BACHELOR OF COMPUTER APPLICATIONS

SEMESTER 3

DCA2104

BASICS OF DATA COMMUNICATION

Unit 8

Digital Data Communication Techniques

Table of Contents

1. INTRODUCTION

This unit introduces the concept of digital data communication techniques. The transmission of a stream of bits from one device to another require synchronization. The receiver should know the rate at which bits are received so that it can sample the line at appropriate intervals to determine the value of each received bit. Asynchronous transmission and synchronous transmission techniques are used for this purpose. Error detection is performed by calculating an error detecting code that is a function of the bits being transmitted. Errors in the transmitted bit stream is corrected using error correction methods.

In this unit, we will discuss synchronous and asynchronous transmission. Data must be transmitted one bit at a time and these timings should be same for transmitter and receiver. Synchronous and asynchronous are two common techniques for controlling this timing. In the next section we will discuss the problem of bit errors. After the discussion of different type of errors, we will discuss error detection and error correction techniques.

1.1 Objectives

After studying this unit, you should be able to:

- *describe asynchronous and synchronous transmission*
- *explain types of errors*
- *describe error detection*
- *explain error correction*
- *describe line configuration*

2. ASYNCHRONOUS AND SYNCHRONOUS TRANSMISSION

During serial transmission, signaling elements are sent down the line one at a time. Reception of the digital data involves sampling the incoming signal once per bit time to determine the binary value. One of the difficulties here is that various transmission impairments will corrupt the signal so that there is a chance of errors. Receiver must know the arrival time and duration of each bit that it receives in order to sample incoming bits properly.

Asynchronous Transmission

In order to achieve desired synchronization, two approaches are commonly used. They are asynchronous transmission and synchronous transmission. In asynchronous transmission, the strategy used is to avoid timing problem by not sending long, uninterrupted streams of bits. Instead, data are transmitted one character at a time, where each character is five to eight bits in length. Timing and synchronization must only be maintained within each character; the receiver has the opportunity to resynchronize at the beginning of each new character.

Figure 8.1 illustrates asynchronous transmission. We can see in figure that when no character is being transmitted, the line between transmitter and receiver is in an idle state. In case of NRZ-L (non-return-to-zero-level) signaling, which is common for asynchronous transmission, idle would be the presence of a negative voltage on the line. The beginning of a character is signaled by a *start bit* with a value of binary 0. This is followed by the 5 to 8 bits that actually make up the character. The bits of the character are transmitted beginning with the least significant bit. The data bits are followed by a parity bit, which therefore is in the most significant bit position. The parity bit is set by the transmitter such that the total number of ones in the character, including the parity bit, is either even (even parity) or odd (odd parity), depending on the convention being used. The receiver uses this bit for error detection. The final element is a stop element, which is a binary 1. A minimum length for the *stop element* is specified, and this is usually 1, 1.5, or 2 times the duration of an ordinary bit. No maximum value is specified. Because the stop element is the same as the idle state, the transmitter will continue to transmit the stop element until it is ready to send the next character. The timing requirements for this scheme are modest. For example, suppose we sent characters as 8 bits units as shown in figure 8.1, If the receiver is 5% slower or faster than the transmitter, the sampling of the eighth character bit will be displaced by 45% and still be correctly sampled.

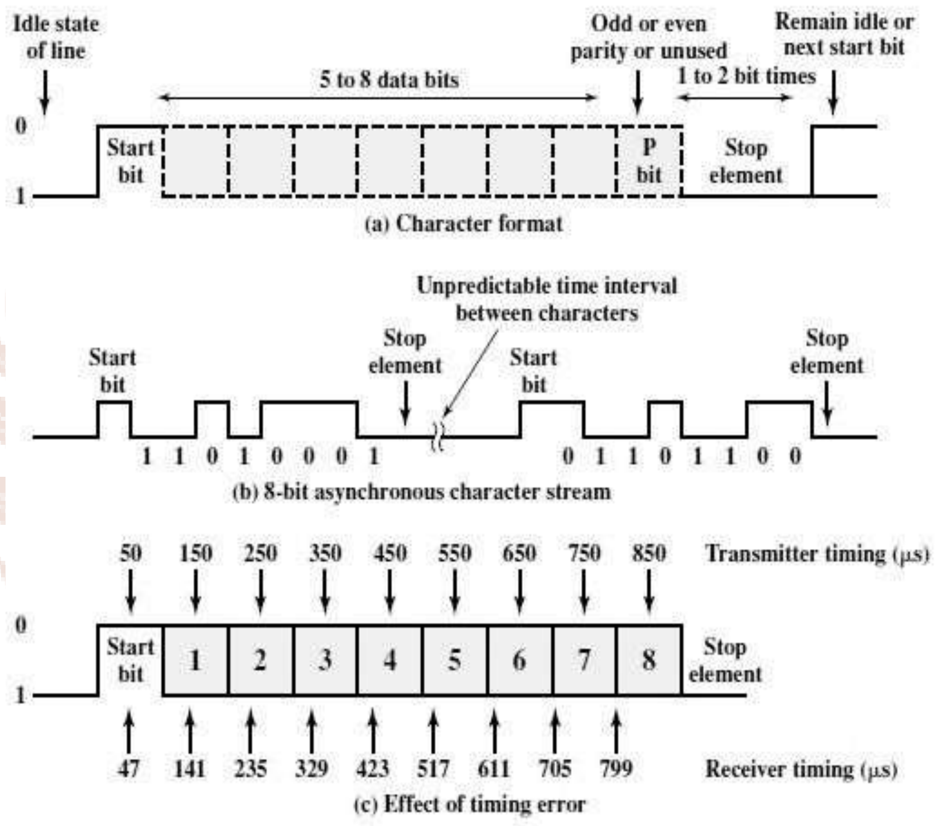


Figure 8.1: Asynchronous transmission

Figure 8.1 (c) shows the effects of a timing error of sufficient magnitude to cause an error in reception. In this example we assume a data rate of 10,000 bits per second (10 kbps); therefore, each bit is of 0.1 millisecond (ms), or 100 μs duration. Assume that the receiver is fast by 6%, or 6 μs per bit time. Thus, the receiver samples the incoming character every 94 μs (based on the transmitter's clock). We can see, the last sample is erroneous.

The error which we described results in two errors. First, the last sampled bit is incorrectly received. Second, the bit count may now be out of alignment. If bit 7 is a 1 and bit 8 is a 0, bit 8 could be mistaken for a start bit. This condition is termed a *framing error*, as the character plus start bit and stop element are sometimes referred to as a frame. A framing error can also occur if some noise condition causes the false appearance of a start bit during the idle state.

Asynchronous transmission is simple and cheap. The drawback of this transmission is that, it can transmit small block of data at a time. For example, while transmitting an 8-bit character, two out of every ten bits is used for synchronization only. Thus the overhead is 20%. We can reduce the percentage overhead by sending larger blocks of bits between the start bit and stop element. But the problem in that case is the larger the block of bits, the greater the cumulative timing error. So in order to achieve greater efficiency, a different form of synchronization known as *synchronous transmission* is used.

Synchronous Transmission

By using synchronous transmission, we can transmit large block of bits in a steady stream without start and stop codes. The block may be many bits in length. To prevent timing change between transmitter and receiver, their clocks must be synchronized. One option is to provide a separate clock line between transmitter and receiver. Either transmitter or receiver pulses the line regularly with one short pulse per bit time. Other side uses these regular pulses as a clock. Over short distances, this technique works well. But for long distances, the clock pulses are subject to the same impairments as the data signal, and timing errors can occur.

Another alternative is to insert the clocking information in the data signal. For digital signals, this can be accomplished with Manchester or differential Manchester encoding. For analog signals, a number of techniques can be used. For instance, the carrier frequency itself can be used to synchronize the receiver based on the phase of the carrier. In case of synchronous transmission, there is another level of synchronization required to allow the receiver to determine the beginning and end of a block of data. To achieve this, each block begins with a *preamble* bit pattern and ends with a *postamble* bit pattern. Also, other bits are added to the block that convey control information used in the data link control procedures. The data plus preamble, postamble, and control information are called a **frame**. The exact format of the frame depends on which data link control procedure is being used.

Figure 8.2 shows the typical frame format for synchronous transmission. The frame starts with a preamble called a flag, which is 8 bits long. The same flag is used as a postamble. The receiver looks for the occurrence of the flag pattern to signal the start of a frame. This is followed by some number of control field, then a data field, more control fields and finally the flag is repeated.

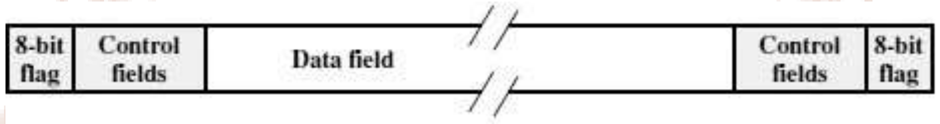


Figure 8.2: synchronous frame format

For sizable blocks of data, synchronous transmission is far more efficient than asynchronous. Asynchronous transmission requires 20% or more overhead. The control information, preamble, and postamble in synchronous transmission are typically less than 100 bits.

Self-Assessment Questions - 1

1. In ______________ transmission, data are transmitted one character at a time.
2. _________________ and ____________ are two common techniques for controlling this timing.
3. Asynchronous transmission can transmit small block of data at a time. State true or false. (a) True (b) False
4. We can transmit large block of bits in a steady stream without start and stop codes using ____________ transmission.

3. TYPES OF ERRORS

In digital transmission systems, an error occurs when a bit is altered between transmission and reception. That means, a binary 1 is transmitted and a binary 0 is received, or a binary 0 is transmitted and a binary 1 is received. Two general types of errors can occur. They are, *single-bit errors* and *burst errors*. A single-bit error is an isolated error condition that alters one bit but does not affect nearby bits. A burst error of length B is a contiguous sequence of B bits in which the first and last bits and any number of intermediate bits are received in error.

Based on IEEE Std 100 and ITU-T recommendation Q.9 an error burst is defined as *a group of bits in which two successive erroneous bits are always separated by less than a given number x of correct bits. The last erroneous bit in the burst and the first erroneous bit in the following burst are accordingly separated by x correct bits or more.*

Thus, in an error burst, there is a cluster of bits in which a number of errors occur, although not necessarily all of the bits in the cluster suffer an error. A single-bit error can occur in the presence of white noise, when a slight random deterioration of the signal-to-noise ratio is sufficient to confuse the receiver's decision of a single bit. Burst errors are more common and more difficult to deal with. Burst errors can be caused by impulse noise (unwanted noise generally created by electromagnetic interference, scratches on the recording disks etc.). Another cause is fading (deviation of the weakening affecting a signal over a transmission media) in a mobile wireless environment.

Note that the effects of burst errors are greater at higher data rates. For example, an impulse noise event or a fading event of 1μs occurs. At a data rate of 10 Mbps, there is a resulting error burst of 10 bits. At a data rate of 100 Mbps, there is an error burst of 100 bits.

Self-Assessment Questions - 2

5. Error condition that alters one bit but does not affect nearby bits is known as_________ .
6. __________errors can be caused by impulse noise.

4. ERROR DETECTION

There is a chance of errors to occur during transmission regardless of the design of the transmission system. We assume that data is transmitted as continuous sequence of bits called frames. We can define following probabilities with respect to errors in transmitted frames.

- P_b is the probability that a bit is received in error; also known as the bit error rate (BER)
- P_1 is the probability that a frame arrives with no bit errors
- P_2 is the probability that with an error-detecting algorithm in use, a frame arrives with one or more undetected errors
- P_3 is the probability that, with an error-detecting algorithm in use, a frame arrives with one or more detected bit errors but no undetected bit errors

If there is no measures to detect errors, then the probability of detected errors (P_3) is zero. To express the remaining probabilities, assume the probability that any bit is in error (P_b) is constant and independent for each bit. Then we have,

$$P_1 = (1-P_b)^F$$

$$P_2 = 1-P_1$$

Where F is the number of bits per frame. The probability that a frame arrives with no bit errors decreases when the probability of a single bit error increases. Also, the probability that a frame arrives with no bit errors decreases with increasing frame length. The longer the frame, the more bits it has and the higher the probability that one of these is in error. Figure 8.3 shows error detection process.

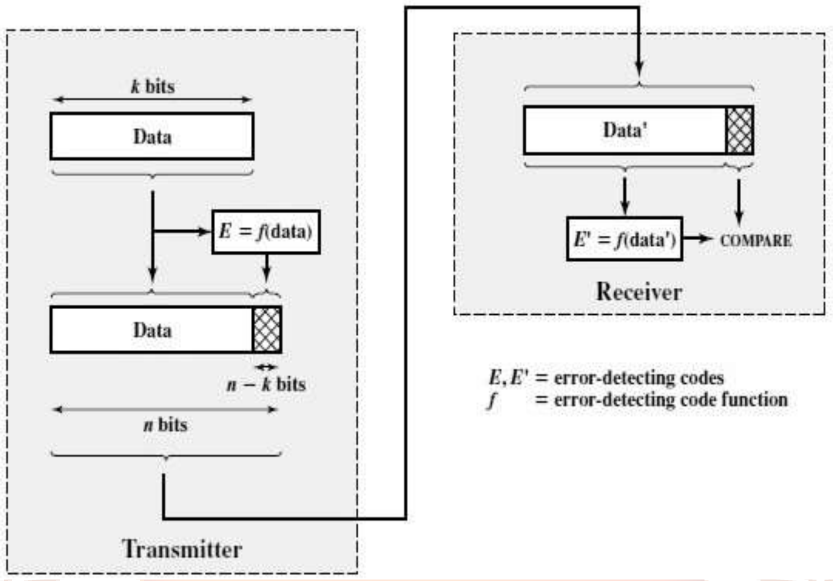


Figure 8.3: Error detection process

As shown in figure 8.3, for a given frame of bits, additional bits that constitute an **error-detecting code** are added by the transmitter. This code is calculated as a function of the other transmitted bits. For a data block of k bits, the error-detecting algorithm yields an error-detecting code of n-k bits where (n-k) < k. The error-detecting code, also referred to as the **check bits**, is appended to the data block to produce a frame of n bits, which is then transmitted. The receiver separates the incoming frame into the k bits of data and n-k bits of the error-detecting code. The receiver performs the same error-detecting calculation on the data bits and compares this value with the value of the incoming error-detecting code. A detected error occurs if and only if there is a mismatch. So P_3 is the probability that a frame contains errors and that the error-detecting scheme will detect that errors. P_2 is known as the residual error rate and is the probability that an error will be undetected regardless of the use of an error-detecting scheme.

Parity Check

The simplest error-detecting scheme is to append a parity bit to the end of a block of data. A typical example is character transmission, in which a parity bit is attached to each 7-bit data. The value of this parity bit is selected such a way that the character has an even number of 1s (even parity) or an odd number of 1s (odd parity).

If the transmitter is transmitting data bits 1110001 and using odd parity, it will append a 1 and transmit 11110001. The receiver examines the received character and, if the total number of 1s is odd, assumes that no error has occurred. If one bit (or any odd number of bits) is erroneously inverted during transmission (for example, 11100001), then the receiver will detect an error. If two (or any even number) of bits are inverted due to error, an undetected error occurs. Typically, even parity is used for synchronous transmission and odd parity for asynchronous transmission.

Cyclic Redundancy Check (CRC)

Cyclic redundancy check is one of the most common, powerful error detecting code. The idea behind this is that, suppose there is a k bit block of bits or message. The transmitter generates an (n-k) bit sequence, known as a frame check sequence (FCS) such that the resulting frame, consisting of *n* bits, is exactly divisible by some predetermined number. The receiver then divides the incoming frame by that number and, if there is no remainder, assumes there was no error.

In order to make it more clearly, we present the procedure in three equivalent ways. They are modulo 2 arithmetic, polynomials and digital logic.

Modulo 2 Arithmetic

Modulo 2 arithmetic uses binary addition with no carries, which is just the exclusive-OR (XOR) operation. Binary subtraction with no carries is also interpreted as the XOR operation: For example,

```
  1111        1111
 +1010       - 0101
 -----       ------
  0101         1010
```

Suppose,

T = n bit frame to be transmitted,

D =k bit block of data, or message, the first k bits of T

F = (n-k) bit FCS, the last (n-k) bits of T

P = pattern of n – k + 1 bits; this is the predetermined divisor

We would like T/P to have no remainder. It should be clear that

$$T = 2^{n-k} D + F$$

That is, by multiplying D by 2^{n-k}, we have in effect shifted it to the left by n–k bits and padded out the result with zeroes. Adding F yields the concatenation of D and F which is T. We want T to be exactly divisible by P. suppose that we divide 2^{n-k} D by P

$$\frac{2^{n-k}D}{P} = Q + \frac{R}{P}$$

... (8.1)

There is a quotient and a remainder. Because division is modulo 2, the remainder is always at least one bit shorter than the divisor. We will use this remainder as our FCS. Then

$$T = 2^{n-k} D + R \quad \textbf{.... (8.2)}$$

Consider,

$$\frac{T}{P} = \frac{2^{n-k}D + R}{P} = \frac{2^{n-k}D}{P} + \frac{R}{P}$$

Substituting equation 8.1, we have

$$\frac{T}{P} = Q + \frac{R}{P} + \frac{R}{P}$$

However, any binary number added to itself modulo 2 yields zero. Thus

$$\frac{T}{P} = Q + \frac{R + R}{P} = Q$$

There is no remainder, and therefore *T* is exactly divisible by *P*. Thus, the FCS is easily generated: Simply divide 2^{n-k}D by *P* and use the (n-k) bit remainder as the FCS. On reception, the receiver will divide *T* by *P* and will get no remainder if there have been no errors.

Polynomials

A second way of viewing the CRC process is to express all values as polynomials in a dummy variable X, with binary coefficients. The coefficients correspond to the bits in the binary number. Thus, for D = 110011, we have D(X) = $X^5 + X^4 + X + 1$ and for P=11001, we have P(X) = $X^4 + X^3 + 1$. Arithmetic operations are again modulo 2. The CRC process can now be described as

$$\frac{X^{n-k}D(X)}{P(X)} = Q(X) + \frac{R(X)}{P(X)}$$
$$T(X) = X^{n-k}D(X) + R(X)$$

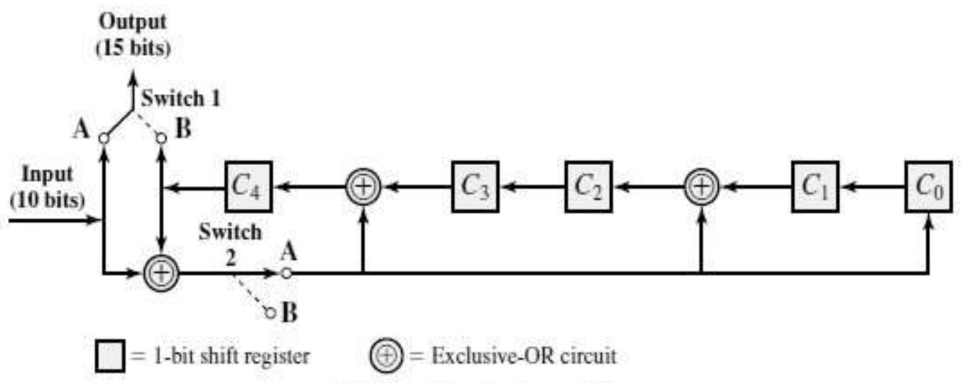
Compare these equations with equations 8.1 and 8.2.

An error E(X) will only be undetectable if it is divisible by *P*(X). It can be shown that all of the following errors are not divisible by a suitably chosen *P*(X) and hence are detectable.

- All single-bit errors, if P(X) has more than one nonzero term
- All double-bit errors, as long as P(X) is a special type of polynomial, called a primitive polynomial, with maximum exponent L, and the frame length is less than $2^L - 1$.
- Any odd number of errors, as long as P(X) contains a factor (X+1)
- Any burst error for which the length of the burst is less than or equal to n-k, that is, less than or equal to the length of the FCS
- A fraction of error bursts of length n-k+1, the fraction equals $1-2^{-(n-k-1)}$
- A fraction of error bursts of length greater than n-k+1; the fraction equals $1-2^{-(n-k)}$

Digital Logic

The CRC process can be implemented as, a dividing circuit consisting of XOR gates and a shift register. The shift register is a string of 1-bit storage devices. Each device has an output line, which indicates the value currently stored, and an input line. At discrete time instants known as clock times, the value in the storage device is replaced by the value indicated by its input line. The entire register is clocked simultaneously, causing a 1-bit shift along the entire register. Figure 8.4 shows shift register implementation.



(a) Shift-register implementation

	C_4	C_3	C_2	C_1	C_0	$C_4 \oplus C_3 \oplus I$	$C_4 \oplus C_1 \oplus I$	$C_4 \oplus I$	I = input	
Initial	0	0	0	0	0	1	1	1	1	Message to be sent
Step 1	1	0	1	0	1	1	1	1	0	
Step 2	1	1	1	1	1	1	1	0	1	
Step 3	1	1	1	1	0	0	0	1	0	
Step 4	0	1	0	0	1	1	0	0	0	
Step 5	1	0	0	1	0	1	0	1	0	
Step 6	1	0	0	0	1	0	0	0	1	
Step 7	0	0	0	1	0	1	0	1	1	
Step 8	1	0	0	0	1	1	1	1	0	
Step 9	1	0	1	1	1	0	1	0	1	
Step 10	0	1	1	1	0					

(b) Example with input of 1010001101

Figure 8.4: Circuit with shift registers for dividing by the polynomial $X^5 + X^4 + X^2 + 1$

The circuit is implemented as follows

- The register contains n-k bits, equal to the length of the FCS
- There are up to n-k XOR gates
- The presence or absence of a gate corresponds to the presence or absence of a term in the divisor polynomial P(X), excluding the terms 1 and X^{n-k}.

Self-Assessment Questions - 3

7. The error detecting code is also known as_________ .
8. Typically,_____________ parity is used for synchronous transmission and ___________parity for asynchronous transmission.
9. Cyclic redundancy check can be explained in three different ways. They are__________ , _____________ and ____________ .

5. ERROR CORRECTION

The techniques that we have discussed so far can detect errors, but do not correct them. Error Correction can be handled in two ways. One is when an error is discovered; the receiver can ask the sender retransmit the entire data unit. This is known as backward error correction. Second method is that, receiver can use an error-correcting code, which automatically corrects certain errors. This is known as forward error correction.

Error-correcting codes are more advanced than error detecting codes and require more redundant bits. The number of bits required to correct multiple- bit or burst error is so high that in most of the cases, it is inefficient to do so. For this reason, most error correction is limited to one, two or at the most three-bit errors.

Single-bit error correction

Concept of error-correction can be easily understood by examining the simplest case of single-bit errors. As we have already seen that a single-bit error can be detected by addition of a parity bit (VRC) with the data, which needed to be send. A single additional bit can detect error, but it's not sufficient to correct that error too. For correcting an error one has to know the exact position of error, i.e. exactly which bit is in error (to locate the invalid bits). For example, to correct a single-bit error in an ASCII character, the error correction must determine which one of the seven bits is in error. To this, we have to add some additional redundant bits. If a frame consist of d data bits and r redundant bits, then the resulting n=d+r bit unit is called a codeword. The number of bits by which two codewords differ is called hamming distance. So we have (d+r) as the total number of bits, which are to be transmitted; then r must be able to indicate at least d+r+1 different values. Of these, one value means no error, and remaining d+r values indicate error location of error in each of d+r locations. So, d+r+1 states must be distinguishable by r bits, and r bits can indicates 2r states. Hence, 2r must be greater than d+r+1.

$$2^r >= d+r+1$$

The value of r must be determined by putting in the value of d in the relation. For example, if d is 7, then the smallest value of r that satisfies the above relation is 4. So the total bits, which are to be transmitted is 11 bits (d+r = 7+4 =11). Now let us examine how we can manipulate these bits to discover which bit is in error. A technique developed by R.W.Hamming provides a practical solution. The solution or coding scheme he developed is commonly

known as Hamming Code. Hamming code can be applied to data units of any length and uses the relationship between the data bits and redundant bits as discussed. Figure 8.8 shows positions of redundancy bits in hamming code.

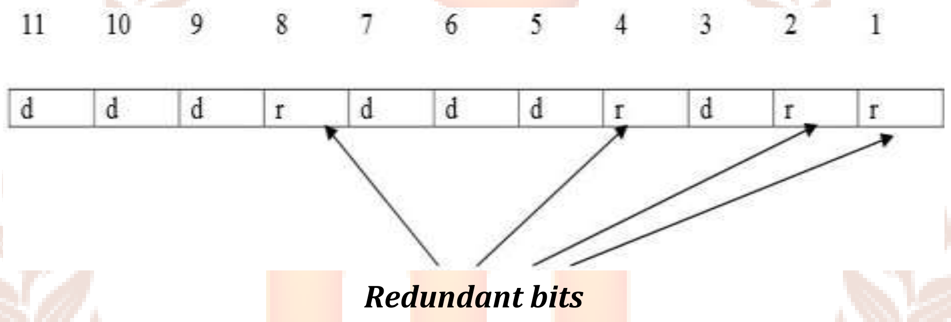
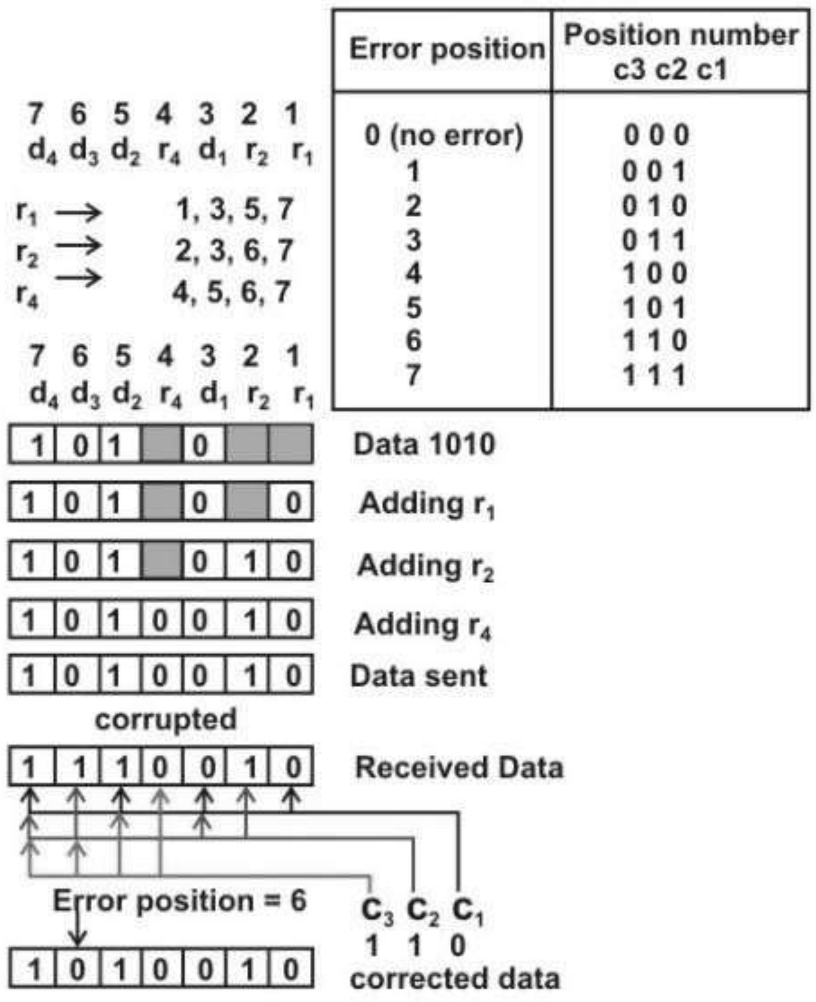


Figure 8.8: positions of redundancy bits in hamming code

Basic approach for error detection by using hamming code is as follows.

- To each group of m information bits, K parity bits are added to form (m+k) bit code as shown in figure 8.8.
- Location of each of the (m+k) digits is assigned a decimal value
- The K parity bits are replaced in positions 1, 2, …., 2^{k-1} positions. K parity checks are performed on selected digits of each codeword.
- At the receiving end, the parity bits are recalculated. The decimal value of the k parity bits provides the bit position in error, if any.

Figure 8.9 shows the use of hamming code for error detection for a 4-bit data.



Error position	Position number c3 c2 c1
0 (no error)	0 0 0
1	0 0 1
2	0 1 0
3	0 1 1
4	1 0 0
5	1 0 1
6	1 1 0
7	1 1 1

Figure 8.9: Use of hamming code for error correction for a 4-bit data

Figure 8.9 shows how hamming code is used for correction for 4-bit numbers ($d_4d_3d_2d1$) with the help of three redundant bits ($r_3r_2r_1$). For example, data 1010, first r_1 (0) is calculated considering the parity of the bit positions, 1, 3, 5 and 7. Then the parity bits r_2 is calculated considering bit positions 2, 3, 6 and 7. Finally, the parity bits r_4 is calculated considering bit positions 4, 5, 6 and 7 as shown. If any corruption occurs in any of the transmitted code 1010010, the bit position in error can be found out by calculating $r_3r_2r_1$ at the receiving end. For example, if the received code word is 1110010, the recalculated value of $r_3r_2r_1$ is 110, which indicates that bit position in error is 6, the decimal value of 110.

Self-Assessment Questions - 4

10. Two methods of error correction are__________ and __________ .
11. An error correcting technique developed by R.W.Hamming is known as_________.
12. If a frame consist of d data bits and r redundant bits, then the resulting n=d+r bit unit is called ______________

6. LINE CONFIGURATION

Two characteristics that distinguish different data link configurations are topology and whether the link is half duplex or full duplex.

Topology

The topology of a data link refers to the physical arrangement of stations on a transmission medium. If there are only two stations (a terminal and a computer or two computers), the link is point to point. If there are more than two stations, then it is a multipoint topology. Traditionally, a multipoint link has been used in the case of a computer (primary station) and a set of terminals (secondary stations). In today's environments, the multipoint topology is found in local area networks.

Traditional multipoint topologies are made possible when the terminals are only transmitting a fraction of the time. Figure 8.10 illustrates the advantages of the multipoint configuration. If each terminal has a point-to- point link to its computer, then the computer must have one I/O port for each terminal. Also there is a separate transmission line from the computer to each terminal. In a multipoint configuration, the computer needs only a single I/O port and a single transmission line, which saves costs.

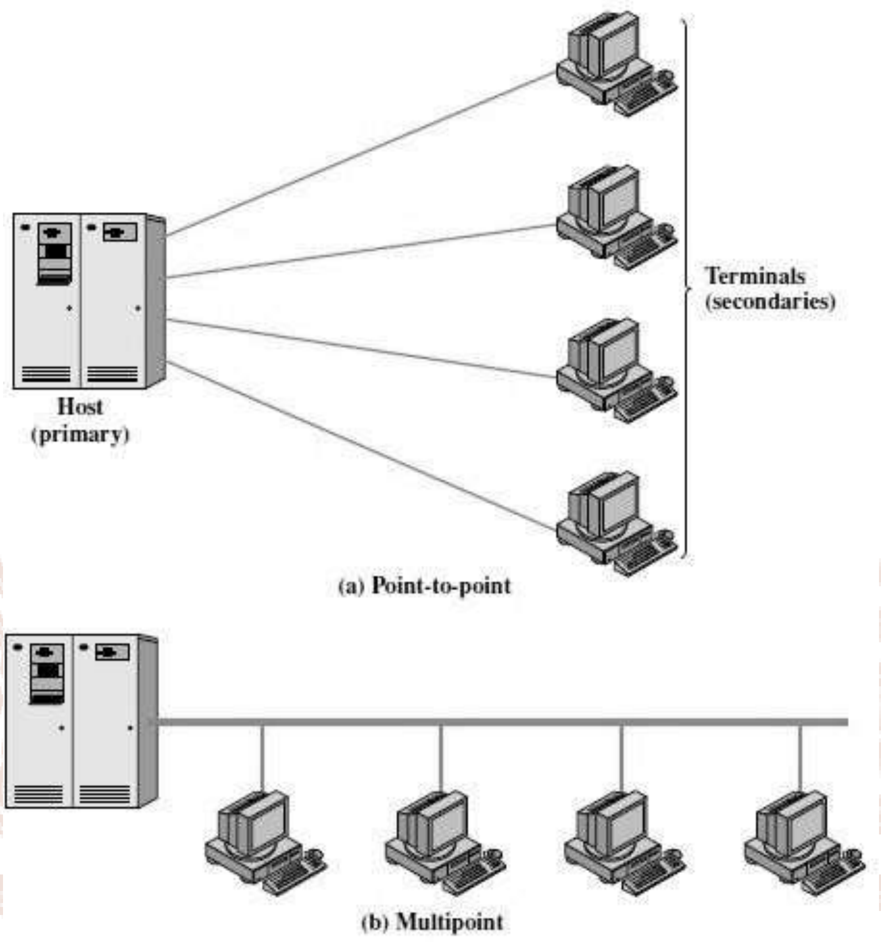


Figure 8.10: Traditional computer or terminal configurations

Full Duplex and Half Duplex

Data exchanges over a transmission line can be classified as full duplex or half duplex. With *half-duplex transmission*, only one of two stations on a point-to-point link may transmit at a time. This mode is also referred to as *two-way alternate*, considering the fact that two stations must alternate in transmitting. This can be compared to a one-lane, two-way bridge. This form of transmission is often used for terminal-to-computer interaction. While a user is entering and transmitting data, the computer is prevented from sending data to the terminal, which would appear on the terminal screen and cause confusion. With digital signaling, which requires guided transmission, full-duplex operation usually requires two separate transmission paths (e.g., two twisted pairs), while half duplex requires only one. For analog signaling, transmission path depends on frequency. If a station transmits and receives on the same frequency, it must operate in half-duplex mode for wireless transmission. Even though stations operate in full-duplex mode for guided transmission using two separate transmission lines. If a station transmits on one frequency and receives on another, it may operate in full-duplex mode for wireless transmission and in full-duplex mode with a single line for guided transmission. For full-duplex transmission, two stations can simultaneously send and receive data from each other. This mode is known as two-way simultaneous and may be compared to a *two-lane, two-way bridge*. For computer-to-computer data exchange, this form of transmission is more efficient than half-duplex transmission.

Self-Assessment Questions - 5

13. The physical arrangement of stations on a transmission medium is known as________.
14. Data exchanges over a transmission line can be classified as __________ or____________.
15. For _________ transmission, two stations can simultaneously send and receive data from each other

7. SUMMARY

Let us recapitulate the important concepts discussed in this unit:

- Asynchronous and synchronous transmission techniques are used for digital data communication to know the rate at which bits are received.
- In asynchronous transmission, data are transmitted one character at a time, where each character is five to eight bits in length.
- Asynchronous transmission is simple and cheap.
- By using synchronous transmission, we can transmit large block of bits in a steady stream without start and stop codes.
- The data plus preamble, postamble, and control information are called a frame.
- Two general types of errors are single-bit errors and burst errors.
- The simplest error-detecting scheme is to append a parity bit to the end of a block of data.
- Cyclic redundancy check is one of the most common, powerful error detecting code.
- Cyclic redundancy check procedure is defined in three ways. They are modulo 2 arithmetic, polynomials and digital logic.
- Error Correction can be handled in two ways. They are backward error correction and forward error correction.
- The topology of a data link refers to the physical arrangement of stations on a transmission medium.
- Data exchanges over a transmission line can be classified as full duplex or half duplex.

8. TERMINAL QUESTIONS

1. Differentiate between synchronous and asynchronous transmission.
2. Explain different types of errors.
3. Analyse Error-detecting processand Cyclic redundancy check.
4. Explain Error Correction and analyse types Error correction mechanisms.
5. Explain data link configuration in topology and analyse half duplex and full duplex.
6. How the Data exchanges in transmission line and explain different types of classifications?

9. ANSWERS

Self-Assessment Questions

1. Asynchronous transmission
2. Synchronous and asynchronous
3. (a) True
4. Synchronous transmission
5. Single bit error
6. Burst
7. Check bits
8. Even, odd
9. Modulo 2 arithmetic, polynomials, digital logic
10. Forward error correction, backward error correction
11. Hamming code
12. Code word
13. Topology
14. Full duplex or half duplex
15. Full duplex

Terminal Questions

1. In asynchronous transmission, the strategy used is to avoid timing problem by not sending long, uninterrupted streams of bits. Instead, data are transmitted one character at a time, where each character is five to eight bits in length. By using synchronous transmission, we can transmit large block of bits in a steady stream without start T and stop codes. (Refer section 2 for detail).
2. In digital transmission systems, an error occurs when a bit is altered between transmission and reception. That means, a binary 1 is transmitted and a binary 0 is received, or a binary 0 is transmitted and a binary 1 is received. Two general types of errors can occur. They are, *single-bit errors* and *burst errors*. (Refer section 3 for detail).
3. The simplest error-detecting scheme is to append a parity bit to the end of a block of data. Cyclic redundancy check is one of the most common, powerful error detecting code. (Refer section 4 for detail).
4. Error Correction can be handled in two ways. One is when an error is discovered; the receiver can ask the sender retransmit the entire data unit. This is known as backward error correction. Second method is that, receiver can use an error-correcting code, which automatically corrects certain errors. This is known as forward error correction. (Refer section 5 for detail).
5. The topology of a data link refers to the physical arrangement of stations on a transmission medium. (Refer section 6 for detail).
6. Data exchanges over a transmission line can be classified as full duplex or half duplex. With *half-duplex transmission*, only one of two stations on a point-to-point link may transmit at a time. This mode is also referred to as *two-way alternate*, considering the fact that two stations must alternate in transmitting. This can be compared to a one-lane, two-way bridge (Refer section 6 for detail).

References

1. Behrouz A. Forouzan, Sophia Chung Fegan, “Data Communications and Networking”, Fourth edition.
2. William Stallings, “Data and Computer Communications”, Sixth edition, Pearson Education, Delhi, 2002.
3. Taub and Schilling, “Principles of Communication Systems”, Tata Mc Graw Hill, Delhi, 2002.
4. S. Tanenbaum, ”Computer Networks”, Pearson Education, Fourth Edition.
5. N. Olifer, V. Olifer, ”Computer Networks: Principles, technologies and Protocols for Network design”, Wiley India Edition, First edition.
6. Simon Poulton (2003), packet switching and X.25 Networking, Pitman publishing.
7. Walrand, P. Varaiya, “high performance communication networks”, Morgan kaufmann.