1

CHAPTER 9

# Error Detection and Correction

Networks must be able to transfer data from one device to another with complete accuracy. A system that cannot guarantee that the data received by one device are identical to the data transmitted by another device is essentially useless. Yet anytime data are transmitted from source to destination, they can become corrupted in passage. In fact, it is more likely that some part of a message will be altered in transit than that the entire contents will arrive intact. Many factors, including line noise, can alter or wipe out one or more bits of a given data unit. Reliable systems must have a mechanism for detecting and correcting such **errors.**

> Data can be corrupted during transmission. For reliable communication, errors must be detected and corrected.

Error detection and correction are implemented either at the data link layer or the transport layer of the OSI model.

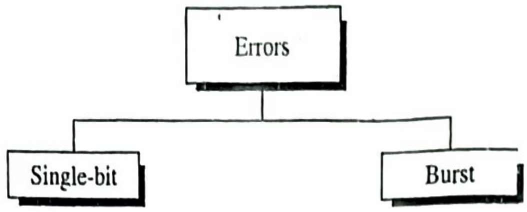
## 9.1 TYPES OF ERRORS

Whenever an electromagnetic signal flows from one point to another, it is subject to unpredictable interference from heat, magnetism, and other forms of electricity. This interference can change the shape or timing of the signal. If the signal is carrying encoded binary data, such changes can alter the meaning of the data. In a single-bit error, a 0 is changed to a 1 or a 1 to a 0. In a burst error, multiple bits are changed. For example, a 0.01-second burst of impulse noise on a transmission with a data rate of 1200 bps might change all or some of 12 bits of information (see Figure 9.1).

### Single-Bit Error

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

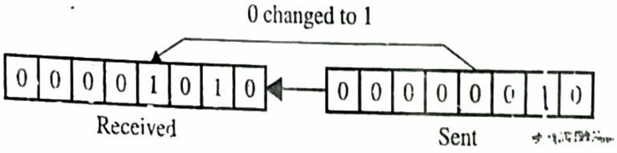
Figure 9.1 *Types of errors*



In a single-bit error, only one bit in the data unit has changed.

Figure 9.2 shows the effect of a single-bit error on a data unit. To understand the impact of the change, imagine that each group of eight bits is an ASCII character with a 0 bit added to the left. In the figure, 00000010 (ASCII *STX*) was sent, meaning *start of text,* but 00001010 (ASCII *LF*) was received, meaning *line feed*. (For more information about ASCII code, see Appendix A.)

**Figure 9.2** *Single-bit error*



Single-bit errors are the least likely type of error in serial data transmission. To see why, imagine a sender sends data at 1Mbps. This means that each bit lasts only 1/1,000,000 second, or 1 μs. For a single-bit error to occur, the noise must have a duration of only 1 μs, which is very rare; noise normally lasts much longer than this.

However, a single-bit error can happen if we are sending data using parallel transmission. For example, if eight wires are used to send all of the eight bits of a byte at the same time and one of the wires is noisy, one bit can be corrupted in each byte. Think of parallel transmission inside a computer, between CPU and memory, for example.

## Burst Error

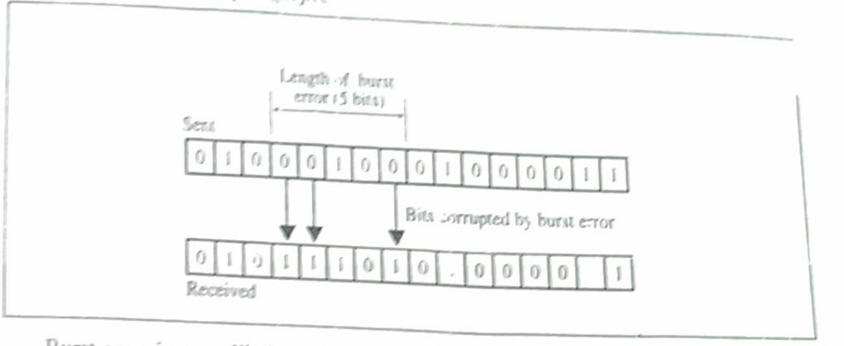
The term **burst error** means that two or more bits in the data unit have changed from 1 to 0 or from 0 to 1

A burst error means that two or more bits in the data unit have changed.

Figure 9.3 shows the effect of a burst error on a data unit. In this case, 0100010001000011 was sent, but 0101110101000011 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.

?

Figure 9.3 *Burst error of length five*



Burst error is most likely to happen in a serial transmission. The duration of noise is normally longer than the duration of a bit, which means that when noise affects data, it affects a set of bits. The number of bits affected depends on the data rate and duration of noise. For example, if we are sending data at 1 Kbps, a noise of 1/100 seconds can affect 10 bits; if we are sending data at 1 Mbps, the same noise can affect 10,000 bits.

## 9.2 DETECTION

Even if we know what types of errors can occur, will we recognize one when we see it? If we have a copy of the intended transmission for comparison, of course we will. But what if we don't have a copy of the original? Then we will have no way of knowing we have received an error until we have decoded the transmission and failed to make sense of it. For a machine to check for errors this way would be slow, costly, and of questionable value. We don't need a system where computers decode whatever comes in, then sit around trying to decide if the sender really meant to use the word *glbrshnif* in the middle of an array of weather statistics. What we need is a mechanism that is simple and completely objective.

### Redundancy

One error detection mechanism that would satisfy these requirements would be to send every data unit twice. The receiving device would then be able to do a bit-for-bit comparison between the two versions of the data. Any discrepancy would indicate an error, and an appropriate correction mechanism could be set in place. This system would be completely accurate (the odds of errors being introduced onto exactly the same bits in both sets of data are infinitesimally small), but it would also be insupportably slow. Not only would the transmission time double, but the time it takes to compare every unit bit by bit must be added.

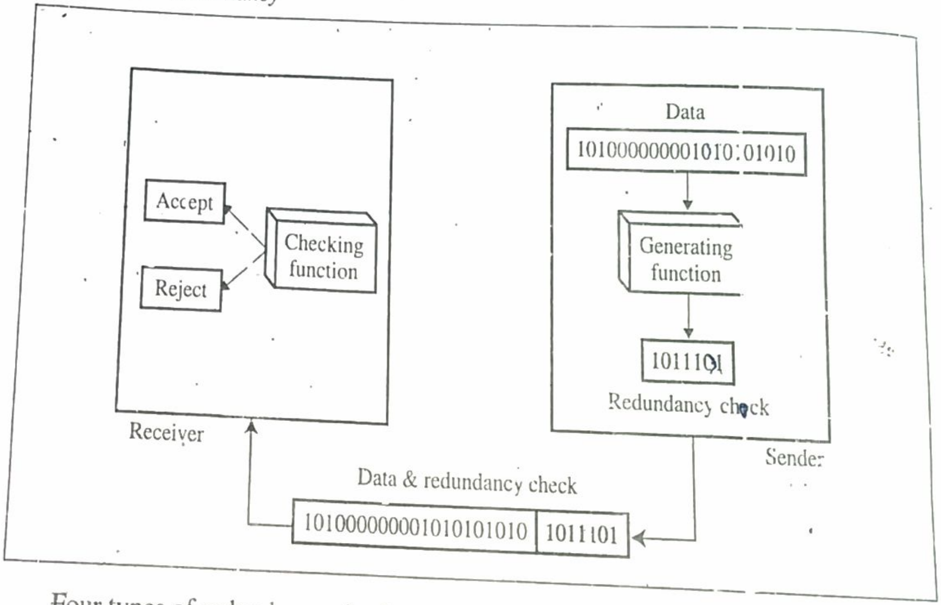
The concept of including extra information in the transmission solely for the purposes of comparison is a good one. But instead of repeating the entire data stream, a shorter group of bits may be appended to the end of each unit. This technique is called **redundancy** because the extra bits are redundant to the information; they are discarded as soon as the accuracy of the transmission has been determined.

4

> Error detection uses the concept of redundancy, which means adding extra bits for detecting errors at the destination.

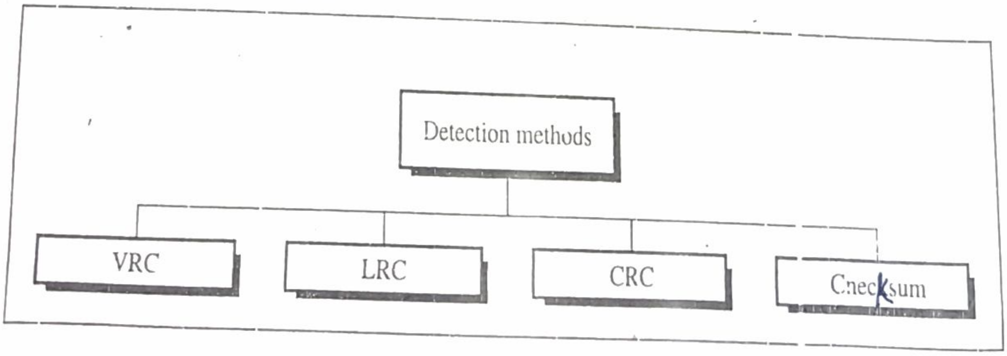
Figure 9.4 shows the process of using redundant bits to check the accuracy of a data unit. Once the data stream has been generated, it passes through a device that analyzes it and adds on an appropriately coded redundancy check. The data unit, now enlarged by several bits (in this illustration, seven), travels over the link to the receiver. The receiver puts the entire stream through a checking function. If the received bit stream passes the checking criteria, the data portion of the data unit is accepted and the redundant bits are discarded.

**Figure 9.4** *Redundancy*



Four types of redundancy checks are used in data communications: vertical redundancy check (VRC) (also called parity check), longitudinal redundancy check (LRC), cyclical redundancy check (CRC), and checksum. The first three, VRC, LRC, and CRC, are normally implemented in the physical layer for use in the data link layer. The fourth, checksum, is used primarily by upper layers (see Figure 9.5).

**Figure 9.5** *Detection methods*



S

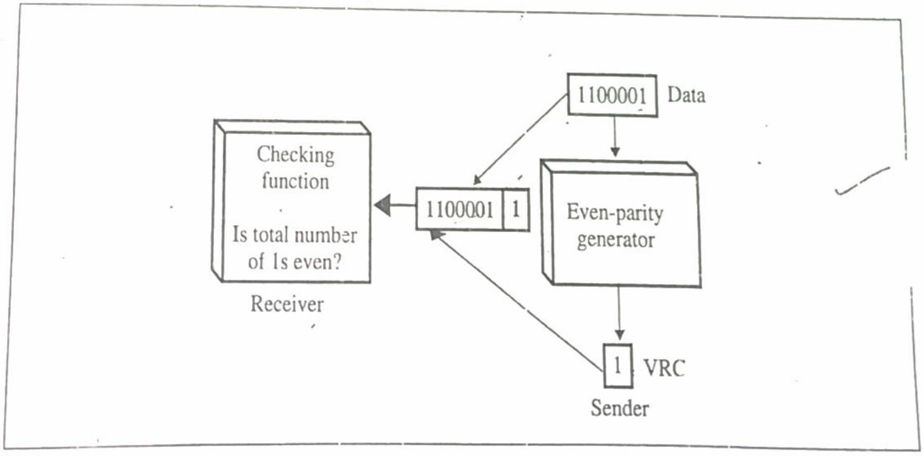
## 9.3 VERTICAL REDUNDANCY CHECK (VRC)

The most common and least expensive mechanism for error detection is the **vertical redundancy check (VRC)**, often called a **parity check.** In this technique, a redundant bit, called a **parity bit,** is appended to every data unit so that the total number of 1s in the unit (including the parity bit) becomes even.

Suppose we want to transmit the binary data unit 1100001 [ASCII *a* (97)]; see Figure 9.6. Adding together the number of 1s gives us 3, an odd number. Before transmitting, we pass the data unit through a parity generator. The parity generator counts the 1s and appends the parity bit (a 1 in this case) to the end. The total number of 1s is now four, an even number. The system now transmits the entire expanded unit across the network link. When it reaches its destination, the receiver puts all eight bits through an **even-parity** checking function If the receiver sees 11100001, it counts four 1s, an even number, and the data unit passes. But what if the data unit has been damaged in transit? What if, instead of 11100001, the receiver sees 11100101? Then, when the parity checker counts the 1s, it gets 5, an odd number. The receiver knows that an error has been introduced into the data somewhere and therefore rejects the whole unit.

In vertical redundancy check (VRC), a parity bit is added to every data unit so that the total number of 1s becomes even.

**Figure 9.6** *Even parity VRC concept*



Note that for the sake of simplicity, we are discussing here even-parity checking, where the number of 1s should be an even number. Some systems may use **odd-parity** checking, where the number of 1s should be odd. The principle is the same; the calculation is different.

6

### Example 9.1

Imagine the sender wants to send the word "world." In ASCII (see Appendix A), the five characters are coded as

| ← | 1110111 | 1101111 | 1110010 | 1101100 | 1100100 |
|---|---|---|---|---|---|
| | w | o | r | l | d |

Each of the first four characters has an even number of 1s, so the parity bit is a 0. The last character ("d"), however, has three 1s (an odd number), so the parity bit is a 1 to make the total number of 1s even. The following shows the actual bits sent (the parity bits are underlined).

← 11101110 11011110 11100100 11011000 11001001

### Example 9.2

Now suppose the word "world," in the previous example, is received by the receiver without being corrupted in transmission.

← 11101110 11011110 11100100 11011000 11001001

The receiver counts the 1s in each character and comes up with even numbers (6, 6, 4, 4, 4). The data would be accepted.

### Example 9.3

Now suppose the word "world," in Example 9.1, is received by the receiver but corrupted during transmission.

← 111*1*1110 11011110 1110*1*100 11011000 11001001

The receiver counts the 1s in each character and comes up with even and odd numbers (7, 6, 5, 4, 4). The receiver knows that the data are corrupted, discards them, and asks for retransmission.

### Performance

VRC can detect all single-bit errors. It can also detect burst errors as long as the total number of bits changed is odd (1, 3, 5, etc.). Let's say we have an even-parity data unit where the total number of 1s, including the parity bit, is 6: 1000111011. If any three bits change value, the resulting parity will be odd and the error will be detected: 1*111*111011:9, *011*0111011:7, 1*1*00*01*0011:5—all odd. The VRC checker would return a result of 1 and the data unit would be rejected. The same holds true for any odd number of errors.

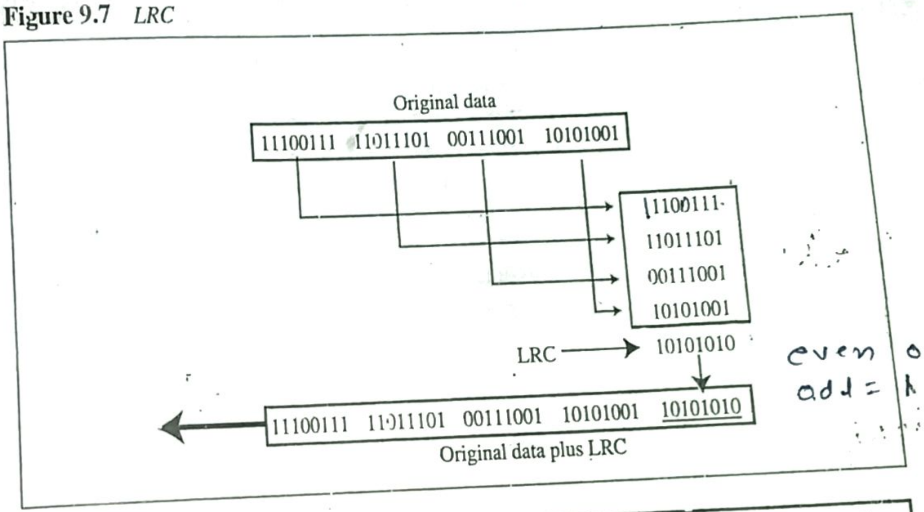
Suppose, however, that two bits of the data unit are changed: 1*11*0111011:8, 1*1*0*0*011011:6, 1000*0*1101*0*:4. In each case the number of 1s in the data unit is still even. The VRC checker will add them and return an even number although the data unit contains two errors. VRC cannot detect errors where the total number of bits changed is even. If any two bits change in transmission, the changes cancel each other and the data unit will pass a parity check even though the data unit is damaged. The same holds true for any even number of errors.

> VRC can detect all single-bit errors. It can detect burst errors only if the total number of errors in each data unit is odd.

## 9.4 LONGITUDINAL REDUNDANCY CHECK (LRC)

In **longitudinal redundancy check (LRC)**, a block of bits is organized in a table (rows and columns). For example, instead of sending a block of 32 bits, we organize them in a table made of four rows and eight columns, as shown in Figure 9.7. We then calculate the parity bit for each column and create a new row of eight bits, which are the parity bits for the whole block. Note that the first parity bit in the fifth row is calculated based on all first bits. The second parity bit is calculated based on all second bits, and so on. We then attach the eight parity bits to the original data and send them to the receiver.

**Figure 9.7** *LRC*



even 0
odd = 1

In longitudinal redundancy check (LRC), a block of bits is divided into rows and a redundant row of bits is added to the whole block.

**Example 9.4**

Suppose the following block is sent:

← 10101001 00111001 11011101 11100111 10101010 (LRC)

However, it is hit by a burst noise of length eight and some bits are corrupted.

← 10100011 10001001 11011101 11100111 10101010 (LRC)

When the receiver checks the LRC, some of the bits do not follow the even-parity rule and the whole block is discarded (the nonmatching bits are shown in bold).

← 10100011 10001001 11011101 11100111 **10101010** (LRC)

8

### Performance

LRC increases the likelihood of detecting burst errors. As we showed in the previous example, an LRC of *n* bits can easily detect a burst error of *n* bits. A burst error of more than *n* bits is also detected by LRC with a very high probability. There is, however, one pattern of errors that remains elusive. If two bits in one data unit are damaged and two bits *in exactly the same positions* in another data unit are also damaged, the LRC checker will not detect an error. Consider, for example, two data units: 11110000 and 11000011. If the first and last bits in each of them are changed, making the data units read *0*111000*1* and *0*100001*0*, the errors cannot be detected by LRC

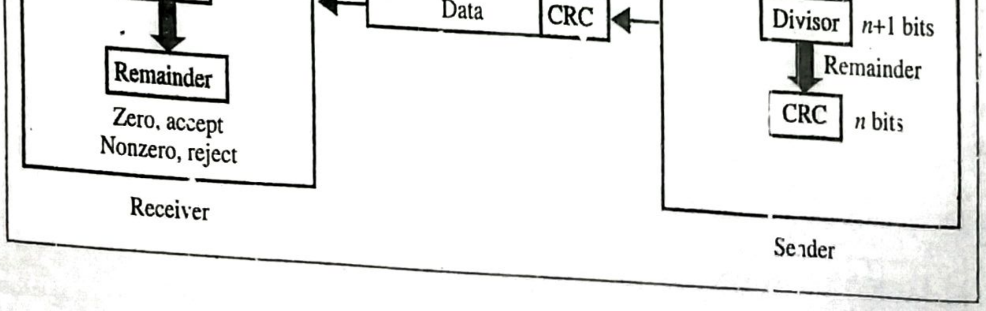
## 9.5 CYCLIC REDUNDANCY CHECK (CRC)

The third and most powerful of the redundancy checking techniques is the **cyclic redundancy check (CRC).** Unlike VRC and LRC, which are based on addition, CRC is based on binary division. In CRC, instead of adding bits together to achieve a desired parity, a sequence of redundant bits, called the CRC or the CRC remainder, is appended to the end of a data unit so that the resulting data unit becomes exactly divisible by a second, predetermined binary number. At its destination, the incoming data unit is divided by the same number. If at this step there is no remainder, the data unit is assumed to be intact and is therefore accepted. A remainder indicates that the data unit has been damaged in transit and therefore must be rejected.

The redundancy bits used by CRC are derived by dividing the data unit by a predetermined divisor; the remainder is the CRC. To be valid, a CRC must have two qualities: it must have exactly one less bit than the divisor, and appending it to the end of the data string must make the resulting bit sequence exactly divisible by the divisor.

Both the theory and the application of CRC error detection are straightforward. The only complexity is in deriving the CRC. In order to clarify this process, we will start with an overview and add complexity as we go. Figure 9.8 provides an outline of the three basic steps.

**Figure 9.8** *CRC generator and checker*



First, a string of $n$ 0s is appended to the data unit. The number $n$ is one less than the number of bits in the predetermined divisor, which is $n + 1$ bits.

Second, the newly elongated data unit is divided by the divisor using a process called binary division. The remainder resulting from this division is the CRC.

Third, the CRC of $n$ bits derived in step 2 replaces the appended 0s at the end of the data unit. Note that the CRC may consist of all 0s.

The data unit arrives at the receiver data first, followed by the CRC. The receiver treats the whole string as a unit and divides it by the same divisor that was used to find the CRC remainder.

If the string arrives without error, the CRC checker yields a remainder of zero and the data unit passes. If the string has been changed in transit, the division yields a nonzero remainder and the data unit does not pass.

### The CRC Generator

A CRC generator uses modulo-2 division. Figure 9.9 shows this process. In the first step, the four-bit divisor is subtracted from the first four bits of the dividend. Each bit of the divisor is subtracted from the corresponding bit of the dividend without disturbing the next higher bit. In our example, the divisor, 1101, is subtracted from the first four bits of the dividend, 1001, yielding 100 (the leading 0 of the remainder is dropped off).

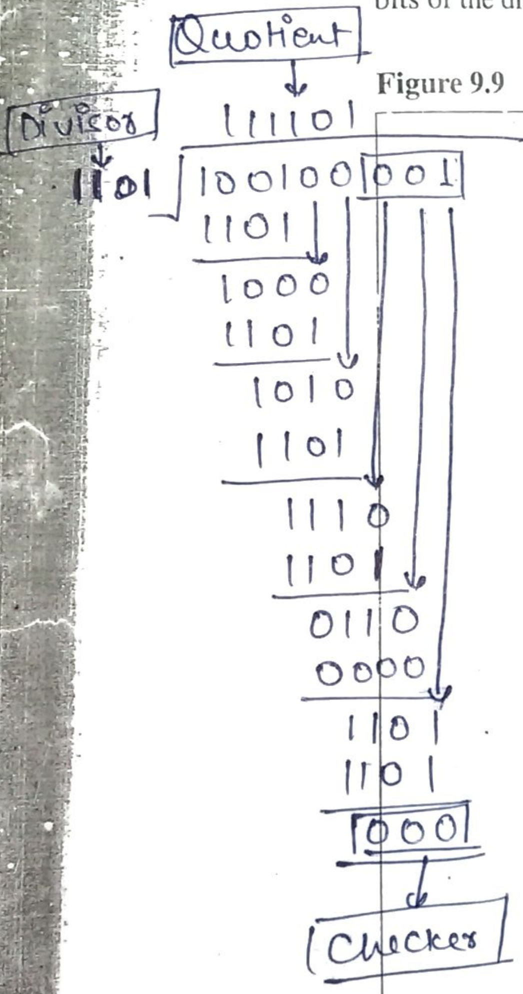


**Figure 9.9** *Binary division in a CRC generator*

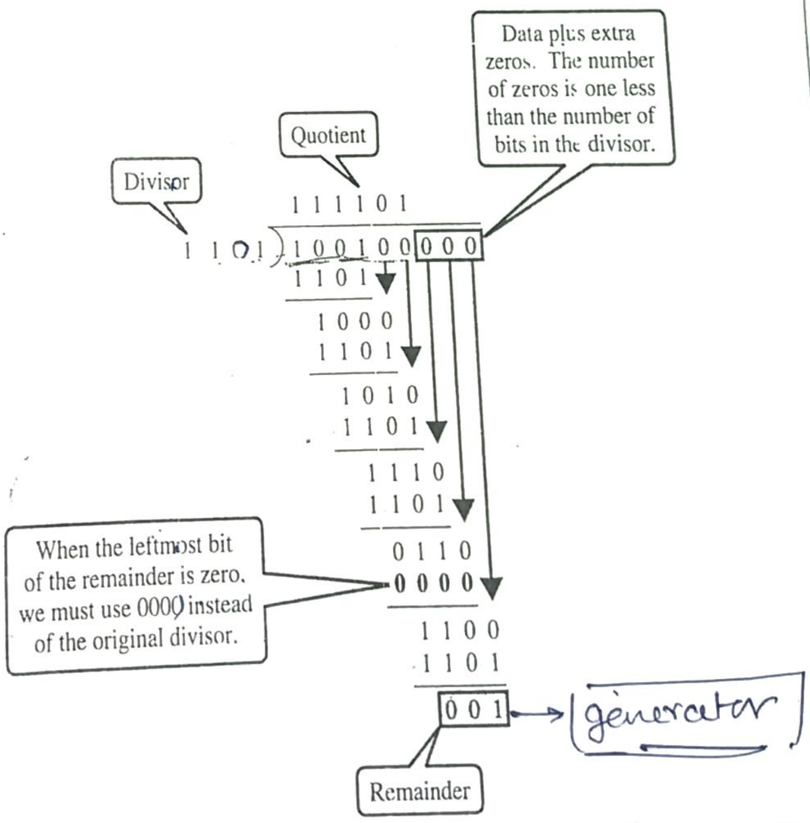

Data plus extra zeros. The number of zeros is one less than the number of bits in the divisor.
Quotient
Divisor
111101
1101
100100 000
1101
1000
1101
1010
1101
1110
1101
0110
0000
1100
1101
001
generator
When the leftmost bit of the remainder is zero, we must use 0000 instead of the original divisor.
Remainder

Figure 9.13 *Standard polynomials* 10

CRC-12

$$\boxed{x^{12}+x^{11}+x^{3}+x+1}$$

CRC-16

$$\boxed{x^{16}+x^{15}+x^{2}+1}$$

CRC-ITU-T

$$\boxed{x^{16}+x^{12}+x^{5}+1}$$

CRC-32

$$\boxed{x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^{8}+x^{7}-x^{5}+x^{4}+x^{2}-x+1}$$

## Performance

CRC is a very effective error detection method. If the divisor is chosen according to the previously mentioned rules,

a. CRC can detect all burst errors that affect an odd number of bits.

b. CRC can detect all burst errors of length less than or equal to the degree of the polynomial.

c. CRC can detect with a very high probability burst errors of length greater than the degree of the polynomial.

### Example 9.6

The CRC-12 ($x^{12}+x^{11}+x^{3}+x+1$), which has a degree of 12, will detect all burst errors affecting an odd number of bits, will detect all burst errors with a length less than or equal to 12, and will detect 99.97 percent of the time burst errors with a length of 12 or more.

## 9.6 CHECKSUM

The error detection method used by the higher-layer protocols is called **checksum**. Like VRC, LRC, and CRC, checksum is based on the concept of redundancy.

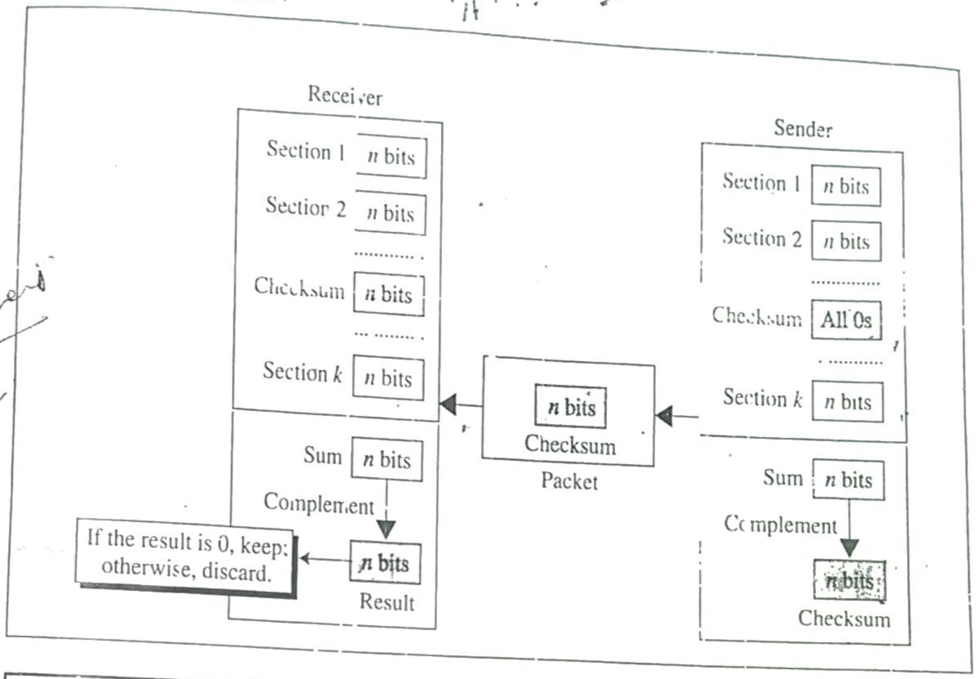
### Checksum Generator

In the sender, the checksum generator subdivides the data unit into equal segments of $n$ bits (usually 16). These segments are added together using **one's complement** arithmetic (see Appendix C) in such a way that the total is also $n$ bits long. That total (sum) is then complemented and appended to the end of the original data unit as redundancy bits, called the checksum field. The extended data unit is transmitted across the network. So if the sum of the data segment is $T$, the checksum will be $-T$ (see Figures 9.14 and 9.15).

### Checksum Checker

The receiver subdivides the data unit as above and adds all segments together and complements the result. If the extended data unit is intact, the total value found by adding the data segments and the checksum field should be zero. If the result is not zero, the packet contains an error and the receiver rejects it (see Appendix C).

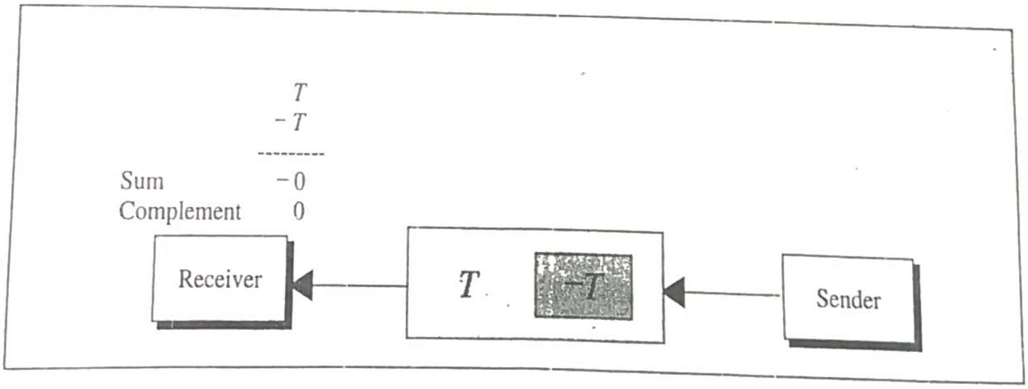
Figure 9.14 *Checksum*



**The sender follows these steps:**

- The unit is divided into *k* sections, each of *n* bits.
- All sections are added together using one's complement to get the sum.
- The sum is complemented and becomes the checksum.
- The checksum is sent with the data.

Figure 9.15 *Data unit and checksum*



**The receiver follows these steps:**

- The unit is divided into *k* sections, each of *n* bits.
- All sections are added together using one's complement to get the sum.
- The sum is complemented.
- If the result is zero, the data are accepted: otherwise, they are rejected.

### Example 9.7

12

Suppose the following block of 16 bits is to be sent using a checksum of 8 bits.

```
        ←  10101001 00111001
```

The rumbers are added using one's complement arithmetic (see Appendix C).

```
                    10101001
                    00111001
                 --------------
          Sum       11100010
          Checksum  00011101   ✓ 1's [illegible]
```

S

The pattern sent is:

```
        ←  10101001  00111001  00011101
                               Checksum
```

### Example 9.8

Now suppose the receiver receives the pattern sent in Example 9.7 and there is no error.

```
        10101001  00111001  00011101
```

When the receiver adds the three sections together, it will get all 1s, which, after complementing, is all 0s and shows that there is no error.

```
                      10101001
                      00111001
                      00011101
                    ---------------
        Sum           11111111
        Complement    00000000 ✓  means that the pattern is OK.
```

### Example 9.9

Now suppose there is a burst error of length five that affects four bits.

10101*111* *11*111001 00011101

When the receiver adds the three sections together, it gets

```
                          10101111
                          11111001
                          00011101
                    ---------------------
        Result      1     11000101
        Carry                    1
                    ---------------------
        Sum               11000110
        Complement        00111001      means that the pattern is corrupted.
```

13

### Performance

The checksum detects all errors involving an odd number of bits, as well as most errors involving an even number of bits. However, if one or more bits of a segment are damaged and the corresponding bit or bits of opposite value in a second segment are also damaged, the sums of those columns will not change and the receiver will not detect a problem. If the last digit of one segment is a 0 and it gets changed to a 1 in transit, then the last 1 in another segment must be changed to a 0 if the error is to go undetected. In LRC, two 0s could both change to 1s without altering the parity because carries were discarded. Checksum retains all carries; so, although two 0s becoming 1s would not alter the value of their own column, they would change the value of the next higher column. But anytime a bit inversion is balanced by an opposite bit inversion in the corresponding digit of another data segment, the error is invisible.

## 9.7 ERROR CORRECTION

The mechanisms that we have covered up to this point detect errors but do not correct them. **Error correction** can be handled in two ways. In one, when an error is discovered, the receiver can have the sender retransmit the entire data unit. In the other, a receiver can use an error-correcting code, which automatically corrects certain errors.

In theory, it is possible to correct any binary code errors automatically. Error-correcting codes, however, are more sophisticated than error-detection codes and require more redundancy bits. The number of bits required to correct a multiple-bit or burst error is so high that in most cases it is inefficient to do so. For this reason, most error correction is limited to one-, two-, or three-bit errors.

### Single-Bit Error Correction

The concept underlying error correction can be most easily understood by examining the simplest case: single-bit errors.

As we saw earlier, single-bit errors can be detected by the addition of a redundant (parity) bit to the data unit (VRC). A single additional bit can detect single-bit errors in any sequence of bits because it must distinguish between only two conditions: error or no error. A bit has two states (0 and 1). These two states are sufficient for this level of detection.

But what if we want to correct as well as detect single-bit errors? Two states are enough to detect an error but not to correct it. An error occurs when the receiver reads a 1 bit as a 0 or a 0 bit as a 1. To correct the error, the receiver simply reverses the value of the altered bit. To do so, however, it must know which bit is in error. The secret of error correction, therefore, is to locate the invalid bit or bits.

For example, to correct a single-bit error in an ASCII character, the error correction code must determine which of the seven bits has changed. In this case, we have to distinguish between eight different states: no error, error in position 1, error in position 2, and so on, up to error in position 7. To do so requires enough redundancy bits to show all eight states.

At first glance, it looks like a three-bit redundancy code should be adequate because three bits can show eight different states (000 to 111) and can therefore

1.4

indicate the locations of eight different possibilities. But what if an error occurs in the redundancy bits themselves? Seven bits of data (the ASCII character) plus three bits of redundancy equals 10 bits. Three bits, however, can identify only eight possibilities. Additional bits are necessary to cover all possible error locations.

### Redundancy Bits

r - Redu. ban. bits
no.
m= data bits

To calculate the number of redundancy bits ($r$) required to correct a given number of data bits ($m$), we must find a relationship between $m$ and $r$. Figure 9.16 shows $m$ bits of data with $r$ bits of redundancy added to them. The length of the resulting code is $m + r$.

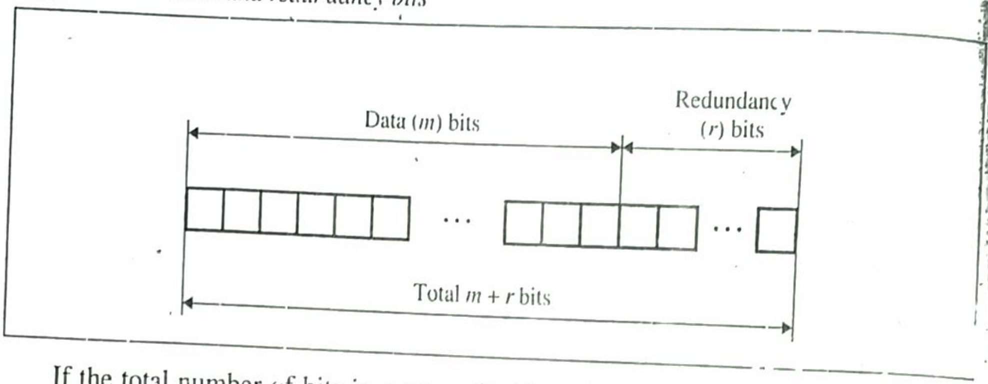


**Figure 9.16** *Data and redundancy bits*

If the total number of bits in a transmittable unit is $m + r$, then $r$ must be able to indicate at least $m + r + 1$ different states. Of these, one state means no error and $m + r$ states indicate the location of an error in each of the $m + r$ positions.

So, $m + r + 1$ states must be discoverable by $r$ bits; and $r$ bits can indicate $2^r$ different states. Therefore, $2^r$ must be equal to or greater than $m + r + 1$:

$$2^r \geq m + r + 1$$

The value of $r$ can be determined by plugging in the value of $m$ (the original length of the data unit to be transmitted). For example, if the value of $m$ is 7 (as in a seven-bit ASCII code), the smallest $r$ value that can satisfy this equation is 4:

$$2^4 \geq 7 + 4 + 1$$

Table 9.1 shows some possible $m$ values and the corresponding $r$ values.

**Table 9.1** *Relationship between data and redundancy bits*

| *Number of Data Bits* (m) | *Number of Redundancy Bits* (r) | *Total Bits* (m + r) |
|---|---|---|
| 1 | 2 | 3 |
| 2 | 3 | 5 |
| 3 | 3 | 6 |
| 4 | 3 | 7 |
| 5 | 4 | 9 |
| 6 | 4 | 10 |
| 7 | 4 | 11 |

IS

## Hamming Code

So far, we have examined the number of bits required to cover all of the possible single-bit error states in a transmission. But how do we manipulate those bits to discover which state has occurred? A technique developed by R. W. Hamming provides a practical solution.

### Positioning the Redundancy Bits

The Hamming code can be applied to data units of any length and uses the relationship between data and redundancy bits discussed above. For example, a seven-bit ASCII code requires four redundancy bits that can be added to the end of the data unit or interspersed with the original data bits. In Figure 9.17, these bits are placed in positions 1, 2, 4, and 8 (the positions in an 11-bit sequence that are powers of 2). For clarity in the examples below, we refer to these bits as $r_1$, $r_2$, $r_4$ 4 and $r_8$.

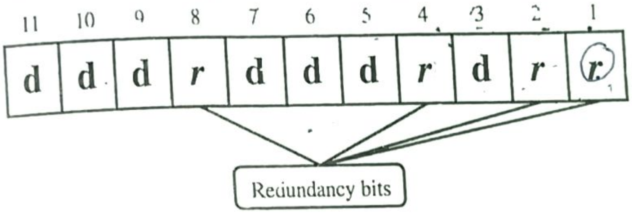
4 5 6 7

1 3 5 7 9 11

2 3 6 7 10 11

**Figure 9.17** *Positions of redundancy bits in Hamming code*

0 1



0000 – 0
0001 – 1
0010 – 2
0011 – 3
0100 – 4
0101 – 5
0110 – 6
0111 – 7
1000 – 8
1001 – 9
1010 – 10
1011 –

In the Hamming code, each $r$ bit is the VRC bit for one combination of data bits: $r_1$ is the VRC bit for one combination of data bits, $r_2$ is the VRC bit for another combination of data bits, and so on. The combinations used to calculate each of the four $r$ values for a seven-bit data sequence are as follows:

$r_1$: bits 1, 3, 5, 7, 9, 11 — one position is 1

$r_2$: bits 2, 3, 6, 7, 10, 11 — second position is 1

$r_4$: bits 4, 5, 6, 7 — ____ is 1

$r_8$: bits 8, 9, 10, 11 — ____ is 1

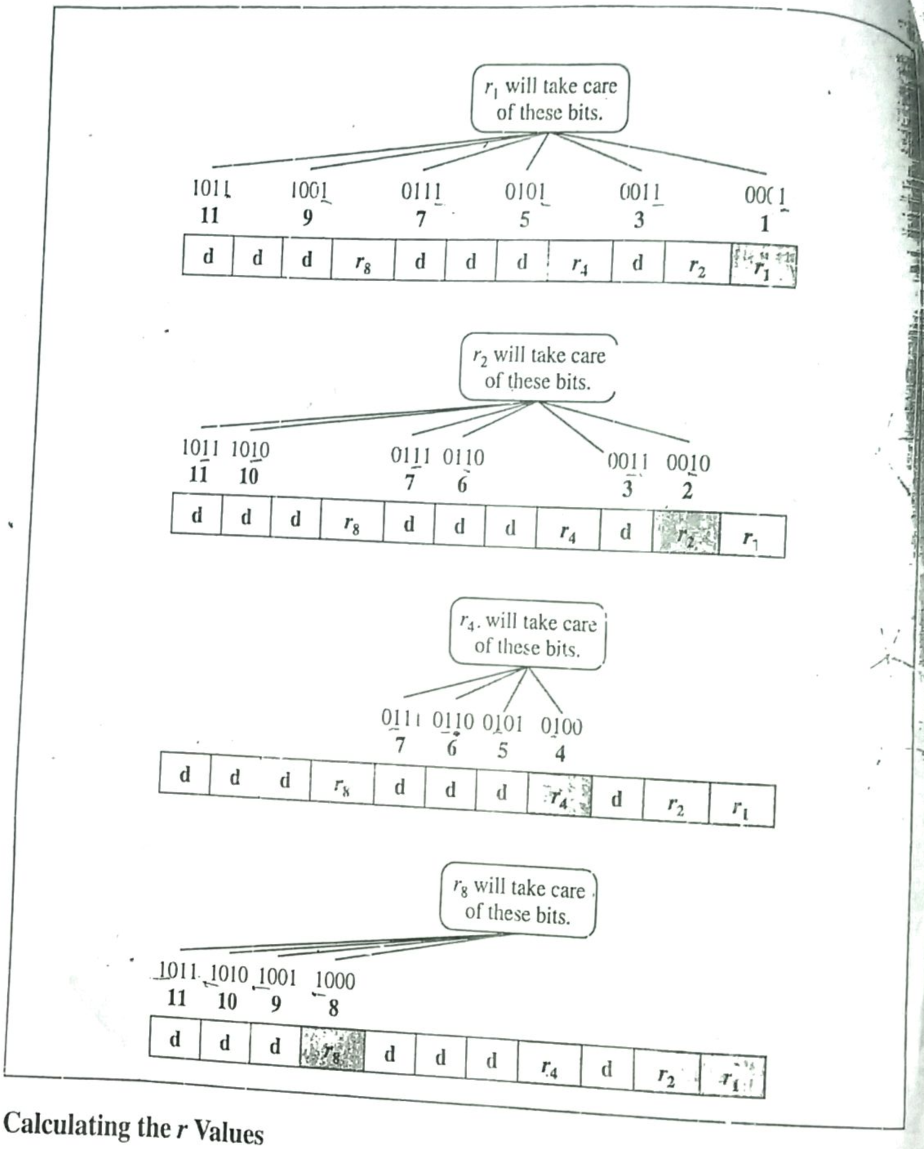
Each data bit may be included in more than one VRC calculation. In the sequences above, for example, each of the original data bits is included in at least two sets, while the $r$ bits are included in only one.

To see the pattern behind this strategy, look at the binary representation of each bit position. The $r_1$ bit is calculated using all bit positions whose binary representation includes a 1 in the rightmost position. The $r_2$ bit is calculated using all bit positions with a 1 in the second position, and so on (see Figure 9.18).

0 00
1 01 ✓
2 10
3 11 ✓
4 00
5 01 ✓

16

Figure 9.18 *Redundancy bits calculation*



$r_1$ will take care of these bits.
1011 11
1001 9
0111 7
0101 5
0011 3
0001 1
d d d $r_8$ d d d $r_4$ d $r_2$ $r_1$
$r_2$ will take care of these bits.
1011 11
1010 10
0111 7
0110 6
0011 3
0010 2
d d d $r_8$ d d d $r_4$ d $r_2$ $r_1$
$r_4$ will take care of these bits.
0111 7
0110 6
0101 5
0100 4
d d d $r_8$ d d d $r_4$ d $r_2$ $r_1$
$r_8$ will take care of these bits.
1011 11
1010 10
1001 9
1000 8
d d d $r_8$ d d d $r_4$ d $r_2$ $r_1$

### Calculating the *r* Values

Figure 9.19 shows a Hamming code implementation for an ASCII character. In the first step, we place each bit of the original character in its appropriate position in the 11-bit unit. In the subsequent steps, we calculate the even parities for the various bit combinations. The parity value for each combination is the value of the corresponding *r* bit. For example, the value of $r_1$ is calculated to provide even parity for a combination of bits 3, 5, 7, 9, and 11. The value of $r_2$ is calculated to provide even parity with bits 3, 6, 7, 10, and 11, and so on. The final 11-bit code is sent through the transmission line.

### Error Detection and Correction

Now imagine that by the time the above transmission is received, the number 7 bit has been changed from 1 to 0 (see Figure 9.20).

Figure 9.19 *Example of redundancy bit calculation*

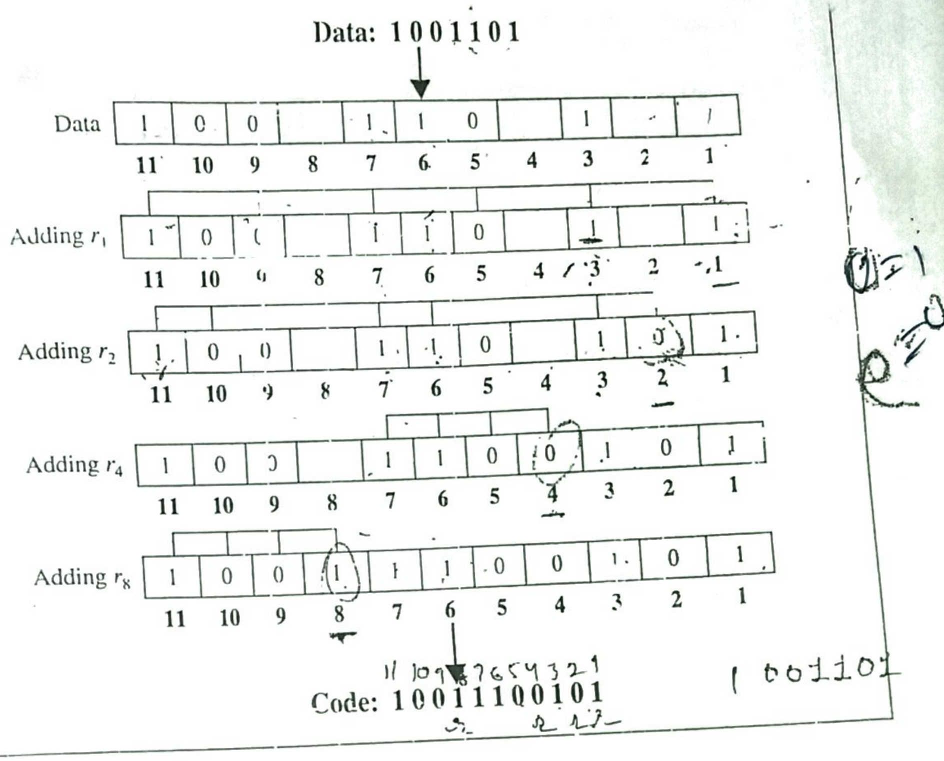
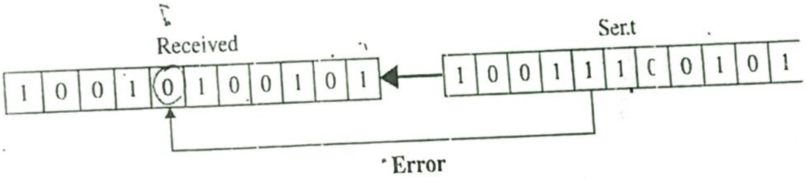


Figure 9.20 *Single-bit error*



The receiver takes the transmission and recalculates four new VRCs using the same sets of bits used by the sender plus the relevant parity (r) bit for each set (see Figure 9.21). Then it assembles the new parity values into a binary number in order of r position ($r_8$, $r_4$, $r_2$, $r_1$). In our example, this step gives us the binary number 0111 (7 in decimal), which is the precise location of the bit in error.

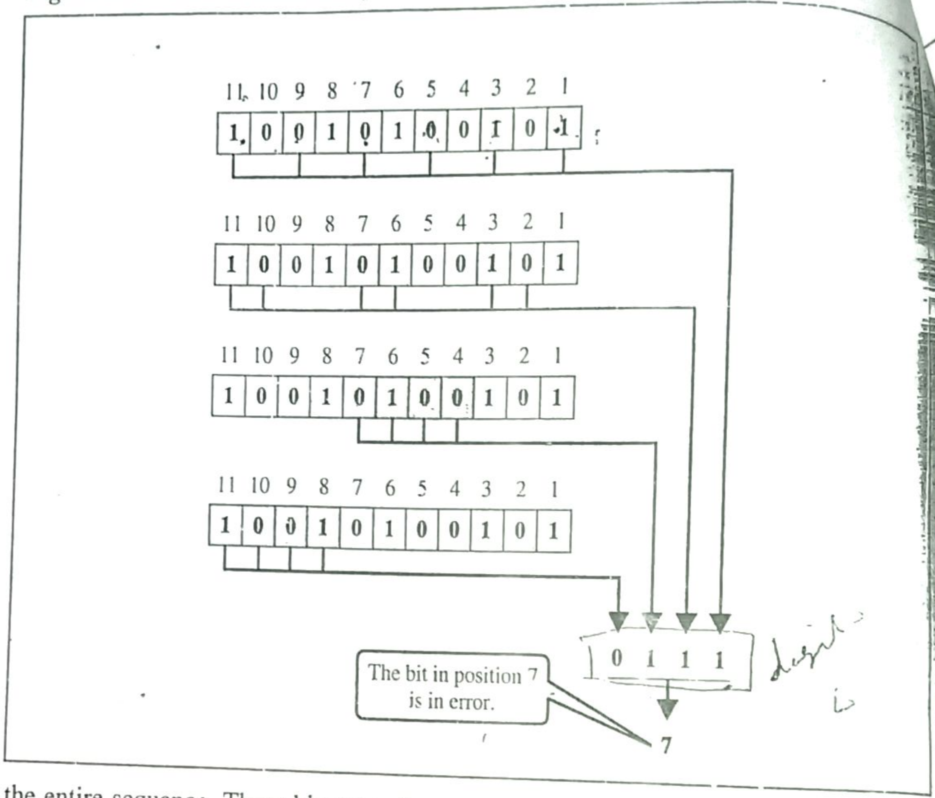
Once the bit is identified, the receiver can reverse its value and correct the error.

## Burst Error Correction

A Hamming code can be designed to correct burst errors of certain lengths. The number of redundancy bits required to make these corrections, however, is dramatically higher than that required for single-bit errors. To correct double-bit errors, for example, we must take into consideration that the two bits can be a combination of any two bits in

18

Figure 9.21 *Error detection using Hamming code*



the entire sequence. Three-bit correction means any three bits in the entire sequence, and so on. So the simple strategy used by the Hamming code to correct single-bit errors must be redesigned to be applicable for multiple-bit correction. We leave the details of these more sophisticated schemes to advanced books on **error handling.**

## 9.8 KEY TERMS AND CONCEPTS

burst error

checksum

cyclic redundancy check (CRC)

error

error correction

error detection

error handling

even parity

Hamming code

longitudinal redundancy check (LRC)

odd parity

one's complement

19

parity bit

parity check

redundancy

single-bit error

vertical redundancy check (VRC)

## 9.9 SUMMARY

- Transmission errors are usually detected at the physical layer of the OSI model.
- Transmission errors are usually corrected at the data link layer of the OSI model.
- Errors can be categorized as follows:
  - a. Single-bit: one bit error per data unit.
  - b. Burst: two or more bit errors per data unit.
- Redundancy is the concept of sending extra bits for use in error detection.
- Four common methods of error detection are the following:
  - a. Vertical redundancy check (VRC).
  - b. Longitudinal redundancy check (LRC).
  - c. Cyclic redundancy check (CRC).
  - d. Checksum.
- In VRC an extra bit (parity bit) is added to the data unit.
- VRC can detect only an odd number of errors; it cannot detect an even number of errors.
- In LRC a redundant data unit follows *n* data units.
- CRC, the most powerful of the redundancy checking techniques, is based on binary division.
- Checksum is used by the higher-layer protocols (TCP/IP) for error detection.
- To calculate a checksum:
  - a. Divide the data into sections.
  - b. Add the sections together using one's complement arithmetic.
  - c. Take the complement of the final sum; this is the checksum.
- At the receiver, when using the checksum method, the data and checksum should add up to zero if there are no errors present.
- The Hamming code is a single-bit error correction method using redundant bits. The number of bits is a function of the length of the data bits.
- In the Hamming code, for a data unit of *m* bits, use the formula $2r \geq m + r - 1$ to determine *r*, the number of redundant bits needed.

1

Note:- Bandwidth - is the information carrying capacity of a communication channel.

Attenuation - The loss of signal's energy due to the resistance of the medium

Distortion - Any changes in a signal due to noise, attenuation or other influences.

Noise - Random electrical signals that can be picked up by the transmission medium and result in degradation or distortion of data

EMI - Electromagnetic interference refer to external influence having an impact on the signal that is being transmitted over the media.

## Transmission Media

→ Physical connection that the computer have to one another.

→ Provide pathways conducts electrical, radio, microwave or light energy as waves or pulses.

→ E.M. Spectrum is used to transmit signal from one device to another. It can travel through a vacuum, air or other transmission media.

→ E.M. energy, a combination of electrical and magnetic field vibrating in relation to each other.

→ Voice band frequencies, transmitted as a current over metal cables.

→ Radio fre. travel through air or space, but require transmitting and receiving mechanism

→ Visible light - E.M. energy use fiber-optic cable.

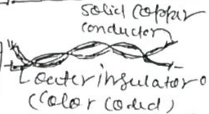
→ Provide conduit from one device to another device.

### Guided media

Provides physical pathway for transmitting the signals.

### Unguided media

### Twisted pair cable



→ Twisted around each other from wire pairs

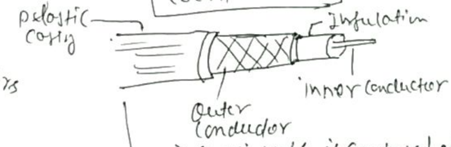
→ 22 to 26 gauge

→ Two pair = four wire

→ nearby pairs of wires carrying signals can interface with each other. This is called CROSS TALK.

→ To reduce cross-talk or other EMF, the wire are twisted.

### Coaxial cable

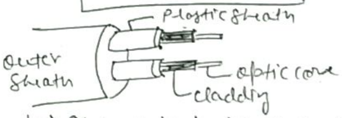


→ coaxial cable is composed of two conductors that share the same axis.

→ The center cable is insulated by plastic foam,

→ a second conductor, foil wrap and an external plastic tube.

### Fiber optics

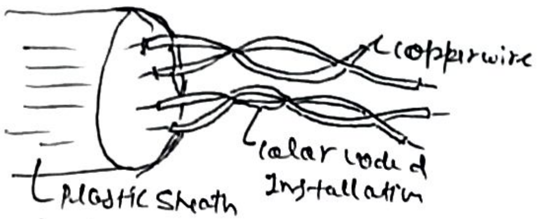


→ It comprised of light-conducting glass or plastic fibers surrounded by a protective cladding and a durable outer sheath.

→ It convert computer signals to/from light pulses.

## Twisted pair cable

### Unshielded Twisted pair (UTP)



→ A set of twisted-pairs within a plastic sheath

→ eg:- Telephone wire

→ different type of UTP cabeling are suitable for different speed communication

→ It's frequency range is suitable for transmitting both data and voice.

→ Twisted pair consist of two conductors (usually copper), each with it's own colored plastic insulation. The plastic insulation is color banded for identification.

→ A number of wiring classification schemes are used —

⇒ category-1: Tradition telephone wire carrying voice but not data, low speed data communication.

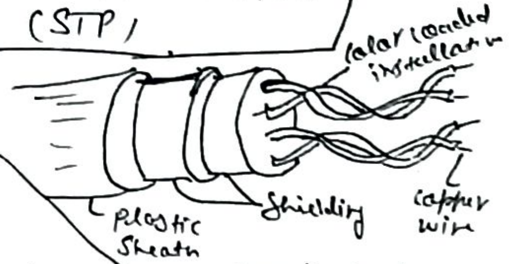
⇒ cat-2: Containing four T.P., data transmission speed upto 4Mbps, Phoneline

⇒ Cat-3: containg four T.P., data transmission speed upto 10Mbps, Phoneline, 3 Twist per foot., Networking

⇒ cat-4: Containg 4 T.P., data transmission speed upto 16 mbps, 3 Twisted per foot.

⇒ cat-5:- Containg - 4 T.P., 100 mbps, Network

### Shielded Twisted pair (STP)



→ STP includes a protetive sheathing around the copper wire.

→ The twisted pair is wrapped in foil to cut down on outside interference and E.M. radiation.

Cost — Moderately expensive

Ease of Installation — more difficult than UTP.

Capacity 500 mbps, most common is 16 mbps.

Attenuation Similar to UTP, limited to 100 m. to 500 m~~eters~~.

EMI immunity Still suffer from outside interference.

## Coaxial Cable

| Use | cable type | Termination |
|---|---|---|
| 10 Base5 (Ethernet) | RG8 and RG11 | 50 Ω |
| 10 Base2 | RG 58 | 50 Ω |
| Cable TV | RG 59 | 75 ohm |
| ARCnet | RG 62 | 93 Ω |

Advantages —

→ EMI Resistance

→ Resilience

Disadvantage

→ ~~Some~~ Some EMI Sensitivity

→ Bulky Installation

→ Relatively expensive

fiber optics

P.T.O.

UTP.

ost - Low cost

se of installation - Relatively inexpensive to other media.

apacity - Data transfer rate 1 to 100 mbps with 10 mbps the most common.

Attenuation - Rapid attenuation, distance limit to 100 of meters.

EMI - Very susceptible to EMI.

... fiberoptics

→ The light pulses are generated by light emitting diodes (LEDs) or injection laser diodes (ILDs)

→ Photodiode reconvert the light pulses to electrical signals.

→ data rates - 100 Mbps to 2 Gbps at distance from 2 to 25 km

→ doesn't carry electricity, ideal for high voltage or secure envim

→ fiberoptics classified based on the diameter of their core. Thicker core allow the signal to reflect from side to side ~~and~~ called multimode, while narrow core fiber cable is reffered to single mode.

Advantage

→ Support extremely high bandwidth.

→ EMI immunity

→ Reliability and security

→ High bandwidth performance

→ Extremely low attenuation

Disadvantage

→ Relatively expensive component parts

→ Installation relatively complex

→ Careful Handling

→ Installation is tedius and Expensive.

Wireless media   wireless media is any media that does not

2 use electrical or optical conductors to transmit and receive electromagnetic signals.

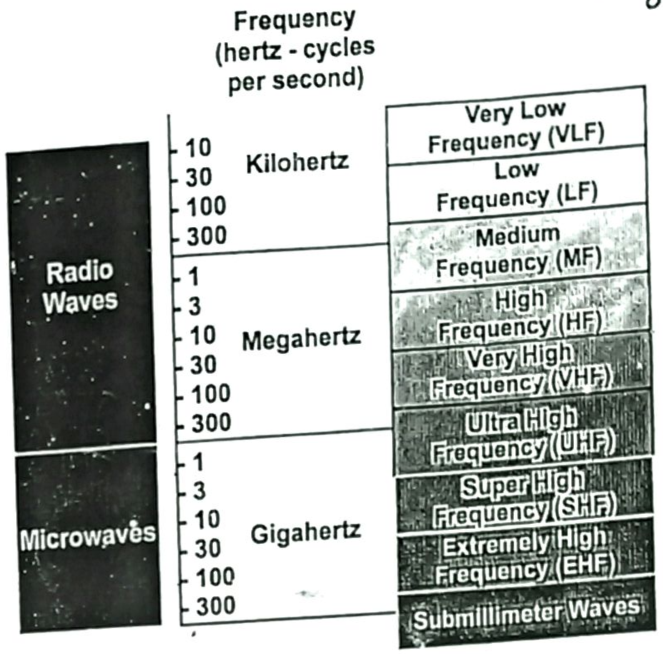
There are three type of energy used to signal over wireless media -

✓ Radio wave

✓ Microwave

✓ Infrared

## Radio Frequency (RF)



### Radio Frequencies

Radio frequency, or RF, refers to broadcast systems that transport electromagnetic waves in the frequency ranges from 10 KHz to 1 GHz. Parts of this frequency range are used for the following signals:

- Short-wave radio
- Very High Frequency television (VHF)
- Frequency Modulation (FM) radio
- Ultra-high frequency radio and television (UHF)

Radio frequencies have been divided into two categories:

Regulated

Users in this category must receive a license from the regulating agency (in the US, the FCC) to maintain transmissions at the desired frequency. Having licensed frequencies guarantees clear transmissions. The licensing process can be cumbersome and costly. RF does not lend itself to flexibility.

Unregulated

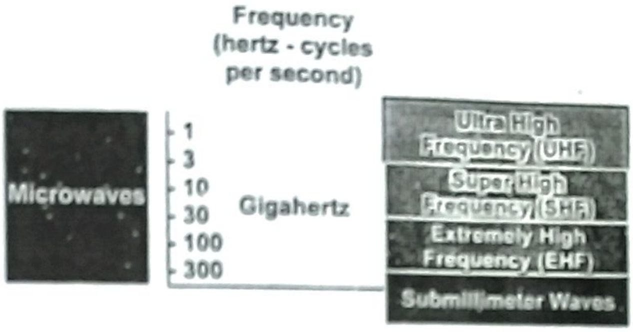
These frequencies (902 to 928 MHz, 2.4 GHz and 5.72 to 5.85 GHz in the US) are not regulated through government agencies. Interference can be met on these frequencies. However, the equipment used for unregulated frequencies must operate with a limited power output. Thus the area covered by each system is small and unlikely to overlap others.

RF waves can either be broadcast in one direction (unidirectional) or all directions (omni-directional). The antenna and the associated transmitter will determine at which frequency the broadcast will be sent. Local systems generally use line-of-site systems, while global systems use short-wave.

There are three classes of radio transmissions:

- Single frequency, low-power
- Single frequency, high-power
- Spread spectrum

## Microwave


Frequency (hertz - cycles per second)
Microwaves
1
3
10
30
100
300
Gigahertz
Ultra High Frequency (UHF)
Super High Frequency (SHF)
Extremely High Frequency (EHF)
Submillimeter Waves

Microwave Frequencies

### Terrestrial Microwave

Using terrestrial microwave requires the installation of directional line-of-site parabolic antennas. This topology is used when cabling would be troublesome or impossible. Terrestrial microwave uses regulated frequencies that add time and expense to the installation. Omnidirectional antennas can be used for microwave transmission inside buildings as well.

| | |
|---|---|
| Frequency Range | Generally operates between 4 to 6 or 21 to 23 GHz. |
| Cost | Depends on whether the network will be using short- or long-distance equipment. Long-distance equipment (including licensing fees) can be very expensive. |
| Ease of Installation | Very difficult and expensive to install because of the line-of-site limitation. |
| Capacity | Data rates range from less than 1 Mbps up to 10 Mbps. |
| Attenuation | Long-distance systems will be affected by rain and fog. Short-distance systems are not affected. |
| EMI Immunity | All air transmitted systems are highly susceptible to EMI. Microwave is susceptible to atmospheric conditions as well. |

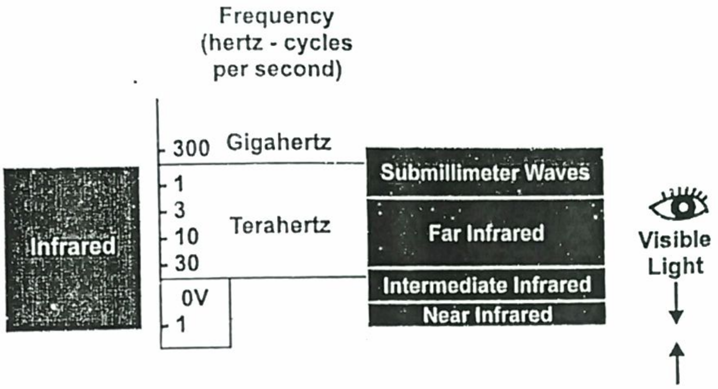
### Satellite Microwave

Satellite microwave requires the installation of beamed line-of-site parabolic antennas on the ground and on geosynchronous satellites. A basic system includes the antennas/receiving devices located at source and destination sites plus a licensed frequency to an orbiting satellite.

Because the signal must travel to a satellite 22,300 miles above the earth and back, some time is required. This time is known as the propagation delay. In satellite systems, this delay can range from half a second to five seconds or more.

| | |
|---|---|
| Frequency Range | Generally operates in the 11- to 14-GHz range. |
| Cost | Relatively high cost. |
| Ease of Installation | Extremely difficult. |
| Capacity | Data rates for a single-frequency system generally range from less than 1 to 10 Mbps. Because of the distance traversed there will be noticeable propagation delays. |
| Attenuation | Atmospheric conditions affect the attenuation. |
| EMI Immunity | All air transmitted systems are highly susceptible to EMI. Microwave is susceptible to atmospheric conditions as well. |

## Infrared Systems



### Infrared Frequencies

Infrared systems use light emitting diodes (LEDs) or injection laser diodes (ILDs) to transmit data between two sites. The light beams can be line-of-site or can be reflected off walls or ceilings. The signals cannot penetrate either solid or opaque surfaces and are affected by strong light sources. High-frequency systems can be used to transmit data at high transmission rates. There are four types of infrared systems:

- Line of sight networks

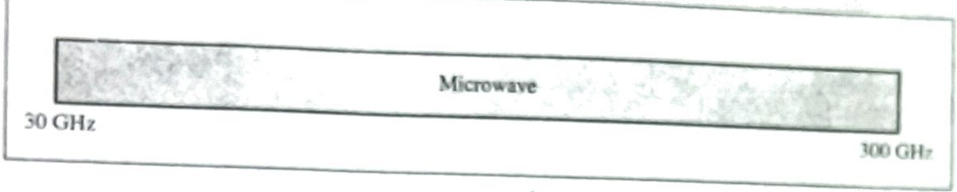
  The path of the infrared signal must be unubstructed between the transmitter and receiver.

- Scatter infrared networks

  Transmissions are reflected by floors, walls and ceilings. One hundred feet is the limitation for this technology.

**EHF** **Extremely high frequency (EHF)** waves use space propagation. Uses for EHF are predominantly scientific and include radar, satellite, and experimental communications (see Figure 7.30).

**Figure 7.30** *Frequency range for EHF*



## Terrestrial Microwave

**Microwaves** do not follow the curvature of the earth and therefore require line-of-sight transmission and reception equipment. The distance coverable by a line-of-sight signal depends to a large extent on the height of the antenna: the taller the antennas, the longer the sight distance. Height allows the signal to travel farther without being stopped by the curvature of the planet and raises the signal above many surface obstacles, such as low hills and tall buildings that would otherwise block transmission. Typically, antennas are mounted on towers that are in turn often mounted on hills or mountains.

Microwave signals propagate in one direction at a time, which means that two frequencies are necessary for two-way communication such as a telephone conversation. One frequency is reserved for **microwave transmission** in one direction and the other for transmission in the other. Each frequency requires its own transmitter and receiver. Today, both pieces of equipment usually are combined in a single piece of equipment called a transceiver, which allows a single antenna to serve both frequencies and functions.

### Repeaters

To increase the distance served by **terrestrial microwave,** a system of repeaters can be installed with each antenna. A signal received by one antenna can be converted back into transmittable form and relayed to the next antenna (see Figure 7.31). The distance required between repeaters varies with the frequency of the signal and the environment in which the antennas are found. A repeater may broadcast the regenerated signal either at the original frequency or at a new frequency, depending on the system.

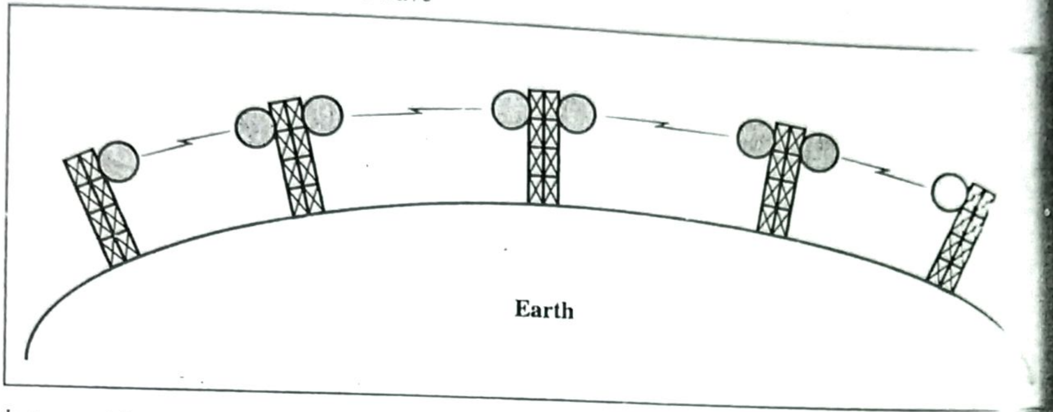
Terrestrial microwave with repeaters provides the basis for most contemporary telephone systems worldwide.

### Antennas

Two types of antennas are used for terrestrial microwave communications: parabolic dish and horn.

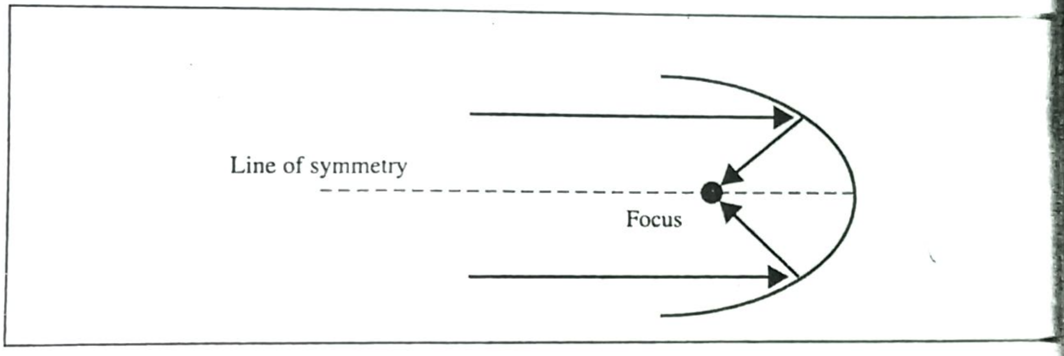
A **parabolic dish antenna** is based on the geometry of a parabola: every line parallel to the line of symmetry (line of sight) reflects off the curve at angles such that they

**Figure 7.31** *Terrestrial microwave*



intersect in a common point called the focus (see Figure 7.32). The parabolic dish works like a funnel, catching a wide range of waves and directing them to a common point. In this way, more of the signal is recovered than would be possible with a single-point receiver.

**Figure 7.32** *Parabolic dish antenna*



Outgoing transmissions are broadcast through a horn aimed at the dish. The microwaves hit the dish and are deflected outward in a reversal of the receipt path.

A **horn antenna** looks like a gigantic scoop. Outgoing transmissions are broadcast up a stem (resembling a handle) and deflected outward in a series of narrow parallel beams by the curved head (see Figure 7.33). Received transmissions are collected by the scooped shape of the horn, in a manner similar to the parabolic dish, and are deflected down into the stem.

## Satellite Communication

Satellite transmission is much like line-of-sight microwave transmission in which one of the stations is a satellite orbiting the earth. The principle is the same as terrestrial microwave, with a satellite acting as a supertall antenna and repeater (see Figure 7.34). Although in satellite transmission signals must still travel in straight lines, the limitations imposed on distance by the curvature of the earth are reduced. In this way, satellite relays allow microwave signals to span continents and oceans with a single bounce.

Figure 7.33 *Horn antenna*

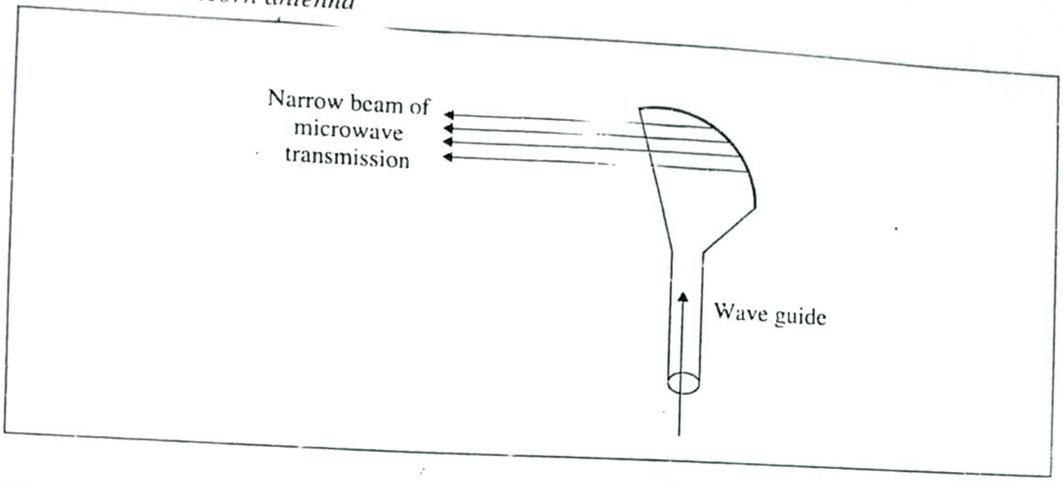
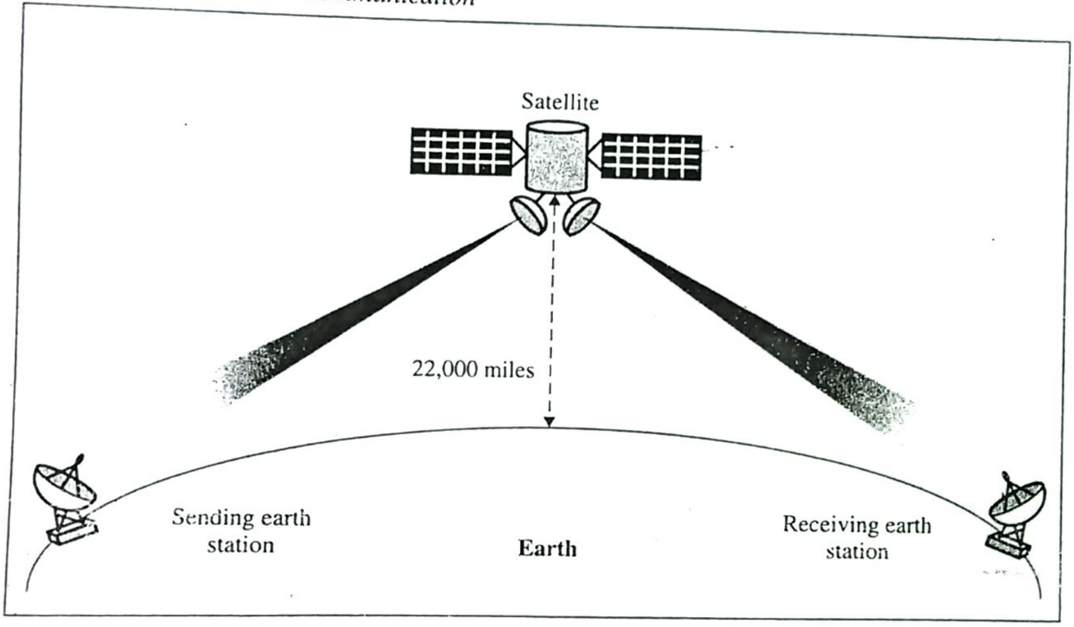


Figure 7.34 *Satellite communication*



Satellite microwave can provide transmission capability to and from any location on earth, no matter how remote. This advantage makes high-quality communication available to undeveloped parts of the world without requiring a huge investment in ground-based infrastructure. Satellites themselves are extremely expensive, of course, but leasing time or frequencies on one can be relatively cheap.

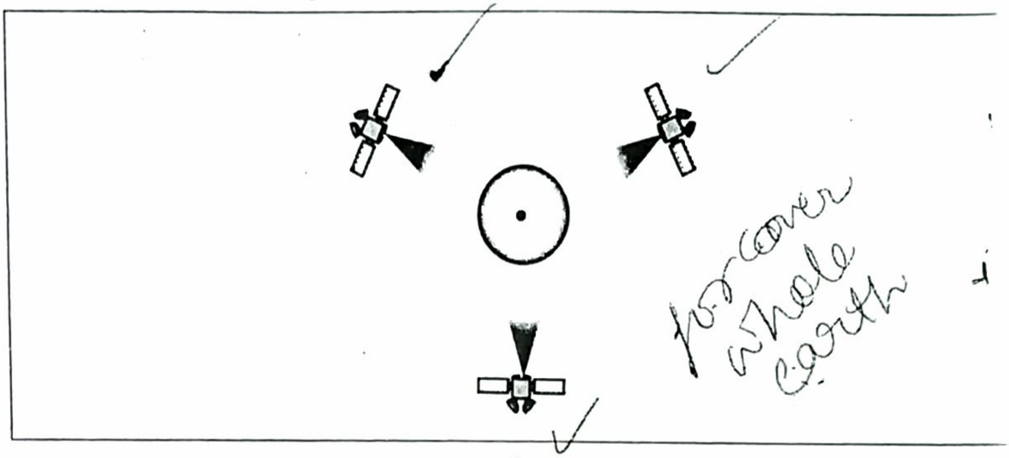
### Geosynchronous Satellites

Line-of-sight propagation requires that the sending and receiving antennas be locked onto each other's location at all times (one antenna must have the other in sight). For this reason, a satellite that moves faster or slower than the earth's rotation is useful only for short periods of time (just as a stopped clock is accurate twice a day). To ensure constant communication, the satellite must move at the same speed as the earth so that it seems to remain fixed above a certain spot. Such satellites are called geosynchronous.

Because orbital speed is based on distance from the planet, only one orbit can be geosynchronous. This orbit occurs at the equatorial plane and is approxima 22,000 miles from the surface of the earth.

But one geosynchronous satellite cannot cover the whole earth. One satelli orbit has line-of-sight contact with a vast number of stations, but the curvature of earth still keeps much of the planet out of sight. It takes a minimum of three sate equidistant from each other in **geosynchronous orbit** to provide full global transmission. Figure 7.35 shows three satellites, each 120 degrees from another in geosynchronous orbit around the equator. The view is from the North Pole.

**Figure 7.35** *Satellites in geosynchronous orbit*



for cover
whole
earth

#### Frequency Bands for Satellite Communication

The frequencies reserved for satellite microwave communication are in the gigahertz (GHz) range. Each satellite sends and receives over two different bands. Transmission from the earth to the satellite is called **uplink.** Transmission from the satellite to the earth is called **downlink.** Table 7.2 gives the band names and frequencies for each range.

**Table 7.2** *Satellite frequency bands*

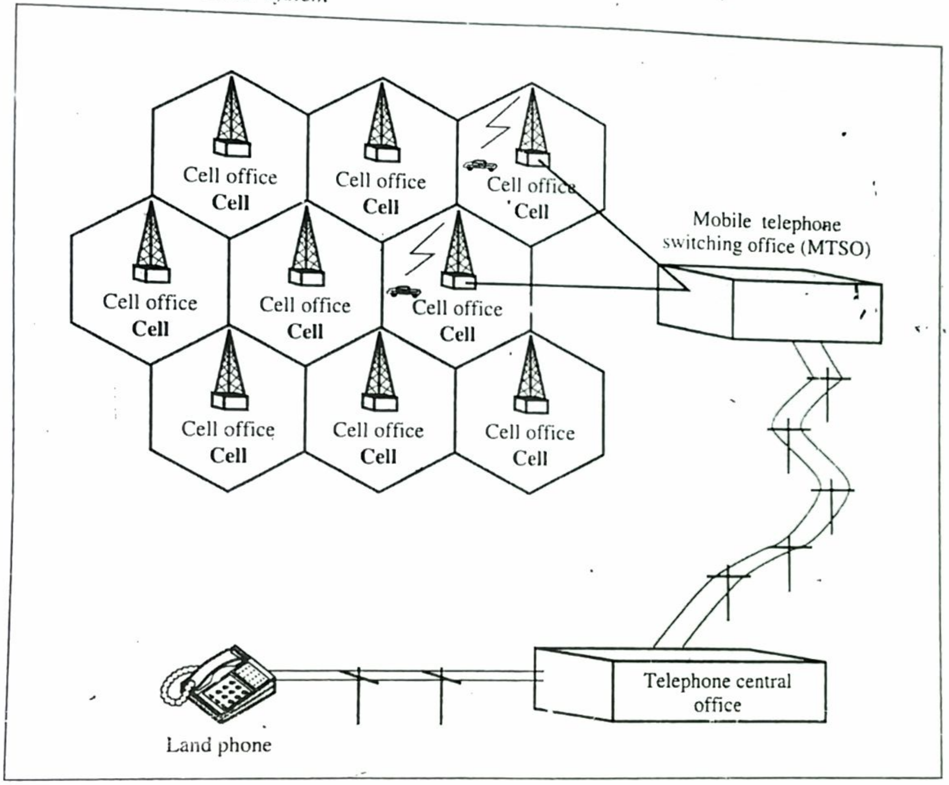
| *Band* | *Downlink* | *Uplink* |
|---|---|---|
| C | 3.7 to 4.2 GHz | 5.925 to 6.425 GHz |
| Ku | 11.7 to 12.2 GHz | 14 to 14.5 GHz |
| Ka | 17.7 to 21 GHz | 27.5 to 31 GHz |

## Cellular Telephony

**Cellular telephony** is designed to provide stable communications connections two moving devices or between one mobile unit and one stationary (land) u vice provider must be able to locate and track a caller, assign a channel to the transfer the signal from channel to channel as the caller moves out of the range channel and into the range of another.

To make this tracking possible, each cellular service area is divided regions called cells. Each cell contains an antenna and is controlled by a small called the cell office. Each cell office, in turn, is controlled by a switching office called a **mobile telephone switching office (MTSO).** The MTSO coordinates communication between all of the cell offices and the telephone central office. It is a computerized center that is responsible for connecting calls as well as recording call information and billing (see Figure 7.36).

**Figure 7.36** *Cellular system*

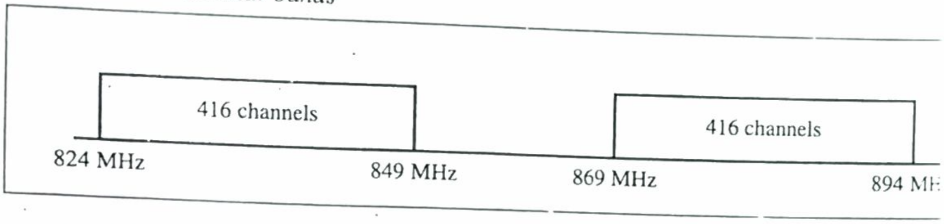


Cell size is not fixed and can be increased or decreased depending on the population of the area. The typical radius of a cell is 1 to 12 miles. High-density areas require more, geographically smaller cells to meet traffic demands than do lower density areas. Once determined, cell size is optimized to prevent the interference of adjacent cell signals. The transmission power of each cell is kept low to prevent its signal from interfering with those of other cells.

## Cellular Bands

Traditional cellular transmission is analog. To minimize noise, frequency modulation (FM) is used for communication between the mobile telephone itself and the cell office. The FCC has assigned two bands for cellular use (see Figure 7.37). The band between 824 and 849 MHz carries those communications that initiate from mobile phones. The band between 869 and 894 MHz carries those communications that initiate from land phones. Carrier frequencies are spaced every 30 KHz, allowing each band to support up to 833 carriers. However, two carriers are required for full-duplex communic which doubles the required width of each channel to 60 KHz and leaves only 416 nels available for each band.

**Figure 7.37** *Cellular bands*



Each band, therefore, is divided into 416 FM channels (for a total of 832 chann Of these, some are reserved for control and setup data rather than voice commur tion. In addition, to prevent interference, channels are distributed among the cel such a way that adjacent cells do not use the same channels. This restriction means each cell normally has access to only 40 channels.

### Transmitting

To place a call from a mobile phone, the caller enters a code of 7 or 10 digits (a ph number) and presses the send button. The mobile phone then scans the band, seeki setup channel with a strong signal, and sends the data (phone number) to the clos cell office using that channel. The cell office relays the data to the MTSO. The MT sends the data on to the telephone central office. If the called party is available, a nection is made and the result is relayed back to the MTSO. At this point, the MT assigns an unused voice channel to the call and a connection is established. The mo phone automatically adjusts its tuning to the new channel and voice communica can begin.

### Receiving

When a land phone places a call to a mobile phone, the telephone central office se the number to the MTSO. The MTSO searches for the location of the mobile phone sending query signals to each cell in a process called paging. Once the mobile phon found, the MTSO transmits a ringing signal and, when the mobile phone is answer assigns a voice channel to the call, allowing voice communication to begin.

### Handoff

It may happen that, during a conversation, the mobile phone moves from one cel another. When it does, the signal may become weak. To solve this problem, the MT monitors the level of the signal every few seconds. If the strength of the signal dim ishes, the MTSO seeks a new cell that can accommodate the communication better. T MTSO then changes the channel carrying the call (hands the signal off from the channel to a new one). Handoffs are performed so smoothly that most of the time th are transparent to the users.

### Digital

Analog (FM) cellular services are based on a standard called analog circuit switched cellular (ACSC). To transmit digital data using an ACSC service requires a modem with a maximum speed of 9600 to 19,200 bps.

Since 1993, however, several service providers have been moving to a cellular data standard called cellular digital packet data (CDPD). CDPD provides low-speed digital service over the existing cellular network. It is based on the OSI model.

To use the existing digital services, such as 56K switched service, CDPD uses what is called a trisector. A trisector is a combination of three cells each using 19.2 Kbps, for a total of 57.6 Kbps (which can be accommodated on a 56K switched line by eliminating some overhead). Under this scheme, the United States is divided into 12,000 trisectors. For every 60 trisectors, there is one router.

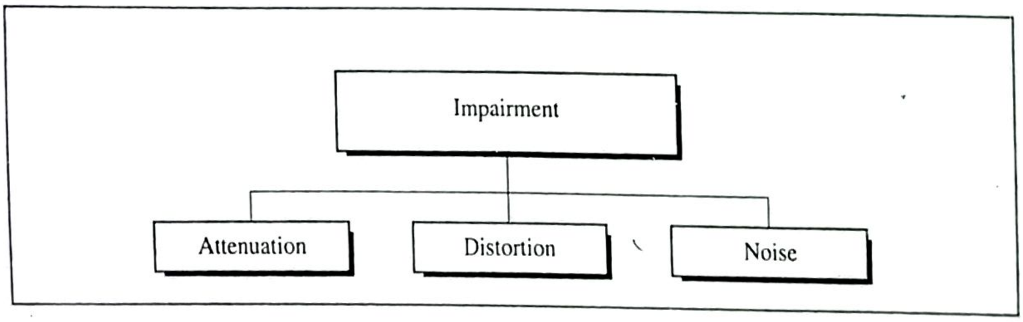
### Integration with Satellites and PCs

Cellular telephony is moving fast toward integrating the existing system with satellite communication. This integration will make it possible to have mobile communication between any two points on the globe. Another goal is to combine cellular.telephony and personal computer communication under a scheme called mobile personal communication to enable people to use small, mobile personal computers to send and receive data, voice, image, and video.

## 7.3 TRANSMISSION IMPAIRMENT

**Transmission media** are not perfect. The imperfections cause impairment in the signal sent through the medium. This means that the signal at the beginning and end of the medium are not the same. What is sent is not what is received. Three types of impairment usually occur: attenuation, distortion, and noise (see Figure 7.38).

**Figure 7.38** *Impairment types*

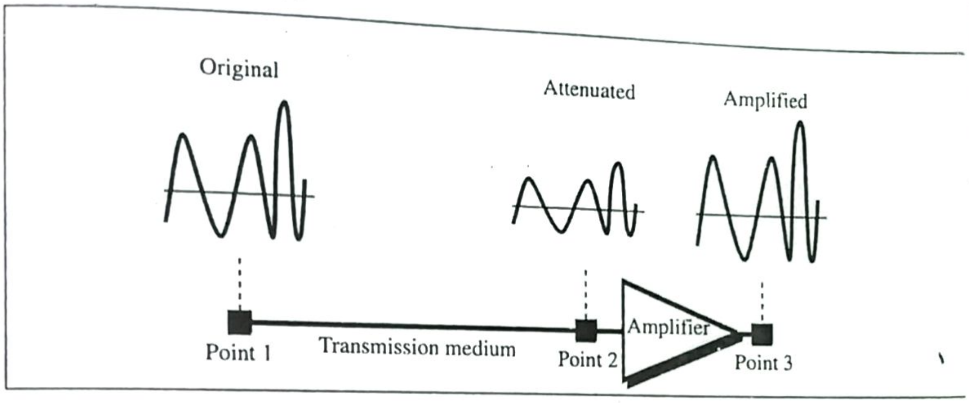


### Attenuation

**Attenuation** means loss of energy. When a signal, simple or complex, travels through a medium, it loses some of its energy so that it can overcome the resistance of the medium. ~~That is why wire carrying electrical~~
~~Signals get~~

medium. That is why a wire carrying electrical signals gets warm, if not hot, after a while. Some of the electrical energy in the signal is converted to heat. To compensate for this loss, amplifiers are used to amplify the signal. Figure 7.39 shows the effect of attenuation and amplification.

**Figure 7.39** *Attenuation*



### Decibel

To show that a signal has lost or gained strength, engineers use the concept of decibel. The **decibel (dB)** measures the relative strengths of two signals or a signal at two different points. Note that the dB is negative if a signal is attenuated and positive if a signal is amplified.

$$\text{dB} = 10 \log_{10} (P_2/P_1)$$

where $P_1$ and $P_2$ are the power of a signal at points 1 and 2.

### Example 7.1

Imagine a signal travels through a transmission medium and its power is reduced to half. This means that $P_2 = (1/2)P_1$. In this case, the attenuation (loss of power) can be calculated as

$$10 \log_{10} (P_2/P_1) = 10 \log_{10} (0.5\, P_1/P_1) = 10 \log_{10} (0.5) = 10\,(-0.3) = -3 \text{ dB}$$

Engineers know that –3 dB, or a loss of 3 dB, is equivalent to losing half the power.

### Example 7.2

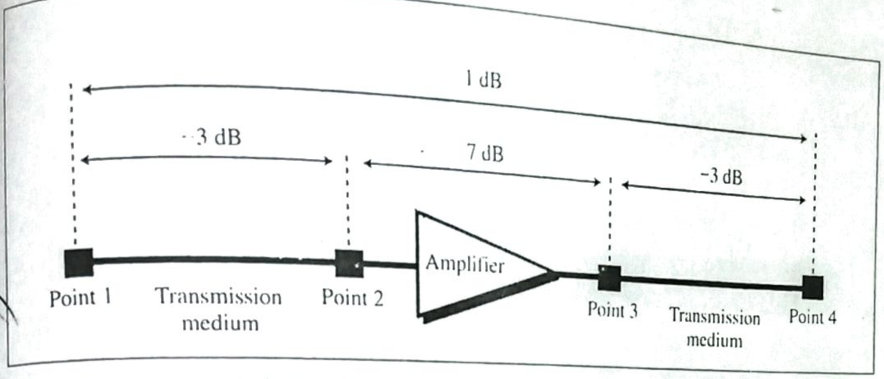
Imagine a signal travels through an amplifier and its power is increased 10 times. This means that $P_2 = 10 \times P_1$. In this case the amplification (gain of power) can be calculated as

$$10 \log_{10} (P_2/P_1) = 10 \log_{10} (10\, P_1/P_1) = 10 \log_{10} (10) = 10\,(1) = 10 \text{ dB}$$

### Example 7.3

One of the reasons that engineers use the decibel to measure the changes in the strength of a signal is that decibel numbers can be added (or subtracted) when we are talking about several points instead of just two (cascading). In Figure 7.40 a signal travels a long distance from point 1 to point 4. The signal is attenuated by the time it reaches point 2. Between points 2 and 3, the signal is amplified. Again, between points 3 and 4, the signal is attenuated. We can find the resultant dB for the signal just by adding the dB measurements between each set of points.

**Figure 7.40** *Example 7.3*



In this case, the decibel can be calculated as

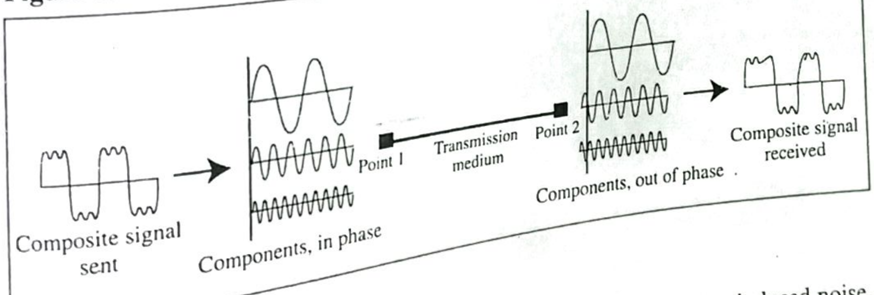
$$dB = -3 + 7 - 3 = +1$$

which means that the signal has gained power.

## Distortion

**Distortion** means that the signal changes its form or shape. Distortion occurs in a composite signal, made of different frequencies. Each signal component has its own propagation speed (see the next section) through a medium and, therefore, its own delay in arriving at the final destination. Figure 7.41 shows the effect of distortion on a composite signal.

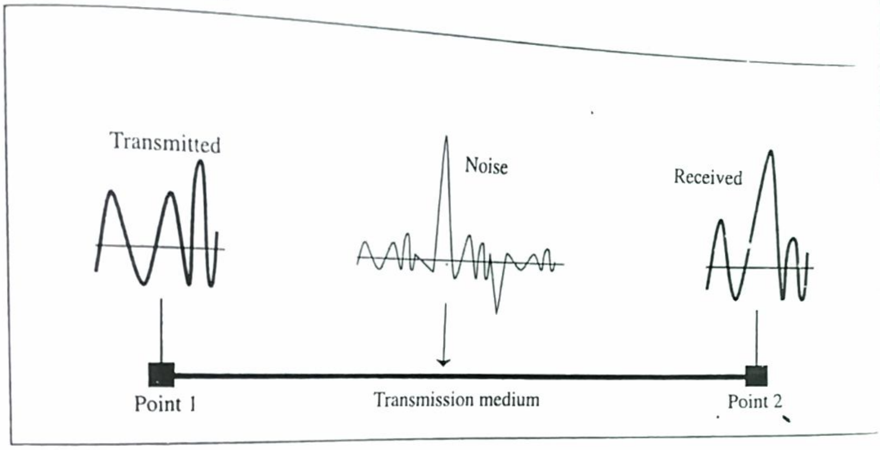
**Figure 7.41** *Distortion*



## Noise

**Noise** is another problem. Several types of noise such as thermal noise, induced noise, crosstalk, and impulse noise may corrupt the signal. Thermal noise is the random motion of electrons in a wire that creates an extra signal not originally sent by the transmitter. Induced noise comes from sources such as motors and appliances. These devices act as a sending antenna and the transmission medium acts as the receiving antenna. Crosstalk is the effect of one wire on the other. One wire acts as a sending antenna and the other as the receiving antenna. Impulse noise is a spike (a signal with high energy in a very short period of time) that comes from power lines, lightning, and so on. Figure 7.42 shows the effect of noise on a signal.

Figure 7.42 *Noise*



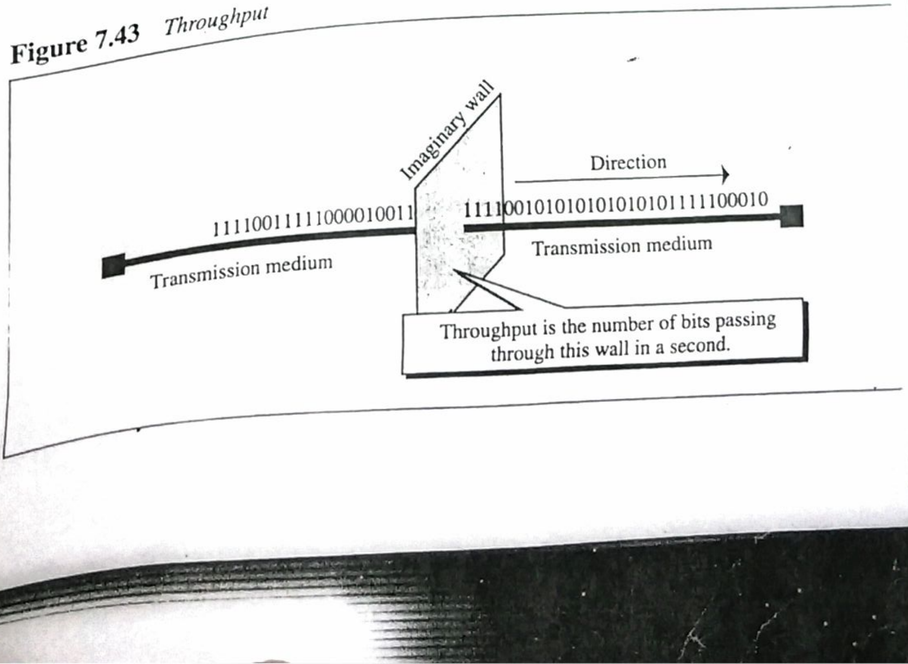
## 7.4 PERFORMANCE

Transmission media are roads on which data travel. To measure the performance of transmission media, we can use three concepts: throughput, propagation speed, and propagation time.

### Throughput

The **throughput** is the measurement of how fast data can pass through a point. In other words, if we consider any point in the transmission medium as a wall through which bits pass, throughput is the number of bits that can pass this wall in one second. Figure 7.43 shows the concept.

Figure 7.43 *Throughput*



### Propagation Speed

**Propagation speed** measures the distance a signal or a bit can travel through a medium in one second. The propagation speed of electromagnetic signals depends on the medium and the frequency of the signal. For example, in a vacuum, light is propagated with a speed of $3 \times 10^8$ m/s. It is almost the same in a twisted-pair cable. However, in coaxial and fiber optic cables, the speed is $2 \times 10^8$ m/s for frequencies in the MHz to GHz range.

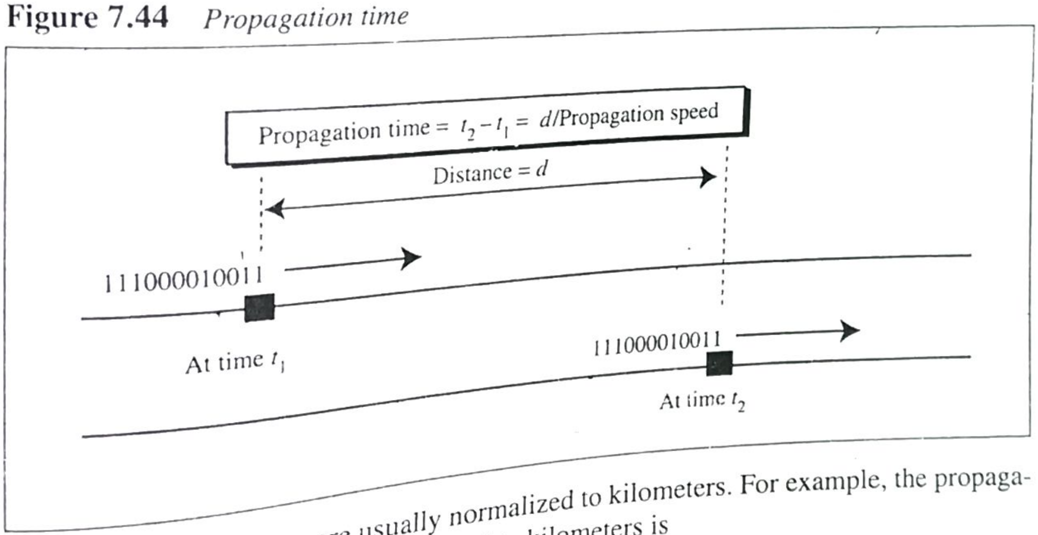
### Propagation Time

**Propagation time** measures the time required for a signal (or a bit) to travel from one point of the transmission medium to another. The propagation time is calculated by dividing the distance by the propagation speed.

$$\text{Propagation time} = \text{Distance}/\text{Propagation speed}$$

Figure 7.44 shows the concept.

**Figure 7.44** *Propagation time*



Propagation times are usually normalized to kilometers. For example, the propagation time for a twisted pair normalized to kilometers is

$$\text{Propagation time} = 1000\text{ m}/(3 \times 10^8\text{ m/s}) = 3.33 \times 10^{-6}\text{ s/m} = 3.33\ \mu\text{s/km}$$

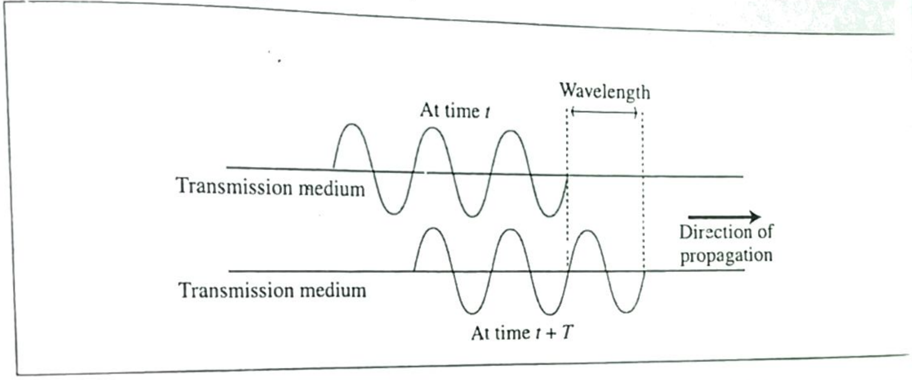
For coaxial or fiber optic cable, it is usually

$$\text{Propagation time} = 1000\text{ m}/(2 \times 10^8\text{ m/s}) = 5 \times 10^{-6}\text{ s/m} = 5\ \mu\text{s/km}$$

## 7.5 WAVELENGTH

**Wavelength** is another characteristic of a signal traveling through a transmission medium. Wavelength binds the period or the frequency of a simple sine wave to the propagation speed of the medium. In other words, while the frequency of a signal is independent of the medium, the wavelength depends on both the frequency and the medium. Although wavelength can be associated with electrical signals, it is customar to use wavelengths when talking about the transmission of light in an optical fiber. The wavelength is the distance a simple signal can travel in one period (see Figure 7.45).

Figure 7.45 *Wavelength*



Wavelength can be calculated given the propagation speed and the period of the signal

$$\text{Wavelength} = \text{Propagation speed} \times \text{Period}$$

However, since period and frequency are related to each other, we can also say

$$\text{Wavelength} = \text{Propagation speed} \times (1/\text{Frequency}) = \text{Propagation speed/Frequency}$$

If we represent wavelength by $\lambda$, propagation speed by $c$, and frequency by $f$, we get

$$\lambda = c/f$$

The wavelength is normally measured in micrometers (microns) instead of meters. For example, the wavelength of red light (frequency = $4 \times 10^{14}$) in air is

$$\lambda = c/f = (3 \times 10^8)/(4 \times 10^{14}) = 0.75 \times 10^{-6} \text{ m} = 0.75 \text{ }\mu\text{m}$$

In a coaxial or fiber-optic cable, however, the wavelength is lower (0.5 μm) because the propagation speed in the cable is less than in the air.

## 7.6 SHANNON CAPACITY

Engineers are often interested in the maximum data rate of a channel. In 1944, Claude Shannon introduced a formula to determine the theoretical highest data rate for a channel:

$$C = B \log_2 (1 + S/N)$$

In this formula, $B$ is the bandwidth of the channel, $S/N$ is the signal to noise ratio, and $C$ is the capacity (called the **Shannon capacity**) of the channel in bps.

## 9.7 Multiplexing

9

In a data communication network, the task of the network designers is to select and co-ordinate the functioning of the network components so that the necessary data are moved to the right place, at the right time, with a minimum of errors and at lowest possible cost.

There are many applications in which several terminals are connected to a computer. If each terminal is operating at a speed of 300 bits per second over a communication channel that can operate at 9600 bits per second, it is obvious that the channel is not being used efficiently.

A channel is an expensive resource, so it should be as fully utilised as possible. To achieve this, a channel is divided into many sub-channels so that multiple signals can be simultaneously transmitted over it. This means that a system has to be designed which allows many individual messages to be transmitted simultaneously over a single communication channel.

The method of dividing a physical channel into many logical channels so that a number of independent signals may be simultaneously transmitted on it is known as *multiplexing*. The electronic device that performs this task is known as a *multiplexer*.

Multiplexing allows several users to share a communication channel. Several data signals are combined into a single signal and transmitted to the sending location.

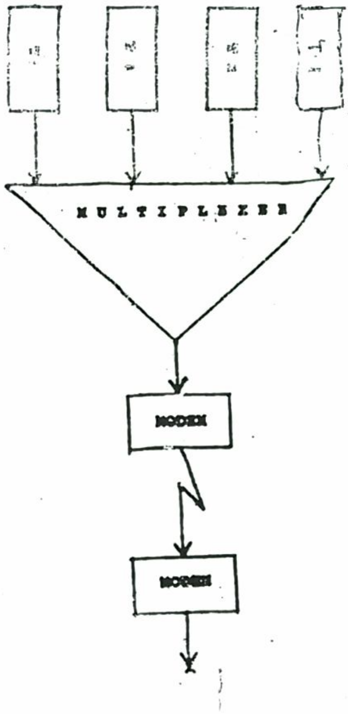


**Figure 9.10**
**Multiplexers at work**

2

a)

In this case four terminals are connected to a multiplexer. The multiplexer takes the signals from the 4 terminals and combines them into a single signal. This signal is then transmitted over the communication line.

At the receiving end, a multiplexer receives this signal and converts it back into the 4 original signals. The process of getting back the original signals from a single signal is called *demultiplexing.*

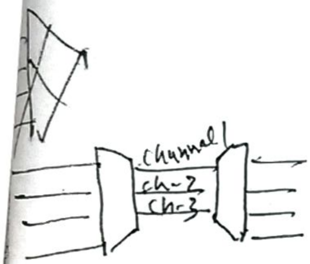
Without multiplexers, four separate communication lines would have been required for simultaneous transmission.

The following describes few of the most commonly used techniques to achieve multiplexing.

## (1) Frequency Division Multiplexing (FDM)

In frequency division multiplexing the available bandwidth of the physical communication channel can be thought of as divided into several smaller disjoint logical channels with each channel having a small bandwidth. Many signals can be transmitted over these available communication lines.

The best example of FDM is the way in which we receive various stations in a radio. Each radio station is assigned a frequency range within a band of radio frequency. A radio receiver's antenna receives signals transmitted by all the stations. Finally the tuning dial in the radio is used to isolate the speech signal of the station tuned.



In order to use FDM

1. The signals to be transmitted must be analog signals. Thus if digital signals are to use FDM, they must be converted to analog form.
2. All the signals in the channel travel simultaneously.
3. The physical communication channel can be thought of as consisting of smaller logical channels. Each logical channel is distinct and carries an independent signal. A modulator is needed for each of the logical channels. For two way communication a modem is placed at each end of the channel.

## (2) Time Division Multiplexing (TDM)

popular method of utilising the capacity of a Time division multiplexing is another popular method of utilising the capacity of a channel effectively. Each user is allotted a time interval during which he may transmit a message. Thus the total time available in the channel is divided and each user is allocated a time slice.

The amount of data that can be transmitted during this time slice is a function of bandwidth of the channel. Suppose the bandwidth of the channel for digital data is 9.6 KBPS and the allocated time slice is 100 ms, then a packet of 960 bits may be transmitted during the allocated time. If there are 10 users, then after sending one packet the next turn for a user to send a packet will come after 1 second.

In FDM a number of messages can be sent in parallel. Whereas in TDM messages are sent sequentially. However, each user can use the full channel bandwidth. The channel capacity is fully utilised in TDM by interleaving a number of messages belonging to different users into one long message. This message sent through the channel must be seperated at the receiving end. individual packets of the messages send by each other should be re-assembled in to a full messages.

TDM can be categarized in two Type

(a) Synchronous TDM

(b) Asynchronous TDM

## a) Synchronous TDM :-

→ Here Synchronous means that the multiplexer ~~allocated~~ allocates exactly the Same time slot to each device at all times, whether or not a device has anything to transmit.

→ Time slot A, for example, is assigned to device A alone and cannot be used by any other device.

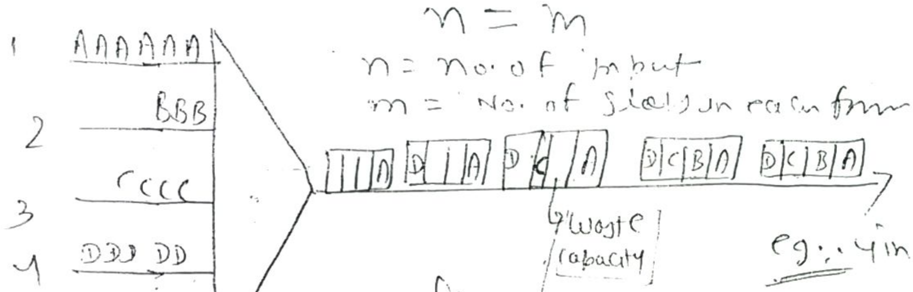
→ Each time it's allocated time slot comes up, a device has the opportunity to send a portion of it's data.

→ If a device is unable to transmit or ~~does~~ does not have data to send, its time slot remain empty.

$n = m$

n = no. of input

m = No. of slots in each fram



eg:: 4 input line = 4 slots in a fram

fig:- Synchronous TDM

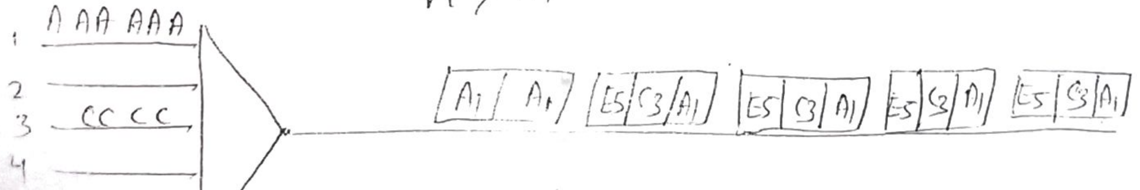
## (b) Asynchronous TDM

→ It avoids waste of capacity as compared than Synchronous TDM

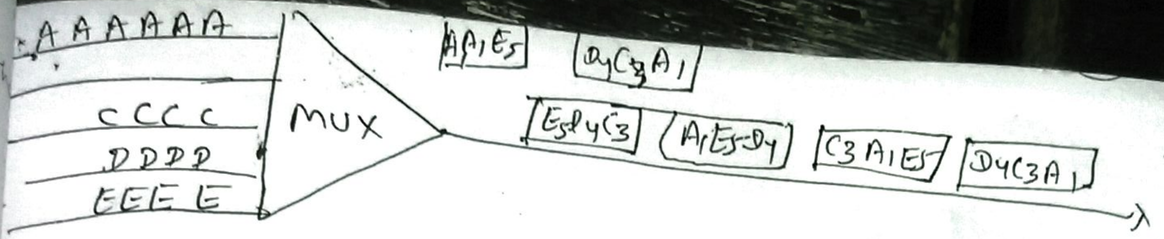
→ Unlike Synchronous TDM however in asynchronous TDM the total speed of the input lines can be greater than the capacity of the path.

→ In an asynchronous system if we have n input lines, the frame contains no more than m slots

$n > m$



Asynchronous TDM



→ Asynchronous TDM can accomodate traffic of varying data rates by varying the length of time slots.

## (3) Code Division Multiplexing (CDM)

Code Division Multiplexing (CDM) uses identifying codes to distinguish one signal from another on a shared medium.

Each signal is assigned a sequence of bits called the spreading code that is combined with the original signal to produce a new stream of encoded data; a receiver that knows the code can retrieve the original signal by subtracting out the spreading code (a process called *dispreading*).

CDM is widely used in digital television and radio broadcasting and in 3G mobile cellular networks. Where CDM allows multiple signals from multiple sources, it is called Code-Division Multiple Access (CDMA).

## (4) Space-division multiplexing

In wired communication, space-division multiplexing is the use of separate point-to-point electrical conductors for each transmitted channel.

In wireless communication, space-division multiplexing is achieved with multiple antenna elements forming a phased array antenna.

Different antennas would give different multi-path propagation (echo) signatures, making it possible for digital signal processing techniques to separate different signals from each other.
These techniques may also be utilized for space diversity (improved robustness to fading) or beamforming (improved selectivity) rather than multiplexing.

## Wave - Division Multiplexing (WDM) (5)

→ It is conceptually the same of FDM, except that the multiplexing and demultiplexing involve light signals transmitted through fiber optic channels.

→ In this combine multible light sources into one single light at the multiplexer and do the reverse at the demultiplexer.

→ combining and spliting of light sources are easily handled by a prism.

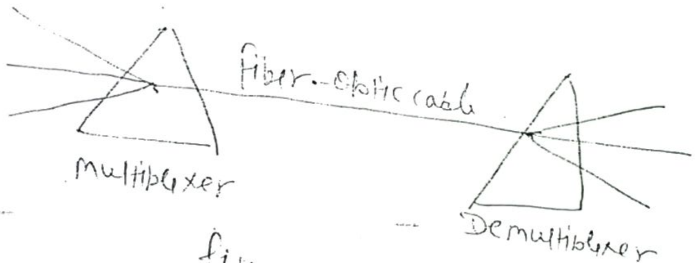


fig:- Prism in WDM multiplexing & Demultiplexing.

AAAAA
CCCC
DDDD
EEEE
→ Async
data

(3) **Code Divisio**
Code Divisio
from anoth

Each signa
combined
receiver th
the spread

CDM is wi
cellular ne
called Cod

(4) **Space-div**
In wired
to-point e

In wirele
antenna

Different
making
signals fr