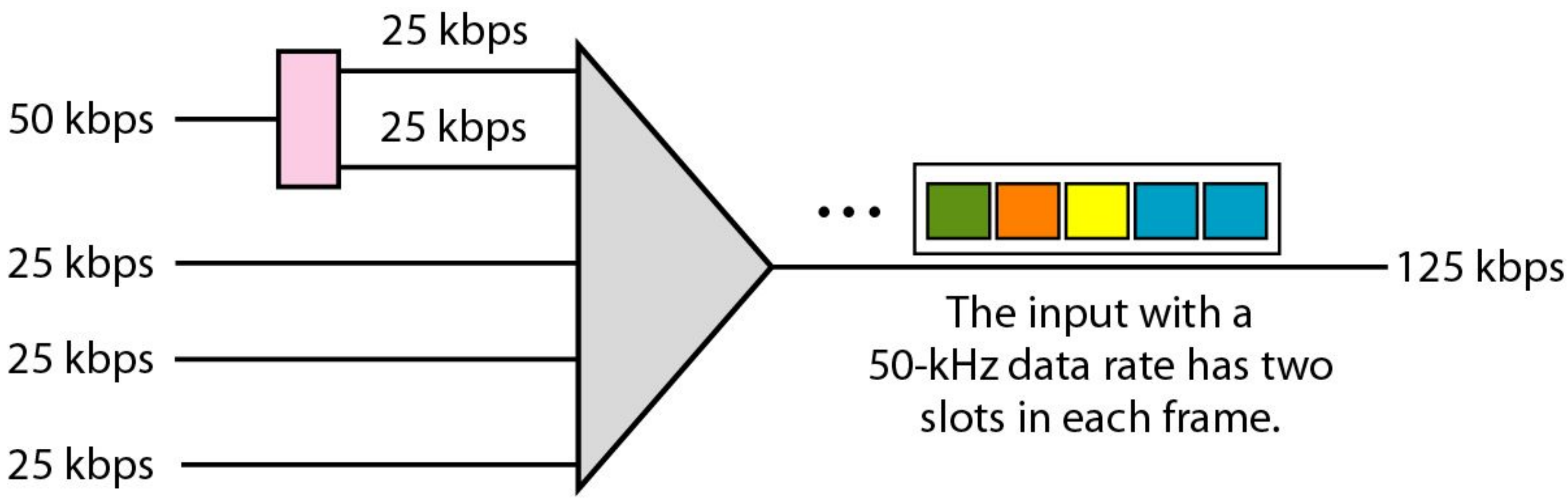
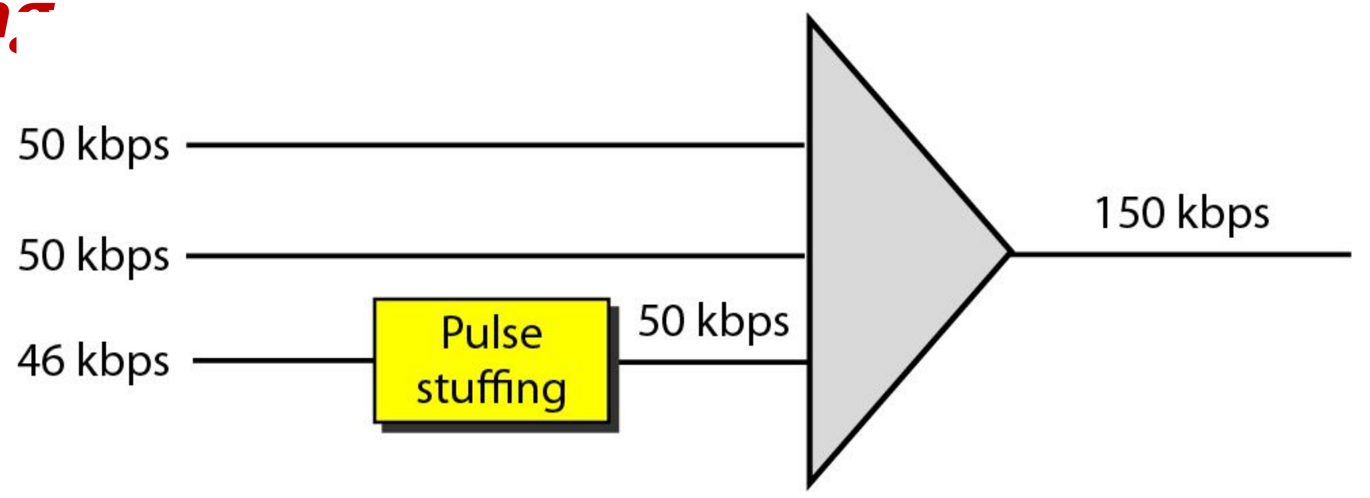
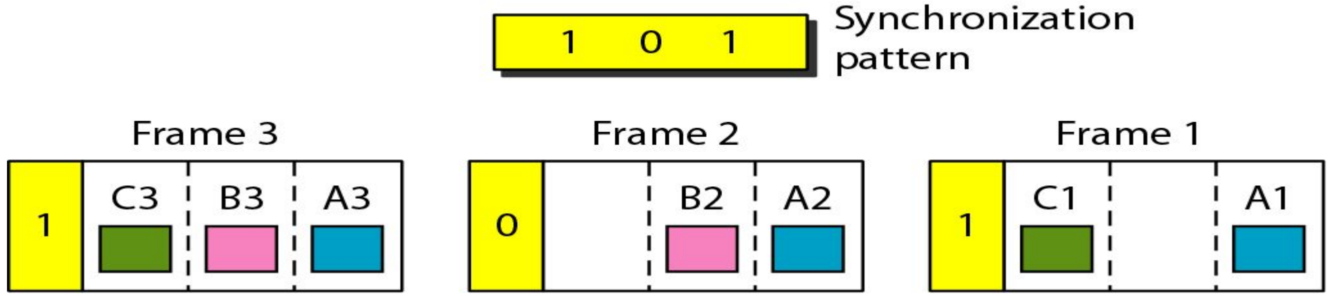


Figure 6.21 *Pulse stuffing*



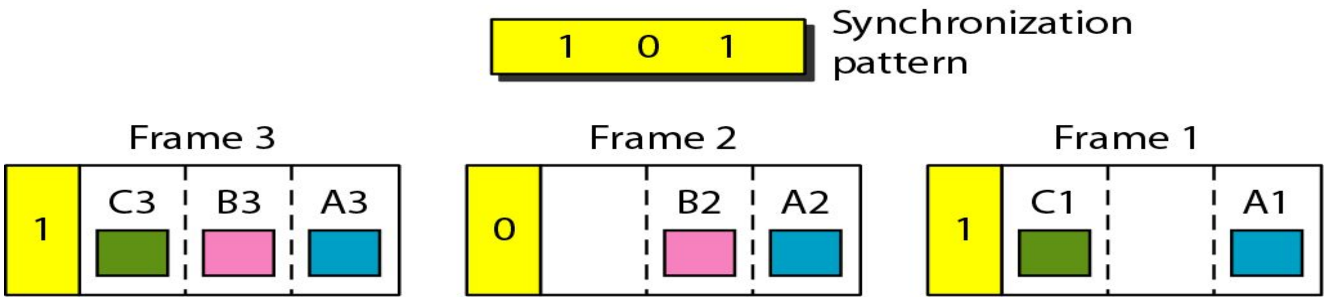
- Sometimes the bit rates of sources are not multiple integers of each other. Therefore, neither of the above two techniques can be applied.
- One solution is to make the highest input data rate the dominant data rate and then **add dummy bits to the** input lines with lower rates. This will increase their rates.
- This technique is called ***pulse stuffing,*** *bit padding,* or *bit stuffing.*

Figure 6.22 *Framing bits*



- The implementation of TDM is not as simple as that of FDM.
- Synchronization between the multiplexer and demultiplexer is a major issue.
- If the multiplexer and the demultiplexer are not synchronized, a bit belonging to one channel may be received by the wrong channel.
- For this reason, one or more synchronization bits are usually added to the beginning of each frame.

Figure 6.22 *Framing bits*



- These bits, called **framing bits,** follow a pattern, frame to frame, that allows the demultiplexer to synchronize with the incoming stream so that it can separate the time slots accurately.
- In most cases, this synchronization information consists of 1 bit per frame, **alternating between 0 and 1 .**

Example 6.10

We have four sources, each creating 250 characters per second. If the interleaved unit is a character and 1 synchronizing bit is added to each frame, find (a) the data rate of each source, (b) the duration of each character in each source, (c) the frame rate, (d) the duration of each frame, (e) the number of bits in each frame, and (f) the data rate of the link.

Solution

We can answer the questions as follows:

a. The data rate of each source is 250 × 8 = 2000 bps = 2 kbps.

Example 6.10 (continued)

b. ***Each source sends 250 characters per second; therefore, the duration of a character is 1/250 s, or 4 ms.***

c. ***Each frame has one character from each source, which means the link needs to send 250 frames per second to keep the transmission rate of each source.***

d. ***The duration of each frame is 1/250 s, or 4 ms. Note that the duration of each frame is the same as the duration of each character coming from each source.***

e. ***Each frame carries 4 characters and 1 extra synchronizing bit. This means that each frame is $4 \times 8 + 1 = 33$ bits.***

Example 6.11

Two channels, one with a bit rate of 100 kbps and another with a bit rate of 200 kbps, are to be multiplexed. How this can be achieved? What is the frame rate? What is the frame duration? What is the bit rate of the link?

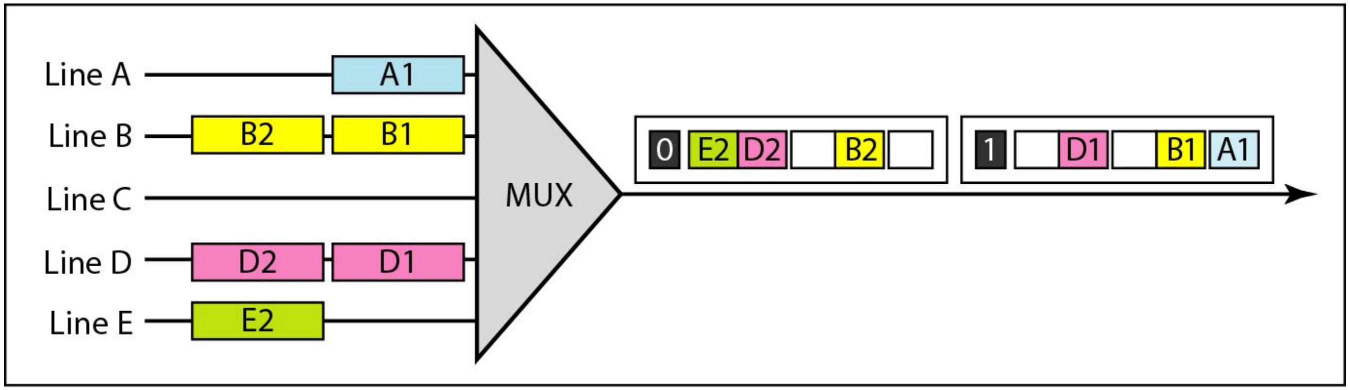
Solution

We can allocate one slot to the first channel and two slots to the second channel. Each frame carries 3 bits. The frame rate is 100,000 frames per second because it carries 1 bit from the first channel. The bit rate is 100,000 frames/s × 3 bits per frame, or 300 kbps.

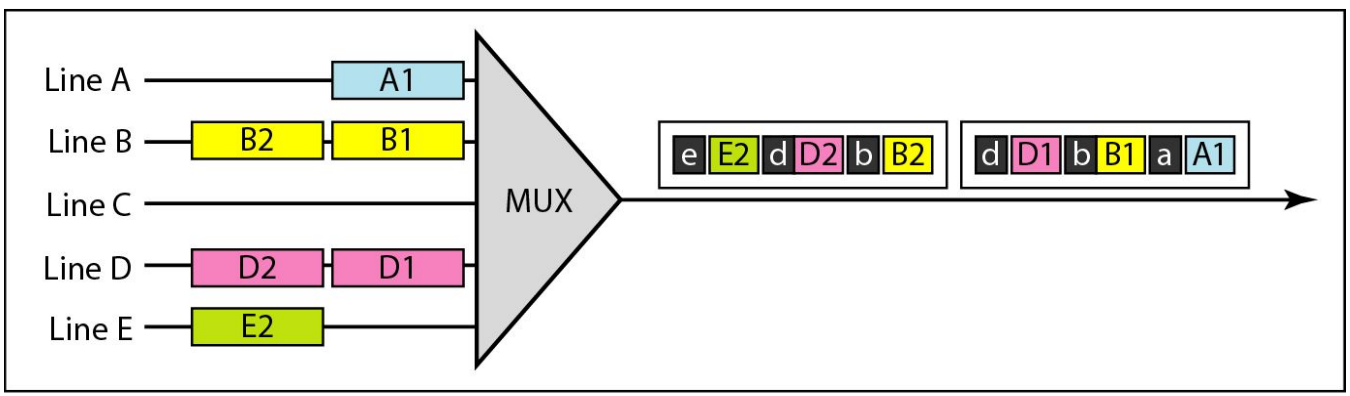
Statistical Time-Division Multiplexing

- As we saw in the previous section, in synchronous TDM, each input has a reserved slot in the output frame.
- This can be inefficient if some input lines have no data to send.
- In statistical time-division multiplexing, slots are dynamically allocated to improve bandwidth efficiency.
- Figure 6.26 shows a synchronous and a statistical TDM example.

Figure 6.26 *TDM slot comparison*



a. Synchronous TDM



b. Statistical TDM

Definitions

Addressing

- In synchronous TDM, there is no need for addressing; synchronization and pre-assigned relationships between the inputs and outputs serve as an address.
- In statistical multiplexing, there is no fixed relationship between the inputs and outputs because there are no preassigned or reserved slots.
- The addressing in its simplest form can be n bits to define N different output lines with $n = \log2 N$.
- For example, for eight different output lines, we need a 3-bit address.

Definitions

Slot Size

- Since a slot carries both data and an address in statistical TDM, the ratio of the data size to address size must be reasonable to make transmission efficient.
- For example, it would be inefficient to send 1 bit per slot as data when the address is 3 bits.
- This would mean an overhead of 300 percent.
- In statistical TDM, a block of data is usually many bytes while the address is just a few bytes.

Definitions

No Synchronization Bit

- There is another difference between synchronous and statistical TDM, but this time it is at the frame level.
- The frames in statistical TDM need not be synchronized, so we do not need synchronization bits.

Definitions

Bandwidth

- In statistical TDM, the capacity of the link is normally less than the sum of the capacities of each channel.
- The designers of statistical TDM define the capacity of the link based on the statistics of the load for each channel.
- If on average only *x* percent of the input slots are filled, the capacity of the link reflects this. Of course, during peak times, some slots need to wait.