UNIT 2 CHAPTER 1 CONCEPTS OF MULTIPLEXING

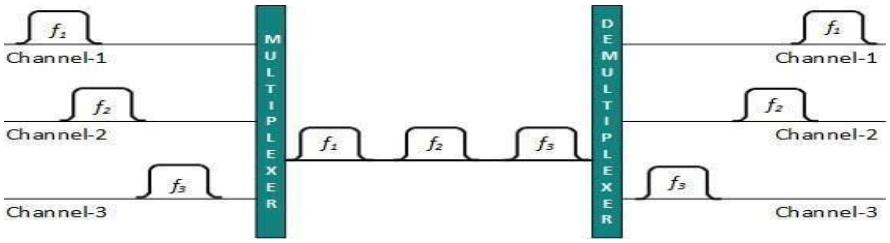
2.1 Multiplexers

It is used in a network for maximum transmission capacity of a high bandwidth line. Multiplexing is technique that allows many communication sources to transmit data over a single physical line. The basic categories in multiplexing are:

1. Frequency division multiplexing(FDM)
2. Wavelength division multiplexing(WDM)
3. Time division multiplexing(TDM)

1. Frequency division multiplexing(FDM):

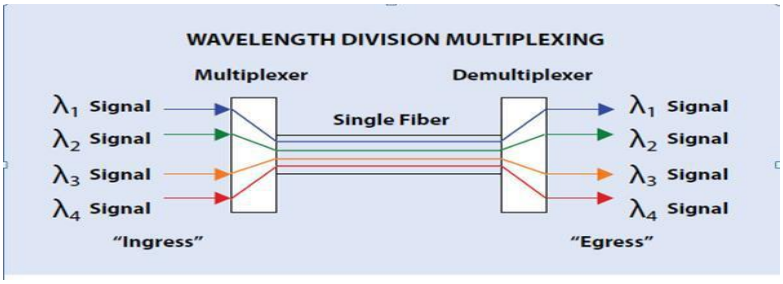
The frequency spectrum is divided into frequency bands or channels, in which each user can be assigned a band.



When many channels are multiplxed together, a certain gaurd band is allowed to keep the channels well separated. To implement a multiplexer, the original frequency at any of 'n' inputs of the multiplexer are raised. Each by different constant amount. Then the 'n' new frequency bands are combined to let no channels occupy the same portion of the spectrum. FDM is normally used over copper wires or microwave channels and is suitable for analog circuitry.

2. Wavelength division multiplexing(WDM)

WDM is inverted as a variation of FDM and is basically a multiplexing method of different wavelengths instead of frequencies.



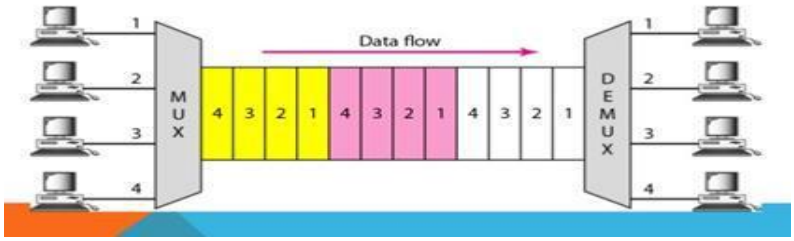
In figure, 'n' optical fiber connected together at an optical multiplexer, each with different wave length. The 'n' optical lines are combined on to a single shared link for transmission to a distant destination. At the

demultiplexer each frame including ‘n’ channels is split up over as many optical fibers as there were on the input side. At output of demultiplexer a tuned filter refines the desired signal at the tuned wavelength and all the other wavelengths are bypassed.

The main issue of WDM compared with FDM is than an optical system using a diffraction grating is completely passive and thus highly reliable.

3. Time division multiplexing (TDM).

Here, time is divided into frames and each frame is further subdivided into time slots. Each channel is allocated to one input. This type of multiplexing can be used only for digital data. Packets arrives on ‘n’ lines of input. Each packet size is variable. To multiplex variable sized packets, additional hardware is needed for efficient scanning and synchronization. Thus TDM can be synchronous or statistical.



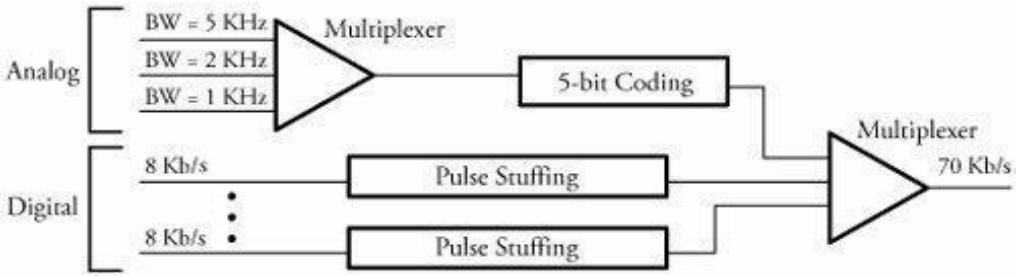
A. Synchronous TDM

In this, the multiplexer scans all lines without exception. The scanning time for each line is pre allocated that is, as long as this time is not altered by system control, the scanner should stay on that line whether or not there is a data for scanning in that time slot. Hence synchronous multiplexer does not operate efficiently, though its complexity is low.

In addition, the bit rates of analog and digital data being combined in a multiplexer sometimes need to be synchronized. In such cases, dummy pulses can be added to an input line with bit rate short coming. This technique of bit synchronization is called pulses stuffing.

Problem 1:

Consider the integration of 3 analog sources and 4 identical digital sources through a TDM as shown in figure.



We would like to use the entire 240 kbps maximum capacity of this multiplexers. The analog lines with bandwidth 5KHz, 2KHz, and 1 KHz respectively are sampled, multiplexed, quantized and 5 bit encoded. The digital lines are multiplexed and each carries 8 kbps. Find the pulse stuffing rate.

Solution: Each analog input is sampled by a frequency two times greater than its corresponding bandwidth according to Nyquist sampling rule:

Therefore we have, 5*2 + 2*2+ 1*2 = 16 k/s

Once it is encoded, we get a total of 16000* 5 = 80 kb/s on analog lines.

The total share of digital lines is then 170 – 80 = 90 kb/s.

Therefore each digital line must generate 90/4 = 22.5 kb/s

While actual rate of 8 kb/s exists, each digital line must add a difference of 22.5 – 8 = 14.5 kb/s pulse stuffing in order to balance the ultimate bit rate of multiplexer bit rate.

B. Statistical TDM

In this, a frame's time slots are dynamically allocated based on demand. This method removes all the empty slots on a frame and makes the multiplexer operate more efficiently. Meanwhile, trade off of such a technique is that requirement that the additional overhead be attached to each outgoing channel. This additional data is needed because each channel must carry information about line it belonged to.

As a result in the statistical multiplexer, the frame length is variable, because of different channel size as well as the possible absence of some channels.

2.2 Line coding methods

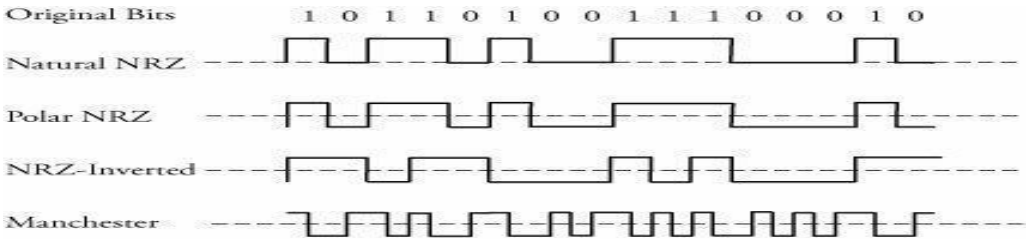
It is a process of converting binary information sequence into a digital code. Encoding process is essential to also recover the bit timing information from digital signal. So that the receiving sample clock can synchronize with the transmitting clock. Reasons for using line coding are:

- Maximize bit rate in digital transmission.
- Reduction in transmitted power and removal of dc voltage from transmission lines.

There are four line coding techniques:

1. Natural non return to zero(NRZ)
2. Polar NRZ
3. NRZ inverted
4. Bipolar NRZ
5. Manchester encoding method

Encoded signals are produced by the line codes for the binary sequence 1011010011100010



1. Natural non return to zero(NRZ)

Here binary 1 is represented by positive voltage level and 0 is represented by a negative voltage. This method is not popular for LAN systems. The average transmitted power is: **(1/2)v^2+(1/2)0^2=v^2/2**.

2. Polar NRZ

This is a more power efficient line coding method. Here, binary 1 is represented by +V/2 and a binary 0 is represented by –V/2. The average power is of: **(+V/2)^2+(-V/2)^2= V^2/4**

In these two methods, the problem is that a priority error can cause all 1's to be delivered as 0's and all 0's as 1's.

3. NRZ Inverted

It is the solution to the above problem. Here the binary 1 is represented by positive transition at a beginning of each interval and 0 as no transition and the signal remains constant during the actual bit time. The method also produces spectrum starting from low frequencies close to zero.

4. Bipolar NRZ

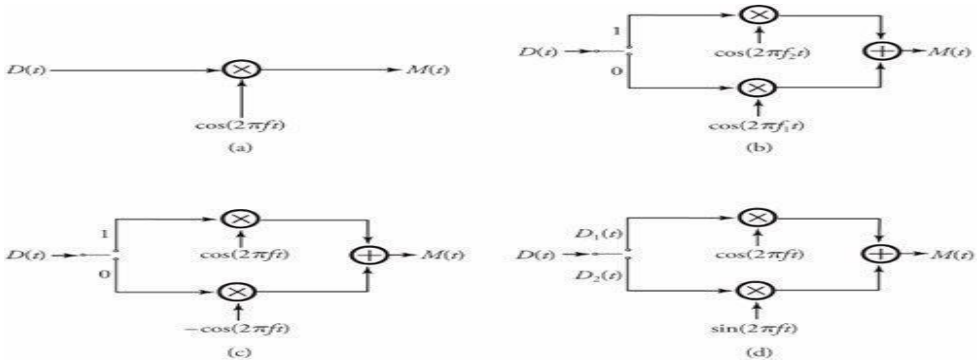
It has a better spectrum distribution but immune to noise. To overcome this issue we use next method.

5. **Manchester Encoding Method**

Here, binary 1 is represented by a 1 plus a transition to 0 and then a 0. And a binary 0 is represented by a 0 plus a transition to 1 and then 1. Advantage of this method is **self-clocking**.

2.3 Digital modulation techniques

In order to reduce the bandwidth of digital signals, digital modulation techniques are required. In modulation process, the amplitude phase or frequency of carrier signal varies with the variations of digital information signal.



1. Amplitude shift keying

In ASK system, incoming data D(t) containing binary 0's and '1's is modulated over a constant frequency and constant amplitude sinusoidal carrier signal **cos2πft** where 'f' is frequency of carrier signal. The resulting modulated signal is represented by a cosine with same frequency 'f' where binary 1 is present and no signal for 0. That is, **M(t) = D(t) cos(2πft).**

When the original data is multiplied by a constant cosine, we obtain ASK modulated version of the original data.

At receiver, the ASK demodulator only needs to determine the presence or absence of sinusoid in a given time interval in order to detect original data.

2**. Frequency shift keying**

In FSK modulator, there are two different sinusoidal carriers: f1 to represent binary 1 and f2 to represent binary 0.

The frequency of the carrier varies according to the information such that we have cos2πf1t instead of binary 1 and cos2πf2t instead of a binary 0.

3. Phase shift keying

In PSK the phase of sinusoidal carrier signal changes according to the information sequence. Binary 1 is represented by cos(2πft) and binary 0 is represented by cos(2πft+π), means – cos(2πft). In PSK modulator multiply the sinusoidal signal by +1 then the information is 1 and by -1 when 0 is present.

4. **Quadrature amplitude modulation**

The original information is split into two equal sequences, d1(t) and d2(t) consisting of the odd and even symbols. Each sequence has a rate of 's' symbols/seconds and the number of bits per symbol is constant.

D1(t) is multiplied by cos(2πft) for t- second interval to produce a modulated signal. Similarly d2(t) is multiplied by sin(2πft) for a t-second interval to produce a modulated signal. The D1(t) is known as **inphase component** and the D2(t) is known as **quadrature phase component**. Therefore at the output of the modulator we have : **M(t)= D1(t) cos(2πft) + D2(t) sin(2πft).**

Here, the original data can be extracted at receiver if M(t) is multiplied by same carrier signal but with doubled amplitude. Since M(t) has two terms in this case, D1(t) cos(2πft) has to be multiplied by 2 cos(2πft) and D2(t) sin(2πft) must be multiplied by 2sin(2πft). Therefore the above equation is arranged as:

$$\mathbf{M(t)= \sqrt{[D1^2(t) + D2^2(t)]}\ [\cos(2\pi ft) + \tan^{-1} D2(t)/ D1(t)]}$$

CHAPTER 2 NETWORKING DEVICES

2.4 Modems

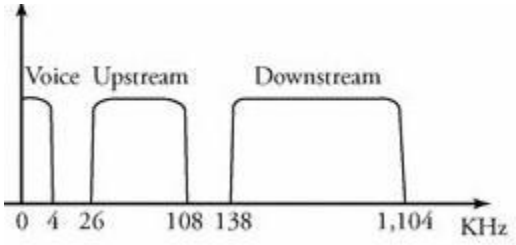
2.4.1 Digital subscriber line modem (DSL)

It uses the existing telephone cable. It is a convenient option for home users to access internet. DSL offers various versions of DSL technology. ADSL, VDSL, HDSL, SDSL are in general xDSL.

1. **ADSL(Asymmetric DSL)**

It is popular and is designed for residential users. A modem is designed to be connected to telephone links. This links are capable of handling bandwidths upto 1.1 MHz. Out of this bandwidth, only 4khz are used for phone conversation, the remaining bandwidth can become available to data communications.

However other factors such as the distance between the users and a switching office and the condition and size of link might restrict this remaining bandwidth from being completely available.



ADSL uses QAM modulation technique. The available bandwidth of 1.1 MHz is divided into 256 channels , each using a bandwidth of approximately 4.312 KHz. Voice communication uses channel 0 , channel 1 - 5 remain ideal and together acts as gaurd band between voice and data communication. Because data communication bandwidth is split into two bandwidth, upstream for communication from the user to internet and downstream for communications from the internet to the users, the technique is said to be asymmetric.

Channels 6 - 30 are allocated to upstream bandwidth with 1 channel dedicated to control and 24 channels are assigned to data transfer. Thus 24 channels with QAM offer

24*4 khz * 15 = 1.44 Mbytes/sec bandwidth in upstream direction.

As QAM requires 15 bit encoding channels 31 - 255 are assigned to the downstream bandwidth with one channel for control and the remaining 224 channels for data with QAM upto

224*4khz * 15 = 13.4 Mbytes/sec is achieved for downstream bandwidth

2. SDSL(Symmetric DSL)

SDSL divides the available bandwidth equally between downstream and upstream data transfer.

3. HDSL(High bit rate DSL)

It is designed to compete with t-1 lines (this t-1 has the capacity of 1.544 Megabytes / sec) this t-1 is susceptible to attenuation at high frequency and thus limits the length of line to 1 km.

HDSL uses two twisted pair wires and 2B1Q -> it is an encoding technique less susceptible to attenuation. HDSL can achieve a data rate of 2 Megabytes/sec without the need of repeaters upto 3.6km.

4. **VDSL(Very high bit rate DSL)**

It is similar to ADSL but uses co-axial or fibre optic cable for a bit rate of 50 Mb/s downstream and 1.5 Mbps to 2.5 Mbps upstream data transfer.

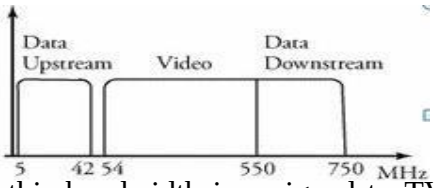
2.4.2Cable modems

DSL modem types are susceptible to errors. Hence cable modems are alternative to provide internet access to residential building with the use of cable TV network.

A cable company layouts a very high speed backbone optical fibre cables all the way to the residential buildings, where each of which can be connected to the optical infrastructure for TV, radio and internet through either co-axial or optical fiber depending on the demand and budget. This network is called

Hybrid Fibre Co-axial(HFC).

Communication in an HFC cable tv network is bidirectional. The cable company divides the bandwidth into video/radio, downstream data and upstream. Co-axial cables can carry signals upto 750 MHz.



About 500MHz of this bandwidth is assigned to TV channels. As the bandwidth of each TV channel is 6MHz, the assigned bandwidth can accommodate more than 80 channels. About 200 MHz, from 550 MHz to 750 MHz is allocated to the downstream data transfer. This bandwidth is also divided into 33 channels each with 6 MHz bandwidth.

The cable modem uses the 64-QAM or 256 QAM modulation technique for the downstream data transfer. These modulation technique use 5-bit encoding, so the bandwidth of the downstream data channel can be: **5 b/Hz * 6 = 30 Mb/s**

The upstream data premises communicate to the internet and occupy 37 MHz. From 5 MHz to 42 MHz, including 6 MHz wide channels. The upstream data is modulated using the QPSK technique, which is susceptible to noise in the lower frequency range. Using 2 b/Hz offers the downstream data rate at:

2 b/Hz * 6 MHz = 12 Mb/s.

The protocol for upstream communication is as follow:

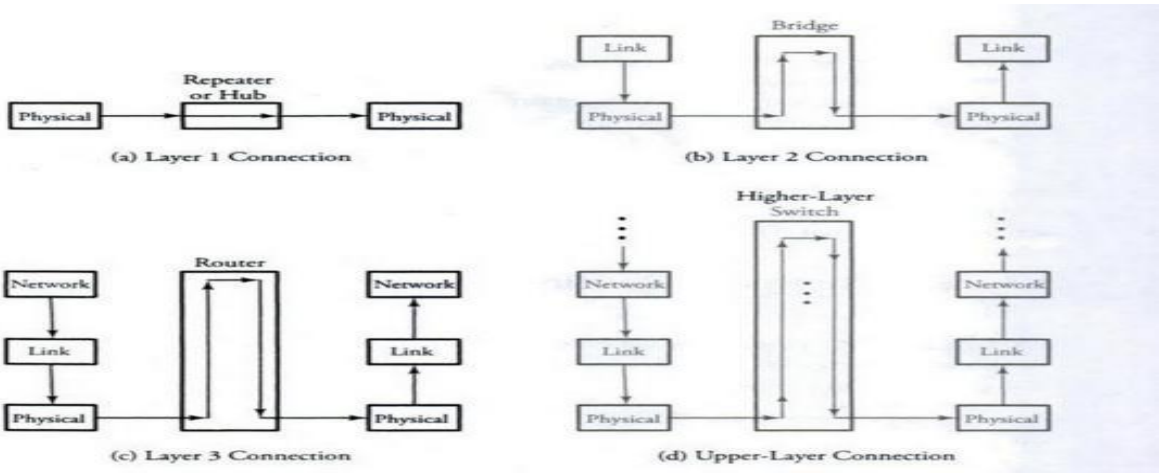
1. The cable modem checks the downstream channel to determine whether any packet periodically sent by the cable company seeks any new modems attaché to the cable.
2. The cable company sends a packet to the cable modem, allocated downstream.
3. The cable modem sends a packet to the cable company, requesting the internet address.

4. The modem and the cable company go through a handshaking process and exchanges some packets.
5. The cable company delivers an identifier to the modem.
6. Internet access in the allocated upstream channel bandwidth can then be granted to the modem in the allocated upstream channel.

2.5 Switching Devices

Repeaters

Repeaters are used to connect two segments of a LAN. Signal generation is the essential function of repeaters. Signal generation is needed when the LAN length is extended. Two LANs are interconnected using a repeaters at the physical layer. Repeater assumes that the connecting LANs have the same protocol and simply accepts bits from one LAN and transmits then onto the other LANs



Hub

It is used to provide connections among multiple users in layer1(Physical layer). A hub is similar to repeater but connects several pieces of LAN. It is a multipoint repeater. Hub has regeneration capability to strengthen incoming data to prevent any decay.

Bridge

It is a switch that connects two LANs. Bridge operates in layer2. A bridge can also be used for signal regeneration. A bridge extends the network and checks the physical address of any destination user and enhances the efficiency of networks by facilitating simultaneous transmissions with in multiple LANs.

Traffic between LAN users is less with the use of bridges compared to repeaters. Bridges can make decisions about where to forward frames. It perform data link functions such as forwarding, formatting and error detection. It can forward only one frame at a time in store and forward fashion.

Switches

It can forward multiple frames simultaneously through multiple data paths. It can also operate as cut through devices.

Routers and High layered switches

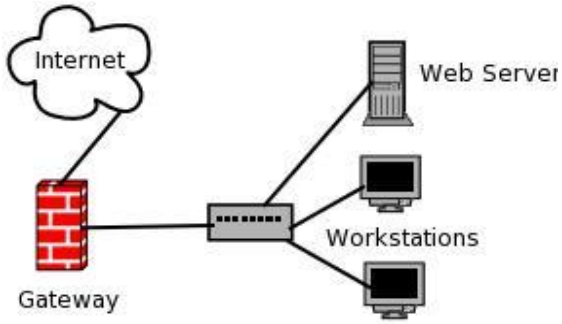
A router is a layer 3 switch that connects other routing nodes. A router is dependent on protocols and establishes physical circuits for individual node pair connection. A router has a routing looking up table for routing packets. Layer 3 switches are of two types:

1. **Packet by packet switches**, by which packet forwarding is handled based on each individual packet.
2. **Flow based switches**, by which a number of packets having the same source and destination are identified and forwarded together.

Layer2 switches are developed to replace routers at a LAN. A layer4 switch uses the information from higher levels for routing decisions. Basically layer4 switches are the application switches for both layer 2 and layer 3. In layer 4 packets are normally forwarded on a connectionless system.

Gateways

A gateway is an internetworking system capable of joining together two networks that use different base protocols. A gateway is the same as a router, except in that it also translates between one network system or protocol and another. A network gateway can be implemented completely in software or in hardware, or as a combination of both. Depending on the types of protocols they support, network gateways can operate at any level of the OSI model.



The most common gateway is the internet gateway, which connects a home or enterprise network to the internet. An internet gateway also often acts as a security node, such as proxy server, firewall or network address translation (NAT) server.

UNIT 2 CHAPTER 3 : ERROR DETECTION AND CORRECTION

2.6 INTRODUCTION

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

Types of Errors :

Flow of data from one point to another are subjected to unpredictable changes because of **interference**. This interference can change the shape of the signal and results in change of bits in data.

There are two types of error:

- Single-Bit Error
- Burst Error

- **Single-Bit Error**

The term single-bit error means that only 1 bit of a given data unit (Byte, packet) is changed from 1 to 0 or from 0 to 1. Figure 10.1 shows the effect of a single-bit error on a data unit. In Figure 00000010 was sent, but 00001010 was received at the destination. i.e fifth bit is changed from 0 to 1. So that single bit error is occurred. Single-bit errors are the least likely type of error in serial data transmission.

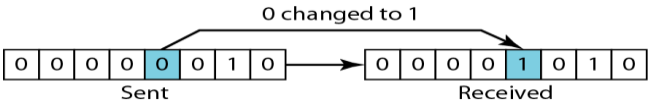


Figure 10.1 Single-bit error

- **Burst Error**

The term burst error means that 2 or more bits in the data unit have changed from 1 to 0 or from 0 to 1. Figure shows the effect of a burst error on a data unit. In this case, 0100010001000011 was sent, but 0101110101100011 was received. The burst error does not necessarily mean that the errors occur in consecutive bits. The length of burst is measured from the first corrupted bit to the last corrupted bit. Some bits in between may not have been corrupted.

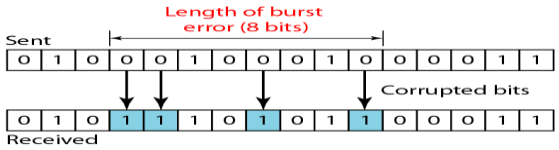


Figure 10.2 Burst error of length 8

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

Redundancy

It is a concept in detecting or correcting errors. In order to do this, we need to send some extra bits with the original data bits. These redundant bits are added by the sender and removed by the receiver. Their presence allows the receiver to detect or correct corrupted bits.

Detection versus Correction

Error detection means that checking if any error has occurred. The answer is a simple yes or no. We are not even interested in the number of errors. A single-bit error is same as a burst error.

The correction of errors is more difficult than the detection. In error correction, it is important to identify the exact number of bits that are corrupted and their location in the message. Here the number of errors and the size of the message are important factors.

Forward Error Correction versus Retransmission

There are two main methods of error correction.

- **Forward error correction**: It is the process in which the receiver tries to guess the message by using redundant bits. This is possible if the number of errors is small.
- **Correction by retransmission**: It is a technique in which the receiver detects the occurrence of an error and asks the sender to resend the message. Resending is repeated until a message arrives that the receiver believes is error-free.

Coding

Redundancy is achieved through various coding schemes. The sender adds redundant bits through a process that creates a relationship between the redundant bits and the actual data bits. The receiver checks the relationships between the two sets of bits to detect or correct the errors. Figure shows the general idea of coding. We can divide coding schemes into two broad categories: **block coding** and convolution coding.

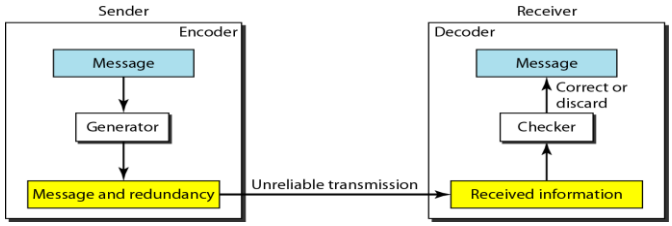


Figure 10.3 The structure of encoder and decoder

Modular Arithmetic

Modular arithmetic is used to error detection and correction. In Modular arithmetic, only a limited range of integers are used. In modulo-**N arithmetic** we use only the integers 0 to N - 1, inclusive. For example, if the modulus is 12, we use only the integers 0 to 11.

In a modulo-N system, if a number is greater than N, it is divided by N and the remainder is the result. If it is negative, as many Ns as needed are added to make it positive.

Addition and subtraction in modulo arithmetic are simple. There is **no carry**, when you add two digits in a column or when you subtract one digit from another in a column.

Modulo-2 Arithmetic

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

Adding: **0+0=0** **0+1=1** **1+0=1** **1+1=0.**

Subtracting: 0-0=0 **0-1=1** **1-0=1** **1-1=0.**

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

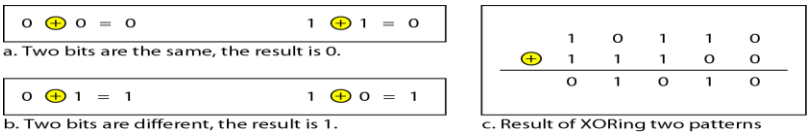


Figure 10.4 XORing of two single bits or two words

Block coding

In block coding, message is divided into blocks, each of k bits, called **datawords.** Add r redundant bits to each block to make the length **n = k + r**. The resulting n-bit blocks are called **codewords**. We have a set of datawords, each of size k, and a set of codewords, each of size of n. With k bits, we can create a combination of $\mathbf{2^k}$ datawords; with n bits, we can create a combination of $\mathbf{2^n}$ codewords.

Error Detection

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

1. The receiver has (or can find) a list of valid codewords.
2. The original codeword has changed to an invalid one.

Figure shows the role of block coding in error detection.

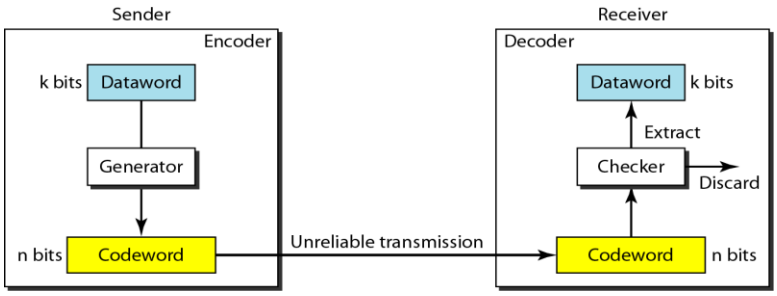


Figure 10.6 Process of error detection in block coding

The sender creates codewords out of datawords by using a generator that applies the rules and procedures of encoding. Each codeword sent to the receiver, may change during transmission. If the received codeword is the same as one of the valid codewords, the word is accepted; the corresponding dataword is extracted for use. If the received codeword is not valid, it is discarded.

Error Correction

Error correction is much more difficult than error detection. In error detection, the receiver needs to know only that the received codeword is invalid; in error correction the receiver needs to find the original codeword sent. Figure shows the role of block coding in error correction. The idea is the same as error detection but the checker functions are much more complex in error correction.

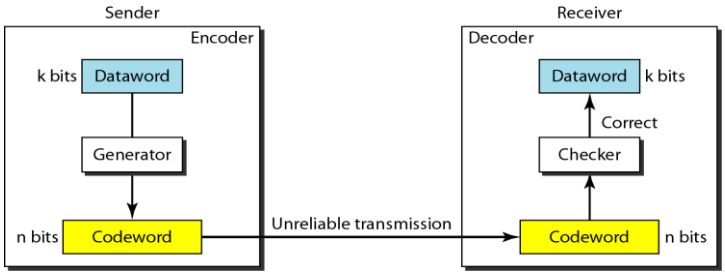


Figure 10.7 Structure of encoder and decoder in error correction

2.7 CYCLIC CODES

Cyclic codes are special linear block codes with one extra property. In a cyclic code, if a codeword is cyclically shifted (rotated), the result is another codeword. For example, if 1011000 is a codeword and we cyclically left-shift, then 0110001 is also a codeword.

Cyclic Redundancy Check

Cyclic codes are created to correct errors. One of the category of cyclic codes is called the cyclic redundancy check (CRC) that is used in networks such as LANs and WANs.

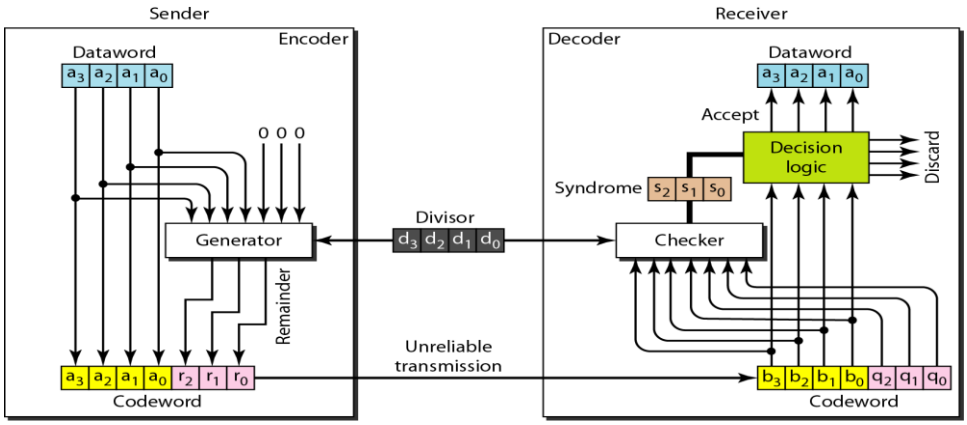


Figure 10.14 CRC encoder and decoder

In the encoder, the dataword has k bits (4 here); the codeword has n bits (7 here). The size of the dataword is augmented by adding **n - k** (3 here) 0's to the right-hand side of the word. The n-bit result is fed into the generator. The generator uses a divisor of size **n - k + 1** (4 here). The generator divides the augmented dataword by the divisor (modulo-2 division). The quotient of the division is discarded; the remainder (r2,rl, ro) is appended to the dataword to create the codeword. i.e **codeword= dataword+ remainder.**

The decoder receives the possibly corrupted codeword. A copy of all n bits is fed to the checker which is a replica of the generator. The remainder produced by the checker is a syndrome of n - k (3 here) bits, which is fed to the decision logic analyzer. The analyzer has a simple function. If the syndrome bits are all 0s, the 4 leftmost bits of the codeword are accepted as the dataword (interpreted as no error); otherwise, the 4 bits are discarded (error).

Encoder

The encoder takes the dataword and augments it with n - k number of 0's. It then divides the augmented dataword by the divisor, as shown in Figure.

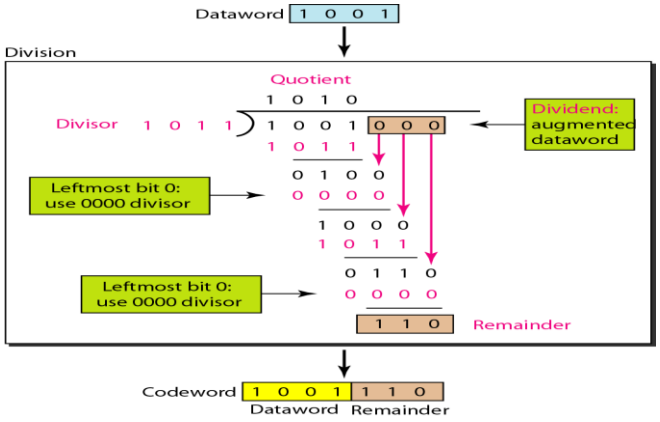


Figure 10.15 Division in CRC encoder

The process of modulo-2 binary division is the same as the familiar division process. In each step, a copy of the divisor is XORed with the 4 bits of the dividend. The result of the XOR operation (remainder) is 3 bits , which is used for the next step after 1 extra bit is pulled down to make it 4 bits long. If the leftmost bit of the dividend is 0, the step cannot use the regular divisor; we need to use an all-0's divisor.

When there are no bits left to pull down, we have a result. The 3-bit remainder forms the check bits (r2,rl, and ro). They are appended to the dataword to create the codeword.

Decoder

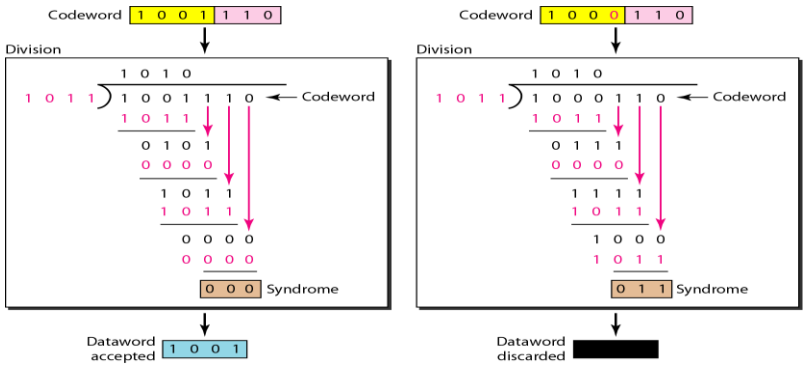


Figure 10.16 Division in the CRC decoder for two cases

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

One of the advantages of a cyclic code is that the encoder and decoder can easily and cheaply be implemented in hardware by using a handful of electronic devices.

Polynomials

A better way to understand cyclic codes and how they can be analyzed is to represent them as polynomials. Cyclic codes can be represented as polynomials with coefficients of 0 and 1. The power of each term shows the position of the bit; the coefficient shows the value of the bit. Figure 10.21 shows a binary pattern and its polynomial representation. A polynomial can be shortened by removing all terms with zero coefficients and replacing x^1 by x and x^0 by 1. The degree of a polynomial is the highest power in the polynomial. For Ex: The degree of the polynomial $x^6 + x + 1$ is 6.

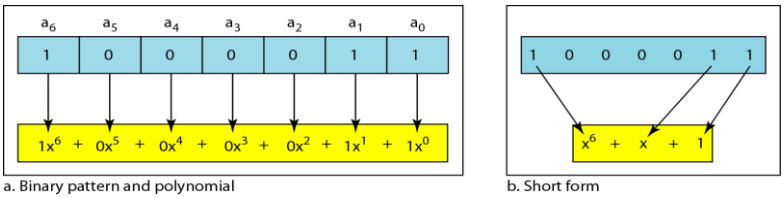


Figure 10.21 A polynomial to represent a binary word

Adding and subtracting of polynomials is in modulo-2. Adding or subtracting is done by combining terms and deleting pairs of identical terms. For example, adding x^5 + x^4 + x^2 and x^6 + x^4 + x^2 gives just x^6 + x^5. The terms x^4 and x^2 are deleted.

Note: If we add, for example, three polynomials and we get x^2 three times, we delete a pair of them and keep the third.

Multiplying a term by another term is just adding the powers. For example, $x^3 \times x^4$ is x^7. Multiplying a polynomial by another is done term by term. Each term of the first polynomial must be multiplied by all terms of the second. The result, is then simplified, by deleting pairs of equal terms.

For Ex: $(x^5 + x^3 + x^2 + x)(x^2 + x + 1)$

$= x^7 + x^6 + x^5 + x^5 + x^4 + x^3 + x^4 + x^3 + x^2 + x^3 + x^2 + x$

$= x^7 + x^6 + x^3 + x.$

For dividing, just subtract the power of the second term from the power of the first. For example, x^5/x^2 is x^3 .

Division of polynomials is the same as the binary division.

Creation of a codeword from a dataword using polynomials.

- The dataword 1001 is represented as $x^3 + 1$.
- The divisor 1011 is represented as $x^3 + x + 1$
- To find the augmented dataword, left-shift the dataword by 3 bits (Multiplying by x^3) . The result is $x^6 + x^3$.
- Divide the first term of the dividend, x^6, by the first term of the divisor, x^3
- The first term of the quotient is then x^6/x^3, or x^3.
- Then we multiply the divisor by x^3 and subtract the result from the dividend. The result is x^4,
- Continue to divide until the degree of the remainder is less than the degree of the divisor.

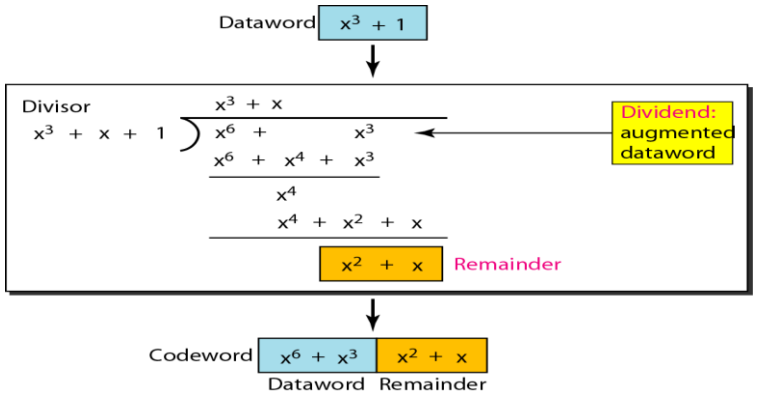


Figure 10.22 CRC division using polynomials

Advantages of Cyclic Codes:

1. Cyclic codes have a very good performance in detecting single-bit errors, double errors, an odd number of errors, and burst errors.
2. They can easily be implemented in hardware and software.
3. They are especially fast when implemented in hardware.
4. Cyclic codes are a good candidate for many networks.

2.7 CHECKSUM

10.5 CHECKSUM

The last error detection method we discuss here is called the checksum. The checksum is used in the Internet by several protocols although not at the data link layer. However, we briefly discuss it here to complete our discussion on error checking.

Like linear and cyclic codes, the checksum is based on the concept of redundancy. Several protocols still use the checksum for error detection as we will see in future chapters, although the tendency is to replace it with a CRe. This means that the CRC is also used in layers other than the data link layer.

Idea

The concept of the checksum is not difficult. Let us illustrate it with a few examples.

Example 10.18

Suppose our data is a list of five 4-bit numbers that we want to send to a destination. In addition to sending these numbers, we send the sum of the numbers. For example, if the set of numbers is (7, 11, 12, 0, 6), we send (7, 11, 12,0,6,36), where 36 is the sum of the original numbers. The receiver adds the five numbers and compares the result with the sum. If the two are the same, the receiver assumes no error, accepts the five numbers, and discards the sum. Otherwise, there is an error somewhere and the data are not accepted.

Example 10.19

We can make the job of the receiver easier if we send the negative (complement) of the sum, called the *checksum*. In this case, we send (7, 11, 12,0,6, -36). The receiver can add all the numbers received (including the checksum). If the result is 0, it assumes no error; otherwise, there is an error.

One's Complement

The previous example has one major drawback. All of our data can be written as a 4-bit word (they are less than 15) except for the checksum. One solution is to use one's complement arithmetic. In this arithmetic, we can represent unsigned numbers between 0 and 2^n - 1 using only n bits.† If the number has more than n bits, the extra leftmost bits need to be added to the n rightmost bits (wrapping). In one's complement arithmetic, a negative number can be represented by inverting all bits (changing a 0 to a 1 and a 1 to a 0). This is the same as subtracting the number from 2^n - 1.

Example 10.20

How can we represent the number 21 in one's complement arithmetic using only four bits?

---

†Although one's complement can represent both positive and negative numbers, we are concerned only with unsigned representation here.

Solution

The number 21 in binary is 10101 (it needs five bits). We can wrap the leftmost bit and add it to the four rightmost bits. We have (0101 + 1) = 0110 or 6.

Example 10.21

How can we represent the number -6 in one's complement arithmetic using only four bits?

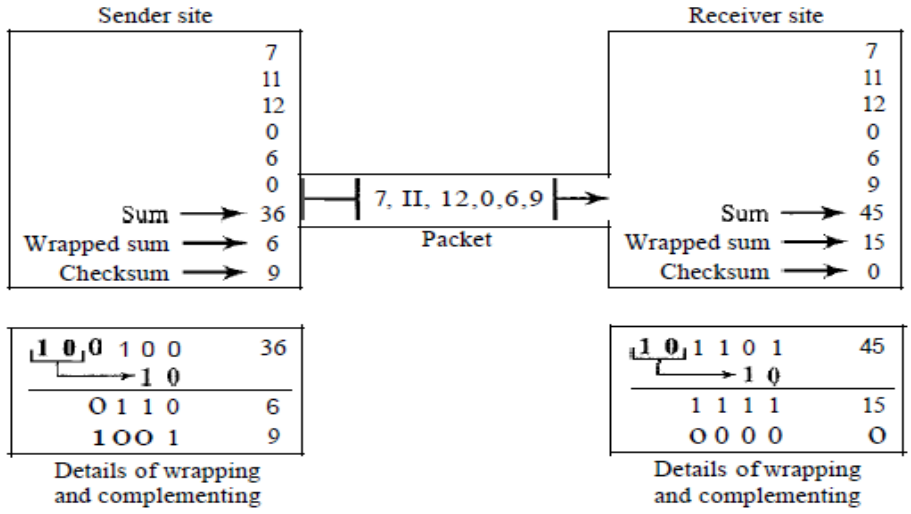
Solution

In one's complement arithmetic, the negative or complement of a number is found by inverting all bits. Positive 6 is 0110; negative 6 is 1001. If we consider only unsigned numbers, this is 9. In other words, the complement of 6 is 9. Another way to find the complement of a number in one's complement arithmetic is to subtract the number from 2^n - 1 (16 - 1 in this case).

Example 10.22

Let us redo Exercise 10.19 using one's complement arithmetic. Figure 10.24 shows the process at the sender and at the receiver. The sender initializes the checksum to 0 and adds all data items and the checksum (the checksum is considered as one data item and is shown in color). The result is 36. However, 36 cannot be expressed in 4 bits. The extra two bits are wrapped and added with the sum to create the wrapped sum value 6. In the figure, we have shown the details in binary. The sum is then complemented, resulting in the checksum value 9 (15 - 6 = 9). The sender now sends six data items to the receiver including the checksum 9. The receiver follows the same procedure as the sender. It adds all data items (including the checksum); the result is 45. The sum is wrapped and becomes 15. The wrapped sum is complemented and becomes 0. Since the value of the checksum is 0, this means that the data is not corrupted. The receiver drops the checksum and keeps the other data items. If the checksum is not zero, the entire packet is dropped.

Figure 10.24



Internet Checksum

Traditionally, the Internet has been using a 16-bit checksum. The sender calculates the checksum by following these steps.

Sender site:

1. The message is divided into 16-bit words.
2. The value of the checksum word is set to 0.
3. All words including the checksum are added using one's complement addition.
4. The sum is complemented and becomes the checksum.
5. The checksum is sent with the data.

The receiver uses the following steps for error detection.

Receiver site:

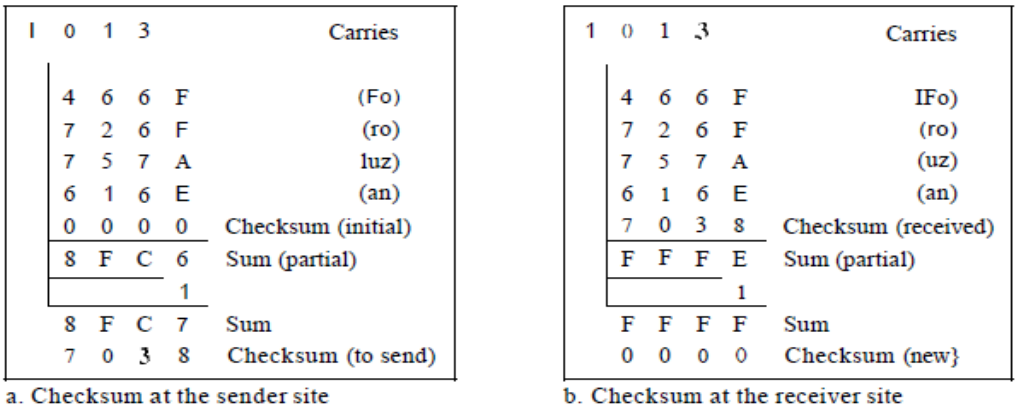
1. The message (including checksum) is divided into 16-bit words.
2. All words are added using one's complement addition.
3. The sum is complemented and becomes the new checksum.
4. If the value of checksum is 0, the message is accepted; otherwise, it is rejected.

The nature of the checksum (treating words as numbers and adding and complementing them) is well-suited for software implementation. Short programs can be written to calculate the checksum at the receiver site or to check the validity of the message at the receiver site.

Example 10.23

Let us calculate the checksum for a text of 8 characters ("Forouzan"). The text needs to be divided into 2-byte (16-bit) words. We use ASCII (see Appendix A) to change each byte to a 2-digit hexadecimal number. For example, F is represented as 0x46 and o is represented as 0x6F. Figure 10.25 shows how the checksum is calculated at the sender and receiver sites. In part a of the figure, the value of partial sum for the first column is 0x36. We keep the rightmost digit (6) and insert the

Figure 10.25



a. Checksum at the sender site

b. Checksum at the receiver site

leftmost dight (3) as the carry in the second column. The process is repeated for each column. Hexadecimal numbers are reviewed in Appendix B.

Note that if there is any corruption, the checksum recalculated by the receiver is not all as. We leave this an exercise.

Performance

The traditional checksum uses a small number of bits (16) to detect errors in a message of any size (sometimes thousands of bits). However, it is not as strong as the CRC in error-checking capability. For example, if the value of one word is incremented and the value of another word is decremented by the same amount, the two errors cannot be detected because the sum and checksum remain the same. Also if the values of several words are incremented but the total change is a multiple of 65535, the sum and the checksum does not change, which means the errors are not detected. Fletcher and Adler have proposed some weighted checksums, in which each word is multiplied by a number (its weight) that is related to its position in the text. This will eliminate the first problem we mentioned. However, the tendency in the Internet, particularly in designing new protocols, is to replace the checksum with a CRC.

32. A sender needs to send the four data items Ox3456, OxABCC, Ox02BC, and OxEEEE. Answer the following:
    a. Find the checksum at the sender site.
    b. Find the checksum at the receiver site if there is no error.
    c. Find the checksum at the receiver site if the second data item is changed to OxABCE.
    d. Find the checksum at the receiver site if the second data item is changed to OxABCE and the third data item is changed to Ox02BA.
33. This problem shows a special case in checksum handling. A sender has two data items to send: Ox4567 and OxBA98. What is the value of the checksum?

2.8 FRAMING

- The two main functions of the data link layer are data link control and media access control.
  - Data link control deals with the design and procedures for communication between two adjacent nodes.
  - Media access control deals with the sharing of the link.
- Data link control functions include framing, flow control, error control and protocols that provide reliable transmission of frames.

Data transmission in the physical layer is in the form of a signal from the source to the destination. But in The data link layer, data is transmitted in the form of frames. So that each frame is distinguishable from another. Our postal system practices a type of framing. The simple act of inserting a letter into an envelope separates one piece of information from another; the envelope serves as the delimiter. It adds the source and destination addresses.

Frames can be of fixed or variable size.

A. In **fixed-size framing,** the size itself can be used as a delimiter, there is no need for defining the boundaries of the frames. Ex: ATM network uses frames of fixed size called cells.

B. In **variable-size framing,** the end of the frame and the beginning of the next frame need to be defined. There are two approaches for this: Character-oriented approach and Bit-oriented approach.

Character-oriented protocol: (Byte stuffing)

In a **character-oriented protocol**, data to be carried are 8-bit ASCII characters. Refer Fig. 11.1 for the frame format.

a. The **header** carries source, destination addresses and other control information.
b. The **trailer** carries error detection or error correction redundant bits.
c. An **8-bit flag** at the beginning and the end of a frame acts as a delimiter separating one frame from the next.

Character-oriented framing was popular when only text was exchanged by the data link layers. If the flag pattern appears in the data, then the receiver on encountering this pattern in the middle of the data wrongly interprets that it has reached the end of frame. In order to avoid this Byte stuffing or character stuffing is used**.**

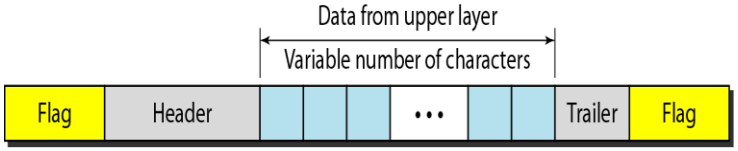


Figure 11.1 A frame in a character-oriented protocol

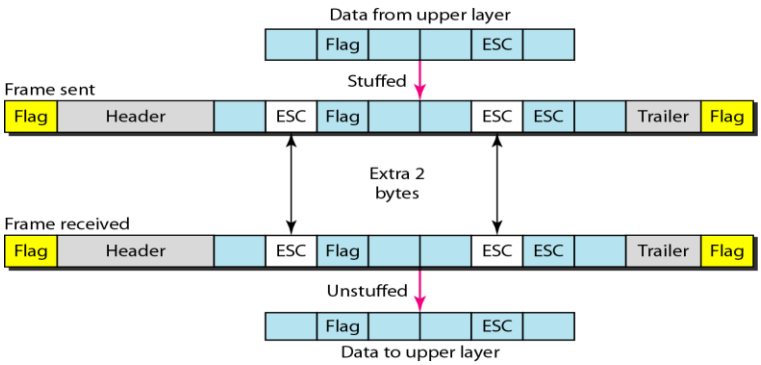


Figure 11.2 Byte stuffing and unstuffing

Byte stuffing or character stuffing is the process of adding a special byte (called the **escape character**) to the data section of the frame whenever there is a character with the same pattern as the flag. The receiver on encountering the escape character, removes it from the data section and treats the next character as data, not a flag. The escape characters that are part of the text must be marked by another escape character.

Byte stuffing is the process of adding 1 extra byte whenever there is a flag or escape character in the text.

Bit-oriented protocols: (Bit stuffing)

In **bit-oriented** protocols, the data section is a sequence of bits interpreted as text, audio, video, graphic and so on.

An 8-bit pattern 01111110 is used as a flag to define the beginning and the end of the frame. If the flag pattern appears in the data, **bit stuffing** is done: If a 0 and five consecutive 1 bits are encountered, an extra 0 is added regardless of the value of the next bit. The receiver on encountering, a 0 and five consecutive 1 bits and a zero removes the stuffed 0. This guarantees that the flag field sequence does not inadvertently appear in the frame.

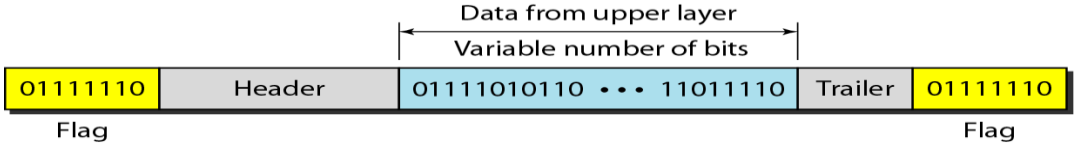


Figure 11.3 A frame in a bit-oriented protocol

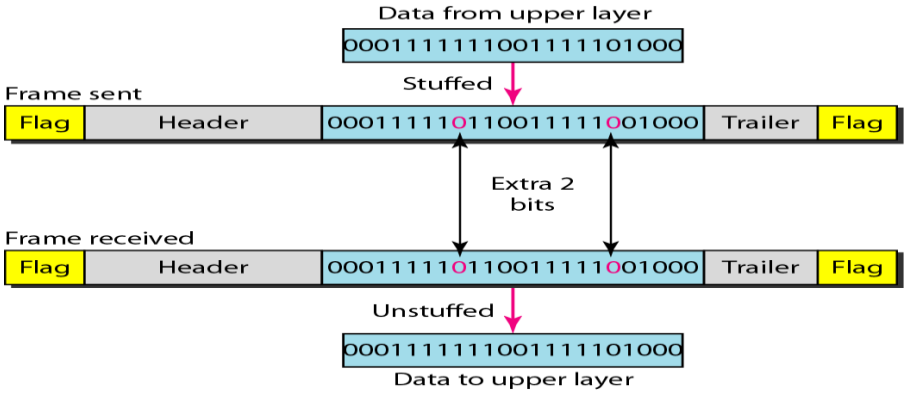


Figure 11.4 Bit stuffing and unstuffing

Bit stuffing is the process of adding one extra 0 whenever five consecutive 1s follow a 0 in the data, so that the receiver does not mistake the pattern 0111110 for a flag.

2.9 POINT-TO-POINT PROTOCOL

Although HDLC is a general protocol that can be used for both point-to-point and multipoint configurations, one of the most common protocols for point-to-point access is the Point-to-Point Protocol

(PPP). PPP is a byte-oriented protocol. Today, millions of Internet users who need to connect their home computers to the server of an Internet service provider use PPP. **PPP is a byte-oriented protocol using byte stuffing with the escape byte 01111101.**

PPP provides several services:

1. PPP defines the format of the frame to be exchanged between devices.
2. PPP defines how two devices can negotiate the establishment of the link and the exchange of data.
3. PPP defines how network layer data are encapsulated in the data link frame.
4. PPP defines how two devices can authenticate each other.
5. PPP provides multiple network layer services supporting a variety of network layer protocols.
6. PPP provides connections over multiple links.
7. PPP provides network address configuration. This is particularly useful when a home user needs a temporary network address to connect to the Internet.

PPP has a certain limitations:

1. PPP does not provide flow control.
2. PPP has a very simple mechanism for error control.
3. PPP does not provide a sophisticated addressing mechanism to handle frames in a multipoint configuration.

PPP frame format

PPP is a byte-oriented protocol. Figure 11.32 shows the format of a PPP frame.

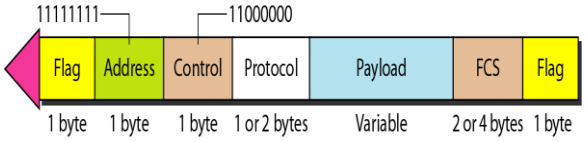


Figure 11.32 PPP frame format

- **Flag**: A PPP frame starts and ends with a 1-byte flag with the bit pattern 01111110.
- **Address**: The address field in this protocol is a constant value and set to 11111111 (broadcast address).
- **Control**: This field is set to the constant value 11000000. PPP does not provide any flow control. This means that this field is not needed at all.
- **Protocol**: The protocol field defines what is being carried in the data field: either user data or other information.
- **Payload field**: This field carries either the user data or other information.
- **FCS**: The frame check sequence (FCS) is simply a 2-byte or 4-byte standard CRC.

Transition Phases

A PPP connection goes through phases which can be shown in a transition phase diagram.

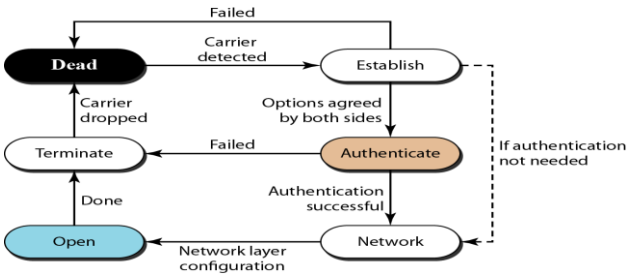


Figure 11.33 Transition phases

- **Dead**: In the dead phase the link is not being used.
- **Establish**: When one of the nodes starts the communication, the connection goes into this phase. In this phase, options are negotiated between the two parties. If the negotiation is successful, the system goes to the authentication phase or directly to the networking phase.
- **Authenticate**: The authentication phase is optional; the two nodes may decide, during the establishment phase, not to skip this phase. However, if they decide to proceed with authentication, they send several authentication packets. If the result is successful, the connection goes to the networking phase; otherwise, it goes to the termination phase.
- **Network**: In the network phase, negotiation for the network layer protocols takes place. PPP specifies that two nodes establish a network layer agreement before data at the network layer can be exchanged.
- **Open:** In the open phase, data transfer takes place. When a connection reaches this phase, the exchange of data packets can be started. The connection remains in this phase until one of the endpoints wants to terminate the connection.
- **Terminate:** In the termination phase the connection is terminated. Several packets are exchanged between the two ends and closing the link.

Multiplexing

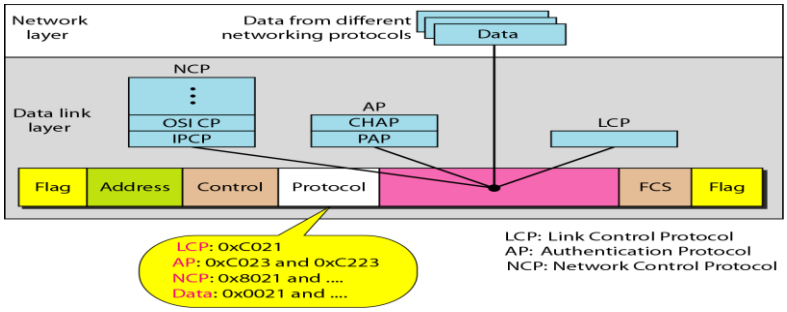


Figure 11.34 Multiplexing in PPP

PPP uses another set of protocols to establish the link, authenticate the parties involved, and carry the network layer data. Three sets of protocols are defined to make PPP powerful: **the Link Control Protocol**

(LCP), two Authentication Protocols (APs), and several Network Control Protocols (NCPs). At any moment, a PPP packet can carry data from one of these protocols in its data field, as shown in Figure 11.34. Note that there is one LCP, two APs, and several NCPs. Data may also come from several different network layers.

Link Control Protocol

The **Link Control Protocol** (LCP) is responsible for establishing, maintaining, configuring, and terminating links. It also provides negotiation mechanisms to set options between the two endpoints. Both endpoints of the link must reach an agreement about the options before the link can be established. See Figure 11.35. All LCP packets are carried in the payload field of the PPP frame.

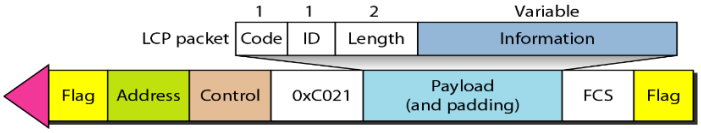


Figure 11.35 LCP packet encapsulated in a frame

Authentication Protocols

Authentication plays a very important role in PPP because PPP is designed for use over dial-up links where verification of user identity is necessary. **Authentication** means validating the identity of a user who needs to access a set of resources. PPP has created two protocols for authentication: **Password Authentication Protocol (PAP) and Challenge Handshake Authentication Protocol (CHAP).**

Network Control Protocols

PPP is a multiple-network layer protocol. It can carry a network layer data packet from protocols defined by the Internet, OSI, Xerox, DECnet, AppleTalk, Novel, and so on. To do this, PPP has defined a specific Network Control Protocol for each network protocol.

For example, IPCP (Internet Protocol Control Protocol) configures the link for carrying IP data packets. Xerox CP does the same for the Xerox protocol data packets, and so on.

Chapter – 12 Introduction

When all the nodes or stations are connected to a common link, called a multipoint or broadcast link, we need a multiple-access protocol to coordinate access to the link. The problem of controlling the access to the medium is similar to the rules of speaking in an assembly. The procedures guarantee that the right to speak is upheld and ensure that two people do not speak at the same time, do not interrupt each other, and do not monopolize the discussion, and so on.

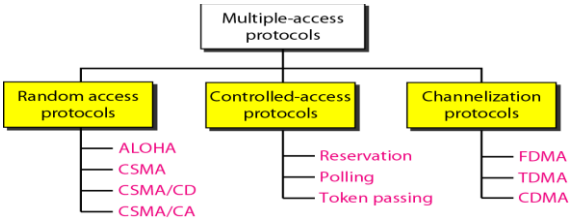


Figure 12.2 Taxonomy of multiple-access protocols

RANDOM ACCESS

In random access or contention methods, no station is superior to another station and none is assigned the control over another. No station permits, or does not permit, another station to send. At each instance, a station that has data to send uses a procedure defined by the protocol to make a decision on whether or not to send. This decision depends on the state of the medium (idle or busy).

Two features give this method its name. First, there is no scheduled time for a station to transmit. Transmission is random among the stations. That is why these methods are called **random access**. Second, no rules specify which station should send next. Stations compete with one another to access the medium. That is why these methods are also called **contention methods**.

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

In this method, a station monitors the medium after it sends a frame to see if the transmission was successful. If so, the station is finished. If, there is a collision, the frame is sent again.

In the figure, the first bits transmitted by the two stations involved in the collision. Although each station continues to send bits in the frame until it detects the collision. In Figure, stations A and C are involved in the collision.

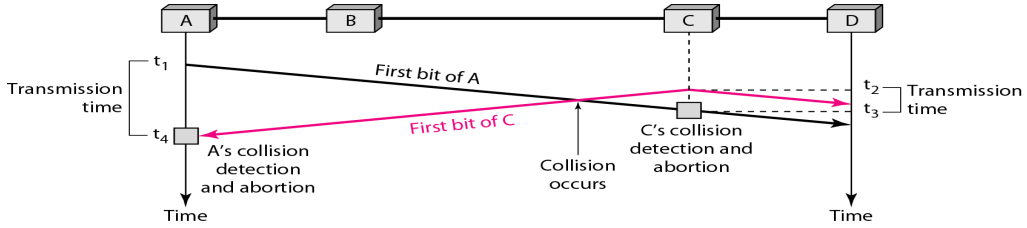


Figure 12.12 Collision of the first bit in CSMA/CD

At time t1, station A has executed its persistence procedure and starts sending the Bits of its frame. At time t2, station C has not yet sensed the first bit sent by A. Station C executes its persistence procedure and starts sending the bits in its frame, which propagate both to the left and to the right. The collision occurs sometime after time t2 Station C detects a collision at time t3 when it receives the first bit of A's frame. Station C immediately aborts transmission. Station A detects collision at time t4 when it receives the first bit of C's frame; it also immediately aborts transmission.

At time t4, the transmission of A's frame, though incomplete, is aborted; at time t3, the transmission of C's frame, though incomplete, is aborted. Figure shows the clear graph.

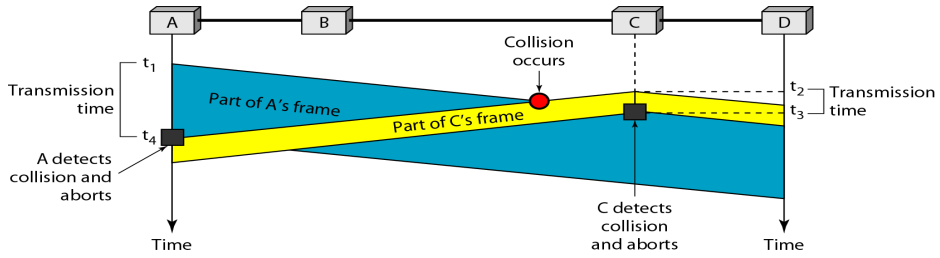


Figure 12.13 Collision and abortion in CSMA/CD

Minimum Frame Size

We need a restriction on the frame size. Before sending the last bit of the frame, the sending station must detect a collision, if any, and abort the transmission. This is so because the station, once the entire frame is sent, does not keep a copy of the frame and does not monitor the line for collision detection. Therefore, the frame transmission time T_{fr} must be at least two times the maximum propagation time T_p.

Procedure

Figure 12.14 Flow diagram for the CSMA/CD. Whenever there is a frame to send, set the number of attempts, k = 0. Sense the channel before starting to send the frame by using one of the persistence processes. The second difference is the frame transmission. In CSMA/CD, transmission and collision detection is a continuous process. We do not send the entire frame and then look for a collision. The station transmits and receives continuously and simultaneously. We use a loop to show that transmission is a continuous process.

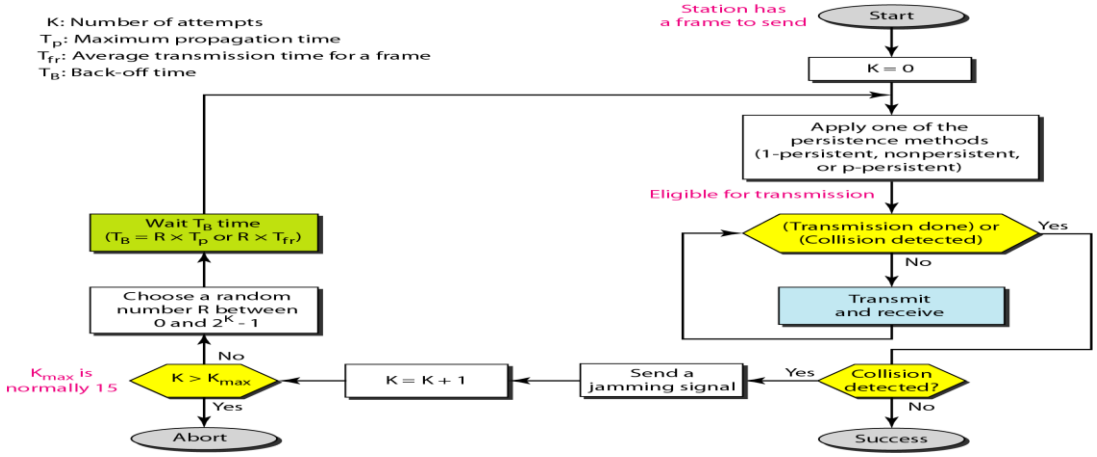


Figure 12.14 Flow diagram for the CSMA/CD

We constantly monitor one of two conditions: either transmission is finished or a collision is detected. Either of these events, stops transmission. When we come out of the loop, if a collision has not been detected, it means that transmission is complete; the entire frame is transmitted. Otherwise, a collision has occurred.

If collision is detected, a short **jamming signal** is sent that enforces the collision in case other stations have not yet sensed the collision. Then the k value is incremented by 1 that is k=k+1. The maximum number of attempts, $\mathbf{k_{max}}$ to send the frame is predefined and is set to **15**. If the k value is greater then k_{max} the transmission is aborted else station choose a random number to wait for the channel to send the frame.

Energy Level

We can say that the level of energy in a channel can have three values: zero, normal, and abnormal is shown in fig 12.15.

- At the zero level, the channel is idle.
- At the normal level, a station has successfully captured the channel and is sending its frame.
- At the abnormal level, there is a collision and the level of the energy is twice the normal level.

The throughput of CSMA/CD is greater than that of pure or slotted ALOHA

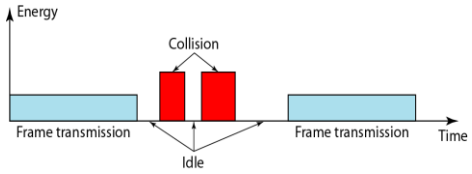


Figure 12.15 Energy level during transmission, idleness, or collision

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

In CSMA/CD, when there is no collision, the station receives one signal: its own signal. When there is a collision, the station receives two signals: its own signal and the signal transmitted by a second station. To distinguish between these two cases, the signal from the second station needs to add a significant amount of energy to the one created by the first station.

In a wireless network, much of the energy is lost in transmission. The received signal has very little energy. Therefore, a collision may add only 5 to 10 percent additional energy. This is not useful for effective collision detection. We need to avoid collisions on wireless networks because they cannot be detected. Collisions are avoided through the use of CSMA/CA's three strategies:

- interframe space
- contention window
- acknowledgments

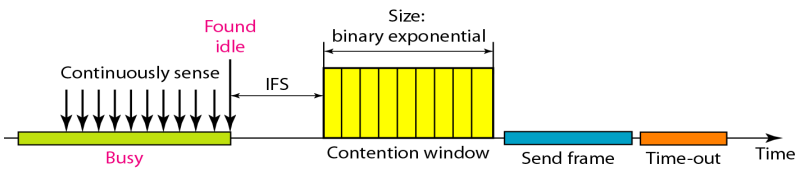


Figure 12.16 Timing in CSMA/CA

- **Interframe Space (IFS)**

First, collisions are avoided by deferring transmission even if the channel is found idle. When an idle channel is found, the station does not send immediately. It waits for a period of time called the **interframe space or IFS.(** it is shown in fig 12.16). Even though the channel may appear idle when it is sensed, a distant station may have already started transmitting.

The distant station's signal has not yet reached this station. The IFS time allows the front of the transmitted signal by the distant station to reach this station. If after the IFS time the channel is still idle, the station can send, but it still needs to wait a time equal to the contention time. A station that is assigned a shorter IFS has a higher priority.

In CSMA/CA, the IFS can also be used to define the priority of a station or a frame.

- **Contention Window**

In CSMA/CA, the contention window is an amount of time divided into slots. A station that is ready to send chooses a random number of slots as its wait time. The number of slots in the window changes according to the **binary exponential back-off strategy**. This means that it is set to one slot the first time and then doubles each time the station cannot detect an idle channel after the IFS time. The station needs to sense the channel after each time slot.

In CSMA/CA, if the station finds the channel busy, it does not restart the timer of the contention window; it stops the timer and restarts it when the channel becomes idle. This gives priority to the station with the longest waiting time.

- **Acknowledgment**

The **positive ack**nowledgment and the **time-out** timer can help guarantee that the receiver has received the frame.

Procedure

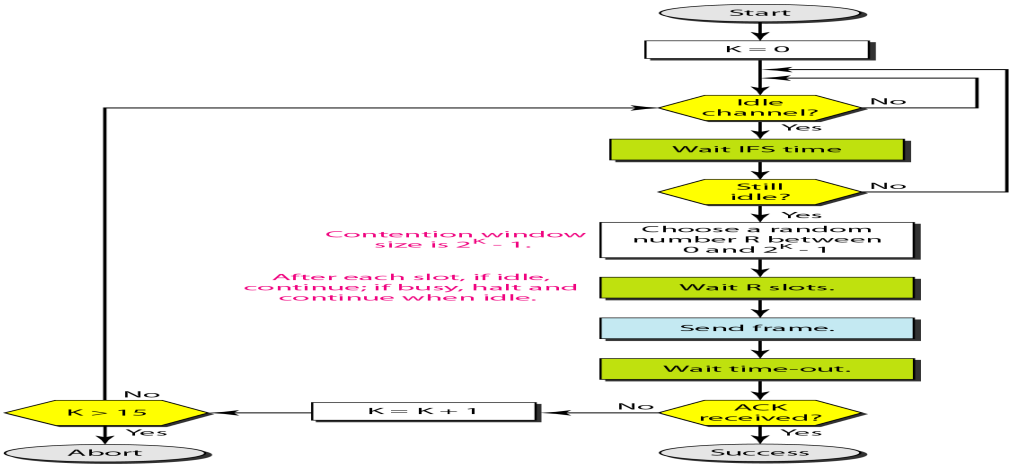


Figure 12.17 Flow diagram for CSMA/CA

Figure 12.17 shows the procedure. The channel needs to be sensed before and after the IFS. The channel also needs to be sensed during the contention time. For each time slot of the contention window, the channel is sensed. If it is found idle, the timer continues; if the channel is found busy, the timer is stopped and continues after the timer becomes idle again.

2.12 CONTROLLED ACCESS

In controlled access, the stations consult one another to find which station has the right to send. A station cannot send unless it has been authorized by other stations. Three popular controlled-access methods are:

1. Reservation

In the reservation method, a station needs to make a reservation before sending data. Time is divided into intervals. In each interval, a reservation frame precedes the data frames sent in that interval. If there are N stations in the system, there are exactly N reservation minislots in the reservation frame. When a station needs to send a data frame, it makes a reservation in its own minislot. The stations that have made reservations can send their data frames after the reservation frame.

Figure shows a situation with five stations and a five-mini slot reservation frame. In the first interval, only stations 1, 3, and 4 have made reservations. In the second interval, only station 1 has made a reservation.

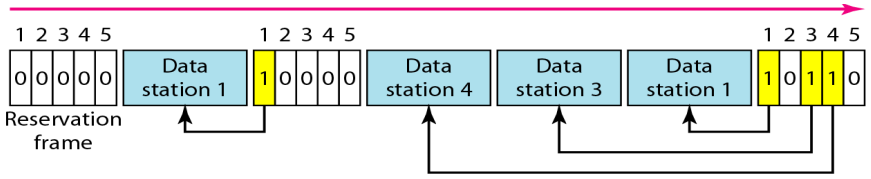


Figure 12.18 Reservation access method

2. Polling

Polling works with topologies in which one device is designated as a **primary station** and the other devices are **secondary stations.** All data exchanges must be made through the primary device even when the ultimate destination is a secondary device.

The primary device controls the link; the secondary devices follow its instructions. It is up to the primary device to determine which device is allowed to use the channel at a given time.

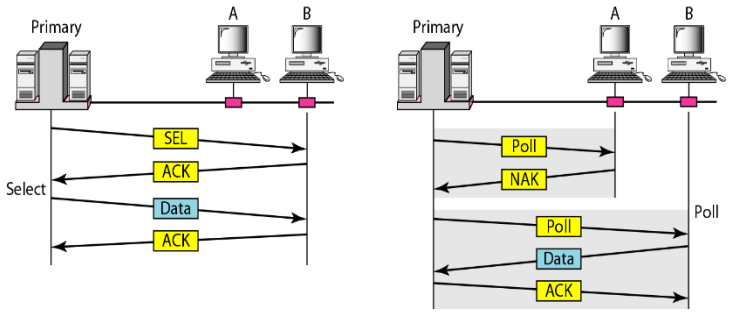


Figure 12.19 Select and poll functions in polling access method

If the primary wants to receive data, it asks the secondaries if they have anything to send; this is called **poll function**. If the primary wants to send data, it tells the secondary to get ready to receive; this is called **select function**.

3. Token Passing

In the token-passing method, for each station, there is a **predecessor and a successor**. The predecessor is the station which is logically before the station in the ring; the successor is the station which is after the station in the ring. The current station is the one that is accessing the channel now. The right to access, this device has been passed from the predecessor to the current station. The right will be passed to the successor when the current station has no more data to send.

A special packet called a **token** circulates through the ring. The possession of the token gives the station the right to access the channel and send its data. When a station has some data to send, it waits until it receives the token from its predecessor. It then holds the token and sends its data. When the station has no more data to send, it releases the token, passing it to the next logical station in the ring. The station cannot send data until it receives the token again in the next round.

In this process, when a station receives the token and has no data to send, it just passes the data to the next station. Token management is needed for this access method. Stations must be limited in the time they can have possession of the token.

Logical Ring

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

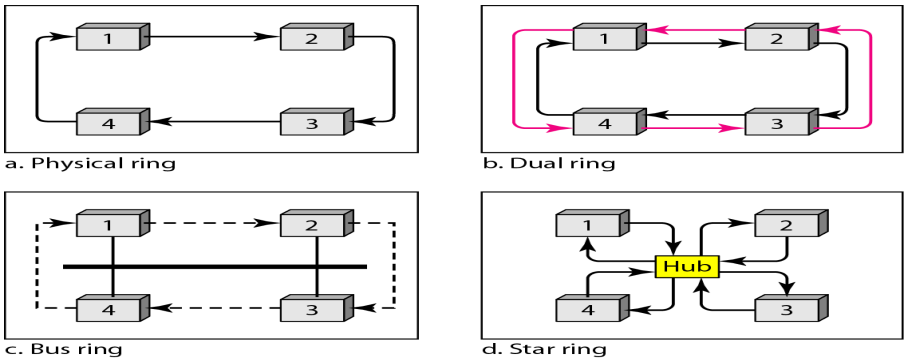


Figure 12.20 Logical ring and physical topology in token-passing access method

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

2. The **dual ring topology** uses a second (auxiliary) ring which operates in the reverse direction compared with the main ring. The second ring is for emergencies only. If one of the links in the main ring fails, the system automatically combines the two rings to form a temporary ring. After the failed link is restored, the auxiliary ring becomes idle again.

3. In the **bus ring topology/ token bus**, the stations are connected to a single cable called a bus. It make a logical ring, because each station knows the address of its successor. When a station has finished sending its data, it releases the token and inserts the address of its successor in the token. Only the station with the address matching the destination address of the token gets the token to access the shared media.

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