Computer Science & IT

Computer Networks

Comprehensive Theory

with **Solved Examples** and **Practice Questions**

MADE EASY
India's Best Institute for IES, GATE & PSUs

CS

Contents

Computer Networks

Chapter 1

Networking Fundamentals and Physical layer 3

1.1 Introduction 3
1.2 Signal 4
1.3 Transmission Media 5
1.4 Noise 8
1.5 Transmission of Signals 11
1.6 IP Addressing 14
1.7 Subnetting 19
1.8 CIDR (Classless Inter Domain Routing) 20
1.9 Network Address Translation (NAT) 27

Chapter 2

Data Link Layer 32

2.1 Delays in Computer Networks 32
2.2 Protocol Layering 33
2.3 Circuit-Switched 36
2.4 Packet Switching 36

Chapter 3

MAC Sublayer 39

3.1 Introduction 39
3.2 Data Link Layer Functions 40
3.3 Data Link Layer Framing 41
3.4 Error Control 45
3.5 Data Link Layer: Error Detection/Correction 45
3.6 Error-Detecting and Correcting Codes 50
3.7 Sliding Window Protocols (SWP) 53
3.8 Repeaters 59
3.9 Hubs 60
3.10 Bridges 60
3.11 Switches 65
3.12 Routers 66
3.13 Gateways 67
3.14 IEEE Standard 802.4: Token Bus 67

Chapter 4

Network Layer 76

4.1 Introduction 76
4.2 Channel Allocation Problem 76
4.3 Multiple Access Protocols 77
4.4 CSMA with Collision Detection (CSMA/CD) 79
4.5 Routing Algorithms 83
4.6 RIP – Routing Information Protocol 89
4.7 Open Shortest Path First (OSPF) 95
4.8 Border Gateway Protocol (BGP) 100
4.9 Internet Protocol (IP) 101
4.10 Address Resolution Protocol 105
4.11 Reverse ARP (RARP) 106
4.12 DHCP 107
4.13 ICMP 109
4.14 IPV6 110
4.15 Transition from IPv4 to IPv6 118

Chapter 5

Transport Layer 121

5.1 Introduction 121
5.2 Transport Layer Services 121
5.3 Transmission Control Protocol (TCP) 122
5.4 Introduction to UDP 138
5.5 Congestion Control 139
5.6 Congestion Control in Virtual Circuit 143

Chapter 6

Application Layer 149

6.1 Inroduction 149
6.2 Application Layer Protocols 150
6.3 Application Layer Protocols and Services Examples 152
6.4 WWW Service and HTTP 155
6.5 E-mail Services and SMTP/POP Protocols 157
6.6 FTP (File Transfer Protocol) 159
6.7 Telnet Services and Protocol 160

Chapter 7

Network Security 164

7.1 Cryptography 164
7.2 Public Key Cryptography 167
7.3 Secured Communication 168
7.4 Key Management 172
7.5 Application Layer Security 173
7.6 Firewalls, Tunnels, and Network Intrusion Detection 174
7.7 Basics of WiFi 180

■■■■

Computer Networks

Goal of the Subject

The main goal of networking is "Resource sharing", and it is to make all programs, data and equipment available to anyone on the network without the regard to the physical location of the resource and the user.

A second goal is to provide high reliability by having alternative sources of supply. For example, all files could be replicated on two or three machines, so if one of them is unavailable, the other copies could be available.

Another goal is saving money. Small computers have a much better price/performance ratio than larger ones. This goal leads to networks with many computers located in the same building. Such a network is called a LAN (local area network).

Another closely related goal is to increase the systems performance as the work load increases by just adding more processors. With central mainframes, when the system is full, it must be replaced by a larger one, usually at great expense and with even greater disruption to the users.

Computer networks provide a powerful communication medium. A file that was updated/ modified on a network can be seen by the other users on the network immediately.

Computer Networks

INTRODUCTION

Although Computer network is a vast subject on its own, in this book we tried to keep it around the GATE syllabus. Each topic required for GATE is crisply covered with illustrative examples and each chapter is provided with Student Assignment at the end of each chapter so that the students get the thorough revision of the topics that he/she had studied. This subject is carefully divided into eight chapters as described below.

1. **Networking Fundamentals and Physical layer:** In this chapter we discuss transmission medium, noise that cause bit errors, types of transmission media, concept of protocol layering. Finally we discuss the IP addressing, Subnetting and Network address translation.
2. **Data Link layer:** In this chapter we discuss Delays in computer networks, Protocol layering, Circuit-switched and Packet switching.
3. **MAC-sub layer:** In this chapter we discuss the data link layer functions, farming methods, error correction and detection methods, Sliding window protocols for flow control the Static and dynamic channel allocation methods. Then we finally discuss Networking devices like Repeaters, Hubs, Bridges, Switches, Routers and Gateways.
4. **Network Layer & Protocols:** In this chapter we discuss the classification of routing algorithms, Distance vector and Link state routing protocols. We also cover congestion control algorithms at network layer, Internet protocol, and finally we cover the network layer protocols namely ARP, RARP, ICMP and IPv4 & IPv6 header format and their functionality.
5. **Transport Layer & Protocols:** In this chapter we discuss the TCP protocol as connected oriented service and reliable service provider, TCP congestion control, TCP timers and finally we discuss UDP.
6. **Application Layer & Protocols:** In this chapter we discuss the various protocols used at application layer: DNS, HTTP, SMTP, Telnet, UDP, FTP etc.
7. **Network Security:** In this chapter we discuss the symmetric and asymmetric key cryptography techniques, RSA encryption algorithm, diffie-hellman key exchange algorithm, firewalls, security services and finally we discuss Basics of WiFi.

■■■■

CHAPTER 01

Networking Fundamentals and Physical Layer

1.1 Introduction

Source (as shown in the following figure) is where the data is originated. Typically it is a computer, but it can be any other electronic equipment such as telephone handset, video camera, etc., which can generate data for transmission to some destination.

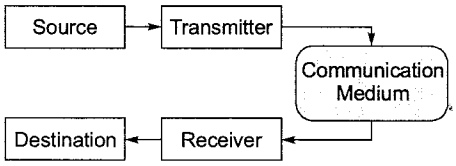


Figure: *Simplified model of a data communication system*

Transmitter

As data cannot be sent in its native form, it is necessary to convert it into signal. This is performed with the help of a transmitter such as modem.

Communication Medium

The signal can be sent to the receiver through a communication medium, which could be a simple twisted-pair of wire, a coaxial cable, optical fiber or wireless communication system. It may be noted that the signal that comes out of the communication medium is different from that was sent by the transmitter (received data may not be same as it was send).

The receiver receives the signal and converts it back to data before forwarding to the destination. Destination is where the data is absorbed. Again, it can be a computer system, a telephone handset, a television set and so on.

Data

Data refers to information that conveys some meaning based on some mutually agreed up rules or conventions between a sender and a receiver and today it comes in a variety of forms such as text, graphics, audio, video and animation.

Data can be of two types; analog and digital. *Analog data* take on continuous values on some interval. *Digital data* take on discrete values.

1.2 Signal

It is electrical, electronic or optical representation of data, which can be sent over a communication medium.

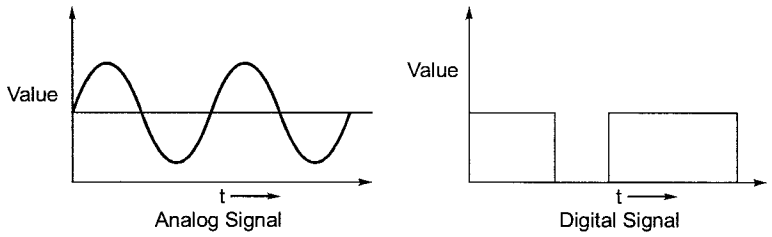


Figure: *Analog signal and Digital signal*

Digital signal can have only a limited number of defined values, usually two values 0 and 1. Analog signals are continuous with the wave forms as shown in the figure.

1.2.1 Signal Characteristics

A signal can be represented as a function of time, i.e. it varies with time. However, it can be also expressed as a function of frequency, i.e. a signal can be considered as a composition of different frequency components. Thus, a signal has both time-domain and frequency domain representation.

Bandwidth

The range of frequencies over which most of the signal energy of a signal is contained is known as **bandwidth** or effective bandwidth of the signal. The term 'most' is somewhat arbitrary. Usually, it is defined. The frequency spectrum and spectrum of a signal is shown in Figure.

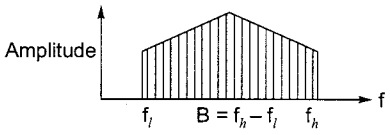


Figure:
Frequency spectrum and bandwidth of a signal

Bandwidth (B) = $F_h - f_l$

1.2.2 Digital Signal

Most digital signals are aperiodic and thus, period or frequency is not appropriate. Two new terms, *bit interval* (instead of period) and *bit rate* (instead of frequency) are used to describe digital signals. The bit interval is the time required to send one single bit. The bit rate is the number of bit interval per second. This mean that the bit rate is the number of bits sent in one second, usually expressed in bits per second (bps) as shown in Figure.

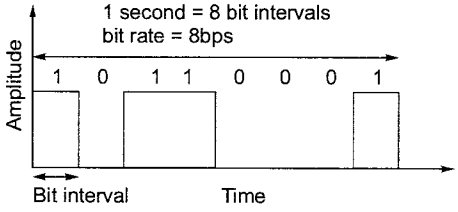


Figure: *Bit Rate and Bit Interval*

A digital signal can be considered as a signal with an infinite number of frequencies and transmission of digital requires a low-pass channel as shown in Figure. On the other hand, transmission of analog signal requires band-pass channel shown in Figure.

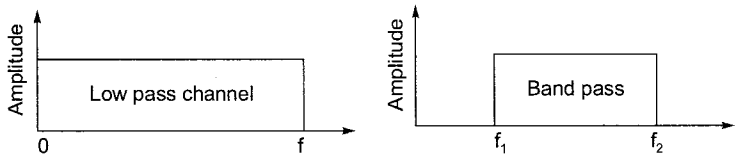


Figure: *Low pass channel required for transmission of digital and Analog signal*

Digital transmission has several advantages over analog transmission. That is why there is a shift towards digital transmission despite large analog base. Some of the advantages of digital transmission are highlighted below:

- Analog circuits require amplifiers, and each amplifier adds distortion and noise to the signal. In contrast, digital amplifiers regenerate an exact signal, eliminating cumulative errors. An incoming (analog) signal is sampled, its value is determined, and the node then generates a new signal from the bit value; the incoming signal is discarded. With analog circuits, intermediate nodes amplify the incoming signal, noise and all.
- Voice, data, video, etc. can all by carried by digital circuits. What about carrying digital signals over analog circuit? The modem example shows the difficulties in carrying digital over analog. A simple encoding method is to use constant voltage levels for a "1" and a ``0''. Can lead to long periods where the voltage does not change.
- Easier to multiplex large channel capacities with digital.
- Easy to apply encryption to digital data.
- Better integration if all signals are in one form. Can integrate voice, video and digital data.

Base-band and Broadband Signals

Base-band is defined as one that uses digital signaling, which is inserted in the transmission channel as voltage pulses. In baseband LANs, the entire frequency spectrum of the medium is utilized and transmission is bi-directional. Baseband systems used for small distance communication.

Broadband systems are those, which use analog signaling to transmit information using a carrier of high frequency.

Since broadband systems use analog signaling, frequency division multiplexing is possible, where the frequency spectrum of the cable is divided into several sections of bandwidth. These separate channels can support different types of signals of various frequency ranges to travel at the same instance.

Broadband is a unidirectional medium where the signal inserted into the media propagates in only one direction. Two data paths are required, which are connected at a point in the network called *headend*. All the stations transmit towards the headend on one path and the signals received at the headend are propagated through the second path.

1.3 Transmission Media

Transmission media can be defined as physical path between transmitter and receiver in a data transmission system. Classified as:

- **Guided:** Transmission capacity depends critically on the medium, the length, and whether the medium is point-to-point or multipoint (e.g. LAN). Examples are co- axial cable, twisted pair, and optical fiber.
- **Unguided:** Provides a means for transmitting electro-magnetic signals but do not guide them. Example wireless transmission.

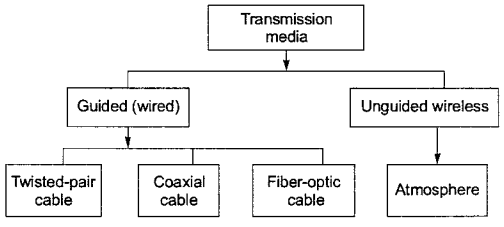


Figure: *Classification of the transmission media*

1.3.1 Guided Media Transmission

(a) Twisted Pair

In twisted pair technology, two copper wires are strung between two points:

- The two wires are typically "twisted" together in a helix (as shown in Figure) to reduce interference and cross- talk interference.
- Can carry both analog and digital signals.
- Data rates of several Mbps common.
- Spans distances of several kilometers.
- Data rate determined by wire thickness and length.
- Good, low-cost communication.

Figure: *CAT5 cable (twisted cable)*

Typical Characteristics

The data rate that can be supported over a twisted-pair is inversely proportional to the square of the line length. Maximum transmission distance of 1 km can be achieved for data rates up to 1 Mb/s.

To reduce interference, the twisted pair can be shielded with metallic braid. This type of wire is known as *Shielded Twisted-Pair* (STP) and the other form is known as *Unshielded Twisted-Pair* (UTP). Used for telephone lines small LAN(100m).

(b) Base Band Coaxial

With "coax", the medium consists of a copper core surrounded by insulating material and a braided outer conductor. The term *base band* indicates digital transmission (as opposed to *broadband* analog).

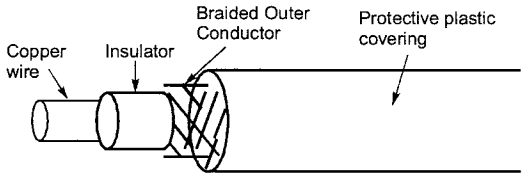


Figure: *Co-axial cable*

Characteristics

Co-axial cable has superior frequency characteristics compared to twisted-pair and can be used for both analog and digital signaling. In baseband LAN, the data rates lies in the range of 1 KHz to 20 MHz over a distance in the range of 1 km. Coaxial cables are used both for *baseband* and *broadband* communication.

In broadband signaling, signal propagates only in one direction, in contrast to propagation in both directions in baseband signaling. Needed for every kilometer or so. Data rate depends on physical properties of cable, but 10 Mbps is typical.

Use: One of the most popular use of co-axial cable is in cable TV (CATV) for the distribution of TV signals. Another importance use of co-axial cable is in LAN.

(c) Broadband Coaxial

The term *broadband* refers to analog transmission over coaxial cable. (Note, however, that the telephone folks use broadband to refer to any channel wider than 4 kHz). The technology:

- Typically bandwidth of 300 MHz, total data rate of about 150 Mbps.
- Operates at distances up to 100 km (metropolitan area!).
- Uses analog signaling.
- Technology used in cable television. Thus, it is already available at sites such as universities that may have TV classes.
- Total available spectrum typically divided into smaller channels of 6 MHz each.

That is, to get more than 6MHz of bandwidth, you have to use two smaller channels and somehow combine the signals. Requires amplifiers to boost signal strength; because amplifiers are one way, data flows in only one direction.

Two types of systems have emerged:

1. **Dual cable systems use two cables, one for transmission in each direction:**
   (a) One cable is used for receiving data.
   (b) Second cable used to communicate with *headend*. When a node wishes to transmit data, it sends the data to a special node called the *headend*. The headend then resends the data on the first cable. Thus, the headend acts as a root of the tree, and all data must be sent to the root for redistribution to the other nodes.
2. **Midsplit** systems divide the raw channel into two smaller channels, with each sub channel having the same purpose as above.

Which is better, broadband or base band? There is rarely a simple answer to such questions. Base band is simple to install, interfaces are inexpensive, but doesn't have the same range. Broadband is more complicated, more expensive, and requires regular adjustment by a trained technician, but offers more services (e.g., it carries audio and video too).

(d) Fiber Optics

In fiber optic technology, the medium consists of a hair-width strand of silicon or glass, and the signal consists of pulses of light. For instance, a pulse of light means "1", lack of pulse means "0". It has a cylindrical shape and consists of three concentric sections: the ***core***, the ***cladding***, and the ***jacket*** as shown in Figure.

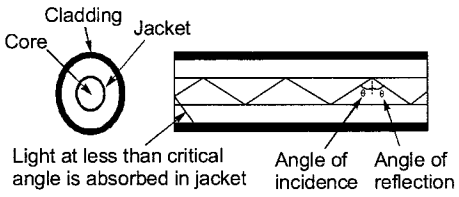


Figure: *Optical Fiber*

The core, innermost section consists of a single solid dielectric cylinder of diameter d_1 and of refractive index n_1. The core is surrounded by a solid dielectric cladding of refractive index n_2 that is less than n_1. As a consequence, the light is propagated through multiple total internal reflection. The core material is usually made of ultra pure fused silica or glass and the cladding is either made of glass or plastic. The cladding is surrounded by a jacket made of plastic. The jacket is used to protect against moisture, abrasion, crushing and other environmental hazards. Three components are required:

1. **Fiber medium:** Current technology carries light pulses for tremendous distances (e.g., 100s of kilometers) with virtually no signal loss.
2. **Light source:** Typically a Light Emitting Diode (LED) or laser diode. Running current through the material generates a pulse of light.
3. A photo diode light detector, which converts light pulses into electrical signals.

Advantages

1. Very high data rate, low error rate. 1000 Mbps (1 Gbps) over distances of kilometers common. Error rates are so low they are almost negligible.
2. Difficult to tap, which makes it hard for unauthorized taps as well. This is responsible for higher reliability of this medium.
   How difficult is it to prevent coax taps? Very difficult indeed, unless one can keep the entire cable in a locked room!
3. Much thinner than existing copper circuits.
4. Not susceptible to electrical interference (lightning) or corrosion (rust).
5. Greater repeater distance than coax.

Disadvantages

- Difficult to tap. It really is point-to-point technology. In contrast, tapping into coax is trivial. No special training or expensive tools or parts are required.
- One-way channel. Two fibers needed to get full duplex (both ways) communication.

Fiber Uses: Because of greater bandwidth (2 Gbps), smaller diameter, lighter weight, low attenuation, immunity to electromagnetic interference and longer repeater spacing, optical fiber cables are finding widespread use in long-distance telecommunications. Especially, the single mode fiber is suitable for this purpose. Fiber optic cables are also used in high-speed LAN applications. Multi-mode fiber is commonly used in LAN.

- Long-haul trunks-increasingly common in telephone network (Sprint ads)
- Metropolitan trunks-without repeaters (average 8 miles in length)
- Rural exchange trunks-link towns and villages
- Local loops-direct from central exchange to a subscriber (business or home)
- Local area networks-100Mbps ring networks.

1.3.2 Unguided Transmission

Unguided transmission is used when running a physical cable (either fiber or copper) between two end points is not possible. For example, running wires between buildings is probably not legal if the building is separated by a public street.

Difficulties

1. **Weather interferes with signals:** For instance, clouds, rain, lightning, etc. may adversely affect communication.
2. **Radio transmissions easy to tap:** A big concern for companies worried about competitors stealing plans.
3. Signals bouncing off of structures may lead to out-of-phase signals that the receiver must filter out.

1.4 Noise

As signal is transmitted through a channel, undesired signal in the form of noise gets mixed up with the signal, along with the distortion introduced by the transmission media. Noise can be categorised into the following four types:

- Thermal Noise
- Intermodulation Noise
- Cross talk
- Impulse Noise

The ***thermal noise*** is due to thermal agitation of electrons in a conductor. It is distributed across the entire spectrum and that is why it is also known as *white noise* (as the frequency encompass over a broad range of frequencies). When more than one signal share a single transmission medium, ***intermodulation noise*** is generated. For example, two signals f_1 and f_2 will generate signals of frequencies $(f_1 + f_2)$ and $(f_1 - f_2)$, which may interfere with the signals of the same frequencies sent by the transmitter. Intermodulation noise is introduced due to nonlinearity present in any part of the communication system.

Cross talk is a result of bunching several conductors together in a single cable. Signal carrying wires generate electromagnetic radiation, which is induced on other conductors because of close proximity of the conductors. While using telephone, it is a common experience to hear conversation of other people in the background. This is known as *cross talk*.

Impulse noise is irregular pulses or noise spikes of short duration generated by phenomena like lightning, spark due to loose contact in electric circuits, etc. Impulse noise is a primary source of bit-errors in digital data communication. This kind of noise introduces burst errors.

Bandwidth and Channel Capacity

Bandwidth of a medium decides the quality of the signal at the other end. A digital signal (usually aperiodic) requires a bandwidth from 0 to infinity. So, it needs a low-pass channel characteristic as shown in Figure (a). On the other hand, a band-pass channel characteristic is required for the transmission of analog signals, as shown in Figure (b).

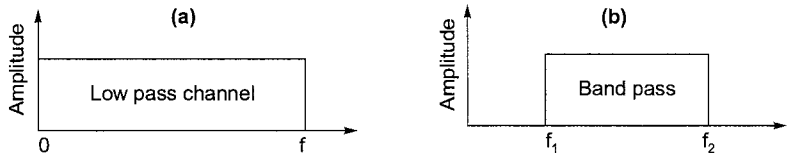


Figure: ***(a)*** *Low-pass channel characteristic required for the transmission of digital signals* ***(b)*** *Band-pass channel characteristic required for the transmission of analog signals*

Nyquist Bit Rate

The maximum rate at which data can be correctly communicated over a channel in presence of noise and distortion is known as its *channel capacity*. Consider first a noise- free channel of Bandwidth B. Based on Nyquist formulation it is known that given a bandwidth B of a channel, the maximum data rate that can be carried is 2B. This limitation arises due to the effect of intersymbol interference caused by the frequency components higher than B. If the signal consists of m discrete levels, then Nyquist theorem states:

Maximum data rate $C = 2B \log_2 m$ bits/sec,
where C is known as the channel capacity,
B is the bandwidth of the channel
and m is the number of signal levels used.

Baud Rate

The baud rate or signaling rate is defined as the number of distinct symbols transmitted per second, irrespective of the form of encoding. For baseband digital transmission m = 2.

So, the maximum baud rate = 1/Element width (in Seconds) = 2B

Bit Rate

The bit rate or information rate I is the actual equivalent number of bits transmitted per second.

$$I = \text{Baud Rate} \times \text{Bits per Baud}$$
$$= \text{Baud Rate} \times N = \text{Baud Rate} \times \log_2 m$$

For binary encoding, the bit rate and the baud rate are the same; i.e., I = Baud Rate.

Example **- 1.1** Let us consider the telephone channel having bandwidth B = 4 kHz. Assuming there is no noise, determine channel capacity for the following encoding levels: (a) 2, and (b) 128.

Solution:

(a) $C = 2B = 2 \times 4000 = 8$ Kbits/s

(b) $C = 2 \times 4000 \times \log_2 128 = 8000 \times 7 = 56$ Kbits/s

Effects of Noise

When there is noise present in the medium, the limitations of both bandwidth and noise must be considered. A noise spike may cause a given level to be interpreted as a signal of greater level, if it is in positive phase or a smaller level, if it is negative phase. Noise becomes more problematic as the number of levels increases.

Shannon Capacity (Noisy Channel)

In presence of Gaussian band-limited white noise, Shannon-Hartley theorem gives the maximum data rate capacity

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

Signal to noise ratio is represented in dB (decibell).

$\log_{10}(S/N)$ is in bells

$10 \times \log_{10}(S/N)$ is in decibells

+ve value of S/N means signal power is dominating, where as –ve value of S/N represents noise power is dominating signal.

Where S and N are the signal and noise power, respectively, at the output of the channel. This theorem gives an upper bound of the data rate which can be reliably transmitted over a thermal-noise limited channel.

Example: Suppose we have a channel of 3000 Hz bandwidth, we need an S/N ratio (i.e. signal to noise ration, SNR) of 30 dB to have an acceptable bit-error rate. Then, the maximum data rate that we can transmit is 30,000 bps. In practice, because of the presence of different types of noises, attenuation and delay distortions, actual (practical) upper limit will be much lower.

NOTE: In case of extremely noisy channel, C = 0. Between the Nyquist Bit Rate and the Shannon limit, the result providing the smallest channel capacity is the one that establishes the limit.

Example-1.2 A channel has B = 4 KHz. Determine the channel capacity for each of the following signal-to-noise ratios: (a) 20 dB, (b) 30 dB, (c) 40 dB.

Solution:

(a) $C = B \log_2 (1 + S/N) = 4 \times 10^3 \times \log_2 (1 + 100) = 4 \times 10^3 \times 3.32 \times 2.004 = 26.6$ kbits/s

(b) $C = B \log_2 (1 + S/N) = 4 \times 10^3 \times \log_2 (1 + 1000) = 4 \times 10^3 \times 3.32 \times 3.0 = 39.8$ kbits/s

(c) $C = B \log_2 (1 + S/N) = 4 \times 10^3 \times \log_2 (1 + 10000) = 4 \times 10^3 \times 3.32 \times 4.0 = 53.1$ kbits/s

Example-1.3 A channel has B = 4 KHz and a signal-to-noise ratio of 30 dB. Determine maximum information rate for 4-level encoding.

Solution:

For B = 4 KHz and 4-level encoding the *Nyquist Bit Rate* is 16 Kbps.

Again for B = 4 KHz and S/N of 30 dB the *Shannon capacity* is 39.8 Kbps.

The smallest of the two values has to be taken as the Information capacity I = 16 Kbps.

Example-1.4 A channel has B = 4 kHz and a signal-to-noise ratio of 30 dB. Determine maximum information rate for 128-level encoding.

Solution:

The *Nyquist Bit Rate* for B = 4 kHz and M = 128 levels is 56 kbits/s.

Again the *Shannon capacity* for B = 4 kHz and S/N of 30 dB is 39.8 Kbps.

The smallest of the two values decides the channel capacity C = 39.8 kbps.

Example-1.5 The digital signal is to be designed to permit 160 kbps for a bandwidth of 20 kHz. Determine (a) number of levels and (b) S/N ratio.

Solution:

(a) Apply *Nyquist Bit Rate* to determine number of levels. $C = 2B \log_2 (M)$,

or $160 \times 10^3 = 2 \times 20 \times 10^3 \log_2 (M)$,

or $M = 2^4$, which means 4bits/baud.

(b) Apply *Shannon capacity* to determine the S/N ratio $C = B \log 2 (1 + S/N)$,

or $160 \times 10^3 = 20 \times 10^3 \log_2 (1 + S/N) \times 10^3 \log_2 (M)$,

or $S/N = 2^8 - 1$, or $S/N = 255$,

or $S/N = 24.07$ dB.

Example-1.6 Assuming there is no noise in a medium of B = 4 kHz, determine channel capacity for the encoding level 4.

Solution:

$$I = 2 \times 4000 \times \log_2 4 = 16 \text{ Kbps}$$

Example-1.7 A channel has B = 10 MHz. Determine the channel capacity for signal-to-noise ratio 60 dB.

Solution:

$$C = B \times \log_2 (1 + S/N) = 10 \times \log_2 (1 + 60)$$

Example-1.8 The digital signal is to be designed to permit 56 kbps for a bandwidth of 4 kHz. Determine (a) number of levels and (b) S/N ratio.

Solution:

Nyquist bit rate $56 \times 10^3 = 2 \times 4 \times 10^3 \times \log_2 M$

$7 = \log_2 M \quad \Rightarrow M = 128$ levels

Shannon capacity $56 = B \times \log (1 + S/N)$

$S = 10^7 - 1 = 70$ dB

1.5 Transmission of Signals

The first approach converts digital data to digital signal, known as line coding, as shown in Figure. Important parameters those characteristics line coding techniques are mentioned below.

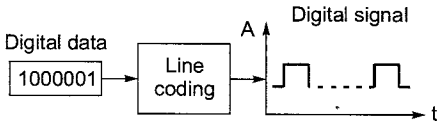


Figure: *Line coding to convert digital data to digital signal*

Number of Signal Levels

This refers to the number values allowed in a signal, known as **signal levels**, to represent data. Figure (a) shows two signal levels, whereas Figure (b) shows three signal levels to represent binary data.

Bit Rate versus Baud Rate

The **bit rate** represents the number of bits sent per second, whereas the **baud rate** defines the number of signal elements per second in the signal. Depending on the encoding technique used, baud rate may be more than or less than the data rate.

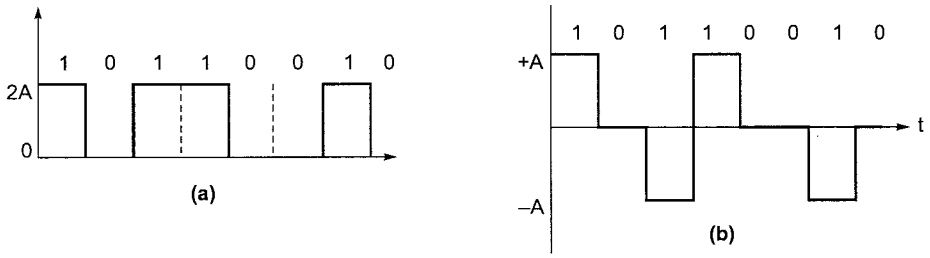


Figure: ***(a)*** *Signal with two voltage levels* ***(b)*** *Signal with three voltage levels*

Line Coding Techniques

Line coding techniques can be broadly divided into three broad categories: Unipolar, Polar and Bipolar, as shown in Figure.

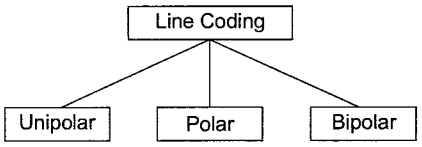


Figure: *Three basic categories of line coding techniques*

Unipolar

In unipolar encoding technique, only two voltage levels are used. It uses only one polarity of voltage level as shown in Figure. In this encoding approach, the bit rate same as data rate. Unfortunately, DC component present in the encoded signal and there is loss of synchronization for long sequences of 0's and 1's. It is simple but obsolete.

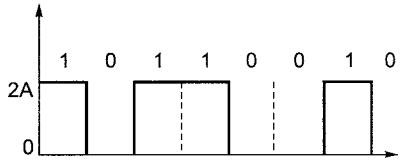


Figure: *Unipolar encoding with two voltage levels*

Polar

Polar encoding technique uses two voltage levels – one positive and the other one negative. Four different encoding schemes shown in Figure under this category discussed below:

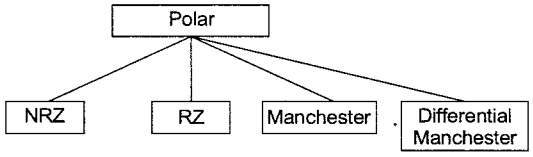


Figure: *Encoding Schemes under polar category*

Non Return to Zero (NRZ)

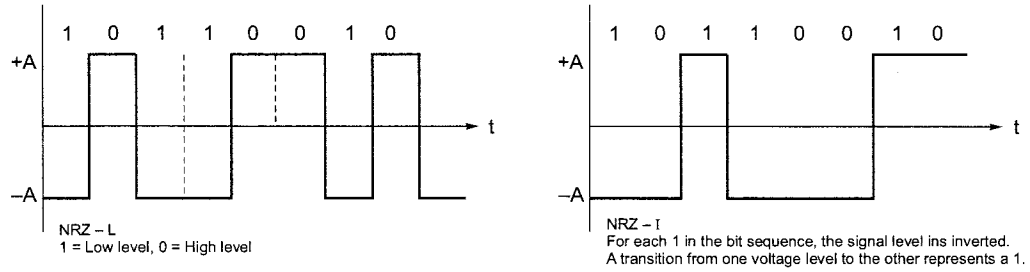


Figure: *NRZ encoding scheme*

The most common and easiest way to transmit digital signals is to use two different voltage levels for the two binary digits. Usually a negative voltage is used to represent one binary value and a positive voltage to represent the other. The data is encoded as the presence or absence of a signal transition at the beginning of the bit time. As shown in the figure below, in NRZ encoding, the signal level remains same throughout the bit-period. There are two encoding schemes in NRZ: NRZ-L and NRZ-I, as shown in Figure.

The **advantages** of NRZ coding are:

- Detecting a transition in presence of noise is more reliable than to compare a value to a threshold.
- NRZ codes are easy to engineer and it makes efficient use of bandwidth.

Return to Zero RZ

To ensure synchronization, there must be a signal transition in each bit as shown in Figure. Key characteristics of the RZ coding are:

- Three levels
- Bit rate is double than that of data rate
- No dc component
- Good synchronization
- Main limitation is the increase in bandwidth

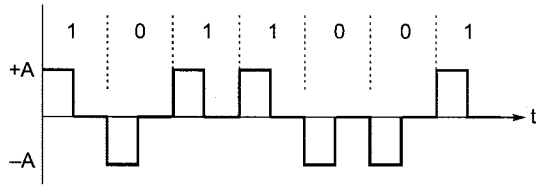


Figure: *RZ encoding technique*

Biphase

To overcome the limitations of NRZ encoding, biphase encoding techniques can be adopted. *Manchester* and *differential Manchester Coding* are the two common Biphase techniques in use, as shown in Figure (a) & (b). In Manchester coding the mid-bit transition serves as a clocking mechanism and also as data.

In the standard Manchester coding there is a transition at the middle of each bit period. A binary **1** corresponds to a ***low-to-high transition*** and a binary **0** to a ***high-to-low transition*** in the middle. In Differential Manchester, inversion in the middle of each bit is used for synchronization. The encoding of a 0 is represented by the presence of a transition both at the beginning and at the middle and 1 is represented by a transition only in the middle of the bit period.

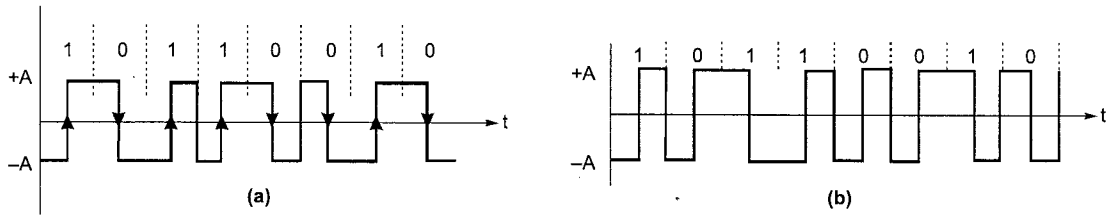


Figure: *(a) Standard Manchester Encoding Scheme (b) Differential Manchester Encoding Scheme*

Key Characteristics

- Two levels
- No DC component
- Good synchronization
- Higher bandwidth due to doubling of bit rate with respect to data rate

The bandwidth required for biphase techniques are greater than that of NRZ techniques, but due to the predictable transition during each bit time, the receiver can synchronize properly on that transition. Biphase encoded signals have no DC components. A Manchester code is now very popular and has been specified for the IEEE.

802.3 standard for base band coaxial cables and twisted pair CSMA/CD bus LANs.

Bipolar Encoding

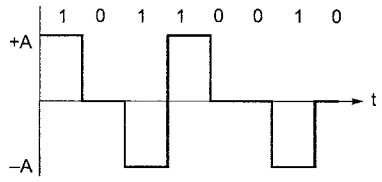


Figure: *Bipolar AMI signal*

Bipolar AMI uses three voltage levels. Unlike RZ, the zero level is used to represent a 0 and a binary 1's are represented by alternating positive and negative voltages, as shown in Figure.

Pseudoternary

This encoding scheme is same as AMI, but alternating positive and negative pulses occur for binary 0 instead of binary 1. Key characteristics are:

- Three levels
- No DC component
- Loss of synchronization for long sequences of 0's
- Lesser bandwidth

Modulation Rate

Data rate is expressed in bits per second. On the other hand, modulation rate is expressed in bauds. General relationship between the two is given below:

$$D = R/b = R/\log_2 L$$

Where, D is the modulation rate in bauds, R is the data rate in bps, L is the number of different signal elements and b is the number of bits per signal element.

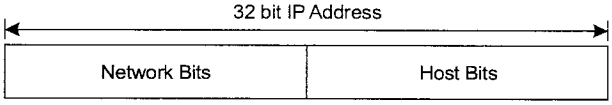
1.6 IP Addressing

The IP address is a 32-bit address that consists of two components. One component is the network portion of the address, consisting of the network bits.

- The network bits make up the left portion of the address.
- They consist of the first bit up to some boundary, to be discussed later.

The second component is the host portion of the address, consisting of the host bits.

- The host bits make up the right portion of the address.
- They consist of the remaining bits not included with the network bits.



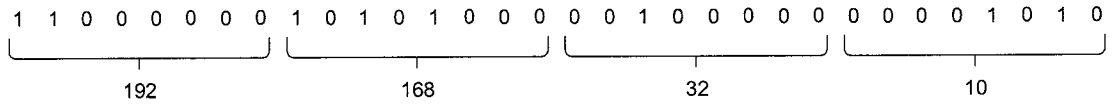
The Mask

- The network portion of the address is separated from the host portion of the address by a mask. The mask simply indicates how many bits are used for the network portion, leaving the remaining bits for the host portion.
- A 24-bit mask indicates that the first 24 bits of the address are network bits, and the remaining 8 bits are host bits.
- A 16-bit mask indicates that the first 16 bits of the address are network bits, and the remaining 16 bits are host bits. And so forth...
- The difference between a network mask and a subnet mask will be explained as this tutorial progresses.

Dotted Decimal Notation

Machines read the IP address as a stream of 32 bits. However, for human consumption, the IP address is written in dotted decimal notation.

- The 32-bit address is divided into 4 groups of 8 bits (an octet or a byte).
- Each octet is written as a decimal number ranging from 0 to 255.
- The decimal numbers are separated by periods, or dots.



For a given IP network...

- The network bits remain fixed and the host bits vary.
- The network address is the one that results when all the host bits are not set (the result of performing an AND operation on the address and its mask).
- The broadcast address is the one that results when all the host bits are set.
- Host addresses are those that result with all remaining combinations of the host bits.

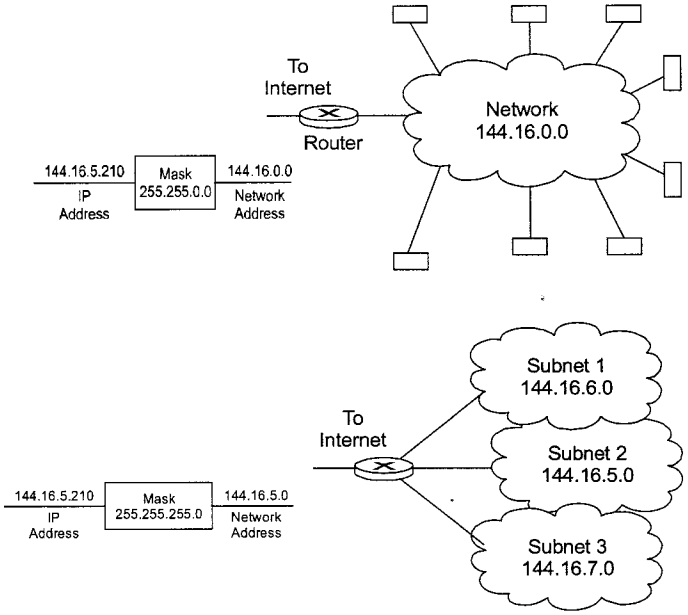
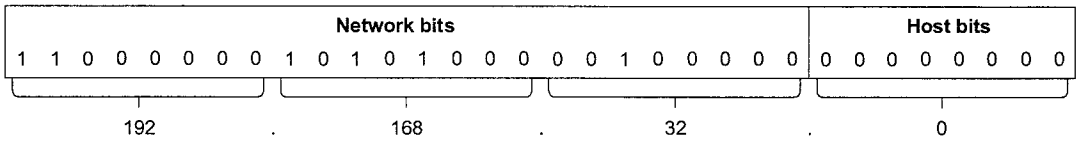


Figure: *Sub netting masking with the help of router*

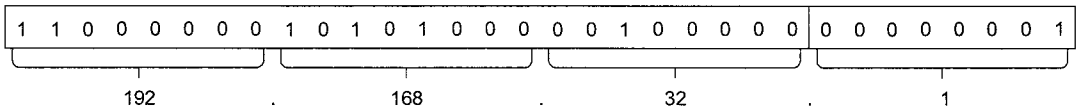
Network, Host and Broadcast Addresses

24-bit mask (255.255.255.0)

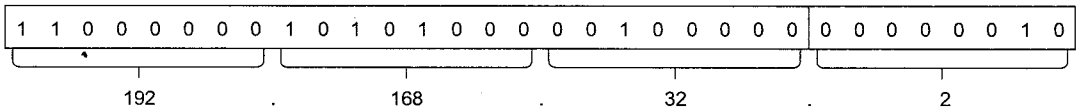
Network address w/24-bit mask 255.255.255.0



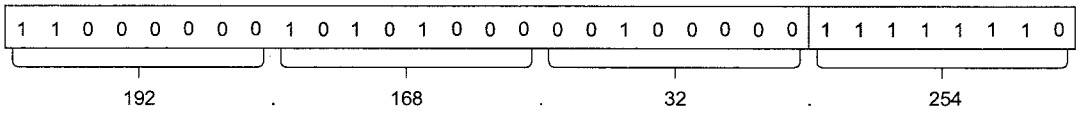
First hot address for this network



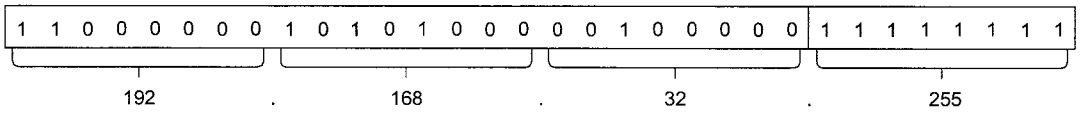
Second hot address for this network



Last hot address for this network

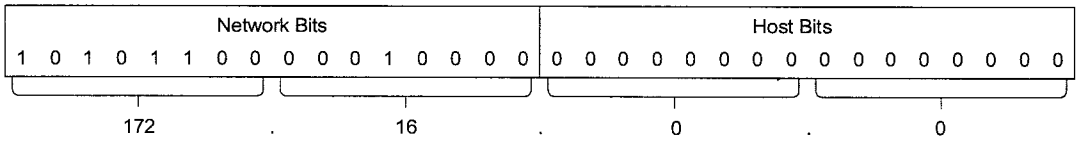


Broadcast address for this network

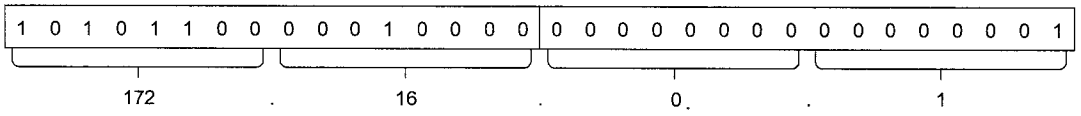


16 bit Mask

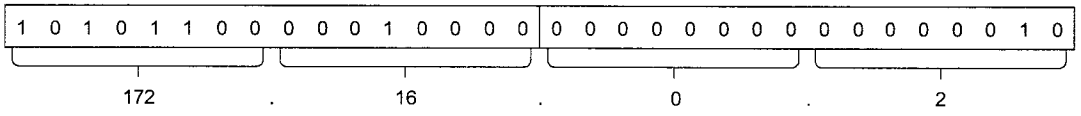
Network address w/ 16-bit mask 255.255.0.0



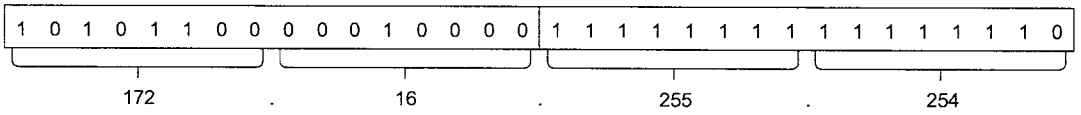
First hot address for this network



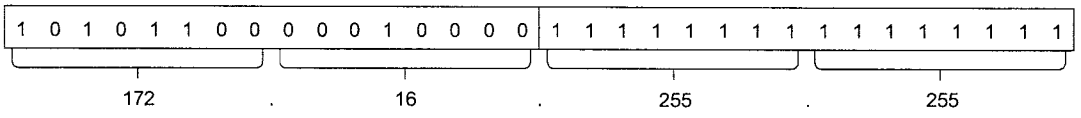
Second hot address for this network



Last hot address for this network



Broadcast address for this network



IP Networks and Hosts

- An IP host is any device with an IP address, such as a PC.
- Multiple hosts reside on a given IP network or subnet (short for subnetwork). Subnets will be discussed later.
- A group of IP networks is an internetwork, with the largest internetwork being the Internet.
- What is typically called a "data network" is technically an internetwork, because multiple IP networks are connected together by routers.
- This internetwork contains 6 IP networks.
- Note that even a link between routers is a network.

Determining Number of hosts in given network

- Given that there are N host bits in an address, the number of hosts for that network is $2^N - 2$. Two addresses are subtracted for the network address and the broadcast address.
- 8 host bits: $2^8 - 2 = 254$ hosts
- 16 host bits: $2^{16} - 2 = 65534$ hosts
- 24 host bits: $2^{24} - 2 = 16777214$ hosts

Public IP Addresses

- Most IP addresses are public addresses. Public addresses are registered as belonging to a specific organization.
- Internet Service Providers (ISP) and extremely large organizations in the U.S. obtains blocks of public addresses from the American Registry for Internet Numbers (ARIN http://www.arin.net). Other organizations obtain public addresses from their ISPs.
- There are ARIN counterparts in other parts of the world, and all of these regional registration authorities are subject to the global Internet Assigned Numbers Authority (IANA http://www.iana.org).
- Public IP addresses are routed across the Internet, so that hosts with public addresses may freely communicate with one another globally.
- No organization is permitted use public addresses that are not registered with that organization!

Private Addresses

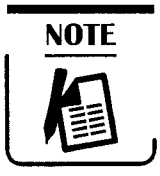
RFC 1918 designates the following as private addresses.

- **Class A range:** 10.0.0.0 through 10.255.255.255.
- **Class B range:** 172.16.0.0 through 172.31.255.255.
- **Class C range:** 192.168.0.0 through 192.168.255.255.

Private addresses may be used by any organization, without any requirement for registration.

Because private addresses are ambiguous - can't tell where they're coming from or going to because anyone can use them - private addresses are not permitted to be routed across the Internet.

- ISPs block private addresses from being routed across their infrastructure.



- The use of private addresses, network address translation (NAT), and proxy servers solved the IP address shortage problem for the short and medium terms. The projected long-term solution is IPv6.

1.6.1 Classful IP Addressing and its Shortcomings

- Class A networks:
  (a) First octet values range from 1 through 126.
  (b) First octet starts with bit **0**.
  (c) Network mask is 8 bits, written /8 or 255.0.0.0.
  (d) 1.0.0.0 through 126.0.0.0 are class A networks with 16777214 hosts each.
- Class B networks:
  (a) First octet values range from 128 through 191.
  (b) First octet starts with binary pattern **10**.
  (c) Network mask is 16 bits, written /16 or 255.255.0.0.
  (d) 128.0.0.0 through 191.255.0.0 are class B networks, with 65534 hosts each.
- Class C networks:
  (a) First octet values range from 192 through 223.
  (b) First octet starts with binary pattern **110**.
  (c) Network mask is 24 bits, written /24 or 255.255.255.0.
  (d) 192.0.0.0 through 223.255.255.0 are class C networks, with 254 hosts each.
- Class D addresses:
  (a) First octet values range from **224 through 239**.
  (b) First octet starts with binary pattern **1110**.
  (c) Class D addresses are multicast addresses, which will not be discussed in this tutorial.
- Class E addresses:
  (a) Essentially everything that's left.
  (b) Experimental class, which will not be discussed in this tutorial.
- Reserved addresses:
  (a) **0.0.0.0** is the default IP address, and it is used to specify a **default route**. The default route will be discussed later.
  (b) Addresses beginning with 127(**127.x.y.z**) are reserved for **internal loopback addresses**.
- It is common to see 127.0.0.1 used as the internal loopback address on many devices. Try pinging this address on a PC or UNIX station.

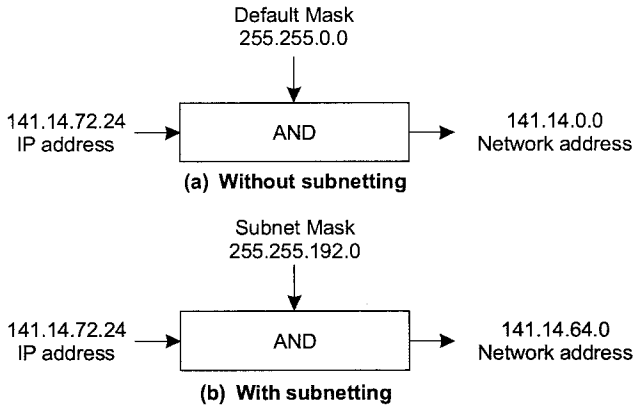
1.6.2 IP Addresses Efficiency

- As IP networking and internetworking progressed, it became very apparent that class A and B networks were simply too large.
- 254 hosts on one network segment are manageable, but 65534 hosts or more on a single network segment is difficult to manage.
  (a) This would result in class A and B networks not being fully utilized, meaning that not all the host addresses would get used.
  (b) Or it would result in more hosts being put onto a single network segment than could reasonably be managed.
- For these and other reasons, there was a need to improve the efficiency of IP addressing. That is, to provide a way to limit the number of host addresses per network segment to what is actually needed, regardless of the network class.
- This need was met progressively through the conceptions of subnet masks, variable-length subnet masks, and classless inter-domain routing.

1.7 Subnetting

Subnet masks are used to make classful networks more manageable and efficient, by creating smaller subnets and reducing the number of host addresses per subnet to what is actually required. Subnet masks were first used on class boundaries.

Default Mask and Subnet Mask



Example:

- Take class A network 10.0.0.0 with network mask 255.0.0.0.
- Add additional 8 subnet bits to network mask.
- New subnet mask is 255.255.0.0.
- New subnets are 10.0.0.0, 10.1.0.0, 10.2.0.0, and so on with 65534 host addresses per subnet.
- Still too many hosts per subnet.

Example:

- Take class A network 10.0.0.0 with network mask 255.0.0.0.
- Add additional 16 subnet bits to network mask.
- New subnet mask is 255.255.255.0
- New subnets are 10.0.0.0, 10.0.1.0, 10.0.2.0... 10.1.0.0, 10.1.1.0, 10.1.2.0... 10.2.0.0, 10.2.1.0, 10.2.2.0, and so on with 254 host addresses per subnet.

Example

- Take class B network 172.16.0.0 with network mask 255.255.0.0. Add additional 8 subnet bits to network mask.
- New subnet mask is 255.255.255.0
- New subnets are 172.16.0.0, 172.16.1.0, 172.16.2.0, and so on with 254 host addresses per subnet.

As shown in these examples...

- A class A network can be subnetted to create 256 (2^8) /16 subnets.
- A class A network can be subnetted to create 65536 (2^{16}) /24 subnets.
- A class B network can be subnetted to create 256 (2^8) /24 subnets.

NOTE

- Technically there really is no such thing as a classful subnet or classful subnet mask. However, terms such as "class C subnet" and "class C subnet mask" are used routinely to describe a class A or B network that has been subnetted with a 24-bit mask.
- It should also be apparent by now that the terms network mask and subnet mask technically mean two different things.

1.8 CIDR (Classless Inter Domain Routing)

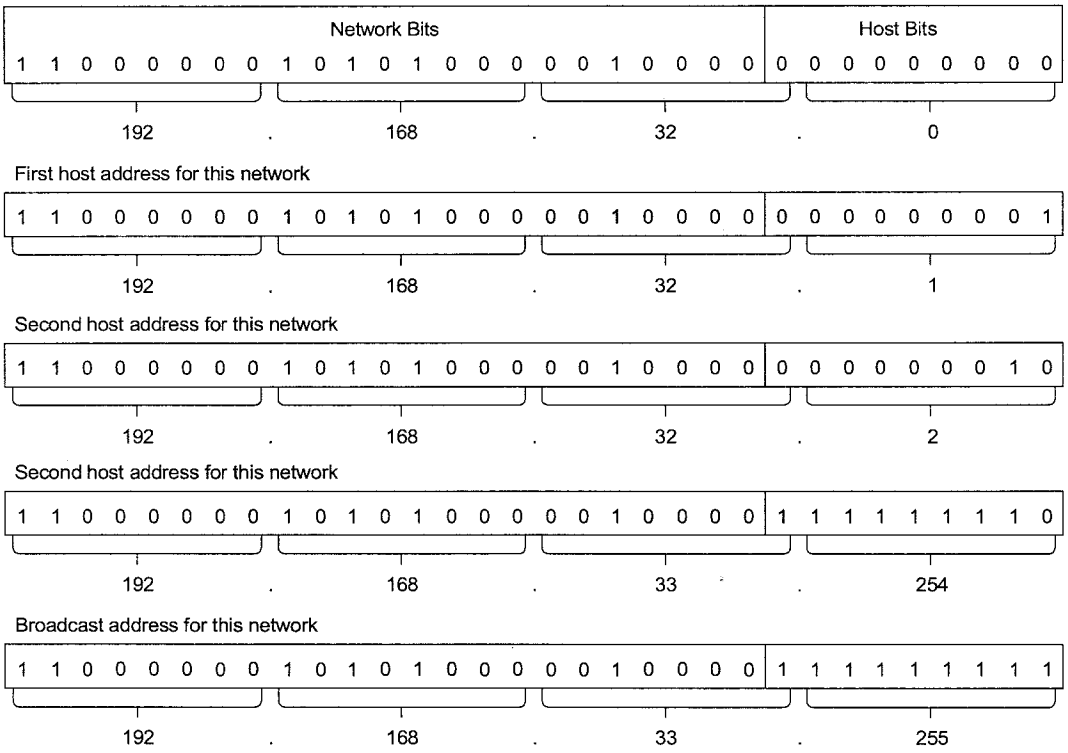
Super Netting

Very simply stated, CIDR is combining two or more classful networks to create a supernetwork.

- The most common use of CIDR for actual addressing is to combine two or more class C networks to create a /23 or /22 supernet. For *example*, the class C networks 192.168.32.0 and 192.168.33.0 could be combined to create 192.168.32.0/23.
- The class C networks 192.168.34.0 and 192.168.35.0 could be combined to create 192.168.34.0/23.

Example:

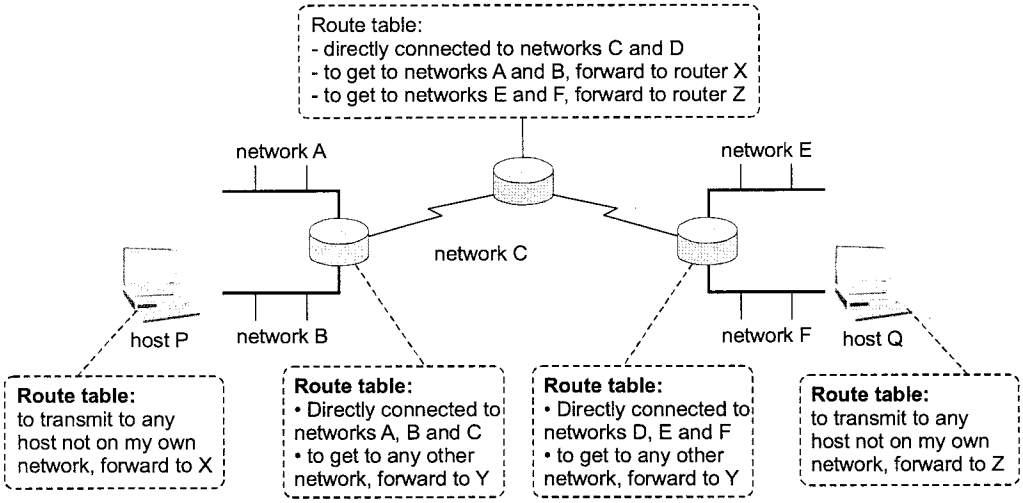
Network address of a supernet w/ 23-bit mask 255.255.254.0



- There really is no significant difference between the /23 supernet examples just shown and the /23 subnet examples shown previously.
  (a) Both networks utilize the same 23-bit mask.
  (b) Both networks have 510 host addresses.
  (c) The only difference is that a class A network was subnetted in one set of examples, and two class C networks were supernetted in the other.
- As with all things in IP addressing, supernets must fall on bit boundaries.
  (a) There must be a power of 2 number (2, 4, 8, ...) of networks in a supernet.
- A /23 supernet must include two class C networks.
- A /22 supernet must include four class C networks.

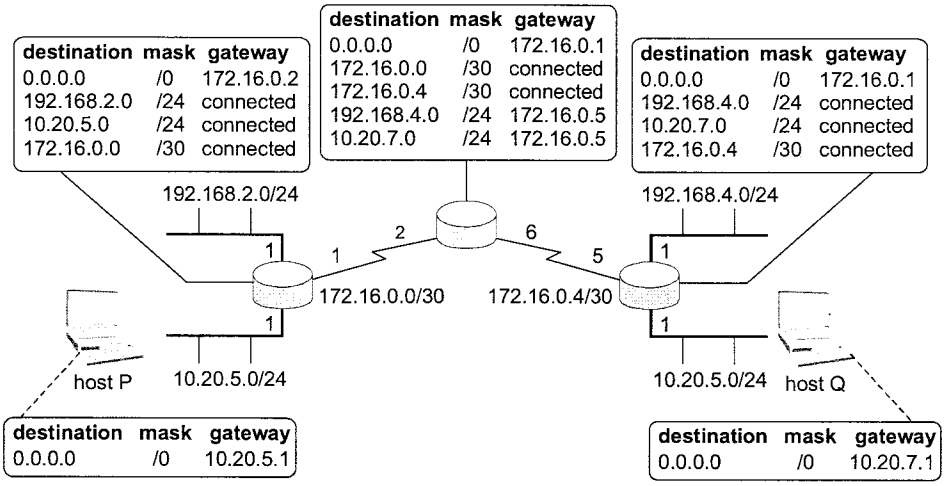
For *example*, it is not possible to create a supernet that includes just 192.168.1.0 and 192.168.2.0. To include both these networks the mask must be 22 bits, and a 22-bit mask must also include the networks 192.168.0.0 and 192.168.3.0.

- Although the term supernet is used in this tutorial to explain CIDR, this term is not commonly used in casual communication. Instead, most will simply use the term network.



- Hosts P and Q
  (a) Hosts P and Q have only one way to get off their networks, and that is to forward traffic to the router connected to their respective networks.
  (b) These hosts each have a default route to a default gateway-router X or Z.
- Router X
  (a) Router X has interfaces directly connected to networks A, B, and C, so it can route traffic between these networks w/o additional configurations.
  (b) This router has only one option to get to other networks, and that is to forward traffic to router Y. So it has a default route to router Y.
- Router Z
  (a) Router Z has interfaces directly connected to networks D, E, and F, so it can route traffic between these networks w/o additional configurations.
  (b) This router has only one option to get to other networks, and that is to forward traffic to router Y. So it has a default route to router Y.
- Router Y
  (a) Router Y automatically knows how to route traffic between networks C and D.
  (b) To get to networks A and B, this router forwards traffic to router X.
  (c) To get to networks E and F, this router forwards traffic to router Z.
  (d) One of these routes could be made the default route.

Same Routing with IP addresses



- In the preceding diagram, the network could not have worked with classful routing.
- Without subnet masks, router Y would have a route to class A network 10.1.1.1 via gateway 172.16.0.5.
  (a) This would result in all 10.x.x.x packets that traverse router Y, being forwarded to router Z.
  (b) For example, router Z would forward destination 10.20.5.x packets to router Y, and router Y would return them to router Z.
  (c) This is because router Y has no way to distinguish between 10.20.5.0 and 10.20.4.0 without a subnet mask.
  (d) Destination 10.20.5.x packets within router X-that is, those sourced from 192.168.2.x - would not be affected by this phenomenon.

***Example*-1.9** What is the subnetwork address if the destination address is 200.45.34.56 and the subnet mask is 255.255.240.0?

Solution:

200	45	34	56
11001000	00101101	00100010	00111000
11111111	11111111	1111 0000	00000000
11001000	00101101	00100000	00000000

The subnet work address is **200.45.32.0**

***Example*-1.10** A company is granted the site address 201.70.64.0 (class C). The company needs six subnets of 25 host each. Design the subnets.

Solution:

The number of 1s in the default mask is 24 (class C).

The company needs six subnets. This number 6 is not a power of 2. The next number that is a power of 2 is 8 (2^3). We need 3 more 1s in the subnet mask. The total number of 1s in the subnet mask is 27 (24 + 3).

The total number of 0s is 5 (32 - 27). The mask is

11111111 11111111 11111111 11100000 or 255.255.255.224

the number of subnets theoretically is 8. But Practically only 6 subnets are possible. (All 1's and all 0's in subnet Id bits because they are Direct Broad Cast address and Network Id respectively). The subnets are

201	70	64		
11001001	01000110	01000000	[000] 00000	Network Id (201.70.64.0)
11001001	01000110	01000000	[001]00000	1st Subnet ID 201.70.64.32
11001001	01000110	01000000	[010]00000	2nd subnet ID 201.70.64.64
11001001	01000110	01000000	[011]00000	3rd subnet ID 201.70.64.96
11001001	01000110	01000000	[100]00000	4th subnet ID 201.70.64.128
11001001	01000110	01000000	[101]00000	5th subnet Id 201.70.64.160
11001001	01000110	01000000	[110]00000	6th subnet ID 201.70.64.224
11001001	01000110	01000000	[111]00000	

[111] cannot be used as subnet id because in last octet all 1's is used as DBA (direct broadcast address) of the network **201.70.64.0 i.e.** 11111111 11111111 11111111 **[111]**11111 **is DBA of the network.**

The number of addresses in each subnet is 25 (5 is the number of 0s) or 32.

***Example*-1.11** A company is granted the site address 181.56.0.0 (class B). The company needs 1000 subnets. What is the DBA of the first subnet?

Solution:

The company needs 1000 subnets. We need 10 bits to represent those 1000 subnets.

The total number of 1s in the subnet mask is 26 (16 + 10). Six bits are left for host bits.

Therefore we can have $2^6 - 2$ hosts. (All 1's in host bits represent DBA and all 0's would be the case of Network ID (181.56.0.0)

The MASK for these subnets is

11111111	11111111	11111111	11000000	or	181.56.255.192

The first subnet ID is

10110101	00111000	00000000	01000000	or	181.56.0.64

The first host of first subnet is

10110101	00111000	00000000	01000001	or	181.56.0.65

2nd host

10110101	00111000	00000000	01000010	or	181.56.0.66

DBA (direct broadcast address) of first subnet

10110101	00111000	00000000	01111111	or	181.56.0.127

The number of subnets is 1024 and the number of addresses in each subnet is 26 (6 is the number of 0s) or 64.

***Example*-1.12** A company is granted the site address 181.56.0.0 (class B). The company needs 1000 subnets. What is the DBA of the 4th subnet?

Solution:

4th subnet id is

10110101	00111000	00000001	00000000	or	181.56.1.0

DBA of 4th subnet (all host bits 1's) is

10110101	00111000	00000001	00111111	or	181.56.1.63

Example-1.13 A company is granted the site address 181.56.0.0 (class B). The company needs 1000 subnets.

(a) What is the DBA of the last subnet?

(b) What is address of 4th host of 2nd subnet in the company?

Solution:

Last subnet ID is

10110101 00111000 **11111111** **11000000** or 181.56.255.192

DBA of last subnet is

10110101 00111000 **11111111** **11111111** or 181.56.255.255

The last four subnet ID's used by the company are

10110101 00111000 **11111110** **11000000** or 181.56.254.192

10110101 00111000 **11111111** **00000000** or 181.56.255.0

10110101 00111000 **11111111** **01000000** or 181.56.255.64 (2nd last)

10110101 00111000 **11111111** **10000000** or 181.56.255.128

4th last host of 2nd last subnet is

10110101 00111000 **11111111** 11111011 181.56.255.251

Example-1.14 Which of the following can be used as both source IP as well as destination IP (a) 192.168.11.255 (b) 143.18.255.255 (c) 255.255.255.255 (d) 19.19.19.25

Solution:

(a) NO it cannot be used as source IP as it is a DBA (direct broadcast address)

(b) NO it is DBA

(c) NO it is Limited broadcast address and it cannot be used as source

(d) YES it can be used as both source and destination IP

Example-1.15 Identify which of the following IP's belong to same subnet. Given subnet mask is 255.255.255.240?

(a) 207.19.36.58 (b) 207.19.36.75

(c) 207.19.36.89 (d) 207.19.36.97

(e) None of them

Solution: (e)

240 (11110000) first four bits of last octet are subnet bits. If first 4 bits match then they both belong to same subnet

58-**0011**1010

75-**0100**1011

89-**0101**1001

97-**0111**0001

None of them belong to same subnet.

Example-1.16 A company needs 600 addresses. Which of the following set of class C blocks can be used to form a supernet for this company?

(a) 198.47.32.0 198.47.33.0 198.47.50.0

(b) 198.47.32.0 198.47.42.0 198.47.52.0 198.47.62.0

(c) 198.47.31.0 198.47.32.0 198.47.33.0 198.47.52.0
(d) 198.47.32.0 198.47.33.0 198.47.34.0 198.47.35.0

Solution: (d)
In order to combine networks into a single Network (super netting) the networks must be continuous and number of networks to be combined should be powers of 2.

***Example*-1.17** We need to make a supernetwork out of 16 class C blocks. What is the supernet mask?

Solution:
We need 16 blocks. For 16 blocks we need to change four 1s to 0s in the default mask. So the mask is

11111111 11111111 11110000 00000000 or 255.255.240.0

***Example*-1.18** A supernet has a first address of 205.16.32.0 and a supernet mask of 255.255.248.0. A router receives three packets with the following destination addresses:

205.16.37.44, 205.16.42.56 and 205.17.33.76

Which packet belongs to the supernet?

Solution:
We apply the supernet mask to see if we can find the beginning address
205.16.37.44 AND 255.255.248.0 ⇒ 205.16.32.0
205.16.42.56 AND 255.255.248.0 ⇒ 205.16.40.0
205.17.33.76 AND 255.255.248.0 ⇒ 205.17.32.0
Only the first address belongs to this supernet.

***Example*-1.19** A supernet has a first address of 205.16.32.0 and a supernet mask of 255.255.248.0. How many blocks are in this supernet and what is the range of addresses?

Solution:
The supernet has 21 1s. The default mask has 24 1s. Since the difference is 3, there are 2^3 or 8 blocks in this supernet. The blocks are 205.16.32.0 to 205.16.39.0. The first address is 205.16.32.0. The last address is 205.16.39.255.

***Example*-1.20** Which of the following can be the beginning address of a block that contains 16 addresses?

(a) 123.45.24.52 (b) 17.17.33.80
(c) 190.16.42.44 (d) 205.16.37.32

Solution:
The condition on the number of addresses in a block; it must be a power of 2 (2, 4, 8 ...)
The beginning address must be evenly divisible by the number of addresses. For example, if a block contains 4 addresses, the beginning address must be divisible by 4.
If the block has less than 256 addresses, we need to check only the rightmost byte. If it has less than 65,536 addresses, we need to check only the two rightmost bytes, and so on.
The address 205.16.37.32 is eligible because 32 is divisible by 16. The address 17.17.33.80 is eligible because 80 is divisible by 16.

Example-1.21 Which of the following can be the beginning address of a block that contains 1024 addresses?

(a) 205.16.37.32 (b) 190.16.42.0
(c) 17.17.32.0 (d) 123.45.24.52

Solution: (c)

To be divisible by 1024, the rightmost byte of an address should be 0 and the second rightmost byte must be divisible by 4. Only the address 17.17.32.0 meets this condition.

Example-1.22 A small organization is given a block with the beginning address and the prefix length 205.16.37.24/29 (in slash notation). What is the range of the block?

Solution:

The beginning address is 205.16.37.24. To find the last address we keep the first 29 bits and change the last 3 bits to 1s.

Beginning:	11001111	00010000	00100101	00011**000**
Ending:	11001111	00010000	00100101	00011**111**

There are only 8 addresses in this block

Example-1.23 What is the network address if one of the addresses is 167.199.170.82/27?

Solution:

The prefix length is 27, which means that we must keep the first 27 bits as is and change the remaining bits (5) to 0s. The 5 bits affect only the last byte. The last byte is 01010010. Changing the last 5 bits to 0s, we get 01000000 or 64. The network address is 167.199.170.64/27.

Example-1.24 An ISP is granted a block of addresses starting with 190.100.0.0/16. The ISP needs to distribute these addresses to three groups of customers as follows:

(a) The first group has 64 customers; each needs 256 addresses.
(b) The second group has 128 customers; each needs 128 addresses.
(c) The third group has 128 customers; each needs 64 addresses.

Design the sub blocks and give the slash notation for each sub block. Find out how many addresses are still available after these allocations.

Solution:

Group 1

For this group, each customer needs 256 addresses. This means the suffix length is 8 ($2^8 = 256$). The prefix length is then 32 – 8 = 24.

01: 190.100.0.0/24 ⇒ 190.100.0.255/24
02: 190.100.1.0/24 ⇒ 190.100.1.255/24...

And so on up to

64: 190.100.63.0/24 ⇒ 190.100.63.255/24

Total = 64 × 256 = 16,384

Group 2

For this group, each customer needs 128 addresses. This means the suffix length is 7 ($2^7 = 128$). The prefix length is then 32 – 7 = 25.

The addresses are:

001: 190.100.64.0/25 ⇒ 190.100.64.127/25

002: 190.100.64.128/25 ⇒ 190.100.64.255/25
003: 190.100.127.128/25 ⇒ 190.100.127.255/25
Total = 128 × 128 = 16,384

Group 3

For this group, each customer needs 64 addresses. This means the suffix length is 6 ($2^6 = 64$). The prefix length is then 32 – 6 = 26.
001:190.100.128.0/26 ⇒ 190.100.128.63/26
002:190.100.128.64/26 ⇒ 190.100.128.127/26...
And so on up to
128: 190.100.159.192/26 ⇒ 190.100.159.255/26
Total = 128 × 64 = 8,192
Number of granted, allocated and available addresses are : 65,536 , 40,960 and 24,576 respectively.

1.9 Network Address Translation (NAT)

With the increasing number of internet users requiring an unique IP address for each host, there is an acute shortage of IP addresses (until everybody moves to IPV6). The *Network Address Translation* (NAT) approach is a quick interim solution to this problem. NAT allows a large set of IP addresses to be used in an internal (private) network and a handful of addresses to be used for the external internet. The internet authorities has set aside three sets of addresses to be used as private addresses as shown in Table. It may be noted that these addresses can be reused within different internal networks simultaneously, which in effect has helped to increase the lifespan of the IPV4. However, to make use of the concept, it is necessary to have a router to perform the operation of address translation between the private network and the internet. As shown in Figure, the NAT router maintains a table with a pair of entries for private and internet address. The source address of all outgoing packets passing through the NAT router gets replaced by an internet address based on table look up. Similarly, the destination address of all incoming packets passing through the NAT router gets replaced by the corresponding private address, as shown in the figure. The NAT can use a pool of internet addresses to have internet access by a limited number of stations of the private network at a time.

Range of addresses		Total number
10.0.0.0	to 10.255.255.255	2^{24}
172.16.0.0	to 172.31.255.255	2^{20}
192.168.0.0	to 192.168.255.255	2^{16}

Addresses for Private Network

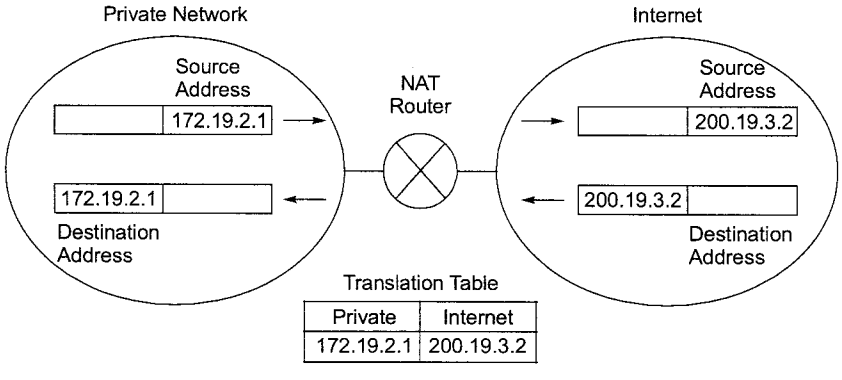


Figure: *NAT Address translation*

Summary

- A data communications system must transmit data to the correct destination in an accurate and timely manner.
- The five components that make up a data communications system are the message, sender, receiver, medium, and protocol. Text, numbers, images, audio, and video are different forms of information. Data flow between two devices can occur in one of three ways: simplex, half-duplex, or full-duplex.
- **Base-band** is defined as one that uses digital signaling, which is inserted in the transmission channel as voltage pulses.
- **Broadband** systems are those, which use analog signaling to transmit information using a carrier of high frequency.
- Co-axial cable has superior frequency characteristics compared to twisted-pair and can be used for both analog and digital signaling.
- The term *broadband* refers to analog transmission over coaxial cable.
- Fibre optics has very high data rate, and low error rate.
- Unguided transmission is used when running a physical cable (either fiber or copper) between two end points is not possible.
- The baud rate or signaling rate is defined as the number of distinct symbols transmitted per second, irrespective of the form of encoding.
- The bit rate or information rate is the actual equivalent number of bits transmitted per second.
- The **bit rate** represents the number of bits sent per second, whereas the **baud rate** defines the number of signal elements per second in the signal. Depending on the encoding technique used, baud rate may be more than or less than the data rate.
- In the standard Manchester coding there is a transition at the middle of each bit period. A binary **1** corresponds to a ***low-to-high transition*** and a binary **0** to a ***high-to-low transition*** in the middle.
- In Differential Manchester, inversion in the middle of each bit is used for synchronization.
- IP defines how computers can get data to each other over a routed, interconnected set of networks. TCP defines how applications can create reliable channels of communication across such a network.

Student's Assignment

Q.1 A researcher wishes to digitally record analog sounds for testing animal hearing with frequencies of up to 100 [kHz]. Use Shannon's formula to find the minimum signal to noise ratio (in dB!) required to sustain the given data rate over a 500 KHz radio channel.
(a) 19.2 (b) 18
(c) 14 (d) 19.5

Q.2 The correct order of the corresponding OSI layers for a Router, Medium Access Control, Repeater and FTP is
(a) Network, Data link, Application and Physical
(b) Physical, Data Link, Session and Transport
(c) Network, Data Link, Physical and Application
(d) Presentation, Network, Transport and Application

Q.3 Which of the following statement is correct with respect to Half Duplex communication?
(a) Error correction is not possible
(b) Always a pair of physical links is necessary
(c) Two connections can be used for sending and receiving data
(d) It cannot be used in broadcast networks

Q.4 What addressing information is shipped with every network interface card?
(a) The Internet protocol (IP) address
(b) The physical (MAC) address
(c) The address resolution protocol (ARP) address
(d) None of the above

Q.5 Which layer deals with
(*i*) File system transfer
(*ii*) Virtual terminals implementation
(*iii*) Inter process communication
(a) Application (b) Presentation
(c) Session (d) Transport

Q.6 The data link layer is responsible for
(a) Flow control (b) Error control
(c) Access control (d) All of these

Q.7 The topology with highest reliability is
(a) BUS (b) STAR
(c) RING (d) MESH

Q.8 Transport layer is responsible for
(a) node-to-node delivery
(b) end-to-end delivery
(c) station-to-station delivery
(d) network-to-network delivery

Q.9 Which network topologies degrades most gracefully in high network load situations?
(a) Ring (b) Star
(c) Mesh (d) Bus

Q.10 You have been asked to wire a conference room with six computers for a demonstration tomorrow. It needs be done as quickly and with the lowest possible expense. What network topology is best suited to this situation?
(a) Star (b) Ring
(c) Mesh (d) Bus

Q.11 Which layer is responsible for data translating?
(a) Application (b) Network
(c) Presentation (d) Data link

Q.12 Which one is the least expensive that can support 100 Mbps?
(a) Coaxial (b) UTP
(c) Fiber-optic (d) STP

Q.13 With the use of which of the following device(s) and cables can a LAN based on star topology be setup?
(a) Router (b) Bridge
(c) Switch (d) Repeater

Q.14 Identify the unequal pair(s).
(a) Physical Address – MAC Address
(b) IP Address – Logical Address
(c) Ethernet – IEEE 802.4
(d) Serial port – COM1

Q.15 In a topology, if there are n devices in a network, each device has $n-1$ ports for cables. Identify it
(a) Mesh (b) Star
(c) Ring (d) Bus

Q.16 In a mesh topology with n devices and half duplex links, if a new device is added, how many new links are needed.

(a) n – 1 (b) n
(c) n + 1 (d) 2 n

Q.17 Match **List-I** (Function) with **List-II** (Layer) and select the correct answer using the codes given below the lists:

List-I

A. Reassembly of packets
B. Responsibility for delivery between adjacent nodes
C. Mechanical electrical and functional interface
D. Error correction and retransmission

List-II

1. Transport
2. Data link
3. Physical

Codes:

	A	B	C	D
(a)	1	2	3	4
(b)	1	3	2	2
(c)	1	1	3	2
(d)	1	2	3	1, 2

Q.18 Consider the OSI standard for LANs,

(a) the OSI network layer is subdivided into a MAC layer and a LLC layer
(b) the OSI data link layer is subdivided into an Ethernet layer and a Token ring layer
(c) the OSI data link layer is subdivided into a MAC layer a LLC layer
(d) the OSI physical layer is subdivided in to an Ethernet layer and a Token Ring layer

Q.19 Match **List-I** with **List-II** and select the correct answer using the codes given below the lists:

List-I

A. Data link layer
B. Physical layer
C. Presentation layer
D. Network layer

List-II

1. The lowest layer whose function is to activate, deactivate and maintain the circuit between DTE & DCE
2. Performs routing & communication
3. Detection and recovery from errors in the transmitted data
4. Provides for the syntax of the data

Codes:

	A	B	C	D
(a)	3	1	4	2
(b)	2	1	4	3
(c)	4	1	2	3
(d)	3	4	1	2

Q.20 The header added by the transport layer to the packet coming from the upper layer includes the

(a) Logical address
(b) Services-point address
(c) Physical address
(d) Network address

Q.21 If there are n devices (nodes) in a network, what is the number of cable links required for a mesh and a star topology respectively.

(a) n, n – 1 (b) n(n–1)/2, n – 1
(c) n – 1, n (d) n – 1, n(n–1)/2

Q.22 A system has an n-layer protocol hierarchy. Applications generate messages of length M bytes. At each of the layers, an h-byte header is added. What fractions of the network bandwidth is filled with headers?

(a) h/M (b) hn/(M + nh)
(c) nh/M (d) 1 – (nh/M)

Q.23 _____ addresses on headers change as a packet moves from network to network but the _____ do not.

(a) Logical, port (b) Logical, network
(c) Logical, physical (d) Physical, logical

Q.24 _____ used in telephone network for bi-directional, real time transfer between computers

(a) Message switching (b) Circuit switching
(c) Packet switching (d) Circular switching

Q.25 If the value of signal changes over a very short span of time, it's frequency is

(a) Short (b) Low
(c) High (d) Long

Q.26 _____ is used to optimize the use of the channel capacity available in a network, to minimize the transmission latency and to increase robustness of communication
(a) Message switching
(b) Linear switching
(c) Circuit switching
(d) Packet switching

Q.27 _____ splits traffic data into chunks
(a) Message switching (b) Linear switching
(c) Circuit switching (d) Packet switching

Q.28 The key concern in design of data transmission system is Data Rate and _____
(a) Data path (b) Data flow
(c) Distance (d) Frequencies

Q.29 The router connecting a company's network to the internet applies the mask 255.255.255.192 to the destination address of incoming IP packets. If one of the incoming packet has a destination address of 154.33.7.220, then find the network ID, subnet bits and host ID bits of incoming packets respectively
(a) 154.33.7, 11, 011100
(b) 154.33, 11000000, 011100
(c) 154.33, 0000011111, 011100
(d) 154.33.7, 011111, 011100

Q.30 If one of the address of a block is 210.69.92.39/26, then find the last host of the 2nd last subnet, where the addresses are referred in lexicographic order.
(a) 210.69.92.127/26
(b) 210.69.92.192/26
(c) 210.69.92.191/26
(d) 210.69.92.254/26

Answer Key:

1.	(a)	**2.**	(c)	**3.**	(a)	**4.**	(b)	**5.**	(a)
6.	(d)	**7.**	(d)	**8.**	(b)	**9.**	(a)	**10.**	(d)
11.	(c)	**12.**	(b)	**13.**	(c)	**14.**	(c)	**15.**	(a)
16.	(d)	**17.**	(d)	**18.**	(c)	**19.**	(a)	**20.**	(b)
21.	(b)	**22.**	(b)	**23.**	(d)	**24.**	(b)	**25.**	(c)
26.	(d)	**27.**	(d)	**28.**	(c)	**29.**	(c)	**30.**	(c)

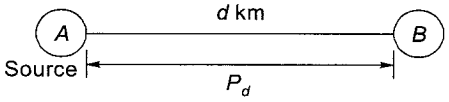
CHAPTER 02

Data Link Layer

2.1 Delays in Computer Networks

(a) Propagation delay (P_d)

It is the time taken by 1 bit to traverse the link



$\boxed{P_d = \frac{d}{V}}$; V is the velocity of signal on the link and usually it's value is given as 2.1×10^8 ms^{-1}

(b) Transmission delay (T_d)

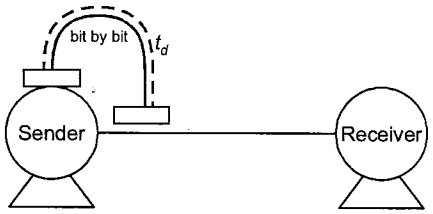
The time taken to push the entire frame

or packet bits into the wire $t_d = \frac{L}{B}$;

Where L - Frame size in bits and B - Bandwidth.

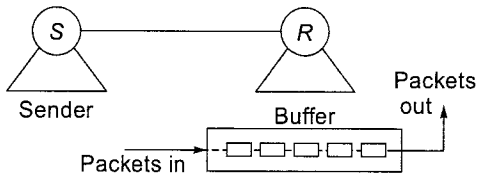
If L increase t_d increases

If B increase t_d decreases.



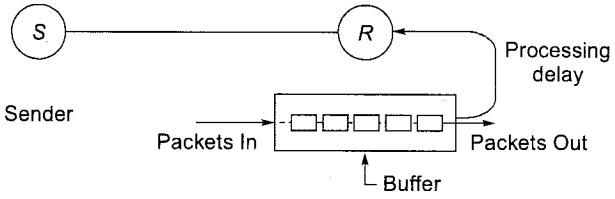
(c) Queuing delay

Time for which packet stays in the buffer is called queuing delay. It is taken by receiver for processing.



(d) Processing delay

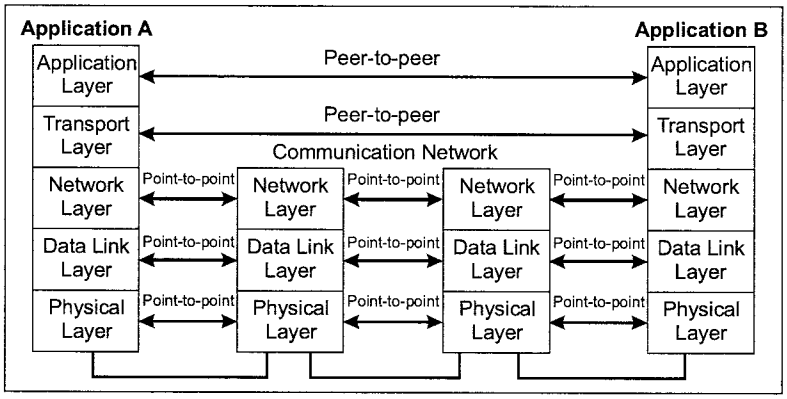
The time router takes to process the packet header is called 'processing delay'. During processing of a packet routers may check a bit level error in packet that occur during transmission.



2.2 Protocol Layering

There are several advantages of protocol layering.

1. Protocol layering enables us to divide a complex task into several smaller simpler tasks. This is referred to as *modularity*.
2. It allows to separate the services from the implementation.
3. Itermediate systems need only some layers but not all layers. Protocol layering helps us in designing the system or devices with required number of layers implemented in it.



OSI layer vs TCP/IP layer

OSI (Open System Interconnection)	TCP (Transmission Control Protocol/ Internet Protocol)
• OSI provide layer function and also defines functions of all layers	• TCP/IP model the transport layer does not guarantees delivery of packets
• IN OSI model the transport layer guarantees the delivery of packets	• Follows vertical approach
• Follows horizontal approach	• TCP/IP does not have a separate presentation layer
• OSI model has a separate presentation layer	• TCP/IP model cannot be used in any other application
• OSI is a general model	• The network layer in TCP/IP model provides connectionless services
• Network layer of OSI model provide both connection oriented and connectionless service	• TCP/IP model does not fit any protocol
• Protocols are hidden in OSI model and are easily replaced as the technology changes	• In TCP/IP replacing protocol is not easy
• OSI model defines services, interfaces and protocols very clearly and makes clear distinction between them	• In TCP/IP it is not clearly separated its services, interfaces and protocols.
• It has 7 layer	• It has 5 layers

The Open Systems Interconnection (OSI) model is a standard "reference model" created by the International Organization for Standardization (ISO) to describe how the different software and hardware components involved in a network communication should divide labor and interact with one another.

It defines a seven-layer set of functional elements, ranging from the physical interconnections at Layer 1 (also known as the physical layer, or PHY interface) all the way up to Layer 7, the application layer.

The Transmission Control Protocol (TCP) and the Internet Protocol (IP) are two of the network standards that define the Internet.

IP defines how computers can get data to each other over a routed, interconnected set of networks. TCP defines how applications can create reliable channels of communication across such a network. Basically, IP defines addressing and routing, while TCP defines how to have a conversation across the link without garbling or

losing data. TCP/IP grew out of research by the U.S. Dept. of Defense and is based on a loose rather than a strict approach to layering. Many other key Internet protocols, such as the Hypertext Transfer Protocol (HTTP), the basic protocol of the Web, and the Simple Mail Transfer Protocol (SMTP), the core email transfer protocol, are built on top of TCP. The User Datagram Protocol (UDP), a companion to TCP, sacrifices the guarantees of reliability that TCP makes in return for faster communications.

TCP/IP doesn't map cleanly to the OSI model, since it was developed before the OSI model and was designed to solve a specific set of problems, not to be a general description for all network communications.

ISO/OSI Layer	TCP/IP Model	Sample Protocols	Devices
7. Application	Application	SOAP, XML	XML Appliances
6. Presentation		HTTP, HTTPS FTP TELNET SMTP, NTP	Content Service Switch Layer 4-7 Switches
5. Session			
4. Transport			
3. Network	Transport	TCP, UDP	Router, Layer-3 Switch
2. Data Link	Network	IP, ICMP, IGMP, IPX	Switches, Bridges
1. Physical	Link	Network Interface: Ethernet, Token Ring, FDDI	Hubs, Repeaters

Two layers in the OSI model, session and presentation, are missing from the TCP/IP protocol suite. These two layers were not added to the TCP/IP protocol suite after the publication of the OSI model. The application layer in the suite is usually considered to be the combination of three layers in the OSI model. The OSI model did not replace the TCP/IP protocol suite because it was completed when TCP/IP was fully in place and because some layers in the OSI model were never fully defined.

Similarities between OSI Model and TCP/IP Model

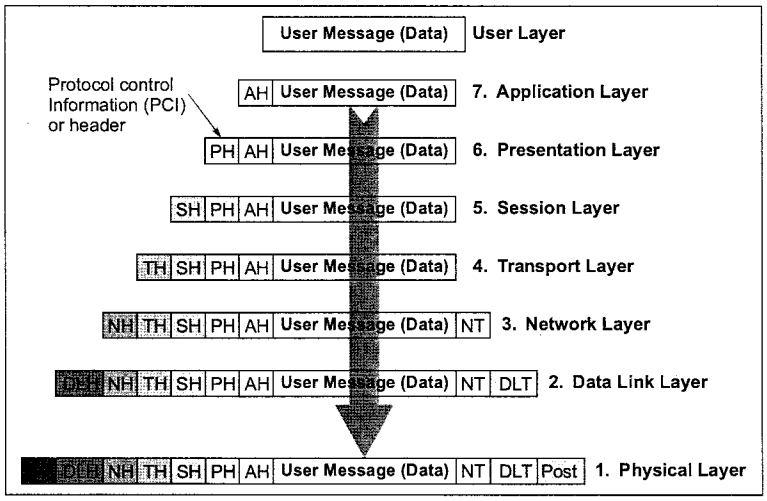
1. Both of them use a layered architecture to explain data communication process in computer networks.
2. Each layer performs well-defined functions in both models.
3. Similar types of protocols are used in both models.
4. OSI and TCP/IP reference models are open in nature.
5. Both models give a good explanation on how various types of network hardware and software interact during a data communication process.
6. Data hiding principle is well maintained on each layer in the two models. The core level functional details of each layer are not revealed to other layers.
7. Transport layer defines end-end data communication process and error-correction techniques in both the models.
8. OSI and TCP/IP reference models process data in the form of packets to perform routing.

Encapsulation and Decapsulation

Messages generated by application are encapsulated by every layer before passing on to the layer below it. Information that is encapsulated where is from layer-to-layer. This encapsulated information is also called as 'header'.

The TCP header information includes the port address, Ack information, checksum and other useful information. The network layer header includes source and destination IP address, header checksum, fragmentation information and other useful information.

Data link layer includes the MAC address information, error control, error detection, error correction and other useful information.



OSI Layer and Message passing

Data Flow

Communication between two devices can be simplex, half-duplex, or full-duplex.

(a) **Simplex:** In *simplex mode*, the communication is unidirectional, as on a one-way street. Only one of the two devices on a link can transmit; the other can only receive.
Example: Keyboards and Traditional Monitors.

(b) **Half-Duplex:** In *half-duplex*, each station can both transmit and receive, but not at the same time. When one device is sending, the other can only receive, and vice versa.
Example: Walkie-talkies and Citizen band (CB) Radio.

(c) **Full-Duplex:** In *full-duplex*, both stations can transmit and receive simultaneously.
Example: Telephone networks.

The simplex mode can use the entire capacity of the channel to send data in one direction. The half-duplex mode is used in cases where there is no need for communication in both direction at the same time; the entire capacity of the channel can be utilized for each direction.

The full-duplex mode is used when communication in both directions is required all the time. The capacity of the channel, however, must be divided between the two directions.

Network Topologies

(a) **Mesh Topology:** In mesh topology, every device has a dedicated point-to-point link to every other device. The term *dedicated* means that the link carries traffic only between the two devices it connects. Node 1 must be connected to n–1 nodes, node 2 must be connected to n–1 nodes, and finally node n must be connected to n – 1 nodes. Hence in total we need n(n–1) physical links. Every node must have n –1 input/output (I/O) ports.

(b) **Star Topology:** In a start topology, each device has a dedicated point-to-point link only to a central controller, usually called a **hub**. The devices are not directly linked to one another. Unlike a mesh topology, a star topology does not allow direct traffic between devices. In a star, each device needs only one link and one I/O port to connect it to any number of others.

(c) **Bus Topology:** A bus topology is multipoint. One long cable acts as **backbone** to link all the devices in a network. A drop line is a connection running between the device and the main cable.

(d) **Ring Topology:** In a ring topology, each device has a dedicated point-to-point connection with only the two devices on either side of it. A signal is passed along the ring in one direction, from device to device, until it reaches its destination. Each device in the ring incorporates a repeater. Each device is linked to only its immediate neighbours.

2.3 Circuit-Switched

A **circuit-switched network** consists of a set of switches connected by physical links. A circuit-switched network is made of a set of switches connected by physical links, in which each link is divided into n channels.

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

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

There is no addressing involved during data transfer. The switches route the data based on their occupied band (FDM) or time slot (TDM). Of course, there is end-to-end addressing used during the setup phase, as we will see shortly.

In circuit switching, the resources need to be reserved during the setup phase; the resources remain dedicated for the entire duration of data transfer until the teardown phase.

Three phase:

(a) **Setup phase:** Before the two parties (or multiple parties in a conference call) can communicate, a dedicated circuit (combination of channel in links) needs to be established.

(b) **Data-transfer phase:** After the establishment of the dedicated circuit (channels), the two parties can transfer data.

(c) **Teardown phase:** When one of the parties needs to disconnect, a signal is sent to each switch to release the resources.

2.4 Packet Switching

1. Refers to protocols in which messages are divided into packets before they are sent. Each packet is then transmitted individually and can even follow different routes to its destination.
2. Once all the packets forming a message arrive at the destination, they are recompiled into the original message.
3. Most modern Wide Area Network (WAN) protocols, including TCP/IP, X.25, and Frame Relay, are based on packet-switching technologies.
4. In contrast, normal telephone service is based on a circuit-switching technology, in which a dedicated line is allocated for transmission between two parties.
5. Circuit-switching is ideal when data must be transmitted quickly and must arrive in the same order in which it's sent.
6. This is the case with most real-time data, such as live audio and video. Packet switching is more efficient and robust for data that can withstand some delays in transmission, such as e-mail messages and Web pages.
7. A new technology, ATM, attempts to combine the best of both worlds — the guaranteed delivery of circuit-switched networks and the robustness and efficiency of packet-switching networks.

Summary

- **Propagation delay:** It is the time taken by 1 bit to traverse the link $P_d = d/V$.
- **Transmission delay:** The time taken to push the entire frame into the wire $t_d = L/B$.
- **Queuing delay:** Time for which packet stays in the buffer is called queuing delay.
- **Processing delay:** The time router takes to process the packet header is called 'processing delay'.
- Protocol layering enables us to divide a complex task into several smaller simpler tasks.
- OSI layer has 7 layer i.e., Application, Presentation, Session, Transport, Network, Data Link and Physical layer.
- TCP/IP has 5 layer i.e., Application, Transport, Network, Data Link and Physical layer.
- Messages generated by application are encapsulated by every layer before passing on to the layer below it.
- The TCP header information includes the port address, Ack information, checksum and other useful information.
- The network layer header includes source and destination IP address, header checksum, fragmentation information and other useful information.
- Data link layer includes the MAC address information, error control, error detection, error correction and other useful information.
- In *simplex mode*, the communication is unidirectional, as on a one-way street. Only one of the two devices on a link can transmit; the other can only receive.
- In *half-duplex*, each station can both transmit and receive, but not at the same time. When one device is sending, the other can only receive, and vice versa.
- In *full-duplex*, both stations can transmit and receive simultaneously.
- In mesh topology, every device has a dedicated point-to-point link to every other device.
- In a start topology, each device has a dedicated point-to-point link only to a central controller, usually called a **hub**.
- A bus topology is multipoint. One long cable acts as **backbone** to link all the devices in a network.
- In a ring topology, each device has a dedicated point-to-point connection with only the two devices on either side of it.
- A circuit-switched network consists of a set of switches connected by physical links.
- A circuit-switched network has three phase: (1) Setup phase (2) Data-transfer phase and (3) Teardown phase.
- In Packet Switching messages are divided into packets before they are sent. Each packet is then transmitted individually and can even follow different routes to its destination.

Student's Assignment

Q.1 Which one of the following statements is false?

(a) Packet switching leads to better utilization of bandwidth resources than circuit switching

(b) Packet switching results in less variation in delay than circuit switching

(c) Packet switching requires more per -packet processing than circuit switching

(d) Packet switching can lead to reordering unlike in circuit switching

Q.2 With respect to Circuit Switching and Packet Switching, which of the following statement is incorrect?

(a) In circuit switching after data transfer begins, no busy conditions take place

(b) In packet switching, each packet of the same message must follow the same route

(c) In packet switching, each packet must contain the addressing information

(d) In circuit switching, a circuit must be established on the network prior to the data transfer

Q.3 Communication via circuit switching involves three phase which are

(a) Circuit establishment, Data transfer, Circuit disconnect

(b) Circuit establishment, Data compression, Circuit disconnect

(c) Data transfer, Data compression, Circuit disconnect

(d) Circuit established, Data compression, Circuit disconnect

Q.4 Which of the following is/are correct

1. Circuit switching is designed for voice communication.
2. In Circuit switching network connection provides for transmission at varying data rate
3. Circuit switching sees all transmission are equal.
4. Circuit switching is less suited to data and non-conversation transmission.

(a) Only 3 (b) 2 and 3

(c) Only 4 (d) 1, 3 and 4

Answer Key:

1. (b) **2.** (b) **3.** (a) **4.** (d)

CHAPTER 03

MAC Sublayer

3.1 Introduction

Communication at the data link layer is node-to-node. The data link layer provides services to the network layer; it receives services from the physical layer. The scope of the data link layer is node-to-node. When a packet is travelling in the Internet, the data link layer of a node (host or router) is responsible for delivering a datagram to the next node in the path.

We encapsulate and decapsulate at each node because IP address information is present in packet, which is encapsulated by frame.

TCP/IP model did not define any protocol for data link layer and physical layer. Therefore we can define our on protocol for data link and physical layer.

Link layer Addressing

The source and destination Ip addresses define the two ends but cannot define which links the datagram should pass through. A link-layer address is sometimes called a *link address*, sometimes a *physical address*, and sometimes a *MAC address*.

When a datagram passes from the network layer to the data-link layer, the datagram will be encapsulated in a frame and two data-link addresses are added to the frame header. These two addresses are changed every time the frame moves from one link to another.

NOTE: MAC addresses are represented using 48 bits (12 hexadecimal digits separated by a colons).

Types of Addresses

1. **Unicast address:** Each host or each interface of a router is assigned a unicast address. A frame with a unicast address destination is destined only for one entity in the link.

   A3: 34: 45: 11: 92: F1

2. **Multicast address:** Multicasting means one-to-many communication.

   A2: 34: 45: 11: 92: F1

3. **Broadcast address:** A frame with a destinations broadcast address is sent to all entities in the link.

   FF: FF: FF: FF: FF: FF

3.2 Data Link Layer Functions

Concerned with reliable, error-free and efficient communication between adjacent machines in the network through the following functions:

Data Framing

The term "frame" refers to a small block of data used in a specific network. The data link layer groups raw data bits to/from the physical layer into discrete frames with error detection/correction code bits added.

Framing methods:

- Character count.
- Starting and ending characters, with character stuffing.
- Starting and ending flags with bit-stuffing.
- Physical layer coding violations.

Error Detection/Correction

Error detection:

- Include enough redundant information in each frame to allow the receiver to deduce that an error has occurred, and to request a retransmission.
- Uses error-detecting codes.

Error correction:

- Include redundant information in the transmitted frame to enable the receiver not only to deduce that an error has occurred but also correct the error.
- Uses error-correcting codes.

Services to the Network Layer

Unacknowledged connectionless service - best effort:

The receiver does not return acknowledgments to the sender, so the sender has no way of knowing if a frame has been successfully delivered.

When would such a service be appropriate?

(a) When high layers can recover from errors with little loss in performance. That is, when errors are so infrequent that there is little to be gained by the data link layer performing the recovery. It is just as easy to have higher layers deal with occasional lost packets.

(b) Independent frames sent without having the destination acknowledge them.

(c) For real-time applications requiring "better never than late" semantics. Old data may be worse than no data. For example, should an airplane bother calculating the proper wing flap angle using old altitude and wind speed data when newer data is already available?. Also used in real time applications such as speech video etc.

Acknowledged connection-less service-acknowledged delivery:

- The receiver returns an acknowledgment frame to the sender indicating that a data frame was properly received.
- Likewise, the receiver may hand received frames to higher layers in the order in which they arrive, regardless of the original sending order.
- Typically, each frame is assigned a unique sequence number, which the receiver returns in an acknowledgment frame to indicate which frame the ACK refers to. The sender must retransmit unacknowledged (e.g., lost or damaged) frames.

Acknowledged connection-oriented service-reliable delivery: Frames are delivered to the receiver reliably and in the same order as generated by the sender. Connection state keeps track of sending order and which frames required retransmission. For example, receiver state includes which frames have been received, which ones have not etc. The data link guarantees that each frame sent is received exactly once and in right order.

Flow Control

There are several protocols to control the rate at which sender transmits frames and at a rate acceptable to the receiver, and the ability to retransmit lost or damaged frames. This ensures that slow receivers are not swamped by fast senders and further aids error detection/correction.

Several flow control protocol exist, but all essentially require a form of feedback to make the sender aware of whether the receiver can keep up.

Stop-and-wait protocols:

- A positive acknowledgment frame is send by the receiver to indicate that the frame has been received and to indicate being ready for the next frame.
- Positive Acknowledgment with Retransmission (PAR); uses timeouts

Sliding window protocols:

- Data frames and acknowledgment frames are mixed in both directions.
- Frames sent contain sequence numbers
- Timeouts used to initiate retransmission of lost frames.

3.3 Data Link Layer Framing

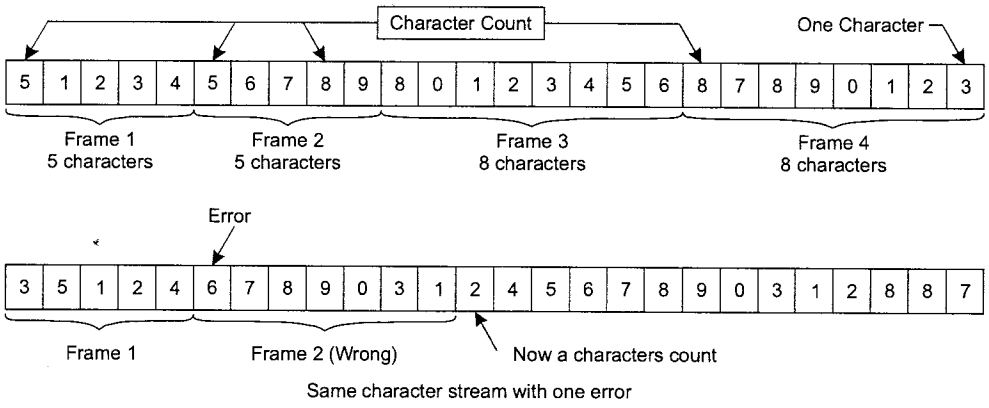
The DLL translates the physical layer's raw bit stream into discrete units (messages) called frames. How can frame be transmitted so the receiver can detect frame boundaries? That is, how can the receiver recognize the start and end of a frame? We will discuss four ways:

3.3.1 Character Count Method

Make the first field in the frame's header be the length of the frame. That way the receiver knows how big the current frame is and can determine where the frame ends.

Disadvantage

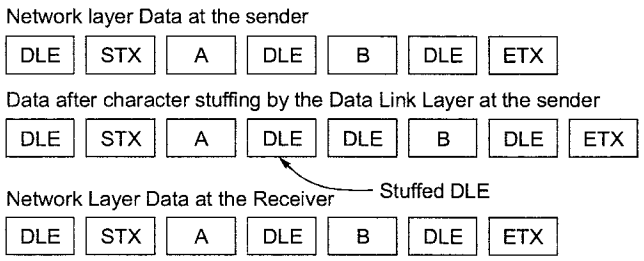
Receiver looses synchronization when bits become garbled. If the bits in the count become corrupted during transmission, the receiver will think that the frame contains fewer (or more) bits than it actually does. Although checksum will detect the frames are incorrect, the receiver will have difficult re-synchronizing to the start of a new frame. This technique is not used anymore, since better techniques are available.



Same character stream with one error

3.3.2 Character Stuffing

Each frame starts with the ASCII character sequence DLE (Data Link Escape) and STX (Start of Text) and ends with DLE ETX (End of Text). When binary data is transmitted where (DLE STX or DLE ETX) can occur in data, character stuffing is used (additional DLE is inserted in the data). Limited to 8-bit characters and ASCII.



3.3.3 Bit-Oriented Using Start/End Flags (Bit stuffing)

Each frame begins and ends with 0 1 1 1 1 1 1 0

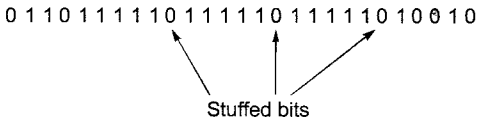
After each five consecutive ones in a data zero is stuffed. Stuffed zero bits are removed by the data link layer at receiving end.

NOTE: When using bit stuffing, locating the start/end of a frame is easy, even when frames are damaged. The receiver simply scans arriving data for the reserved patterns.

The receiver will re-synchronize quickly with the sender as to where frames begin and end, even when bits in the frame get garbled. The main disadvantage with bit stuffing is the insertion of additional bits into the data stream, wasting bandwidth. How much expansion? The precise amount depends on the frequency in which the reserved patterns appear as user data.

The Original Data: 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

Data appearing on the line after bit stuffing



Data received after destuffing: 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

Example **- 3.1** The following character encoding is used in a data link protocol: A: 01000111 B: 11100011 FLAG: 01111110 ESC: 11100000 Show the bit sequence transmitted (in binary) for the four-character frame A B ESC FLAG when each of the following framing methods is used:

(a) Byte count.

(b) Flag bytes with byte stuffing.

(c) Starting and ending flag bytes with bit stuffing.

Solution:

(a) Message is of 4 bytes length. So specify the byte count (00000100) just before the message.
00000100 ABESC FLAG

(b) FLAG AB ESC ESC ESC FLAG FLAG

(c) **01111110** 01000111 110100011 111000000011111010 **01111110**

Example-3.2 The following data fragment occurs in the middle of a data stream for which the bytestuffing algorithm described in the text is used : A B ESC C ESC FLAG FLAG D. What is the output after stuffing?

Solution:

FLAG A B ESC ESC C ESC ESC FLAG ESC FLAG D FLAG

Example-3.3 What is the maximum overhead in byte stuffing algorithm?

Solution:

Maximum overhead occurs when all the bytes are only ESC and FLAG bytes. In that case there will be 100% percent overhead.

Example-3.4 One of your classmates, Suresh Reddy, has pointed out that it is wasteful to end each frame with a flag byte and then begin the next one with a second flag byte. One flag byte could do the job as well, and a byte saved is a byte earned.

Solution:

It's difficult to distinguish between two frames separated by a time gap, whether it is garbage data or actual next frame (First frame ends with FLAG bytes).

Example-3.5 Sixteen-bit messages are transmitted using a Hamming code. How many check bits are needed to ensure that the receiver can detect and correct single-bit errors? Show the bit pattern transmitted for the message 1101001100110101. Assume that even parity is used in the Hamming code.

Solution:

Five check bits are needed as parity bits are placed at 1, 2, 4, 8, 16

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
P_1	P_2	1	P_3	1	0	1	P_4	0	0	1	1	0	0	1	P_5	1	0	1	0	1

To find P_1 take one bit and leave one bit starting from P_1:

P_1	3	5	7	9	11	13	15	17	19	21
0	1	1	1	0	1	0	1	1	1	1
If number of 1's in bits is even then $P_1 = 0$ else $P_1 = 1$										

To find P_2 take two bits and leave two bits starting from P_2 and then take two bits and so on.

P_2	3	6	7	10	11	14	15	18	19
1	1	0	1	0	1	0	1	0	1
Here number of 1's are odd so put $P_2 = 1$ so as to make number of 1's even (even parity means number of 1's should be even)									

To find P_3 take three bits and leave three bits alternatively starting from P_3.

P_3	5	6	10	11	12	16	17	18
P_3	1	0	0	1	1	P_5	1	0

We dont know so first find P_5.

P_5: P_5 1 0 1 0 1 ($P_5 = 0$) $\Rightarrow$ $P_3 = 0$ (even parity)

P_4	9	10	11	16	17	18	19
P_4	0	0	1	P_5	1	0	1

$P_4 = 1$

Now place parity bits in original string and we get the string to be transmitted.

0110 1011 0011 0010 10101

Example-3.6 A block of b its with n rows and k columns uses horizontal and vertical parity bits for error detection. Suppose that exactly 4 bits are inverted due to transmission errors. Derive an expression for the probability that the error will be undetected.

Solution:

Assume N = 4 and K = 4. We observe that 4 bit error goes undetected if those 4 bits form a rectangle or square.

∴ Probability of getting rectangle

$$P = \frac{{}^{k}C_2 \times {}^{n}C_2}{{}^{nk}C_2}$$

1	1	0	0	0
0	1	1	0	0
0	1	0	1	0
1	1	1	1	0
0	0	0	0	0

Example-3.7 Suppose that a message 1001 1100 1010 0011 is transmitted using Internet Checksum (4-b it word). What is the value of the checksum?

Solution:

$$\begin{array}{r} 1001 \\ 1100 \\ 1010 \\ 0011 \\ \hline 1100 \end{array}$$

Example-3.8 What is the remainder obtained by dividing $x^7 + x^5 + 1$ by the generator polynomial $x^3 + 1$?

Solution:

$$\begin{array}{r} x^3+1 \overline{)\, x^7+x^5+1 \,(} x^4+x^2+x \\ x^7+x^4 \\ \hline x^5+x^4+1 \\ x^5+x^2+1 \\ \hline x^4+x^2+1 \\ x^4+x \\ \hline x^2+x+1 \end{array}$$

∴ $x^2 + x + 1$ is remainder.

Example-3.9 A bit stream 10011101 is transmitted using the standard CRC method described in the text. The generator polynomial is $x^3 + 1$. Show the actual bit string transmitted. Suppose that the third bit from the left is inverted during transmission. Show that this error is detected at the receiver's end. Give an example of bit errors in the bit string transmitted that will not be detected by the receiver.

Solution:

$$\begin{array}{r} 1001 \overline{)\, 10011101000 \,(} 10011 \\ 1001 \\ \hline 0001101 \\ 10011 \\ \hline 01000 \\ 1001 \\ \hline 000\mathbf{100} \end{array}$$

3 bits less than divisor so stop.

10011101100 message to be transmitted. If receiver divides this with 1001 should give 0 (correct) If ≠ 0 (error)

3.4 Error Control

Must insure that all frames are eventually delivered (possibly in order) to a destination. Three components are required to do this: Acknowledgements, Timers, and Sequence Numbers.

3.4.1 Acknowledgements

- Reliable delivery is achieved using the "acknowledgements with retransmission" paradigm.
- The receiver returns a special acknowledgement (ACK) frame to the sender indicating the correct receipt of a frame.
- In some systems, the receiver also returns a negative acknowledgements (NACK) for incorrectly-received frames.
- This is only a hint to the sender so that it can retransmit a frame right away without waiting for a timer to expire.

3.4.2 Timers

- What happens if an ACK or NACK becomes lost?
- Retransmission timers are used to resend frames that don't produce an ACK. When sending a frame, schedule a timer at some time after the ACK should have been returned. If the timer goes off, retransmit the frame.

3.4.3 Sequence Numbers

- Retransmissions introduce the possibility of duplicate frames.
- To suppress duplicates, add sequence numbers to each frame, so that a receiver can distinguish between new frames and repeats of old frames.
- Bits used for sequence numbers depend on the number of frames that can be outstanding at any one time.

Flow Control

- Flow control deals with throttling the speed of the sender to match that of the receiver. Usually, this is a dynamic process, as the receiving speed depends on such changing factors as the load, and availability of buffer space.
- One solution is to have to receiver extend credits to the sender. For each credit, the sender may send one frame. Thus, the receiver controls the transmission rate by handing out credits.

Link Initialization

In some cases, the data link layer service must be "opened" before use:

(a) The data link layer uses open operations for allocating buffer space, control blocks, agreeing on the maximum message size, etc.

(b) Synchronize and initialize send and receive sequence numbers with its peer at the other end of the communications channel.

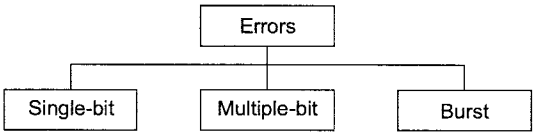
3.5 Data Link Layer: Error Detection/Correction

In data communication, line noise is a fact of life (e.g., signal attenuation, natural phenomenon such as lightning, and the telephone worker). Moreover, noise usually occurs as bursts rather than independent, single bit errors.

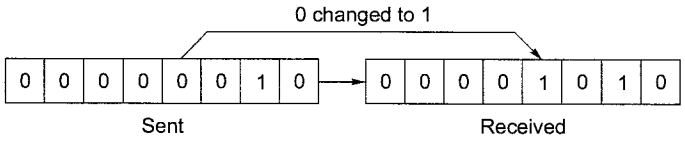
For *example*, a burst of lightning will affect a set of bits for short time after the lightning strike. Detecting and correcting error requires redundancy. Sending additional information along with the data.

There are two types of attacks against errors:

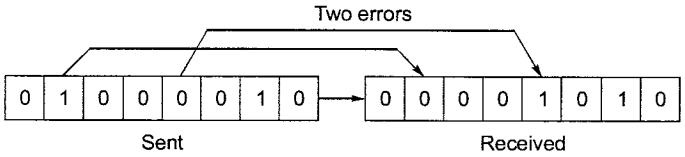
- **Error Detecting Codes:** Include enough redundancy bits to detect errors and use ACKs and retransmissions to recover from the errors.
- **Error correcting Codes:** Include enough redundancy to detect and correct errors.



Single Bit Error

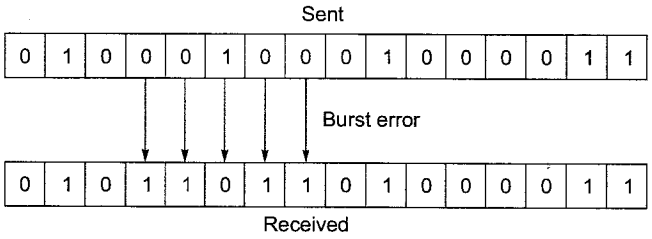


Multiple Bit Error



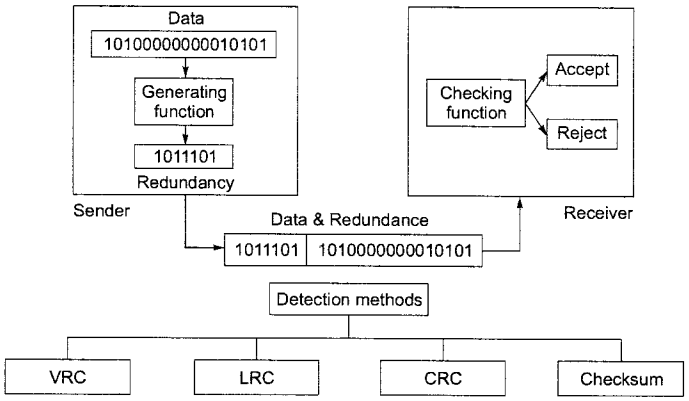
Burst Error

In a burst error two or more bits are changed. The length of the burst error is measured from the first bit corrupted to the last corrupted bit. Some bits in between may not have been corrupted.



Redundancy

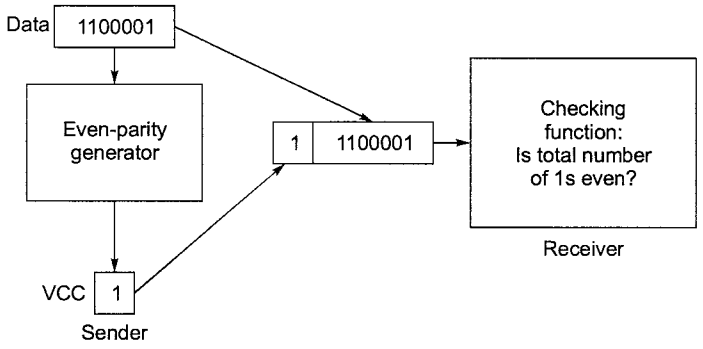
The concept of including extra information in the transmission for error detection is called redundancy. It is called so because the extra bits are redundant to the information, they are discarded as soon as the accuracy of the information has been determined.



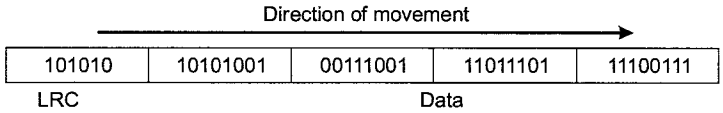
3.5.1 Simplest Error Detection: Party Bits and Checksum (Sum of 1's in Data)

A single parity bit is appended to each data block (e.g. each character in ASCII systems) so that the number of 1 bits always adds up to an even (odd) number.

Vertical Redundancy Check



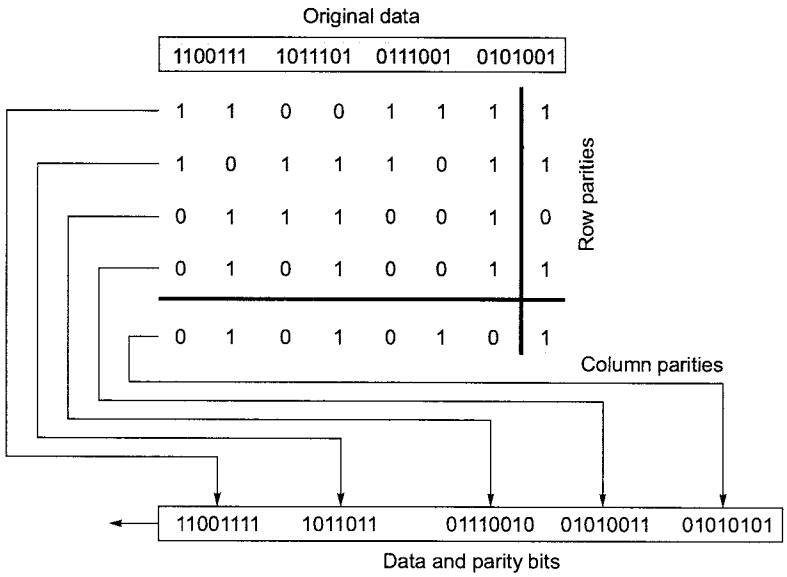
Longitudinal Parity Check



Simple parity check can detect all single bit errors. It can detect burst errors only if the total number of errors in each data unit is odd.

Two dimensional parity check: In this method, a block of bits is organized in a table (rows and columns). First we calculate the parity bit for each data unit. Then we organize them into a table. Thus in this a block of bits is divided into rows and a redundant row of bits is added to the whole block.

Performance: This method increases the likelihood of detecting burst errors. Thus a redundancy of n bits can easily detect a burst error of n bits. There is however one pattern of errors that remain elusive. If 2 bits in one data unit are damaged, the checker will also not detect an error.



3.5.2 Cyclic Redundancy Check (CRC)

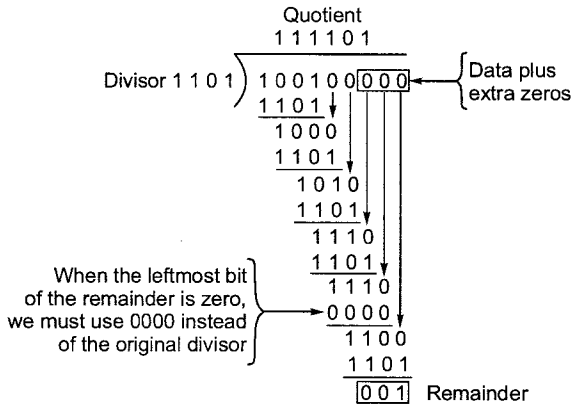
At sender end:

(*i*) Sender has a generator G(x) polynomial.

(*ii*) Sender appends (n – 1) zero bits to the data.
where n = no. of bits in generator.
Example: $G(x) = x^3 + x^2 + 1$
equivalent string for this generator = 1101
Let data stream is 100100
So 100100000 will be the dividend at the sender's end.

(*iii*) Divided appended data with generator G(x) using modulo 2 division (arithmetic).

(*iv*) Remainder of (n – 1) bits will be CRC.



This n – 1 bit remainder in CRC and append it with data and send it.
So sent data = 100100001.

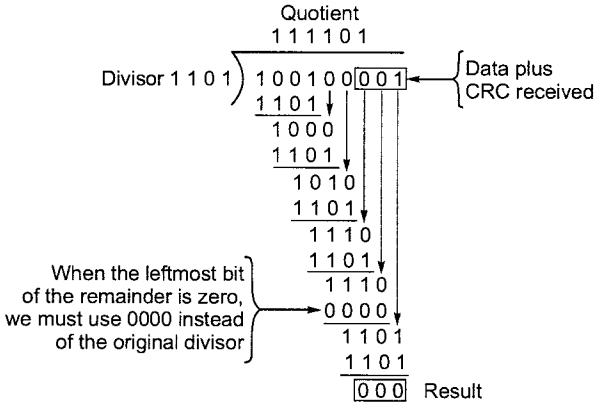
At the receiver end:

(*i*) Receiver has same generator G(x).

(*ii*) Receiver divides received data (data + CRC) with generator.

(*iii*) If remainder is zero, data is correctly received.

(*iv*) Else, Error.



Common CRC generator polynomials:

(*i*) CRC-32:

$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10}$

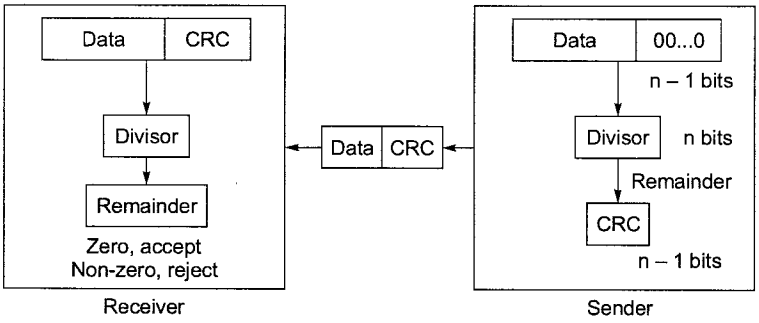
$x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ (Used in FDDI, Ethernet)

(*ii*) CRC-CCITT:

$x^{16} + x^{12} + x^5 + 1$ (Used in HDLC)

(*iii*) CRC-8:

$x^8 + x \mid 2 + x + 1$ (Used in ATM)



Polynomials: The divisor in the CRC generator is most often represented not as a string of 1's and 0's, but as algebraic polynomial. The relationship of a polynomial to its corresponding binary relation. A polynomial should be selected to have atleast the following properties:

1. It should not be divisible by x.
2. It should be divisible by $x + 1$.

The first condition guarantees that all burst errors of a length equal to the degree of the polynomial are detected. The second condition guarantees that all burst errors affecting an odd number of bits are detected.

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

1. CRC can detect all burst errors that affect an odd number of bits.
2. CRC can detect all burst error of length less than or equal to the degree of the polynomial.
3. CRC can detect, with a very high probability, burst errors of length greater than the degree of the polynomial.

3.5.3 Checksum

At sender end:

(*i*) Divide data into K chunks of n bit each.

(*ii*) Add all using 1's complement addition.

(*iii*) Take the complement of the result.

(*iv*) Result will be send with the data as a checksum.

1's complement addition: If result has a carry bit add carry with LSB of result

Example: Data = 01110011, suppose we are taking 4 bit chunks then

$$\begin{array}{r} 0\ 1\ 1\ 1 \\ 0\ 0\ 1\ 1 \\ \hline 1\ 0\ 1\ 0 \end{array}$$

Since result has no carry bit. Now take 1's complement then checksum = 0101.

The data that will be sent = 011100110101.

At receiver end:

(*i*) Receiver adds data end checksum using 1's complement addition.

(*ii*) Take the complement of the result.

(*iii*) If result contains all zero bit, accept data.

(*iv*) Else error: In previous example the receiver computed

$$\begin{array}{r} 0\ 1\ 1\ 1 \\ 0\ 0\ 1\ 1 \\ 0\ 1\ 0\ 1 \\ \hline 1\ 1\ 1\ 1 \end{array}$$

Now taking the complement of result we will get 0000. So in this case there is no error.

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 sum of those columns will not change and the receiver will not detect a problem.

3.6 Error-Detecting and Correcting Codes

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. With m bits of data and r bits of redundancy added to them, the length of the resulting code is m + r.

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:

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

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 7-bit ASCII code), the smallest r value that can satisfy this equation is 4: $2^r >= 7 + 4 + 1$.

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

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

Relationship between data and redundancy bits

3.6.1 Hamming Code

Hamming provides a practical solution. The Hamming code can be applied to data units of any length and uses the relationship between data and redundancy bits that can be added to the end of the data unit or interspersed with the original data bits. In Figure (a), 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 and r_8.

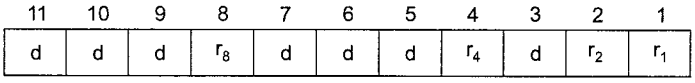


Figure (a)

In the Hamming code, each r bit is the parity bit for one combination of data bits, as shown below:

r_1 : bits 1, 3, 5, 7, 9, 11 r_2 : bits 2, 3, 6, 7, 10, 11

r_4 : bits 4, 5, 6, 7 r_8 : bits 8, 9, 10, 11

Each data bit may be included in more than one 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 (see Figure (b)).

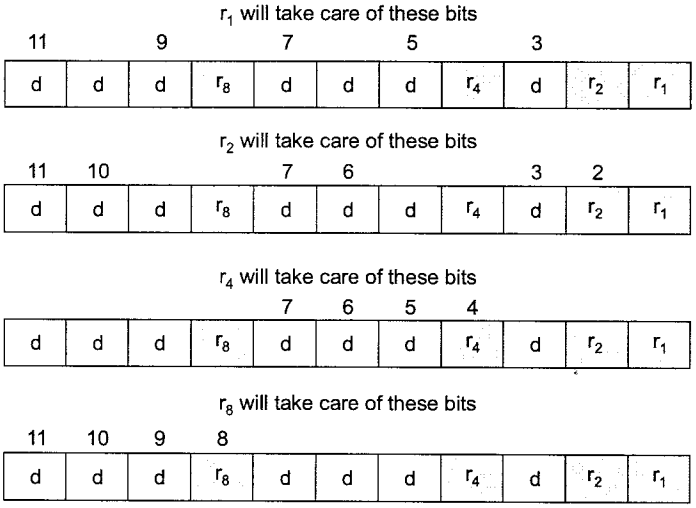


Figure (b)

Calculating the r Values: Figure 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.

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. The receiver takes the transmission and recalculates 4 new parity bits, using the same sets of bits used by the sender plus the relevant parity r bit for each set see fig (c). 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.

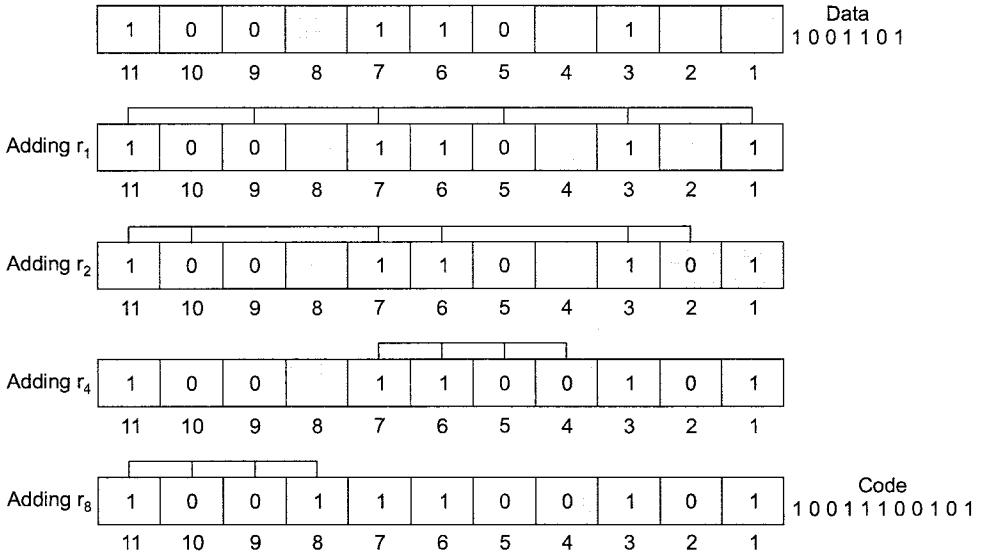


Figure (c)

Once the bit is identified, the receiver can reverse its value and correct the error. The beauty of the technique is that it can easily be implemented in hardware and the code is corrected before the receiver knows about it.

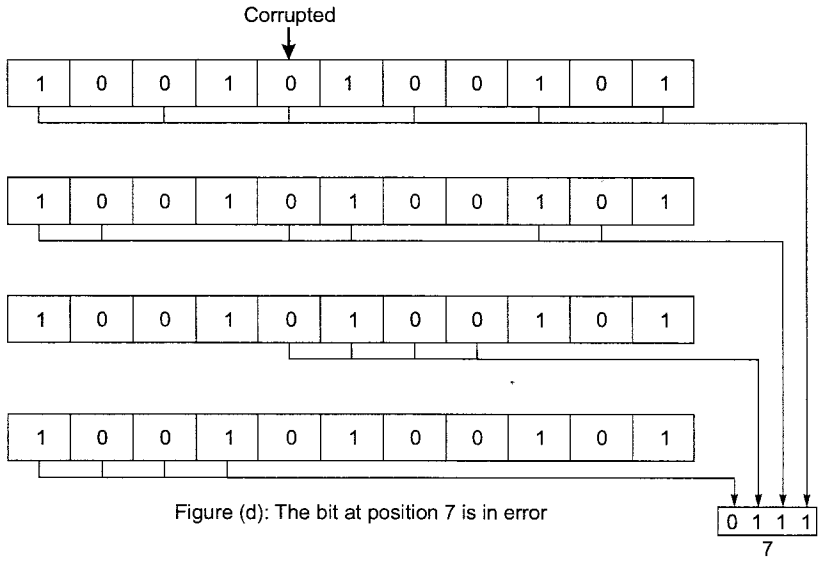


Figure (d): The bit at position 7 is in error

- The Hamming distance-minimum number of positions any two legal code words differ-of a code defines its errors detection/correction ability.
- To detect d errors code Hamming distance = d+1
- To correct d errors code Hamming distance 2d +1
- Some code are more suitable to correct burst errors rather than isolated errors.

3.7 Sliding Window Protocols (SWP)

The SWP protocols allow both link nodes (A, B) to send and receive data and acknowledgments. (Full Duplex). Acknowledgments are piggybacked into an acknowledgment field in the data frame and not as separate frames. This technique of temporarily delaying outgoing Ack at the receive end with an intention to send it along with outgoing data frame is known as piggybacking.

Advantage: Channel bandwidth can be used efficiently.

Each outbound frame contains a sequence number ranging from 0 to 2^{n-1} (n-bit field), n = 1 for stop-and-wait sliding window protocols.

At the receiver side if the new data frames are not ready for transmission in a specified time, a separate Ack frame is generated to avoid time-out. Along with flow control error control ARQ protocol is combined with sliding window protocol. Automatic repeat request includes (ARQ):

(a) Error detection.
(b) Positive Ack.
(c) Retransmission after time out
(d) Negative acknowledgments and retransmissions.

In effect SWP takes care of the following.

(a) Error correction (by retransmissions)
(b) Flow control
(c) Message ordering by sender.

SWP protocols are used by most connection oriented protocol like PPP (Point-to-point), HDLC and TCP etc. Buffer space, channel utilization and effective data rate are the parameters used to compare the sliding window protocols. The three famous flow control protocols along with ARQ are:

(a) STOP and WAIT ARQ
(b) STOP Back N ARQ
(c) Selective repeat ARQ

Sender's Window (W_s)

A set of sequence numbers maintained by the sender and correspond to frame sequence numbers of frames sent out but not acknowledged.

W_S corresponds to the maximum number of frames the sender can transmit before receiver and indicate the frame sequence numbers it is allowed to receive and acknowledge.

Receiving Window (W_R)

A set of sequence numbers maintained by the receiver and indicate the frames sequence numbers it is allowed to receive and acknowledge.

The size of the receiving window is fixed at a specified initial size. Any frame received with a sequence number outside the receiving window is discarded. The sending window and receiving window may not have the same upper or lower limits or have the same size.

Any frame received with a sequence number outside the receiving window is discarded. The W_S and W_R may not have the same upper or lower limits or have same size. GBN and SR are the protocols that implement pipelining. Time to transmit N frames ≥ Round trip time.

$$\boxed{\frac{L}{B} \times N \geq RTT} \quad \text{or} \quad \boxed{\frac{(\text{Data} + \text{Header})}{B} \times N \geq RTT}$$

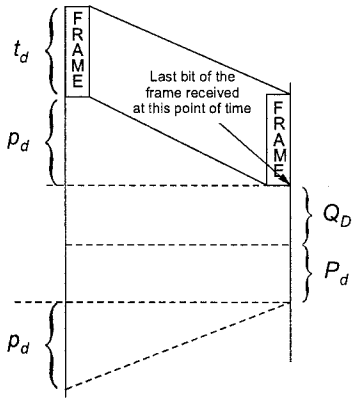
Under this condition.

$$\text{Maximum utilization} \simeq \frac{\text{Data size}}{\text{Data size} + \text{Header}}$$

$$\text{Maximum throughput} \simeq \left(\frac{\text{Data size}}{\text{Data size} + \text{Header}}\right) \times B$$

Where B is Bandwidth, L is Frame size, N is the Window size.

3.7.1 STOP and WAIT



- It uses Half duplex and is very inefficient.
- Using this protocol we can send. One packet per round trip time.

$$\text{Through put} = \frac{\text{Out packet}}{\text{RTT}}$$

- If the "data rates" and "propagation delays" are high, both sender's utilization and channel utilization falls down.
- When bandwidth and delay product is large, then stop and wait is useless.
- Receiver sends a positive Ack frame to transmit the next data frame. Any frame has a sequence number either 0 or 1

Total time taken to send one packet = $t_d + p_d + Q_D$ (queuing delay) + P_D (processing delay) + $t_{dAck}\, p_d$

But Q_D and P_D and $t_{d\,Ack}$ are negligible. Considering overall time.

$$t_d >> t_{d\,Ack}$$

p_d Much greater than Q_D and P_D

$\therefore$ Total time $= t_d + 2\,p_d$

$$\text{Efficiency}\,(\eta) = \frac{\text{Useful time}}{\text{Total time}} = \frac{t_d}{t_d + 2p_d} = \frac{1}{1+2a}, \quad \text{where } a = p_d / t_d$$

NOTE: Normally $t_{d\,(data)} \neq t_{d\,(Ack)}$, but $t_d\,(data) = t_{d\,(Ack)}$ when PIGGY BACKING is used.

$p_{d\,(data)} = p_{d\,(Ack)}$

Throughput (T_H)

Throughput is defined as the number of bits that can be send per second using the current protocol.

$$T_H = \frac{L}{t_d + 2 \times p_d}$$

Throughput is also defined as "Bandwidth utilization" or "Effective Bandwidth".

$$T_H = \frac{(L/B) \times B}{t_d + 2p_d} = \frac{t_d \times B}{t_d + 2_{pd}} = \eta B$$

$$T_H = \eta \times \text{Bandwidth} \qquad \left(\eta = \frac{t_d}{t_d + 2p_d}\right)$$

Remember

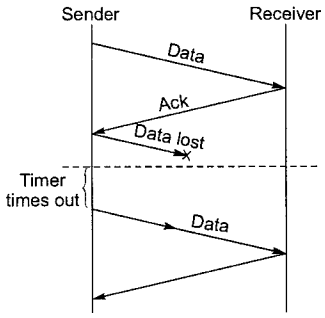
- As the distance increases efficiency (η) decreases. Hence it is use only in LAN and not WAN.
- As the length of the packet increases efficiency (η) also increases. Hence it is use for bursty transmissions
- Throughput (T_h) = $\eta \times B$, where T_h represents the Total Bits transmitted and η represents the percentage of useful time.

Need for ARQ along with STOP & WAIT

(a) **When Data packet is lost:** Both senders and receiver may get into deadlock, so we need a timeout timer. Timeout timer prevents deadlock between sender and receiver.

The timeout timer makes sure that packet is retransmitted after a specified amount of time when the Ack is not received.

As can be seen in the time line diagram between sender and receiver, the sender retransmit after timeout.

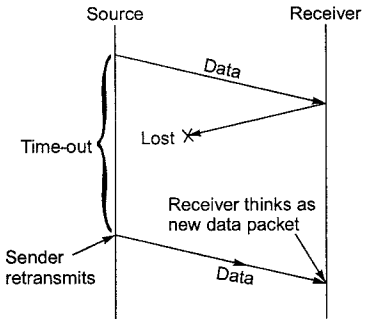


NOTE: STOP & WAIT Protocol + Timeout time (Request for retransmission) = STOP & WAIT ARQ.

(b) **Ack lost problem (duplicate packet problem):** In this case the acknowledgment is lost and the timer for that particular packet times out and sender thinks that the data packet is lost and retransmit after the timeout.

The above problem can be solved by adding sequence number for data.

STOP & WAIT + Timeout + Sequence numbers for data.



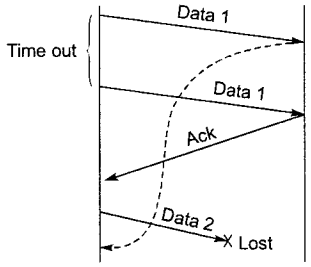
(c) **Ack delayed problem:**

When the acknowledgment for the first packet is delayed for long enough such that it arrives at sender after the transmission of second packet, the sender thinks that it is the Ack for second packet came.

To avoid this problem we use sequence numbering for Ack's.

Ack will contain the next sequence number expected.

So discard A_2 and wait for A_3.

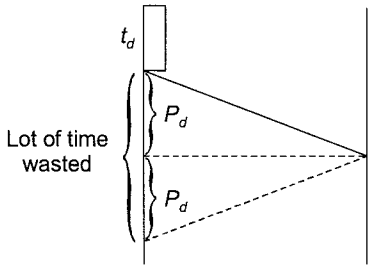


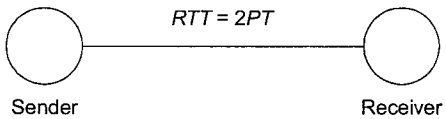
Improving η of STOP & WAIT using pipelining

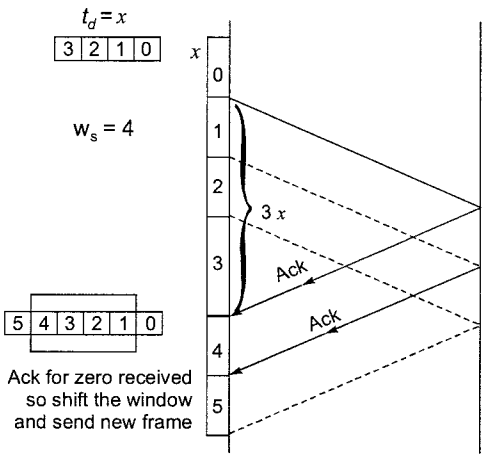
In $t_d \rightarrow 1$ packet

In 1 sec $\rightarrow 1/t_d$ packet

In $(t_d + 2 \times P_d)$ we can send $\frac{t_d + 2 \times P_d}{t_d}$ packets i.e. (1 + 2a) packets can be sent for 100% utilization.







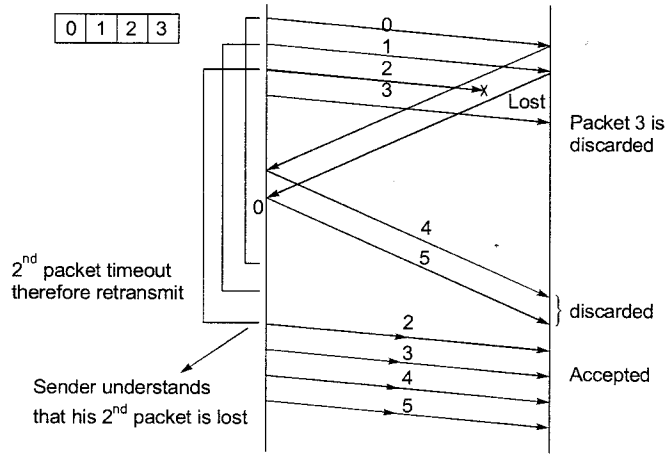
In sliding window protocol for maximum utilization 1 + 2a packets need to be transmitted $W_S = 1 + 2a$.

3.7.2 Go Back N

To improve the efficiency of transmission (to fill the pipe), multiple packet must be in transition while the sender is waiting for acknowledgment. The key to Go-back-N is that we can send several packets before receiving acknowledgment, but the receiver can only buffer one packet. We keep a copy of the sent packets until the acknowledgments arrive. The sequence numbers are modulo 2^m, where m is the size of the sequence number fields in bits.

- Sender window size is N for Go Back N.
- Receiver window size is always equal to 1
- This protocol does not accept the out of order packets.

Example: Sender window size = 4 (GB)



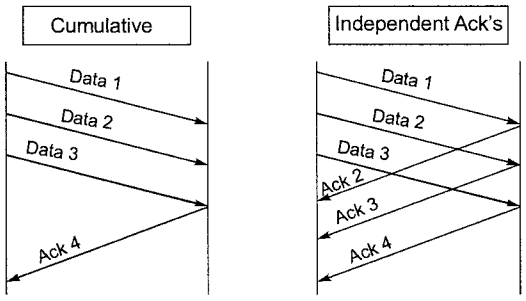
NOTE: Every packet or frame has individual timeout timers. In the Go-Back-N protocol, the acknowledgment number is cumulative and denies the sequence number of the next packet expected to arrive.

In GBN, when ever a particular packet is lost we transmit all the subsequent packets, which were transmitted after the lost one. (Resend the window size).

The sender window is an abstract concept defining an imaginary box of maximum size = $2^m - 1$. The receive window is an abstract concept defining an imaginary box of size 1. The window slides when a correct packet has arrived; sliding occurs one slot at a time.

Two Kind's of Ack's

- Cumulative
- Independent Ack's



Cumulative Ack Implementation

Ack timer is started at the receiver side and when the time expires cumulative Ack is sent for all the packets received before the timer expired

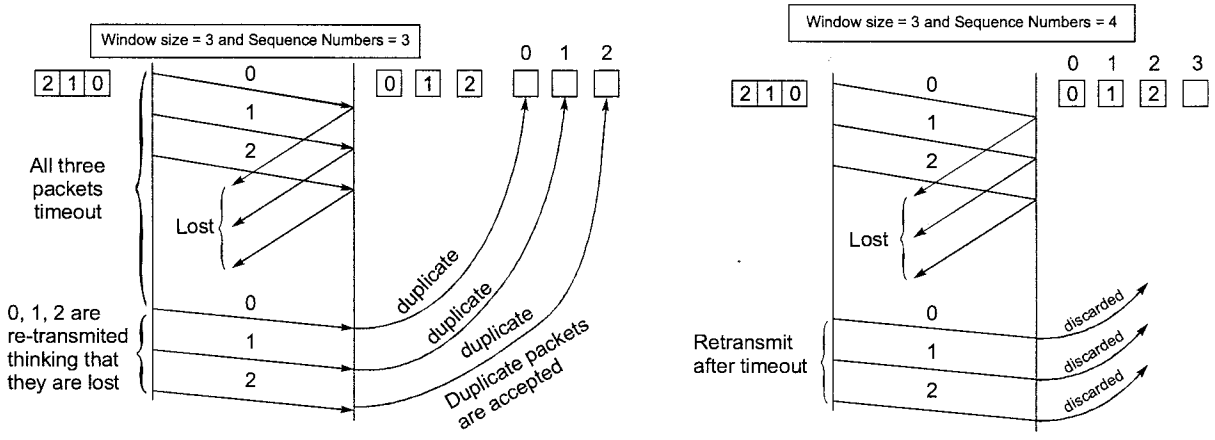
Ack N is the cumulative acknowledgment, then N is the next expected packet.

In GBN $W_S + W_R \leq$ Available sequence numbers.

The last number of sequence numbers required to delete duplicate packets in $(W_S + 1)$.

Receiver may accept duplicate packets.

Consider: $W_S = 3$ Sequence Number = 3



3.7.3 Selective Repeat (SR)

As the name implies, resends only selective packets, those that are actually lost. The *Selective-Repeat* protocol also uses two windows: A send window and a receive window. However, there are differences between the windows in this protocol and the ones in Go-Back-N. First, the maximum size of the send window is much smaller; it is 2^{m-1}. The receive window is the same size as the send window.

The *Selective-Repeat* protocol allows as many packets as the size of the receive window to arrive out of order and be kept until there is a set of consecutive packets to be delivered to the application layer.

1. W_S (Sender Window Size > 1)
2. Receiver Window (W_R) = Sender Window (W_S)
   $W_S = W_R$ because it is inefficient to discard out of order packets. So to accept out of order packets we need $W_S = W_R$.
3. If one packet is lost, the receiving station can accept the subsequent out of order packets.
4. After the timeout, only the selected packet (lost) is retransmitted.
5. In terms of efficiency selective repeat Go Back N is equal to or similar.
6. In terms of retransmissions STOP & WAIT is similar to selective repeat.
7. In the selective-repeat protocol, an acknowledgment number defines the sequence number of the error-free packet received.
8. **Ack's are independent:**
   - Ack for every packet (GBN uses cumulative)
   - If packet is lost then it is retransmitted after time out.
   - If packet is complete, then SR sends NAK (in GBN, even if packet is corrupted, then receiver will silently discards it).
   - Hence the purpose of NAK is that sender need not wait for timeout of packet.
     ∴ Ack timer is not required in SR.

Remember

- In the *selective-repeat* protocol, the size of the sender and receiver window can be at most one-half of 2^m.
- **Piggybacking** is used to improve the efficiency of the bidirectional protocols. When a packet is carrying data from A to B, it can also carry acknowledgment feedback about arrived packets from B; when a packet is carrying data from B to A, it can also carry acknowledgment feedback about the arrived packets from A.

Example-3.10 A 3000 km long link operates at 1.536 Mbps is used to transmit 64 byte frames and uses SWP. If propagation speed is 6 μsec/Km, how many bits should the sequence numbers be

Solution:

$$P_d = 18\text{ ms}$$
$$t_d = 33\text{ ms}$$
$$W_S = 1 + 2a = 1 + 2 \times \left(\frac{18\text{ms}}{33\text{ms}}\right) = 110$$
$$= \log_2(110) = 7 \text{ bits}$$

Example-3.11 The SR protocol is similar to GBN, except in the following way

(a) Frame formats are similar in both protocols
(b) The sender has a window defining maximum number of outstanding frames in both protocols
(c) Both uses piggy backed acknowledgment where possible and does not acknowledge every frame explicitly
(d) Both use piggyback approach that acknowledge the most recently received frame

Solution:

(a) Frame formats are similar

(b) Both are sliding window protocols and hence window defines maximum number of outstanding frames.

(c) Both use piggybacked ack's where ever possible but SR uses individual Ack's and hence acknowledges explicitly (False)

(d) Acknowledging the most recently received frame is same as GBN. But is SR every frame has to acknowledges (False)

***Example*-3.12** A 3000 km long trunk operates at 1536 Mbps and is used to transmit 64 byte frames and uses SWP. If the speed is 6 μsec/km, how many bits should the sequence numbers be

Solution:

d = 3000 km, B = 1536 Mbps, L = 64 B

$$P_d = 6\ \mu\text{sec/km} \times 3000\ \text{km} = 18000\ \mu\text{sec}$$

$$a = \frac{P_d}{t_d} = 5400054$$

$$1 + 2a = 109 \Rightarrow 19 = 7\text{bits}$$

3.8 Repeaters

A network device used to regenerate or replicate a signal. Repeaters are used in transmission systems ***to regenerate analog or digital signals*** distorted by transmission loss. Analog repeaters frequently can only amplify the signal while digital repeaters can reconstruct a signal to near its original quality.

In a data network, a repeater can relay messages between subnetworks that use different protocols or cable types. Hubs can operate as repeaters by relaying messages to all connected computers. A repeater cannot do the intelligent routing performed by bridges and routers.

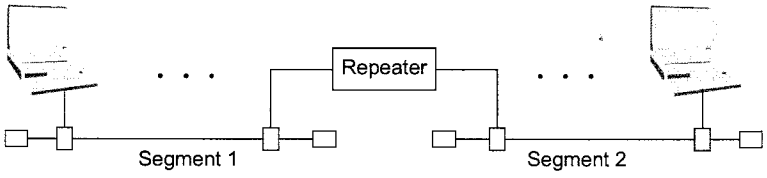
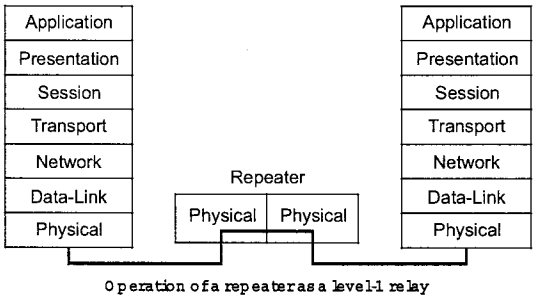


Figure: *Repeater connecting two LAN segments*

The repeater passes the digital signal bit-by-bit in both directions between the two segments. As the signal passes through a repeater, it is amplified and regenerated at the other end. The repeater does not isolate one segment from the other, if there is a collision on one segment; it is regenerated on the other segment. With reference of the ISO model, a repeater is considered as a *level-1.*



Operation of a repeater as a level-1 relay

Important features of a repeater are as follows:

- A repeater connects different segments of a LAN
- A repeater forwards every frame it receives
- A repeater is a regenerator, not an amplifier
- It can be used to create a single extended LAN

3.9 Hubs

A hub is a common connection point for devices in a network. Hubs are commonly used to connect segments of a LAN. A hub contains multiple ports. When a packet arrives at one port, it is copied to the other ports so that all segments of the LAN can see all packets.

3.9.1 What Hubs Do

Hubs and switches serve as a central connection for all of your network equipment and handles the data type known as frames. Frames carry your data. When a frame is received, it is amplified and then transmitted on to the port of the destination PC.

In a hub, a frame is passed along or "broadcast" to every one of its ports. It doesn't matter that the frame is only destined for one port. The hub has no way of distinguishing which port a frame should be sent to. Passing it along to every port ensures that it will reach its intended destination. This places a lot of traffic on the network and can lead to poor network response times.

3.9.2 Passive, Intelligent and Switching Hubs

A *passive hub* serves simply as a conduit for the data, enabling it to go from one device (or segment) to another. So-called *intelligent hubs* include additional features that enables the administrator to monitor the traffic passing through the hub and to configure each port in the hub. Intelligent hubs are also called *manageable hubs.*

A third type of hub, called a *switching hub*, actually reads the destination address of each packet and then forwards the packet to the correct port. It can be used to create multiple levels of hierarchy of stations.

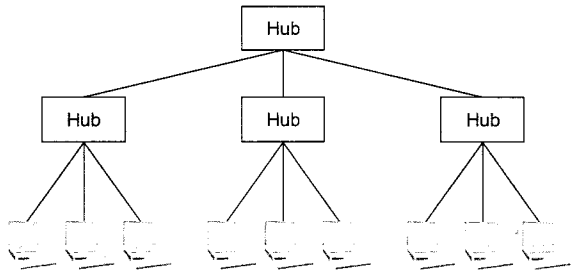


Figure: *Hub as a multi-port Repeater*

Hub as a **multi-port repeater** can be connected in a hierarchical manner to form a single LAN with many nodes.

3.10 Bridges

It is commonly used to connect two similar or dissimilar LANs The bridge operates in layer 2, that is data-link layer and that is why it is called *level-2 relay* with reference to the OSI model. It links similar or dissimilar LANs, designed to store and forward frames, it is protocol independent and transparent to the end stations. Use of bridges offer a number of advantages, such as higher reliability, performance, security, convenience and larger geographic coverage. But, it is desirable that the quality of service (QOS) offered by a bridge should match that of a single LAN. The parameters that define the QOS include *availability, frame mishaps, transit delay, frame lifetime, undetected bit errors, frame size* and *priority*. Key features of a bridge are mentioned below:

- A bridge operates both in physical and data-link layer
- A bridge uses a table for filtering/routing
- A bridge does not change the physical (MAC) addresses in a frame

- Types of bridges:
  (a) Transparent Bridges
  (b) Source routing bridges

A bridge must contain addressing and routing capability. Two routing algorithms have been proposed for a bridged LAN environment. The first, produced as an extension of IEEE 802.1 and applicable to all IEEE 802 LANs, is known as *transparent bridge*. And the other, developed for the IEEE 802.5 token rings, is based on *source routing approach*. It applies to many types of LAN including token ring, token bus and CSMA/CD bus.

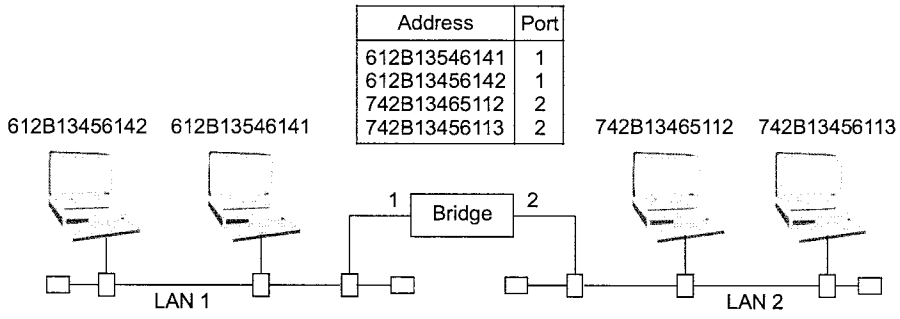


Figure: *A bridge connecting two separate LANs*

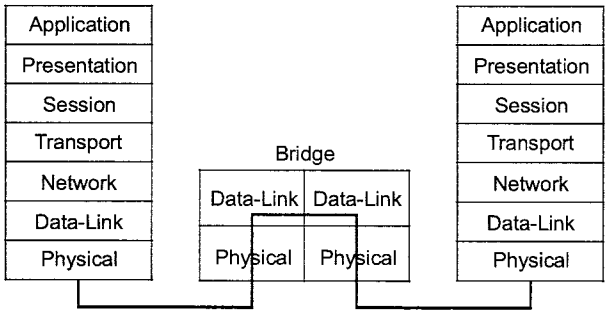


Figure: *Information flow through a bridge*

3.10.1 Transparent Bridges

The transparent bridge uses two processes known as **bridge forwarding** and **bridge learning**. If the destination address is present in the forwarding database already created, the packet is forwarded to the port number to which the destination host is attached. If it is not present, forwarding is done on all parts (flooding). This process is known as *bridge forwarding*. Moreover, as each frame arrives, its source address indicates where a particular host is situated, so that the bridge learns which way to forward frames to that address. This process is known as *bridge learning*. Key features of a transparent bridge are:

- The stations are unaware of the presence of a transparent bridge
- Reconfiguration of the bridge is not necessary; it can be added/removed without being noticed
- It performs two functions:
  (a) Forwarding of frames
  (b) Learning to create the forwarding table

3.10.2 Bridge Forwarding

Bridge forwarding operation is explained with the help of a flowchart in Figure. Basic functions of the bridge forwarding are mentioned below:

- Discard the frame if source and destination addresses are same

- Forward the frame if the source and destination addresses are different and destination address is present in the table
- Use flooding if destination address is not present in the table

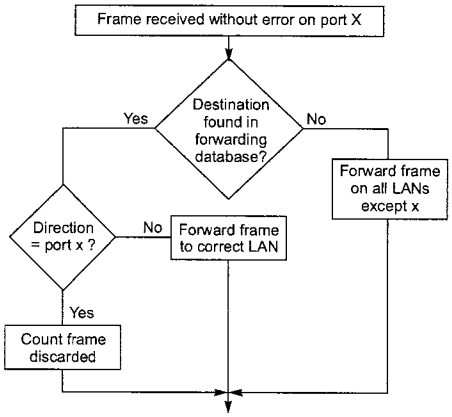


Figure: *Bridge Forwarding*

3.10.3 Bridge Learning

At the time of installation of a transparent bridge, the database, in the form of a table, is empty. As a packet is encountered, the bridge checks its source address and build up a table by associating a source address with a port address to which it is connected.

The flowchart of Figure explains the learning process. The table building up operation is illustrated in Figure.

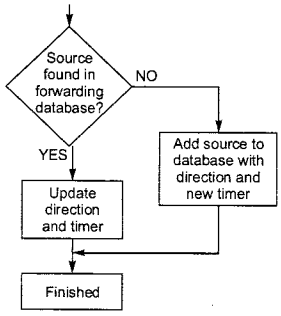
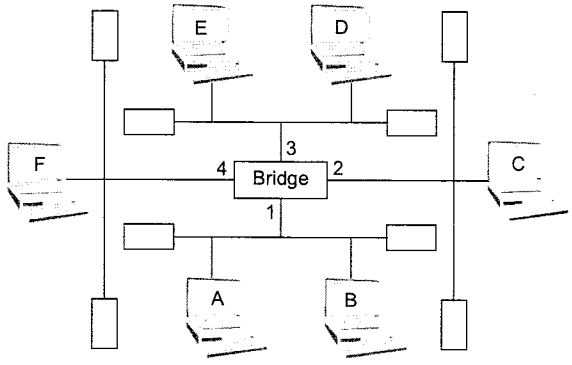


Figure: *Bridge learning*



Address	Port

Initial

Address	Port
A	1
C	2

after A & C sends a frame

Address	Port
A	1
B	1
C	2
D	3
E	3
F	4

After all the stations have sent a frame

Figure: *Creation of a bridge-forwarding table*

Loop Problem

Forwarding and learning processes work without any problem as long as there is no redundant bridge in the system. On the other hand, redundancy is desirable from the viewpoint of reliability, so that the function of a failed bridge is taken over by a redundant bridge. The existence of redundant bridges creates the so-called *loop problem* as illustrated with the help of Figure.

Assuming that after initialization tables in both the bridges are empty let us consider the following steps:

Step-1: Station-A sends a frame to Station-B. Both the bridges forward the frame to LAN Y and update the table with the source address of A.

Step-2: Now there are two copies of the frame on LAN-Y. The copy sent by Bridge-a is received by Bridge-b and vice versa. As both the bridges have no information about Station B, both will forward the frames to LAN-X.

Step-3: Again both the bridges will forward the frames to LAN-Y because of the lack of information of the Station B in their database and again Step-2 will be repeated, and so on.So, the frame will continue to loop around the two LANs indefinitely.

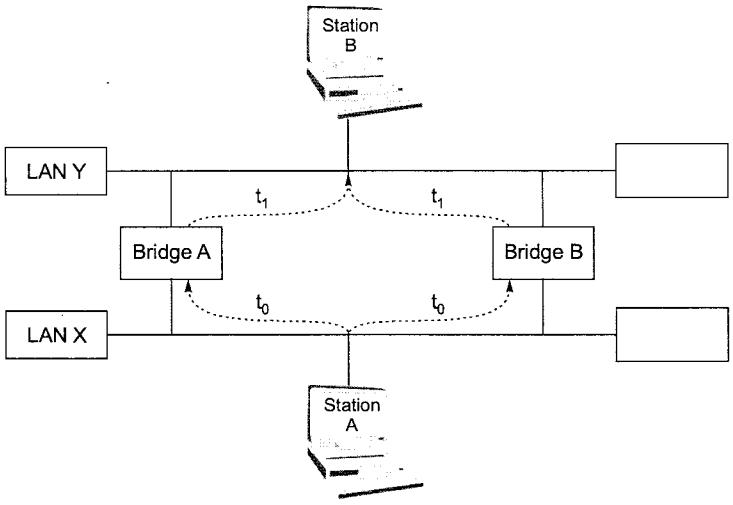


Figure: *Loop problem in a network using bridges*

Spanning Tree

As redundancy creates loop problem in the system, it is very undesirable. To prevent loop problem and proper working of the forwarding and learning processes, there must be only one path between any pair of bridges and LANs between any two segments in the entire bridged LAN. The IEEE specification requires that the bridges use a special topology. Such a topology is known as *spanning tree* (a graph where there is no loop) topology. The methodology for setting up a spanning tree is known as spanning tree algorithm, which creates a tree out of a graph. Without changing the physical topology, a logical topology is created that overlay on the physical one by using the following steps:

- Select a bridge as *Root-bridge,* which has the smallest ID.
- Select *Root ports* for all the bridges, except for the root bridge, which has least-cost path (say minimum number of hops) to the root bridge.
- Choose a *Designated bridge*, which has least-cost path to the Root-bridge, in each LAN.
- Select a port as *Designated port* that gives least-cost path from the Designated bridge to the Root bridge.
- Mark the designated port and the root ports as *Forwarding ports* and the remaining ones as *Blocking ports.*

The spanning tree of a network of bridges is shown in Figure. The forwarding ports are shown as solid lines, whereas the blocked ports are shown as dotted lines.

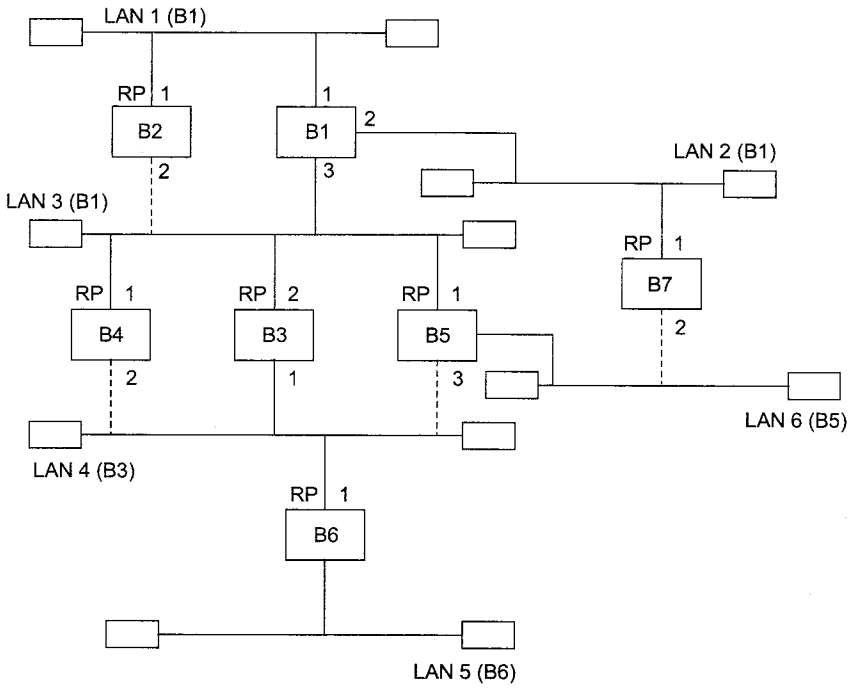


Figure: *Spanning tree of a network of bridges*

3.10.4 Source Routing Bridges

The second approach, known as *source routing,* where the routing operation is performed by the source host and the frame specifies which route the frame is to follow. A host can discover a route by sending a *discovery frame*, which spreads through the entire network using all possible paths to the destination. Each frame gradually gathers addresses as it goes. The destination responds to each frame and the source host chooses an appropriate route from these responses.

For example, a route with minimum hop-count can be chosen. Whereas transparent bridges do not modify a frame, a source routing bridge adds a routing information field to the frame. Source routing approach provides a shortest path at the cost of the proliferation of discovery frames, which can put a serious extra burden on the network. Figure below shows the frame format of a source routing bridge.

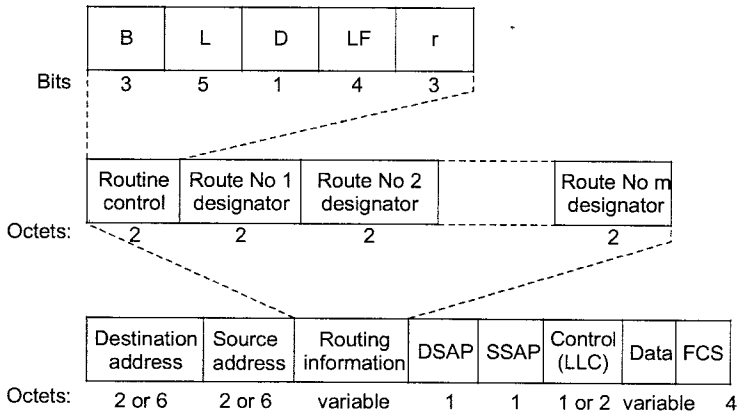


Figure: *Source Routing Frame*

3.11 Switches

A switch is essentially a fast bridge having additional sophistication that allows faster processing of frames. Some of important functionalities are:

- Ports are provided with buffer
- Switch maintains a directory: #address - port#
- Each frame is forwarded after examining the #address and forwarded to the proper port#
- Three possible forwarding approaches: Cut-through, Collision-free and Fully- buffered as briefly explained below.

Cut-through

A switch forwards a frame immediately after receiving the destination address. As a consequence, the switch forwards the frame without collision and error detection.

Collision-free

In this case, the switch forwards the frame after receiving 64 bytes, which allows detection of collision. However, error detection is not possible because switch is yet to receive the entire frame.

Fully Buffered

In this case, the switch forwards the frame only after receiving the entire frame. So, the switch can detect both collision and error free frames are forwarded.

3.11.1 Comparison between a Switch and a Hub

Although a hub and a switch apparently look similar, they have significant differences. As shown in Figure, both can be used to realize physical star topology, the hubs works like a logical bus, because the same signal is repeated on all the ports. On the other hand, a switch functions like a logical star with the possibility of the communication of separate signals between any pair of port lines. As a consequence, all the ports of a hub belong to the same collision domain, and in case of a switch each port operates on separate collision domain. Moreover, in case of a hub, the bandwidth is shared by all the stations connected to all the ports. On the other hand, in case of a switch, each port has dedicated bandwidth. Therefore, switches can be used to increase the bandwidth of a hub- based network by replacing the hubs by switches.

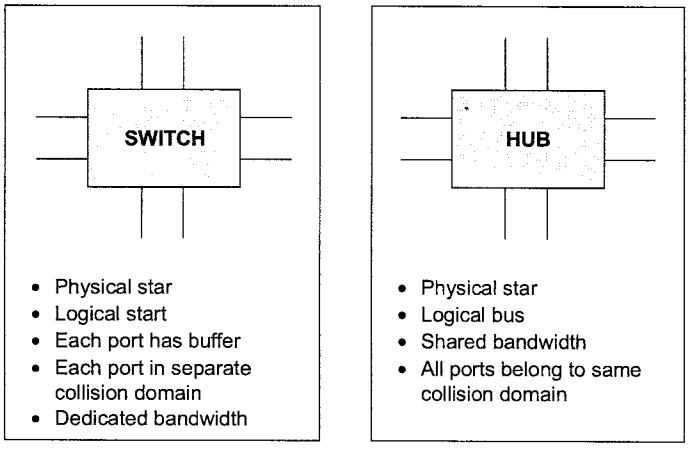


Figure: *Difference between a switch and a bridge*

3.12 Routers

A router is considered as a layer-3 relay that operates in the network layer, that is it acts on network layer frames. It can be used to link two dissimilar LANs. A router isolates LANs in to subnets to manage and control network traffic. However, unlike bridges it is not transparent to end stations. A schematic diagram of the router is shown on Figure.

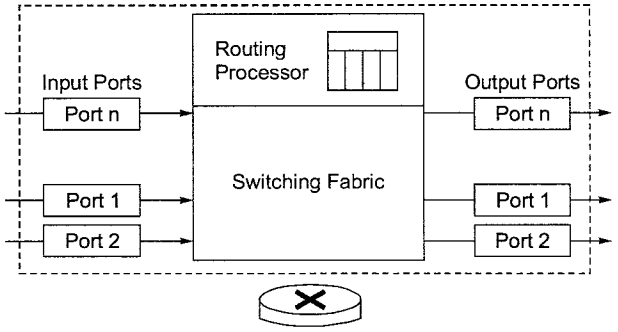


Figure: *Schematic diagram of a router*

A router has four basic components: Input ports, output ports, the routing processor and the switching fabric. The functions of the four components are briefly mentioned below.

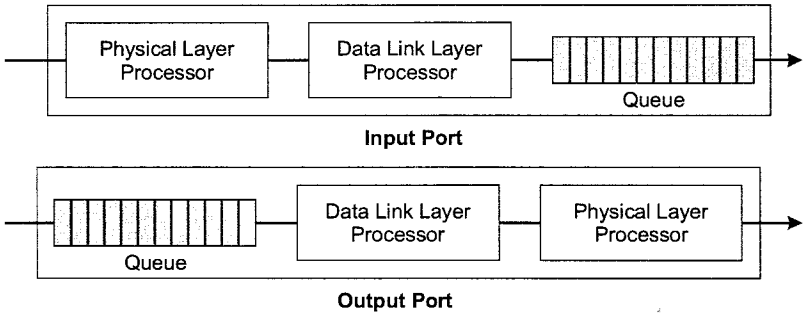


Figure: *Schematic diagram of a router*

- *Input port* performs physical and data-link layer functions of the router. As shown in Figure, the ports are also provided with buffer to hold the packet before forwarding to the switching fabric.
- *Output ports*, as shown in Figure, perform the same functions as the input ports, but in the reverse order.
- The *routing processor* performs the function of the network layer. The process involves table lookup.

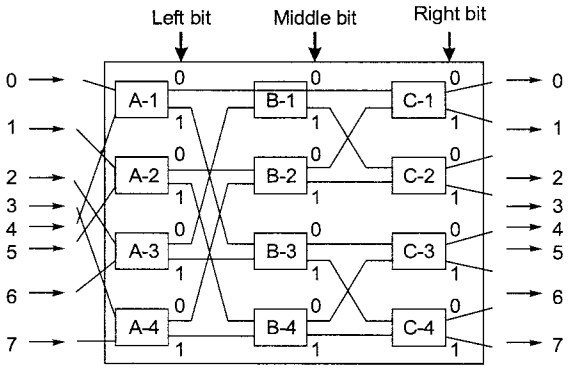


Figure: *Switching fabric of a router*

- The *switching fabric*, shown in Figure, moves the packet from the input queue to the output queue by using specialized mechanisms. The switching fabric is realized with the help of multistage interconnection networks.

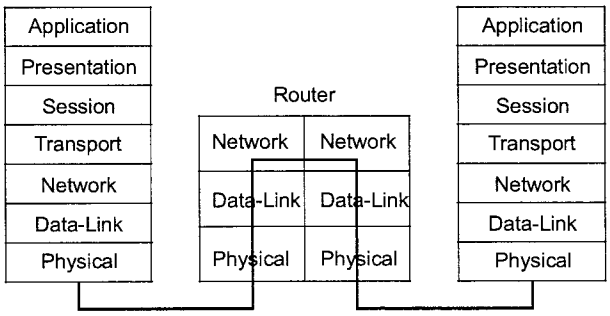


Figure: *Communication through a router*

3.13 Gateways

A gateway works above the network layer, such as application layer as shown in Figure. As a consequence, it is known as a Layer-7 relay. The application level gateways can look into the content application layer packets such as email before forwarding it to the other side. This property has made it suitable for use in Firewalls discussed in the next module.

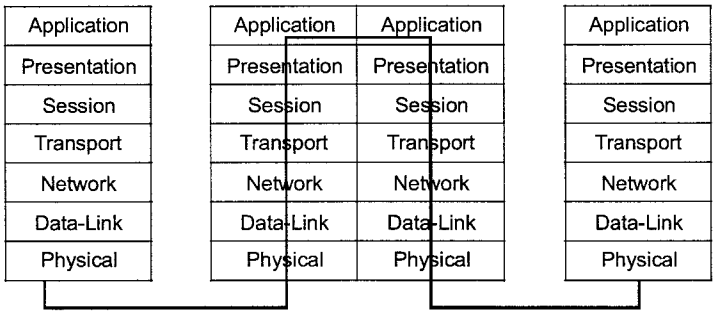
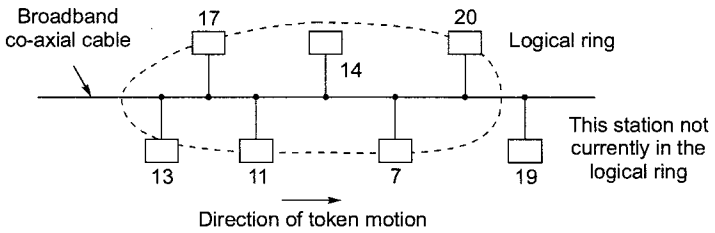


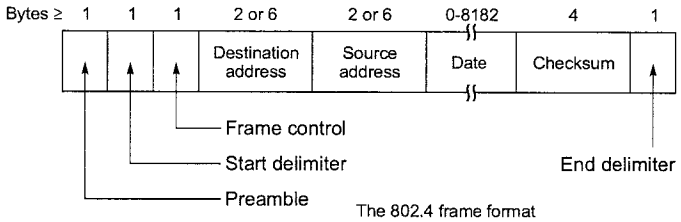
Figure: *Communication through a gateway*

3.14 IEEE Standard 802.4: Token Bus

A linear or tree-shaped bus with N stations. Stations are organized as a logical ring where each station knows the address of its logical right and left neighbors. Only the station in possession of a special control packet "token" is allowed to transmit. Once a station is done transmitting it passes the token to one of its two logical neighbors. A station in possession of the token and data to transmit passes it on. Collision-free protocol; no starvation.



Frame Format of 802.4 (Token Bus)



The 802.4 frame format

The preamble is used to synchronize the receiver's clock, as in 802.3, except that here it may be as short as 1 byte. The starting delimiter and Ending delimiter fields are used to mark the frame boundaries. Both of these fields contain analog encoding of symbols other than 0s and 1s, so that they cannot occur accidentally in the user data. As a result, no length field is needed.

The Frame control field is used to distinguish data frames from control frames. The Destination address and Source address fields are the same as in 802.3. The Data field may be up to 8182 bytes long when 2-byte addresses are used, and up to 8174 bytes long when 6-byte address are used. The Checksum is used to detect transmission errors. It uses the same algorithm and polynomial as 802.3.

Frame control field	Name	Meaning
00000000	Claim_token	Claim token during ring initialization
00000001	Solicit_successor_1	Allow stations to enter the ring
00000010	Solicit_successor_2	Allow stations to enter the ring
00000011	Who_follows	Recover from lost token
00000100	Resolve_contention	Used when multiple stations want to enter
00001000	Token	Pass the token
00001100	Set_successor	Allow station to leave the ring

The token bus control frames

***Example*-3.13** For a standard Ethernet bandwidth is 10 Mbps the minimum frame size should be 64 bytes. To support CSMA/CD, Transmission Time is twice of Propagation Delay. In fast Ethernet, what will be the length of cable to support same frame size of 64 bytes, if L is length of cable in standard Ethernet?

Solution:

$$\text{Transmission Time} = 2 \times \text{Propagation Time} \quad \text{i.e.,} \quad \frac{\text{Data size}}{\text{B.W.}} = 2 \times \frac{d}{v}$$

Velocity is same when media is same, bandwidth for fast ethernet is 100 Mbps. In order to maintain the same frame size since bandwidth is increased from 10 to 100 Mbps the distance will be reduced from L to L/10.

***Example*-3.14** A group of N stations share a 56-kbps pure ALOHA channel. Each station outputs a 1000-bit frame on average once every 100 sec, even if the previous one has not yet been sent (e.g., the stations can buffer outgoing frames). What is the maximum value of N?

Solution:

Maximum bandwidth available = 0.184 × 56 Kbps = 10.3 Kbps

$$\text{Each station bandwidth} = \frac{1000 \text{ bits}}{100 \text{ sec}} = 10 \text{ bps}$$

$$\therefore \quad N = \frac{10.3 \times 10^3}{10} = 1030 \text{ stations}$$

Example-3.15 In an infinite-population s lotted ALOHA system, the mean number of slots a station waits between a collision and a retransmission is 4. Plot the delay versus throughput curve for this system.

Solution:

(a) Chance of success on first attempt in (p) = e^{-G}
'G' Attempts made in time slot, P = e^{-2} = 0.135
Note: Only 2 attempts made in a slot.

(b) Probability of K collisions
$\Rightarrow \quad (1 - e^{-G})^K \times (0.135) = (0.865)^K \times 0.135$

(c) $e^G = e^2 = 7.38$ are the expected numbers of transmissions.

Example-3.16 How long does a station, s, have to wait in t he worst case before it can start transmitting its frame over a LAN that uses the basic bit-map protocol?

Solution:

In CSMA/CD the station has to transmit for 2PT in order to size the channel. (Collision can only be detected if it transmits for 2PT)

(a) Contention slot length = $\frac{2 \times 2 \times 10^3}{2.46 \times 10^8} = 16.2\ \mu\text{sec}$

(b) Contention slot length = $\frac{2 \times 40 \times 10^3}{1.95 \times 10^8} = 410\ \mu\text{sec}$

Example-3.17 A 1-km-long, 10-Mbps CSMA/CD LAN (not 802.3) has a propagation speed of 200 m/isec. Repeaters are not allowed in this system Data frames are 256 bits long, including 32 bits of header, checksum, and other overhead. The first bit slot after a successful transmission is reserved for the receiver to capture the channel in order to send a 32-bit acknowledgement frame. What is the effective data rate, excluding overhead, assuming that there are no collisions?

Solution:

Sender captures channel in 2 PT = 10 μsec.

Sender takes 25.6 μsec to send 256 bits $\Rightarrow$ 256 μsec

Last bit takes 5 μsec to reach receiver.

10 μsec for receiver to size channel.

3.2 use to send Ack.

5 μsec to for last bit of Ack to reach.

$\therefore$ 10 + 25.6 + 5 + 10 + 3.2 + 5 = 58.8 μsec

Total data sent = 256 – 32 = 224 bits

$\therefore$ Throughput (effective bandwidth) = $\frac{224}{58\ \mu\text{sec}} = 3.8$ Mbps

Example-3.18 An IP packet to be transmitted by Ethernet is 60 bytes long, including all its headers. If LLC is not in use, is padding needed in the Ethernet frame, and if so, how many bytes?

Solution:

The minimum ethernet frame is 64 bytes, including both addresses in the ethernet frame header, the type/length field and the checksum.

Since header fields occupy 18 bytes and the packet is 60 bytes the total frame size is 78 bytes, which exceeds the 64 byte minimum. Therefore no padding is used.

***Example*-3.19** Some books quote the maximum size of an Ethernet frame as 1522 bytes instead of 1500 bytes. Are they wrong?

Solution:

The pay load is 1500 bytes

But, Destination address + Source address + Type/Length + Checksum + Other + 1500 B = 1522 B

The maximum frame size is supported by IEEE 802.3 is 1518 bytes.

The maximum frame size i.e. supported by IEEE 802.3 is 1500 bytes.

Summary

- TCP/IP model did not define any protocol for data link layer and physical layer. Therefore we can define our on protocol for data link and physical layer.
- Data link layer concerned with reliable, error-free and efficient communication between adjacent machines in the network.
- The following are the services provide to the network layer:
  (a) Unacknowledged connectionless service-best effort.
  (b) Acknowledged connection-less service-acknowledged delivery.
  (c) Acknowledged connection-oriented service-reliable delivery.
- The DLL translates the physical layer's raw bit stream into discrete units (messages) called frames.
- When using bit stuffing, locating the start/end of a frame is easy, even when frames are damaged. The receiver simply scans arriving data for the reserved patterns.
- Reliable delivery is achieved using the "acknowledgments with retransmission" paradigm.
- Retransmission timers are used to resend frames that don't produce an ACK. When sending a frame, schedule a timer at some time after the ACK should have been returned. If the timer goes off, retransmit the frame.
- Bits used for sequence numbers depend on the number of frames that can be outstanding at any one time.
- The concept of including extra information in the transmission for error detection is called redundancy. It is called so because the extra bits are redundant to the information, they are discarded as soon as the accuracy of the information has been determined.
- The Hamming code can be applied to data units of any length and uses the relationship between data and redundancy bits that can be added to the end of the data unit or interspersed with the original data bits.
- The SWP protocols allow both link nodes (A, B) to send and receive data and acknowledgments. (Full Duplex).
- The three famous flow control protocols along with ARQ are:
  (a) STOP and WAIT ARQ
  (b) STOP Back N ARQ
  (c) Selective repeat ARQ

- In STOP & WAIT as the distance increases efficiency (η) decreases.
- Every packet or frame has individual timeout timers.
- Internetworking creates a single virtual network over which all stations in different network can communicate seamlessly and transparently.
- A repeater operates in the physical layer. Data received on one of its ports is relayed on the remaining port bit-by-bit without looking into the contents.
- A bridge operates in the Data link layer. It looks into various fields of a frame to take various actions. A bridge helps to create a network having different collision domains.
- If there exit more than one path between two LANs through different bridges, there is a possibility of continuous looping of a frame between the LANs. To avoid the loop problem, spanning tree topology is used.
- In the source routing protocol, a host can discover a route by sending a *discovery frame*, which spreads through the entire network using all possible paths to the destination.
- Transparent bridge protocol uses spanning tree algorithm, where a unique path is used for communication between two stations.
- A router overcomes the following limitations of a bridge:
  (a) Linking of two dissimilar networks
  (b) Routing data selectively and efficiently
  (c) Enforcement of security
  (d) Vulnerability to broadcast storm

Student's Assignment

Q.1 In CRC Checking, the divisor is _____ the CRC
(a) One bit less than
(b) One bit more than
(c) The same size as
(d) There bits more than

Q.2 A Go-Back-N ARQ uses a window of size 15. How many bits are needed to define the sequence number?
(a) 15 (b) 4
(c) 16 (d) 5

Q.3 Even-parity checking function
(a) Passes data unit with even number of 1's
(b) Passes data unit with odd number of 1's
(c) Passes data unit with even number of 0's
(d) Passes data unit with odd number of 0's

Q.4 Which of the following statement is correct with respect to Half Duplex communication?
(a) Error correction is not possible
(b) Always a pair of physical links is necessary
(c) Two connections can be used for sending and receiving data
(d) It cannot be used in broadcast networks

Q.5 Sliding Window Protocol is
(a) Used to manage the protocols in the Windows Operating Systems
(b) Used to filter the packets in farewells
(c) Used to control the flow of frames in data communications
(d) Used to exchange Windows among remote hosts

Q.6 In a Go-back NARQ, if the window size is 63, what is the range of sequence numbers?
(a) 0 to 63 (b) 0 to 64
(c) 1 to 63 (d) 1 to 64

Q.7 In Go-back N ARQ, if frames 4, 5 and 6 are received successfully, the receiver may send an ACK _____ to the sender.
(a) 5 (b) 6
(c) 7 (d) Any of these

Q.8 For a sliding window of size n – 1 (n sequence numbers), there can be maximum of _____ frames sent but unacknowledged

(a) 0 (b) n – 1
(c) n (d) n + 1

Q.9 For stop and wait ARQ, for n data packets sent, _____ acknowledgments are needed.

(a) n (b) 2n
(c) n – 1 (d) n + 1

Q.10 What is the remainder obtained by dividing $x^7 + x^5 + 1$ by the generator polynomial $x^3 + 1$?

(a) $x^4 + x^2 - x$ (b) $x^2 + x + 1$
(c) $-x^2 + x + 1$ (d) $x^3 + x - 1$

Q.11 A channel has a bit rate of 20 Kbps and a propagation delay of 100 msec. For what sizes does stop and wait gives an efficiency of 50%?

(a) 250 bits (b) 500 bits
(c) 1000 bits (d) 4000 bits

Q.12 The maximum window size for data transmission using selective reject protocol with n bit frame sequence number is

(a) 2^n (b) 2^{n-1}
(c) $2^n - 1$ (d) 2^{n-2}

Q.13 In a sliding window ARQ scheme, the transmitter's window size is N and the receiver's window size is M. The minimum number of distinct sequence numbers required to ensure correct operation of the ARQ scheme is

(a) Min (M,N) (b) Max (M, N)
(c) M + N (d) MN

Q.14 Which of the following indicates the increasing order of accuracy in error detection?

(a) CRC, Single Parity, Block Sum Check
(b) Block Sum Check, CRC, Single Parity
(c) Single Parity, Block sum Check, CRC
(d) CRC, Block Sum Check, Single Parity

Q.15 The following is a set of codewords:

00000000
00000011
00001001
00001010
00001100
00010100
00101011

Which of the following is the minimum Hamming distance between them?

(a) 4 (b) 2
(c) 1 (d) 3

Q.16 An upper-layer packet is split into 10 frames, each of which has an 80% chance of arriving undamaged. If no error control is done by the data link protocol, how many times must the message be sent on average to get the entire thing through?

Q.17 A channel has a bit rate of 4 kbps and a propagation delay of 20 msec. For what range of frame sizes does stop-and-wait give an efficiency of at least 50%?

Q.18 The distance from earth to a distant planet is approximately 9×10^{10} m. What is the channel utilization if a stop-and-wait protocol is used for frame transmission on a 64 Mbps point-to-point link? Assume that the frame size is 32 KB and the speed of light is 3×10^8 m/s.

Q.19 Compute the fraction of the bandwidth that is wasted on overhead (headers and retransmissions) for protocol 6 on a heavily loaded 50 kbps satellite channel with data frames consisting of 40 header and 3960 data bits. Assume that the signal propagation time from the earth to the satellite is 270 msec. ACK frames never occur. NAK frames are 40 bits . The error rate for data frames is 1%, and the error rate for NAK frames is negligible. The sequence numbers are 8 bits.

Q.20 Consider an error-free 64-kbps satellite channel used to send 512-byte data frames in one direction, with very short acknowledgments coming back the other way. What is the maximum throughput for window sizes of 1, 7, 15, and 127? The earth-satellite propagation time is 270 msec.

Q.21 Which sublayer of data link layer performs data link functions that depend upon the type of medium?
(a) Logical link control sublayer
(b) Media access control sublayer
(c) Network interface control sublayer
(d) None of the above

Q.22 Header of frame generally contains
(a) Synchronization bytes
(b) Addresses
(c) Frame identifies
(d) All of the above

Q.23 When 2 or more bits in a data unit has been changed during the transmission, the error is called
(a) Random error (b) Burst error
(c) Inverted error (d) None of these

Q.24 Which one of the following is a data link protocol?
(a) Ethernet
(b) Point to point protocol
(c) HDLC
(d) All of the above

Q.25 Which one of the following is the multiple access protocol for channel access control?
(a) CSMA/CD (b) CSMA/CA
(c) Both (a) and (b) (d) None of these

Q.26 The Techniques of temporarily delaying outgoing acknowledgments so that they can be hooked onto the next outgoing data frame is called.
(a) Piggybacking
(b) Cyclic redundancy check
(c) Fletcher's checksum
(d) None of these

Q.27 Baud means?
(a) The number of bits transmitted per unit time
(b) The number of bytes transmitted per unit time
(c) The rate at which the signal changes
(d) None of these

Q.28 In OSI model dialogues control and token management are responsibilities of?
(a) Session layer (b) Network layer
(c) Transport layer (d) Data link layer

Q.29 Which of the following connectivity devices typically work at the physical layer of the OSI model?
(a) Routers (b) Bridges
(c) Repeaters (d) Gateways

Q.30 For computers to communicate on a network using TCP/IP, which of the following setting must be unique for each computer
(a) IP Address (b) Subnet Mask
(c) Default Gateway (d) WINS Server

Q.31 Match **List-I** with **List-II** and select the correct answer using the codes given below the lists:

List-I	List-II
A. repeaters	1. Data link layer
B. Bridges	2. Network layer
C. Routers	3. Physical layer

Codes:

	A	B	C
(a)	2	3	1
(b)	3	1	2
(c)	3	2	1
(d)	2	1	3

Q.32 Brouter
(a) Combines the feature of both bridges & routers
(b) It is a type of bridge
(c) It is a type of router
(d) None of the above

Q.33 The correct order of the corresponding OSI layers for a Router, Medium Access Control, Repeater and FTP is
(a) Network, Data link, Application and Physical
(b) Physical, Data Link, Session and Transport
(c) Network, Data Link, Physical and Application
(d) Presentation, Network, Transport and Application

Q.34 With the use of which of the following device(s) and cables can a LAN based on star topology be setup?
(a) Router (b) Bridge
(c) Switch (d) Repeater

Q.35 Two Ethernets can be interconnected by the following device, which operates on the data link layer,
(a) Hub (b) Switch
(c) Router (d) Gateway

Q.36 A _____ is actually a multiport repeater.
(a) Bridge (b) Router
(c) VLAN (d) Hub

Q.37 _____ is a client-server program that provides an IP address, subnet mask, IP address of a router and IP address of a name server to a computer.
(a) NAT (b) CIDR
(c) ISP (d) DHCP

Q.38 Which one of the following statements is false?
(a) Packet switching leads to better utilization of bandwidth resources than circuit switching
(b) Packet switching results in less variation in delay than circuit switching
(c) Packet switching requires more per -packet processing than circuit switching
(d) Packet switching can lead to reordering unlike in circuit switching

Q.39 With respect to Circuit Switching and Packet Switching, which of the following statement is incorrect?
(a) In circuit switching after data transfer begins, no busy conditions take place
(b) In packet switching, each packet of the same message must follow the same route
(c) In packet switching, each packet must contain the addressing information
(d) In circuit switching, a circuit must be established on the network prior to the data transfer

Q.40 Which of the following statement is incorrect?
1. A repeater is a generator, not an amplifier.
2. A repeater must be placed so that signal reaches it before any noise changed the meaning of any bits.

(a) Only 1 (b) Only 2
(c) 1 and 2 (d) None of these

Q.41 Communication via circuit switching involves three phase which are
(a) Circuit establishment, Data transfer, Circuit disconnect
(b) Circuit establishment, Data compression, Circuit disconnect
(c) Data transfer, Data compression, Circuit disconnect
(d) Circuit established, Data compression, Circuit disconnect

Q.42 Which of the following is/are correct
1. Circuit switching is designed for voice communication.
2. In Circuit switching network connection provides for transmission at varying data rate
3. Circuit switching sees all transmission are equal.
4. Circuit switching is less suited to data and non-conversation transmission.

(a) Only 3 (b) 2 and 3
(c) Only 4 (d) 1, 3 and 4

Q.43 Your company uses both ethernet and token ring LAN's in the same building. Users need access to resources and the ability to the transmit data between the LAN's. Which device would you use to enable communications?
(a) Bridge (b) Router
(c) Gateways (d) Repeater

Q.44 Who can send ICMP error-reporting messages?
(a) Rourters (b) Destination hosts
(c) Source host (d) Both (a) and (c)

Q.45 Which of the following is false?
(a) Bridges operate at the MAC layer
(b) Routers filter out unnecessary network traffic and remove it from network segments
(c) Routers cannot act as bridges
(d) Routers are often used to implement farewells

Q.46 Spanning tree protocol for ethernet switches. A network of ethernet switches uses the spanning tree protocol so that

(a) The switch can learn, and build an accurate table of IP address

(b) Packets don't cycle in the network forever

(c) The port of every switch forwards all the packets it receives

(d) Packets will follow the shortest path to a destination

Q.47 Which of the following is NOT true with respect to a transparent bridge and a router?

(a) Both selectively forward data packets

(b) Bridge uses IP addresses while a router uses MAC addresses

(c) A router can connect LAN with WAN

(d) A bridge builds up its routing table by inspecting incoming packets

Q.48 To interconnect two IP classes, Class A and Class C networks,

(a) A class B network is needed

(b) A bridge is needed

(c) A router is needed

(d) A layer 2 Ethernet switch is needed

Answer Key:

1.	(b)	**2.** (b)	**3.** (a)	**4.** (c)	**5.** (c)
6.	(a)	**7.** (c)	**8.** (b)	**9.** (a)	**10.** (b)
11.	(d)	**12.** (b)	**13.** (c)	**14.** (c)	**15.** (b)

16. 9.3 **17.** Min 160 bits **18.** 6.6×10^{-4}

19. 1.9 %

20. (*i*) 6.78 kbps (*ii*) 47.47 kbps (*iii*) 64 kbps (*iv*) 64 kbps

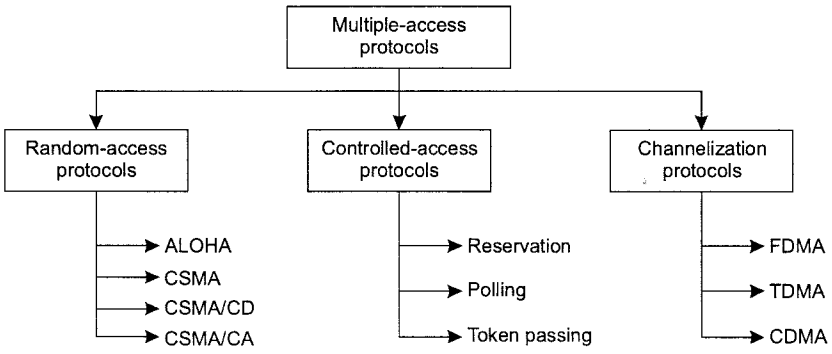
21.	(b)	**22.** (d)	**23.** (b)	**24.** (d)	**25.** (c)
26.	(a)	**27.** (c)	**28.** (a)	**29.** (c)	**30.** (a)
31.	(b)	**32.** (a)	**33.** (c)	**34.** (c)	**35.** (b)
36.	(d)	**37.** (d)	**38.** (b)	**39.** (b)	**40.** (d)
41.	(a)	**42.** (d)	**43.** (a)	**44.** (d)	**45.** (c)
46.	(b)	**47.** (b)	**48.** (c)		

CHAPTER 04

Network Layer

4.1 Introduction

Many protocols have been devisee to handle access to a shared link. All of these protocols belong to a sublayer in the data link layer called *Media Access Control* (MAC).



Broadcast networks with multi-access (or random access) shared channels include the majority of LANS, all wireless and satellite networks.

Static Channel Allocation: Frequency Division Multiplexing (FDM), Time Division Multiplexing (TDM) - too wasteful of available bandwidth.

Dynamic Channel Allocation: No predetermined sender access order to the channel in order to send the data.

4.2 Channel Allocation Problem

4.2.1 Static Channel Allocation in LANs and MANs

The traditional way of allocating a single channel, such as a telephone trunk, among multiple competing users is Frequency Division Multiplexing (FDM). If there are N users, the bandwidth is divided into N equal sized portions each user being assigned one portion. Since each user has a private frequency band, there is no interference between users.

Disadvantages

Inefficient to divide into fixed number of chunks. May not all be used, or may need more.

Doesn't handle burstyness.

T = mean time delay

C = capacity (bps) of channel

λ = arrival rate of frames (frames/sec.)

$1/\mu$ = bits/frame

$$\boxed{T = \frac{1}{\mu C - \lambda}}$$

Now divide this channel into N sub channels, each with capacity C/N. Input rate on each of the N channels is 1/N. Now

$$T(FDM) = \frac{1}{\mu(C/N) - \lambda/N} = \frac{N}{\mu C - \lambda} = NT \frac{1}{\mu(C/N) - \lambda/N}$$

4.2.2 Dynamic Channel Allocation

Possible underlying assumptions include:

- **Station Model:** Assumes that each of N "stations" (packet generators) independently produce frames. The probability of producing a packet in the interval Dt is IDt is where I is the constant arrival rate of new frames. That station generates no new frame until that previous one is transmitted.
- **Single Channel Assumption:** There's only one channel, all stations are equivalent and can send and receive on that channel.
- **Collision Assumption:** If two frames overlap in time-wise, then that's collision. Any collision is an error, and both frames must be retransmitted. Collisions are the only possible error.
- Time can be divided into discrete or continuous slots
  (a) *Continuous Time*: There's no master clock governing transmission. Time is not in discrete chunks.
  (b) *Slotted Time*: Alternatively, frame transmissions always begin at the start of a time slot. Any station can transmit in any slot (with a possible collision.)
- Stations can sense a channel is busy before they try it.

Protocol Assumptions

N independent stations (sender or, computers etc). A station is blocked until its generated frame is transmitted. Probability of a frame being generated in a period of length Dt is IDt where I is the arrival rate of frames. Only a single channel is available. The transmission of two or more frames on the channel at the same time creates a collision and destroyed data.

Time can be either: Continuous or slotted.

Carrier Sense: A station can sense if a channel is busy before transmission.

No Carrier Sense: Timeout used to sense loss of data.

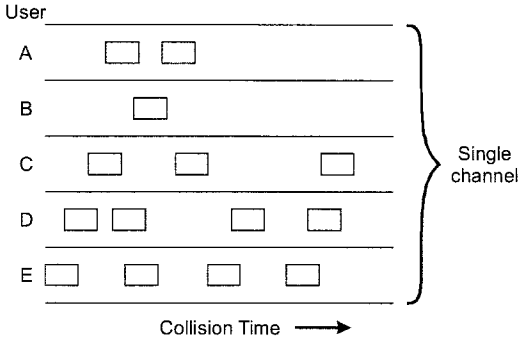
4.3 Multiple Access Protocols

4.3.1 Pure ALOHA

Stations transmit whenever data is available at arbitrary times (forming a contention system). Colliding frames are destroyed. Frame destruction sensed by listening to channel:

- Immediate collision feedback LANs
- 270 msec feedback delay in satellite transmission

When a frame is destroyed the sender waits a random period of time before retransmitting the frame.



In Pure ALOHA, frames are transmitted at completely arbitrary times

Frame throughput of Pure ALOHA

Infinite senders assumed. New frames rate (or frames success rate): Poisson distribution with mean rate S frames/frame time. Combined frame rate with retransmissions G frames/frame time.

$S = GP_0$ where P_0 = probability that a frame is successful.

t = time required to transmit a frame.

A frame is successful if no other frames are transmitted in the vulnerable period from t_0 to $(t_0 + 2t)$. Probability k frames are generated during a frame time:

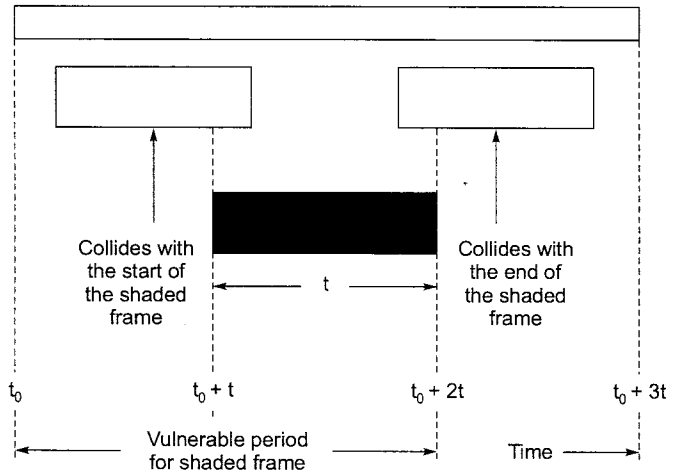
$$P_r[k] = \frac{G^k e^{-G}}{K!}$$

Probability of zero frames in two frame periods is $P_0 = e^{-2G}$

$S = GP_0 = Ge^{-2G}$, Max (S) = 1/2e = 0.184 at G = 0.5 i.e. 18.4% throughput.

Vulnerable Period in Pure ALOHA

For successful frame transmission: No other frame should be on the channel for vulnerable period equal to twice the time to transmit on frame = 2t



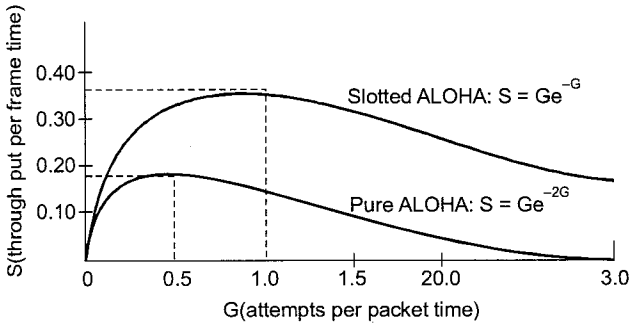
4.3.2 Slotted ALOHA

Time is divided into discrete frame time slots. A station is required to wait for the beginning of the next slot to transmit. Vulnerable period is halved as opposed to pure ALOHA.

$$S = GP_0 = Ge^{-G}$$

Max (s) = 1/e = 0.368 at G = 1, i.e. 36.8% throughput

Expected number of retransmissions: $E = e^{G}$



4.3.3 Carrier Sense Multiple Access (CSMA) Protocols

Medium Access Control (MAC) Protocols for shared channels where a station listens to the channel and has the ability to sense the carrier and thus can detect if the channel is idle before transmitting, and possibly detect the occurrence of a collision after attempting to transmit a frame.

1-Persistent CSMA

A ready station senses to the channel for transmissions until channel is not free. Once it detects an idle channel it transmits a frame immediately. In case of a collision, the stations involved in the collision wait a random period of time before retransmission.

Nonpersistent CSMA

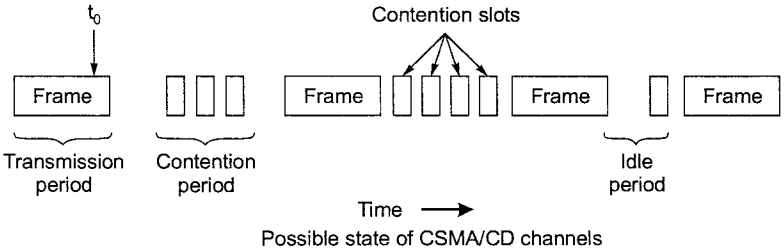
If a ready station senses an idle channel it starts transmission immediately. If a busy channel is sensed a station waits a random period of time before sensing the channel again.

P-Persistent CSMA (Applies to Slotted Channels)

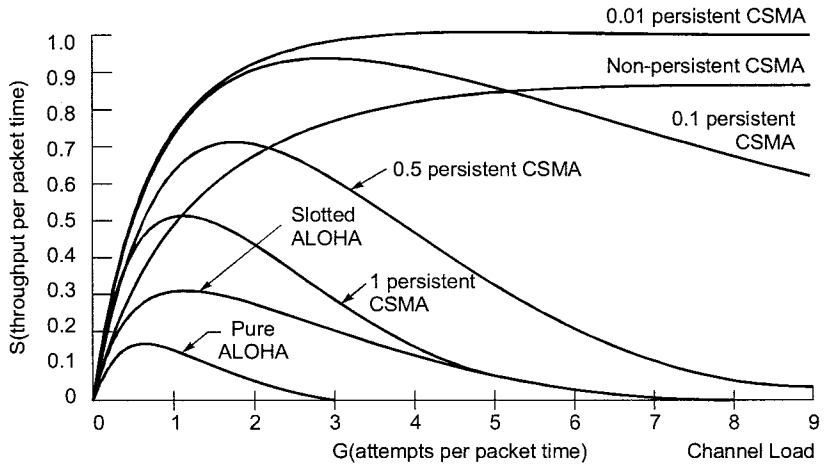
If A ready station senses an idle channel, it transmits with probability p or defers transmission to the next time slot with probability q = 1 – p. If the next slot is idle it transmits and defers again with probabilities p and q. The process continues until the frame has been transmitted or another station has seized the channel. A station can sense the channel at only start of a slot.

4.4 CSMA with Collision Detection (CSMA/CD)

If two stations begin transmitting simultaneously and detect a collision, both stations abort their transmission immediately. Once a collision is detected each ready stations waits a random period of time before attempting to retransmit. Worst-case contention interval (the duration of a collision last) is equal to 2t (t is the propagation time between the two farthest stations).



Channel Utilization Vs. Load for Random Access Protocols



4.4.1 IEEE Standard 802.3 (Ethernet)

1-persistent CSMA/CD LAN. 10-1000Mbps speed. We don't have acknowledgment in LANs. Broadcast to several destinations possible using addresses with high-order bit = 1. Random number of waiting slots upon a collision is chosen by binary exponential backoff algorithm:

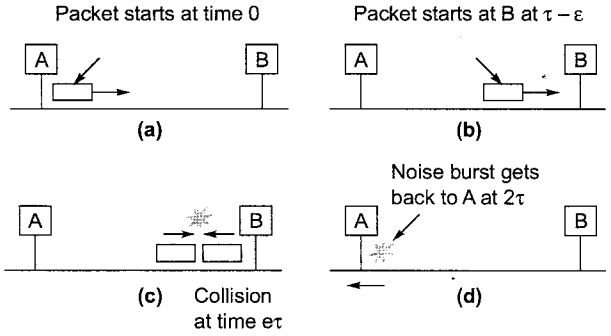
Simplified Channel efficiency = T/(T + 2t/A)

where $A = kp(1-p)^{k-1}$ k stations each with p probability to transmit in a contention slot.

T time to transmit a frame.

τ worst case propagation delay.

Collision detection delay in ethernet, CSMA/CD



As a result, the time to transmit a frame on the media must be longer than 2t otherwise collisions will be undetectable to some stations. This determines the minimum allowed frame size.

Whenever collision occur 48 bit jamming signal is sent by every station that senses collision as a responsibility. So that the transmitting stations can abort their transmission. This jamming signal is sent at different frequency inorder to distinguish from normal signal on the channel. Jamming signal does not affect the minimum frame size.

Minimum frame size $(L) \geq 2 \times P_d \times B$

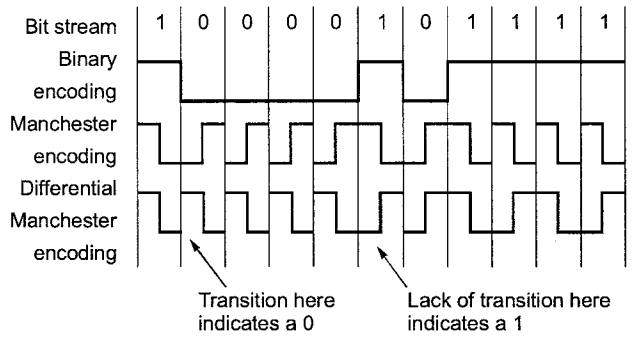
Ethernet IEEE 802.3 standard defines the minimum ethernet frame size as 64 bytes and maximum as 1526. Ethernet MTU is 1500 bytes, meaning the largest IP packet (pay load) and ethernet can contain is 1500 bytes. Adding 26 bytes for ethernet header results in maximum frame size (=MTU) is 1526.

Binary Exponential Backoff Algorithm

After a collision, station waits 0 or 1 slot. If it collides again while doing this send, it picks a time of 0,1,2,3 slots. If again it collides the wait is 0 to $2^3 - 1$ times. Max time is $2^{10} - 1$ (or equal to 10 collisions.) After 16 collisions, an error is reported.

First collision: Wait 0 or 1 slots. After i collisions wait a random number of slots between 0 and $2^i - 1$ with a maximum of 1023. After 16 collisions, failure is reported to higher layers.

IEEE STANDARD 802.3 10BASE-T LANS/MANS Bit Encoding



Encoding ensures that the start, middle, end of each bit is known without using an external clock. Aids in collision detection.

Requires twice the amount of bandwidth of straight binary encoding.

High signal = 0.85 volts, low signal = – 0.85 volts.

802.3 (Ethernet) Frame Format

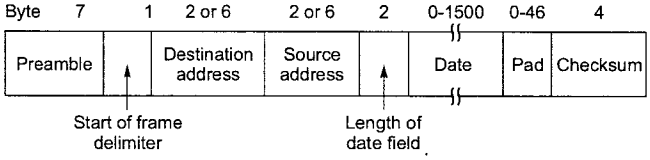
- Preamble: 7 bytes 10101010 to synchronize station clocks.
- Start of frame byte : 10101011
- Destination address.
- Source address.
- Length of data field.
- Data: 0-1500 bytes.
- Pad: padding bits to make frame size more than the minimum size (64 bits)
- Checksum: 4 bytes (CRC-32).

Minimum Frame Size

The time to transmit a frame on the media must be longer than 2τ otherwise collisions will be undetectable to some stations.

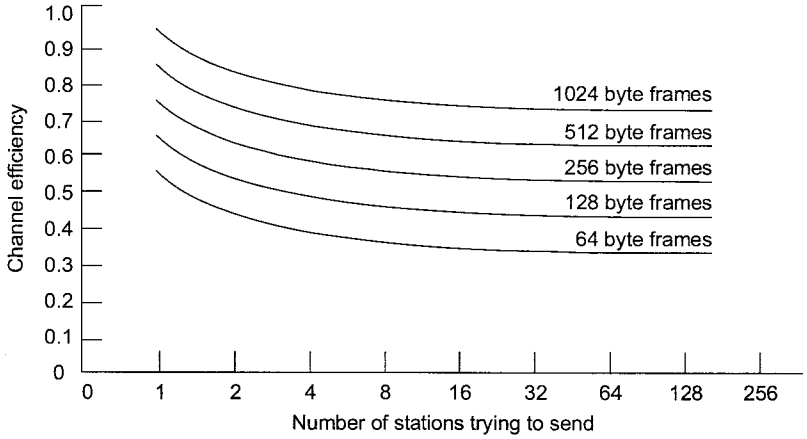
For CSMA/CD to work, we need a restriction on the frame size. Before sending the last bit for the frame, the sending station must detect a collision, if any and abort the transmission, because once the entire frame is sent, station does not keep a copy of the frame and does not monitor the line for collision detection.

Hence, time to transmit 1 frame, $t_d = 2P_d$



The pad ensures transmission takes enough time so it's still being sent when the first bit reaches the destination. The frame needs to still be going out when the noise burst from another stations collision detection gets back to the sender.

Efficiency of 802.3 Ethernet at 10 Mbps with 512 Bit Contention Slot Time



4.4.2 Efficiency (η)

Assume there are 'n' stations, every channel transmits with a probability 'p'. Length of each contention slot is twice the propagation delay.

$$\eta = \frac{\text{use full time}}{\text{Total time}} = \frac{t_d}{k \times 2 \times p_d + p_d + t_d}$$

where, 'k' is the number of contention slots.

Probability of success = ${}^{n}C_1 \times p \times (1-p)^{n-1}$

Maximum value of probability of success is obtained by taking the derivative of above probability and equating to zero. We obtain the value of p as 1/n.

Maximum value of probability of success full transmission = ${}^{n}C_1 \times \frac{1}{n} \times \left(1-\frac{1}{n}\right)^{n-1}$. By applying limit to this probability ($n \to \infty$, where 'n' is the number of stations) the maximum value of probability of success is 1/e.

According to poisson distribution the number of times a station should transmit before getting first success is given as 1/(*Maximum probability of success*) = e

This value of 'e' specifies the number of contention slots

$$\eta = \frac{t_d}{c \times 2 \times p_d + p_d + t_d} = \frac{t_d}{e \times 2 \times p_d + p_d + t_d} = \frac{1}{1+6.44a}$$

where, $a = P_d / t_d$

Remember

- This expression 1/1+6.44*a* for efficiency of CSMA/CD is used when the number of stations is not given or assume to be infinite.
- As the distance, 'd' increases efficiency (h) decreases hence, this CSMA/CD protocol is suitable only for LAN and not WAN
- Increase in the frame size increases the efficiency (small frame size means more number of frames resulting in more overhead because of header). Hence, efficiency decreases with small frame size.

Point-to-Point Vs Shared Channel Communication In LANs

Point-to-point: Computer connected by communication channels that each connect exactly two computers with access to full channel bandwidth. Forms a mesh or point-to-point network. Allows flexibility in communication hardware, packet formats, etc.

- Provides security and privacy because communication channel is not shared.
- Number of channels grows as square of number of computers.
  For n computers: $(n^2 - n)/2$

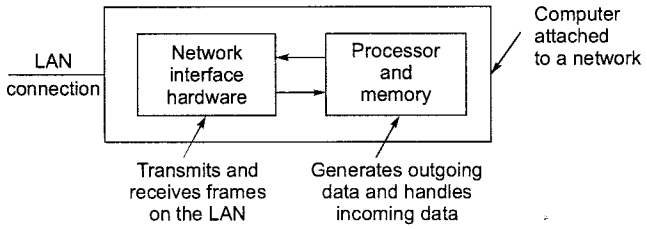
Shared or Broadcast Channel: All computers are connected to a shared broadcast-based communication channel and share the channel bandwidth. Security issues as result of broadcasting to all computers. Cost effective due to reduced number of channels and interface hardware components.

LAN Interface Hardware

LAN interface hardware or Network Interface Card (NIC), handles all details of frame transmission and reception:

- Adds hardware addresses, error detection codes, etc. to outgoing frames.
- May use DMA to copy frame data directly from main memory.
- Obeys access rules (e.g., CSMA/CD) when transmitting.
- Checks error detection codes on incoming frames.
- Checks destination address on incoming frames.

If destination address on incoming frame matches the local station's address, a copy of the frame is passed to the attached computer. Frames not addressed to the local computer are ignored and don't affect the local computer in any way.



4.5 Routing Algorithms

Routing is the act of moving information across an inter-network from a source to a destination (process of selecting best paths in a network).

The primary difference between bridging and routing is that bridging occurs at Layer 2 whereas routing occurs at Layer 3 and hence deal with different information to use in the process of moving information from source to destination.

Routers use routing protocols to decide optimal path between two hosts.

4.5.1 Routing Algorithm Metrics

Routing protocols use metrics to evaluate what path will be the best for a packet to travel. A *metric* (used by routing algorithms to determine the optimal path to a destination) is a standard of measurement. They are:

- Path length
- Delay
- Hop count
- Bandwidth
- Load
- Reliability

Path Length

Network administrators assign arbitrary costs to each network link (path length is the sum of the costs associated with each link traversed).

Routing Delay

Refers to the length of time required to move a packet from source to destination through the internet. Delay depends on bandwidth, queuing delay at each router, network congestion, and the physical distance.

Hop Count

A metric that specifies the number of passes through internetworking products, such as routers, that a packet must pass through in a route from a source to a destination.

Bandwidth (maximum attainable throughput on a link)

Refers to the data rate that the network path or network link can transfer. Greater bandwidth does not necessarily provide better routes. For example, if a faster link is busier, the actual time required to send a packet to the destination could be greater.

Load

Load refers to the degree to which a network resource, such as a router, is busy. Load can be calculated in a variety of ways, including CPU utilization and packets processed per second.

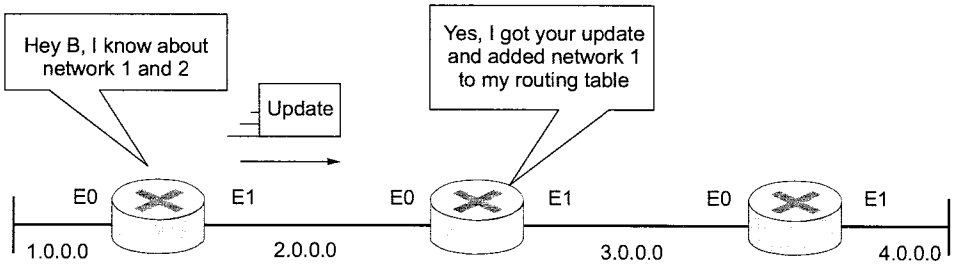
Reliability

Reliability, in the context of routing algorithms, refers to the dependability (usually described in terms of the *bit-error rate*) of each network link.

4.5.2 Routing Table

All routers stores information about other routers and their respective distances or paths to follow in large table called **Routing Table**. To share information among routers, frequently exchange update message. Every router may or may not have the exact path to other router in the network but atleast they know the interface through which they have to be forwarded. Routing protocols (algorithms) maintain Routing tables to store the routing information. When a router receives an incoming packet, it checks the destination address and attempts to associate this address with a next hop.

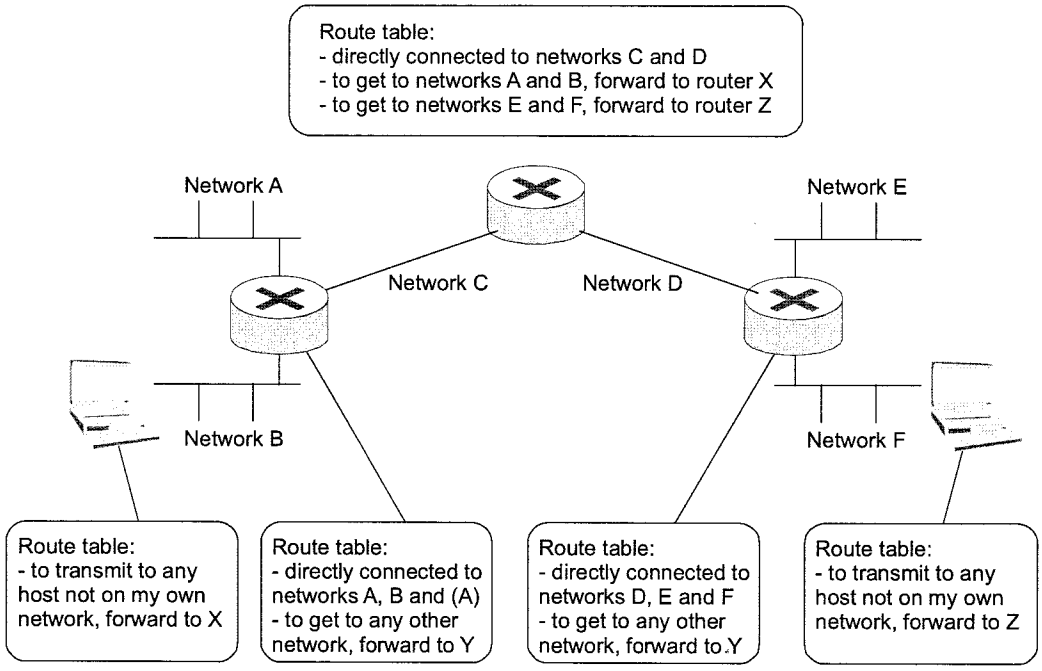
Routers communicate with one another and maintain their routing tables through the transmission of a variety of messages (see figure). By analyzing routing updates from all other routers, a router can build a detailed picture of network topology.



Routing table		
1.0.0.0	E0	0
2.0.0.0	E1	0

Routing table		
2.0.0.0	E0	0
3.0.0.0	E1	0
1.0.0.0	E0	0

Routing table		
3.0.0.0	E0	0
4.0.0.0	E1	0



The routing *update message* is one such message that generally consists of all or a portion of a routing table. By analyzing routing updates from all other routers, a router can build a detailed picture of network topology.

4.5.3 Classification of Routing Algorithms

Static Vs Adaptive

This category is based on how and when the routing tables are set-up and how they can be modified, if at all.

Static	Dynamic
• Static is also known as non-adaptive algorithms • Routing table mappings are established by the network administrator before the beginning of routing • Static routes are simple to design and work well in environments where network traffic is relatively predictable	• Dynamic routing is also referred as Adaptive routing • Routing table adjust to changing network circumstances by analyzing incoming routing update messages

Single-Path Vs Multi-path

This division is based upon the number of paths a router stores for a single destination.

Single-Path	Multi-path
• Single path algorithms are where only a single path (or rather single next hop) is stored in the routing table • Single path algorithms permit traffic multiplexing over multiple lines • Always less reliable than multi-path	• Some sophisticated routing protocols (multi-path algorithms) support multiple paths to the same destination. • Multipath algorithms permit traffic multiplexing over multiple lines • They can provide substantially better throughput and reliability. This is generally called load sharing

Intradomain Vs Interdomain

Intradomain	Interdomain
• Routing algorithms work only within domains.(Interior gateway protocols). • Intradomain routers need to know only about other routers within their domain, their routing algorithms can be simplified. • **Example:** 1. RIP (resource information protocol) which is Distance vector protocol 2. OSPF (open shortest path first) which is a link state protocol	• Work within and between domains (Exterior-gateway protocols) • **Example:** BGP (Border gateway protocol) used to connect two or more Autonomous Systems (AS)

Flat Vs Hierarchical

Flat	Hierarchial
In a flat routing system, the routers are peers of all others	Packets from non-backbone routers travel to the backbone routers, where they are sent through the backbone until they reach the general area of the destination (autonomous system). At this point, they travel from the last backbone router through one or more non-backbone routers to the final destination hierarchical systems, some routers in a domain can communicate with routers in other domains, while others can communicate only with routers within their domain

Link-State Vs Distance Vector

This category is based on the way the routing tables are updated.

Link-State	Distance Vector
• *Link-state algorithms* (also known as shortest path first algorithms) - OSPF • Share the knowledge *only about their neighbors* • Shared with *all the routers in the internet*, by sending small updates using flooding • Information *sharing only when there is a change*, which leads to lesser internet traffic compared to distance vector routing • Convergence takes place more quickly in link-state algorithms and hence are less prone to routing loops than distance vector algorithms	• *Distance vector algorithms* (also known as Bellman-Ford algorithms) - RIP • The routers share the knowledge *about the entire autonomous system* • Sharing of information takes place *only with the neighbors* • Information *sharing at fixed regular intervals*, say every 30 seconds and hence lot of traffic. • Convergence takes long time in distance vector algorithms and high possibility of COUNT to INFINITY problem (infinite loop).

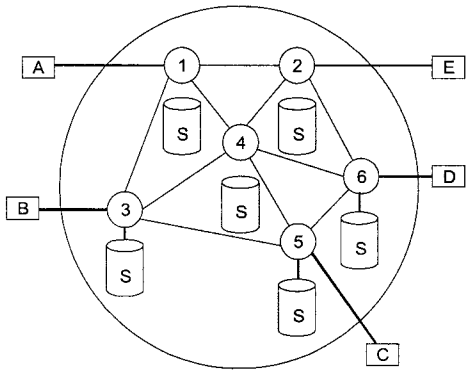
Host-Intelligent Vs Router-Intelligent

This division is on the basis of whether the source knows about the entire route or just about the next-hop where to forward the packet.

Host-Intelligent	Router-Intelligent
• Source end node will determine the entire route (source routing or host-intelligent routing) • Routers merely act as store-and-forward devices, mindlessly sending the packet to the next stop	• Source end hosts knows nothing about routes • Routers determine the path based on their own strategy

4.5.4 Static Routing

In fixed routing a route is selected for each source-destination pair of nodes in the network. The routes are fixed; they may only change if there is a change in the topology of the network. A central routing matrix is created based on least-cost path, which is stored at a network control center. The matrix shows, for each source-destination pair of nodes, the identity of the next node on the route.



Central Routing Directory

From node \ To node	1	2	3	4	5	6
1	-	2	3	2	2	2
2	1	-	1	6	6	6
3	1	1	-	4	5	1
4	6	6	3	-	5	6
5	4	4	3	4	-	4
6	2	2	2	4	4	-

Figure: ***(a)*** *A simple packet switching network with six nodes (routers),* ***(b)*** *The central routing table created based on least-cost path*

Node 1 Directory

Destination	Next Node
2	2
3	3
4	2
5	2
6	2

Node 2 Directory

Destination	Next Node
1	1
3	1
4	6
5	6
6	6

Node 3 Directory

Destination	Next Node
1	1
2	1
4	4
5	5
6	1

Node 4 Directory

Destination	Next Node
1	6
2	6
3	3
5	5
6	6

Node 5 Directory

Destination	Next Node
1	4
2	4
3	3
4	4
6	4

Node 6 Directory

Destination	Next Node
1	2
2	2
3	2
4	4
5	4

Figures: *Routing tables that can be stored in different nodes of the network*

4.5.5 Flooding

Every incoming packet to a node is sent out on every outgoing line except the one it arrived on (no network information required). At least one packet will pass through the shortest route, as all nodes, directly or indirectly connected, are visited.

Generates vast number of duplicate packets (limitation). To overcome this limitation we use *hop-count* (contained in the packet header), which is decremented at each hop, and discarded when the counter becomes zero. Another approach is keep track of packets, which are responsible for flooding using a *sequence number* and avoid sending them out a second time (selective *flooding).*

The routers do not send every incoming packet out on every line, only on those lines that go in approximately in the direction of destination. Some of the important utilities of flooding are:

- Flooding is highly robust, and could be used to send emergency messages (e.g., military applications).
- It may be used to initially set up the route in a virtual circuit.
- Flooding always chooses the shortest path, since it explores every possible path in parallel.
- Can be useful for the dissemination of important information to all nodes (e.g., routing information).

4.5.6 Autonomous Systems

An Autonomous System (AS) is a collection of routers whose prefixes and routing policies are under *common administrative control.*

An AS is identified by an Autonomous System number. AS is a connected segment of a network topology that consists of a collection of subnetworks (with hosts attached) interconnected by a set of routes and these ASs share a *common routing strategy.*

An AS has a single *"interior" routing protocol and policy.* Internal routing information is shared among routers within the AS, but not with systems outside the AS. However, an AS announces the network addresses of its internal networks to other ASs that it is linked to.

4.5.7 Border Gateway Protocols

An autonomous system shares routing information with other autonomous systems using the *Border Gateway Protocol* (BGP).

Previously, the Exterior Gateway Protocol (EGP) was used. When two routers exchange network reach ability information, the message carry the AS identifier (AS number) that router represents.

4.5.8 Interior Gateway Protocols

An Interior Gateway Protocol (IGP) is a type of protocol used for exchanging routing information between gateways (commonly routers) within an Autonomous System (for example, a system of corporate local area networks). This routing information can then be used to route network-level protocols like IP. Two Interior gateway protocols are: (1) Routing Information Protocol (RIP) and (2) Open Shortest path first (OSPF).

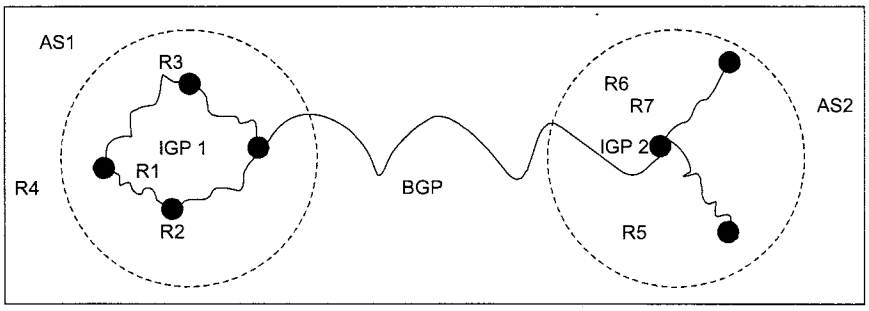


Figure: *Two AS, each of which are using different IGPs internally and one BGP to communicate between each other*

Figure shows a conceptual view of two Autonomous systems (AS1 and AS2), each of which is using a different Interior gateway protocol (IGP1 and IGP2) as a routing protocol internally to the respective AS, while one router from each of the autonomous systems (R1 and R7) communicate among themselves to exchange the

information of their respective Autonomous systems using a Border Gateway protocol, BGP. These two routers (R1 and R7) understands both interior and border gateway protocols.

4.6 RIP – Routing Information Protocol

The **Routing Information Protocol (RIP)** helps a router dynamically adapt to changes of network connections by communicating information about networks *each router can reach* and how far away those networks are. It is a *distance-vector protocol*, which employs **Hop Count** (number of routers the datagram passes though) as the metric (router defined to be one hop from directly connected networks, two hops from networks that are reachable from one other router and so on).

The maximum number of hops allowed with RIP is 15. It runs above Network layer of the Internet protocol suite, using **UDP port 520** to carry its data. RIP uses a distributed version of **Bellman-Ford algorithm.** *Bellman-Ford algorithm* computes single-source shortest paths in a weighted graph (where some of the edge weights may be negative). Bellman Ford runs in O (VE) time, where V and E are the number of vertices and edges.

The algorithm is distributed because each node calculates the distances between itself and all other nodes within the AS and stores this information as a table. Each node *sends its table to all neighboring nodes.*

When a node receives distance tables from its neighbors, *it calculates the shortest routes to all other nodes* and updates its own table to reflect any changes.

Disadvantages of Bellman-Ford algorithm in this setting are:

- Does not scale well
- Changes in network topology are not reflected quickly since updates are spread node-by-node.
- Counting to infinity

RIP partitions the participants (nodes within the AS) into *Active (Routers)* and *Passive* (other than routers) nodes.

Active routers advertise their routes (broadcasts a message or advertisement every 30 seconds) to others while **passive nodes** just listen (do not advertise) and updates their routes based on the advertisements.

Each message consists of pairs, where each pair contains a IP network address and a integer distance to that network. All active and passive nodes listen to the advertisements and updates their route tables.

Destination Address	Hop Count	Next Router	Other Information
115.2.1.00	4	132.35.27.1	
126.3.56.6	5	176.21.11.3	
165.11.12.3	7	173.23.12.5	
188.22.33.2	6	130.22.34.7	
195.23.12.8	3	201.23.11.5	

A distance Vector Routing Table

Lets discuss an example for better understanding. Consider the Autonomous system consisting of 4 routers (R1, R2, R3, R4) shown in Figure.

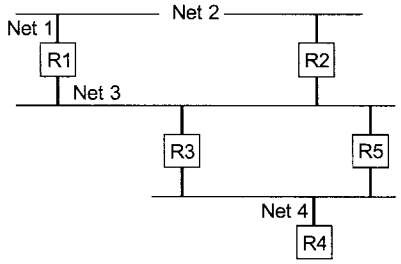


Figure: *Example of an autonomous system*

R2 will broadcast a message on network 3 (Net 3) containing a pair (2, 1), meaning that it can reach network 2 at a cost of 1.

Router R1 and R3 will receive this broadcast and install a route for network 2 (Net 2) in their respective routing tables, through R2 (at a cost of 2, as now there are two routers in between either (R1 or R2) or (R2 and R3). Later on Router R3 will broadcast a message with pair (2, 2) on network 4 (Net 4). Eventually all router will have a entry for Network 2 (Net 2) in their routing tables, and same is the case with the routes for other networks too.

NOTE: RIP specifies that once a router learns a route from another router, it must keep that route until it learns a better one.

In our example, if router R3 and R5 both advertise network 2 (Net 2) or network 1 (Net 1) at cost of 2; router R2 will install a route through the one that happens to advertise first. Hence, to prevent routes from oscillating between two or more equal cost paths, RIP specifies that existing routes should be retained until a new route has strictly lower cost.

```
Distance_Vector_Routing ( )
{
   //Initialize (create initial vectors for the mode)
   D[myself] = 0
   for (y = 1 to N)
   {
      if (y is a neighbor)
         D[y] = c[myself] [y]
      else
         D[y] = ∞
   }
   send vector {D[1], D[2],...D[3]} to all neighbors
      // Update (improve the vector with the vector received from a neghbour)
   repeat (forever)
   {
   wait (for a vector D_W from a neighbor w or any change in the link)
      for (y = 1 to N)
      {
         D[y] = min [D[y], (c[myself][w] + D_W [y])]
         // Bellman-Ford equation
      }
   if (any change in the vector)
      send vector {D[1], D[2],...,D[N]} to all neghbors
   }
} // End of Distance Vector
```

4.6.1 Routing Table Format

As RIP is a *distance vector routing protocol*, it represents the routing information in terms of the cost of reaching the specific destination.

Circuit priorities are represented using numbers between 1 and 15. This scale establishes the order of use of links. The router decides the path to use base on the priority list.

Once the priorities are established, the information is stored in a RIP routing table. Each entry in a RIP routing table provides a variety of information, including the ultimate destination, the next hop on the way to that destination, and a metric.

The metric indicates the distance in number of hops to the destination. RIP maintains only the best route to a destination thus whenever new information provides a better route, it would replaces the old route information. When network topology changes occur, they are reflected in routing update messages.

4.6.2 RIP Timers

Like other routing protocols, RIP uses certain timers to regulate its performance.

The routing-update timer clocks the interval (30 sec) between periodic routing updates, each router periodically transmits it entire routing table to all the other routers on the network.

Route Invalid Timer (or *route-timeout timer*), which determines how much time must expire without a router having heard about a particular route before that route is considered invalid. Each routing table entry has a route-timeout timer associated with it.

This notification of invalid route must occur prior to expiration of the *route flush timer*. When the route flush timer expires, the route is removed from the routing table. Typical initial value for route flush timer is 270 seconds.

4.6.3 Count to Infinity Problem

When a link is down, neighbor routers will detect it and attempt to broadcast route changes after they have calculated the new routes.

This triggered route updates may not arrive at certain network devices and those devices may broadcast a regular update message stating that the route that has gone down is still good to devices that has just been notified of the network failure. As such, the latter devices contains incorrect routing information which they may potentially further advertise.

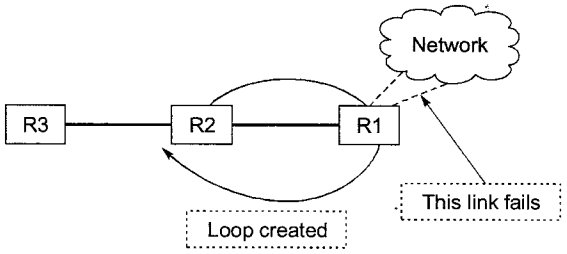


Figure: *Count to Infinity Problem*

If Router 1's link to network A fails, R1 will update its routing table immediately to make the distance 16 (infinite). In the next broadcast, R1 will report the higher cost route. Now suppose r2 advertise a route to Network A via R1 in its normal advertisement message, just after R1's connection to network A fails.

If so R1 will receive this update message and sees that Router 2 has a two-hop link (which is actually via Router 1) to Network A, according to the normal vector-distance algorithm it will install a new route to network A via R2, of length 3.

After this, it would began advertising it has a three-hop link to Network A and then route all traffic to Network A through R2. This would create a routing loop, since when Router 2 (R2) sees that Router 1 get to Network A in three hops, it alters it own routing table entry to show it has a four-hop path to Network A.

This is known as *Count-to Infinity problem*, i.e. *bad news* travel slowly through the network and to advertise a bad news throughout the entire network will take a long time. This problem is also called as *slow convergence problem.*

4.6.4 Solution to Slow Convergence Problem

Hold-Down

This techniques of *Hold down* tell routers to hold on to any changes that might affect recently removed routes for a certain period of time, greater than the period of time necessary to update the entire network with a route change. Let us examine this with an example, say initially all Routers (R1, R2 and R3) knows about a route to network A through Router 1 (R1). Now if the Router 1 (R1) link for network A goes down, and say the link failure message from Router 1 (R1) reaches Router 2 (R2) but not yet reached the Router 3 (R3).

At this point Router 2 (R2) has no entry in its table for a route to network A. Now if a regular update message from Router 3 (R3), about the reach ability information for network A, i.e. the out-dated information, reaches Router 2 (R2). Then Router 2 (R2) will think as if the route to Network A is Up and working, so both the routers- R3, R2 will have wrong information about the network.

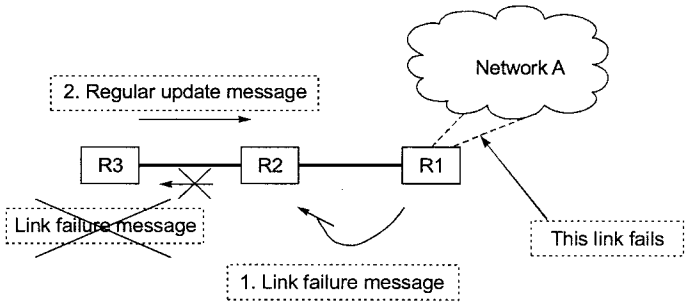


Figure: *Hold down, solution to Slow Convergence problem*

As per our example, it means that once R2 has removed the route to Network A, after receiving a link failure message from R1, It will not change or add any new route to network A, until a certain amount of time (***Hold down time***.) has passed. Typically hold down time is **around 60 sec** (to ensure that all machines receive the link failure news and not mistakenly accepts a message that is out dated).

Split Horizons

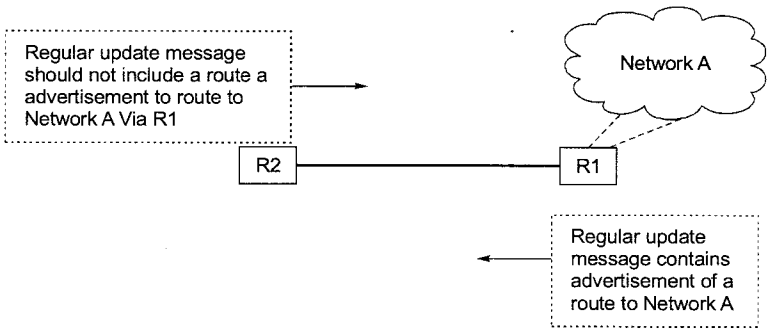


Figure: *Split Horizon, solution to Slow Convergence problem*

It is never useful to send information about a route back in the direction from which it came and thus split horizons is used ***to prevent updates that are redundant*** to the network.

For this purpose Router records the interface over which it received a particular route and does not propagates its information about that route back to the same interface.

Let us consider an example in which Router 1 advertises that it has a route to Network A. If Router 2 is sending traffic to Network A via Router 1, there is no reason for Router 2 to include the route info in its update back to Router 1, because Router 1 is closer to Network A.

Without split horizon rule in place, Router 2 would continue to inform Router 1 that it can actually get to Network A through 2 hops which is via Router 1.

If there is a failed direct connection to Network A, Router 1 may direct traffic to Router 2 thinking it's an alternative route to Network A and thus causing a routing loop. Split horizon in this instance serve as an additional algorithm to achieve stability.

Poison Reverse Updates

Larger routing loops prevented using poison reverse updates.

Once a connection disappears, the router advertising the connection retains the entry for several update periods, and ***include an infinite cost in the broadcast.*** The updates are sent to remove downed route and place it in hold-down.

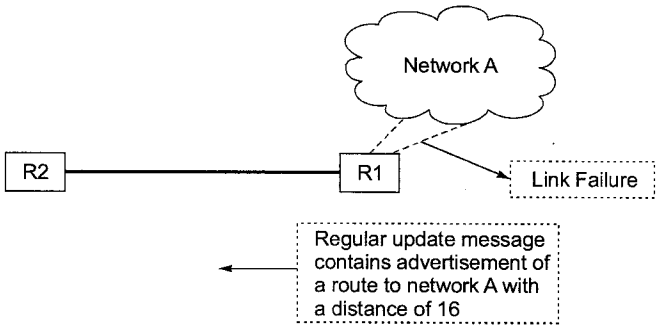


Figure: *Poison Reverse, other solution to Slow Convergence problem*

To make Poison reverse more efficient, it must be combined with *Triggered Updates*. Triggered updates force a router to send an immediate broadcast when receiving bad news, instead of waiting for the next periodic broadcast. By sending an update immediately, a router minimizes the time it is vulnerable to believing in good news.

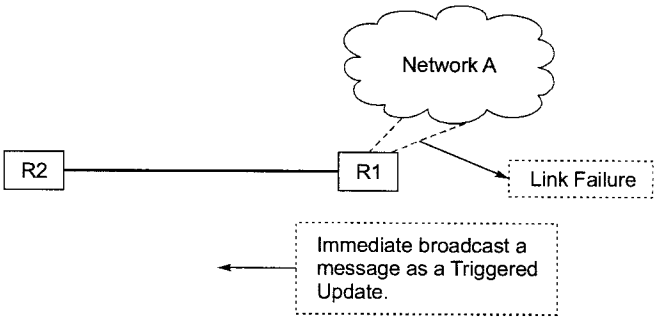
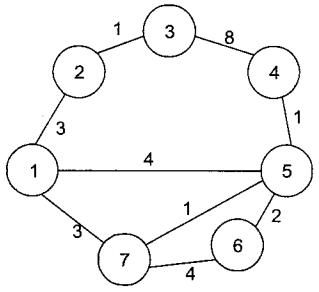


Figure: *Poison Reverse along with triggered Update*

Example-4.1 Consider a network topology as shown in the picture, and a synchronous version of distance vector algorithm (in one iterative step all the nodes compute their distance tables at the same time and then exchange them). Suppose that at each iteration, a node exchanges its minimum cost with its neighbours and receives their minimum cost. Assuming that the algorithm begins with each node knowing only the cost to its immediate neighbours, what is the maximum number of iterations required until the distributed algorithm converges?



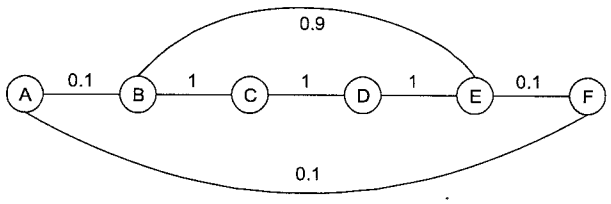
Solution:

Convergence time = length of the longest HOP-path (without loops) between any two nodes in the network.

Max distance from 1 : 2
Max distance from 2 : 3
Max distance from 3 : 3
Max distance from 4 : 2
Max distance from 5 : 2
Max distance from 6 : 3
Max distance from 7 : 3

Hence, we can expect that the algorithm converges after 3 iterations.

Example-4.2 In this problem you will be asked to compute distance vector(s) using the Bellman Ford algorithm for the network below:

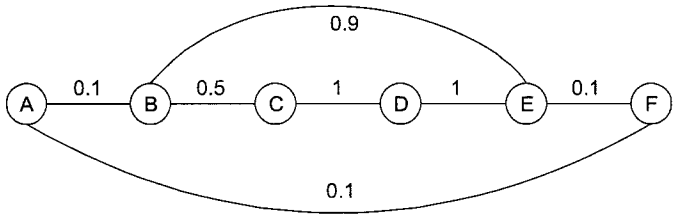


Assume that time is slotted (t = 1, 2, 3,...) and that a node sends its distance vector estimates to its neighbors at the beginning of each slot. A distance vector estimate sent at the beginning of a slot arrives at the end of that slot. All distance estimates are computed using the most recently available estimates. For the above stated problem, what are node A's distance vectors at the beginning of the time slots 1, 2, 3, and 4?

Solution:

	A	B	C	D	E	F
1	0	0.1	1	∞	0.9	∞
2	0	0.1	1	1.9	0.9	0.2
3	0	0.1	1	1.9	0.3	0.2
4	0	0.1	1	1.3	0.3	0.2

Example - 4.3 This time assume that the cost/weight of link AC is 0.5, while the cost/weight of link BF is 1. How many iterations are required, in this case, for A's distance vectors to converge?



Solution

	A	B	C	D	E	F
1	0	0.1	0.5	∞	0.9	∞
2	0	0.1	1	1.5	0.9	1

4.7 Open Shortest Path First (OSPF)

It is a routing protocol developed for Internet Protocol (IP) networks by the Interior Gateway Protocol (IGP). The Routing Information Protocol (RIP) was increasingly incapable of serving large, heterogeneous internetworks. OSPF being a SPF algorithm scales better than RIP. Few of the important features of OSPF are as follows:

- This protocol is *open (*means that anyone can implement it without paying license fees*).*
- OSPF is based on the SPF algorithm ***(Dijkstra's algorithm)***.
- OSPF is a *link-state routing protocol (*exchanges between routers must be *authenticated).*
- It allows a variety of authentication schemes, even different areas can choose different authentication schemes. The idea behind authentication is that only authorized router are allowed to advertise routing information.
- OSPF include Type of service Routing. It can calculate separate routes for each *Type of Service (TOS),* for example it can maintain separate routes to a single destination based on hop-count and high throughput.
- OSPF provides ***Load Balancing.*** When several equal-cost routes to a destination exist, traffic is distributed equally among them.
- OSPF allows supports host-specific routes, Subnet-specific routes and also network-specific routes.
- OSPF allows sets of networks to be grouped together. Such a grouping is called an *Area.* Each Area is self-contained; the topology of an area is hidden from the rest of the Autonomous System and from other Areas too. This information hiding enables a significant reduction in routing traffic.
- OSPF uses different message formats to distinguish the information acquired from within the network (internal sources) with that which is acquired from a router outside (external sources).

4.7.1 Link-State Algorithm

Just like any other Link state routing, OSPF also has the following features:

- **Advertise about neighborhood:** Instead of sending its entire routing table, a router sends information about its neighborhood only.
- **Flooding:** Each router sends this information to every other router on the internetwork, not just to its neighbors. It does so by a process of flooding.
- **Active response:** Each outer sends out information about the neighbor when there is a change.

Initialization

When an SPF router is powered up, it initializes its routing-protocol data structures and then waits for indications from lower-layer protocols that its interfaces are functional. The router sends hello packets to its neighbors and receives their hello packets. These messages are also known as greeting messages.

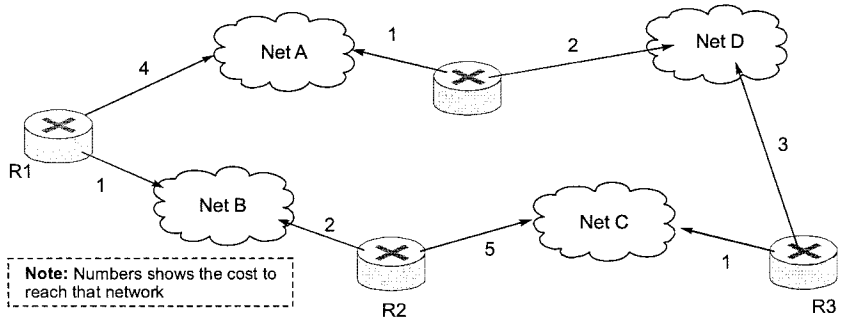


Figure: *An example Internet*

It then prepares an LSP (Link State packet) based on the results of this Hello protocol. An example of an internet is shown in above Figure, where R1 is a neighbor of R2 and R4, R2 is a neighbor of R1, R3 and R4, R3 is a neighbor of R2 and R4, R4 is a neighbor of R1, R2 and R3. So each router will send greeting messages to its entire neighbors.

Information from Neighbors

If neighbor responds to the greeting message as expected, it is assumed to be alive and functioning. If it does not respond the sending router then alerts the rest of the network in its next LSP, about this neighbor being down.

Link State Packet

An LSP usually contains 4 fields: the ID of the advertiser (Identifier of the router which advertises the message), ID of the destination network, The cost, and the ID of the neighbor router.

Link State Database

Every router receives every LSP and then prepares a database, which represents a complete network topology. This Database is known as Link State Database. Table shows the database of our sample internework. These databases are also known as *topological database.*

Advertiser	Network	Cost	Neighbor
R1	A	4	R4
R1	B	1	R2
R2	B	2	R1
R2	C	5	R3
R3	C	1	R2
R3	D	3	R4
R4	A	1	R1
R4	D	2	R3

Link State Database

Because every router receives the same LSPs, every router builds the same database. Every router uses it to calculate its routing table. If a router is added or deleted from the system, the whole database must be changed accordingly in all routers.

Shortest Path Calculation

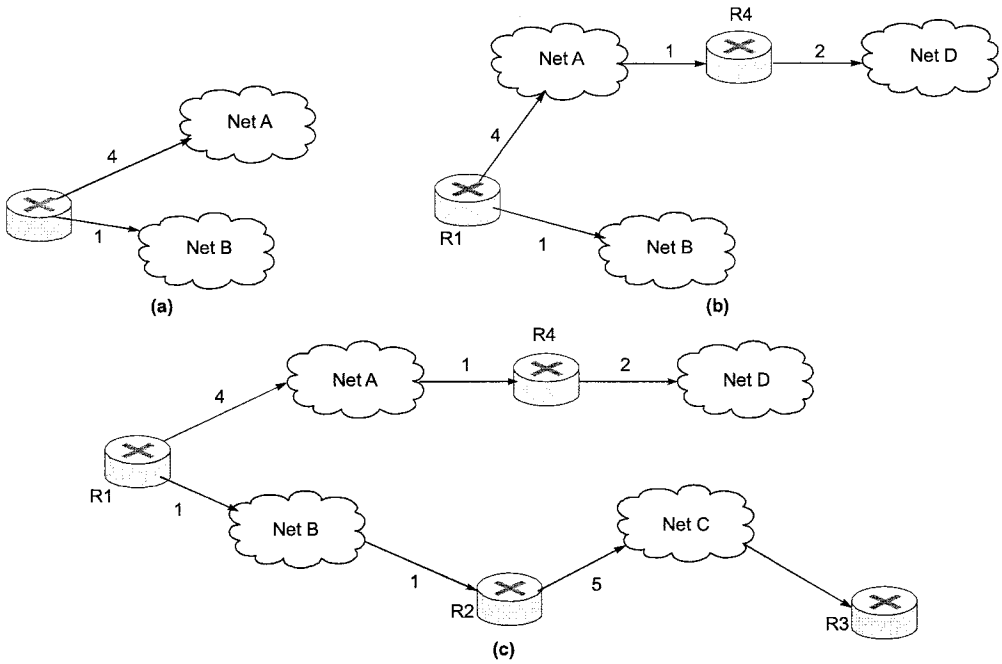


Figure: *Path calculation for router R1*

After gathering the Link State database, each router applies an algorithm called the *Dijkstra algorithm* to calculate the shortest distance between any two nodes. (See the Dijkstra algorithm shown below)

The Dijkstra's algorithm calculates the shortest path between two points on a network using a graph made up of nodes and arcs, where nodes are the Routers and the network, while connection between router and network is refer to as arcs.

The algorithm begins to build a tree by identifying its root as shown in Figure. The router is the root itself. The algorithm then attaches all other nodes that can be reached from that router; this is done with the help of the Link state database.

From this shortest path calculation each router makes its routing table, as per our example internet table for router R1 is given in Table. All other routers too have a similar routing table made up after this point.

Network	Cost	Next Router
A	4	-----
B	1	-----
C	8	R2
D	7	R4

Routing table Example

```
Dijkstra's Algorithm ( )
{  // Initialization
      Tree = {root} // Tree is made only of the root
   for (y = 1 to N) // N is the number of nodes
   {  if (y is the root)
         D[y] = 0 // D[y] is shortest distance from root to node y
      else if (y is a neighbor)
         D[y] = c[root] [y] //c[x] [y] is cost between nodes x and y in LSDB
```

```
        else
            D[y] = ∞
    } // Calculation
    repeat
    {   find a node w, with D[w] minimum among all nodes not in the Tree
        Tree = Tree È {w} // Add w to tree
        // Update distances for all neighbors of w
        for (every node x, which is a neighbor of w and not in the Tree)
        {
            D[x] = min {D[x], (D[w]+c[w][x]}
        }
    }  until (all nodes included in the Tree)
}   // End of Dijkstra
```

4.7.2 Routing Hierarchy in OSPF

Unlike RIP, OSPF can operate within a hierarchy, where the largest entity within the hierarchy is the autonomous system (AS). OSPF is an intra-AS (interior gateway) routing protocol (it can send receive information from other Ass). An AS can be divided into a number of *areas* (groups of contiguous networks and attached hosts).

Routers with multiple interfaces can participate in multiple areas. These routers, which are called *Area Border Routers*, maintain ***separate topological databases for each area***.

The topological database contains the collection of LSAs received from all routers in the same area. Routers within the same area share the same information; hence they have identical topological databases.

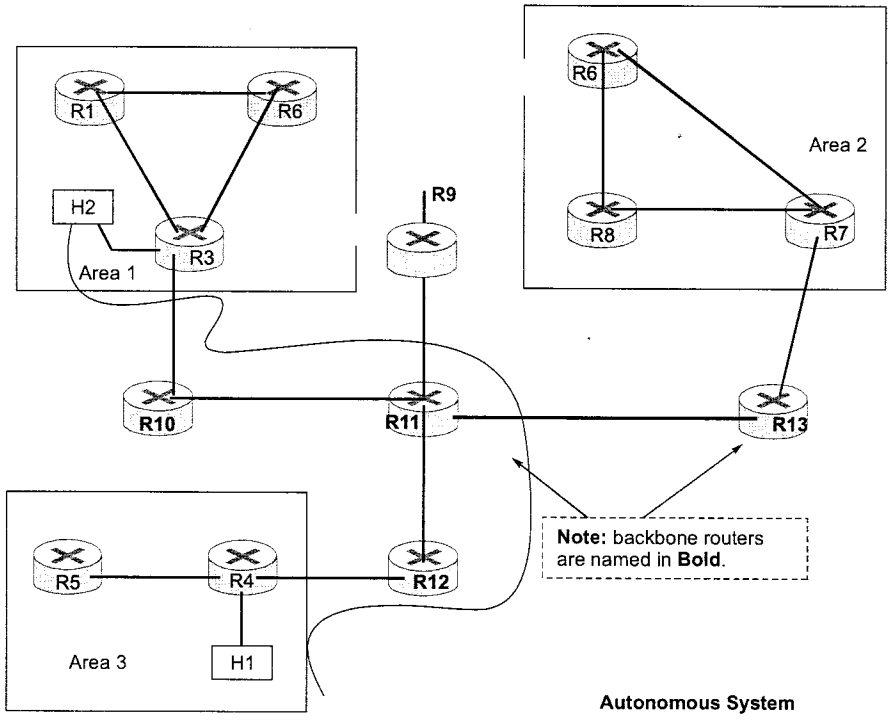


Figure: *Different areas in an Autonomous system*

The term *domain* (Autonomous System) sometimes is used to describe a portion of the network in which all routers have identical topological databases.

An area's topology is invisible to entities outside the area. By keeping area topologies separate, OSPF passes ***less routing traffic*** than it would if the AS were not partitioned.

An OSPF backbone is responsible for distributing routing information between areas. It consists of all Area Border Routers, networks not wholly contained in any area, and their attached routers. Figure shows an example of an internet with several areas. In the Figure, routers 9, 10, 11, 12 and 13 make up the backbone. If host H1 in Area 3 wants to send a packet to host H2 in Area 1, the packet is sent to Router 4, which then forwards the packet along the backbone to Area Border Router 12, which sends the packet to Router 11, and Router 11 forwards it to Router 10. Router 10 then sends the packet through an intra-area router (Router 3) to be forwarded to Host H2.

The backbone itself is an OSPF area, so all backbone routers use the same procedures and algorithms to maintain routing information within the backbone that any area router would. The backbone topology is invisible to all intra-area routers, as are individual area topologies to the backbone. Areas can be defined in such a way that the backbone is not contiguous. In this case, backbone connectivity must be restored through virtual links. Virtual links are configured between any backbone routers that share a link to a nonbackbone area and function as if they were direct links.

Example-4.4 The nodes participating in the Link State algorithm in one network are broadcasting the following link-state packets. Run the Link State (Dijkstra) algorithm to determine the shortest path from D to A.

Router B		Router C		Router D		Router E		Router F	
A	4	B	3	C	3	A	5	B	6
C	3	D	3	F	5	C	2	D	5
F	6	E	2			F	8	E	8

Solution:

Step	N	D(A), p(A)	D(B),p(B)	D(C), p(C)	D(E), p(C)	D(F), p(F)
0	D	∞	∞	3, D	∞	5, D
1	DC	∞	6, C		5, C	5, D
2	DCE	10, E	6, C			5, D
3	DCEF	10, E	6, C			
4	DCEFB	10, E				
5	DCEFBA					

Example-4.5 Consider the Link State Packets (LSPs) entering a router A. Link State Packets:

	Router A		Router B		Router C		Router D		Router E		Router F	
Links	C	1	A	2	A	1	B	5	A	3	C	8
Links	B	2	D	5	F	8	E	3	F	1	E	1
Links	E	3	-	-	-	-	F	1	D	3	D	1

Determine the shortest path A to D. Specify the shortest path between A and D and its respective cost. Using link state protocol.

Solution:

The shortest path between A and D: A-E-F-D. The cost of the shortest path between A and D = 5.

Step	Set S	B	C	D	E	F
0	{A}	2	1	∞	3	∞
1	{A, C}	2	-	∞	3	9
2	{A, C, B}	-	-	7	3	9
3	{A, C, B, E}	-	-	6	-	4
4	{A, C, B, E, F}	-	-	5	-	-
5	{A, C, B, E, F, D}					

4.8 Border Gateway Protocol (BGP)

The **Border Gateway Protocol (BGP)** is an inter-autonomous system routing protocol. BGP is used to exchange routing information for the Internet and is the protocol used between Internet service providers (ISP), which are different ASes.

One of the most important characteristics of BGP is its *flexibility.*

The protocol can connect together any internetwork of autonomous systems using an arbitrary topology. The only requirement is that each AS have at least one router that is able to run BGP and that this router connect to at least one other AS's BGP router. The primary function of a BGP speaking system is to exchange network reachability information with other BGP systems. This network reachability information includes information on the list of Autonomous Systems (ASs) that reachability information traverses. BGP constructs a graph of autonomous systems based on the information exchanged between BGP routers.

As far as BGP is concerned, whole Internet is a graph of ASs, with each AS identified by a Unique AS number. Connections between two ASs together form a path and the collection of path information forms a route to reach a specific destination. BGP uses the path information to ensure the loop-free inter- domain routing. Another important assumption that BGP makes is that it doesn't know anything about what happens within the AS. BGP only takes the information conveyed to it from the AS and shares it with other ASs.

4.8.1 BGP Characteristics

BGP is neither a pure distance vector protocol nor a pure link state protocol.

- **Inter-Autonomous System Configuration:** BGP's primary role is to provide communication between two autonomous systems.
- **Next-Hop paradigm:** Like RIP, BGP supplies next hop information for each destination.
- Coordination among multiple BGP speakers within the autonomous system.
- **Path information:** BGP advertisements also include path information, along with the reachable destination and next destination pair.
- **Policy support:** Unlike most of the distance-vector based routing, BGP can implement policies that can be configured by the administrator. For Example, a router running BGP can be configured to distinguish between the routes that are known from within the Autonomous system and that which are known from outside the autonomous system.
- **Runs over TCP:** BGP uses TCP for all communication. So TCP takes care of reliability issues.
- **Conserve network bandwidth:** BGP doesn't pass full information in each update message. Instead full information is just passed on once and thereafter successive messages only carries the incremental changes called **deltas**. By doing so a lot of network Bandwidth is saved.

- **Support for CIDR:** BGP supports CIDR (supports sending of network mass along with address).
- **Security:** BGP allows a receiver to authenticate messages, so that the identity of the sender can be verified.

4.8.2 BGP Functionality and Route Information Management

BGP peers perform three basic functions.

- The First function consists of initial peer acquisition and authentication. Both the peers establish a TCP connection and perform message exchange that guarantees both sides have agreed to communicate.
- The second function primarily focus on sending of negative or positive reachability information, this step is of major concern.
- The Third function provides ongoing verification that the peers and the network connection between them are functioning correctly. Every BGP speaker is responsible for managing route descriptions according to specific guidelines established in the BGP standards.

BGP Route Information Management Functions

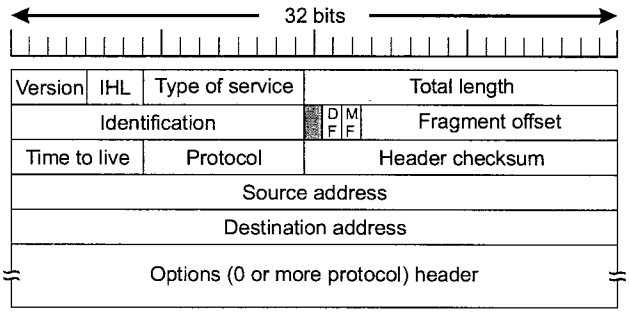
Conceptually, the overall activity of route information management can be considered to encompass four main tasks:

- **Route Storage:** Each BGP stores information about how to reach other networks.
- **Route Update:** Special techniques are applied to determine when and how to use the information received from peers to properly update the device's knowledge of routes.
- **Route Selection:** Each BGP uses the information in its route databases to select good routes to each network on the internetwork.
- **Route Advertisement:** Each BGP speaker regularly tells its peers what it knows about various networks and methods to reach them. This is called *route advertisement* and is accomplished using BGP *Update* messages.

4.9 Internet Protocol (IP)

The goal of IP is to interconnect networks of diverse technologies and create a single, virtual network to which all hosts connect. Hosts communicate with other hosts by handing datagrams to the IP layer;

- The sender doesn't worry about the details of how the networks are actually interconnected.
- IP provides unreliable, connectionless deliver service.
- IP defines a universal packet called an Internet Datagram.



The IP (Internet Protocol) header

Version number (4-bits)

- The current protocol version is 4.
- Including a version number allows a future version of IP be used along side the current version, facilitating migration to new protocols.

Header length (4-bits)

- Length of the datagram header (excluding data) is 32-bit words.
- The minimum length is 5 words = 20 bytes, but can be up to 15 words if options are used.
- In practice, the length field is used to locate the start of the data portion of the datagram.

Type-of-service (8-bits)

The field contains a three bit precedence field, three flags D, T, R and two unused field. A hint to the routing algorithms as to what type of service we desire.

- **Precedence (3-bits):** A priority indication, where 0 is the lowest and means normal service, while 7 is highest and is intended for network control messages (e.g., for interactive control).
- **Delay (1-bit):** An Application can request low delay service (e.g., for interactive use).
- **Throughput (1-bit):** Application requests high throughput.
- **Reliability (1-bit):** Application requests high reliability.

Total length (16-bits)

Total length of the IP datagram (in bytes), including data and header. The size of the data portion of the datagram is the total length minus the size of the header. Maximum length is 65, 535 bytes.

Identification (16-bits), Flags (3-bits), Fragment offset (13-bits)

- **These three fields are used for fragmentation and reassembly:** The Identification field is needed to allow the destination host to determine which datagram a newly arrived fragment belongs to. All the fragments of a datagram contain the same Identification value.
  DF stands for Don't fragment. It is an order to the routers not to fragment the datagram because the destination is incapable of putting the pieces back together again.
  MF stands for More Fragments. All fragments except the last one have this bit set. It is needed to know when all fragments of a datagram have arrived.
- The identification field uniquely identifies fragments of the same original datagram.
- Whenever a host sends a datagram, it sets the identification field of the outgoing datagram and increments its local identification counter.
- The offset field shows order of the fragments.
- When a gateway fragments a datagram, it sets the offset field of each fragment to reflect at what data offset with respect to the original datagram the current fragment belongs.
- Fragmentation occurs in 8-byte chunks, so the offset holds the "chunk number".
- We need to know when we've received all of the fragments. To help with this, the flags field may contain:
  (a) A **Don't Fragment** indication (set by host, honored by gateways). (A 1-bit flag.)
  (b) The **More Fragments** field indicates that another fragment follows this one. This fragment is not the last fragment of the original datagram.
- An unfragmented datagram has an offset of 0, and a More Fragment bit of 0.
- The last fragment bit is needed in order for the recipient host to determine when it has all fragments of a datagram.

Example:
Original Frame: HL = 5, length = 656, Frag Offset = 0, More = 0
Fragment 1: HL = 5, Length = 252, Frag Offset = 0, More = 1
Fragment 2: HL = 5, Length = 252, Frag Offset = 29, More = 1
Fragment 3: HL = 5, Length = 192, Frag Offset = 58, More = 0

Time-to-live (8-bits)

- A counter that is decremented by each geteway
- Should this hopcount reach 0, discard the datagram
- Originally, the time-to-live field was intended to reflect real time.
- In practice, it is now a hopcount.
- The time-to-live field squashes looping packets
- It also guarantees that packets don't stay in the network for longer than 255 seconds, a property needed by higher layer protocols that reuse sequence numbers.

Protocol (8-bits)

- What type of data the IP datagram carries (e.g. TCP, UDP, etc,)
- Needed by the receiving IP to know the higher level service that will next handle the data.

Header Checksum (16-bits)

- A checksum of the IP header (excluding data).
- The IP checksum is computed as follows:
  (a) Treat the data as a stream of 16-bit words (appending a 0 byte if needed).
  (b) Compute the 1's complement sum of the 16-bit words. Take the 1's complement of the computed sum.
- This checksum is much weaker than the CRCs we have studied.
- But, it has the property that the order in which the 16-bit words are summed is irrelevant.
- We can place the checksum in a fixed location in the header, set it to zero, compute the checksum, and store its value in the checksum field.
- On receipt of a datagram, the computed checksum calculated over the received packet should be zero.
- Check summing only the header reduces the processing time at each gateway, but forces transport layer protocols to perform error detection (if desired).
- The header must be recalculated at every router since the time_to_live field is decremented.

Source Address (32-bits)

Original sender's address. This is an IP address, not a MAC address.

Destination Address (32-bits)

Datagram's ultimate destination. The IP embedded datagram contains the source of the original sender (not the forwarding gateway) and the destination address of the ultimate destination.

IP Options

- IP datagram allow the inclusion of optional, varying length fields that need not appear in every datagram. We may sometimes want to send special information, but we don't want to dedicate a field in the packet header for this purpose.
- Options start with a 1-byte option code, followed by zero or more bytes of option data.

- The option code byte contains three parts:
  (a) **Copy flag (1 bit)**: If 1, replicate option in each fragment of a fragmented datagram. That is, this option should appear in every fragment as well. If 0, option need only appear in first fragment.
  (b) **Option class (2 bits):** Purpose of option:
  0 = network control
  1 = reserved
  2 = debugging and measurement
  3 = reserved

Option	Description
Security	Specifies how secret the datagram is
Strict source routing	Gives the complete path to be followed
Loose source routing	Gives a list of routers not to be missed
Record route	Makes each router append its IP address
Timestamp	Makes each router append its address and timestamp

4.9.1 Fragmentation

How to cross networks whose maximum transmission unit (MTU) is smaller than the packet being transmitted.

- **Connection-oriented internets** avoid this problem.
  (*i*) By selecting a maximum packet size at connection set up time.
  (*ii*) That maximum is just min (MTU_1, MTU_2, ...) of the MTUs in the intervening network.
  (*iii*) Once the connection is established, the path never changes, so that sender can select a packet size and never again worry that it will be too large.
- In **connectionless internets**, the appropriate packet size depends on the path used.
  (a) Thus, it can change at any time.
- **Approaches:**
  (*i*) Have router drop packets that are too large to send across a network and return an error message to the sender. The sending host could then retransmit the data in a smaller packet.
  (*ii*) Have router fragment large packets into several fragments, each small enough to traverse the network. There are two flavours called Transparent and non-transparent Fragmentation.
- **Transparent Fragmentation:** With transparent fragmentation, end hosts (sender and receiver) are unaware that fragmentation has taken place. A router fragments a packet, and the next-hop router on the same network reassembles the fragments back into the original packet.
  Drawbacks are:
  (*i*) All fragments must travel through to the same router. They must all be reassembled by the same next-hop router.
  (*ii*) Routers must be careful to avoid re-assembly lockup. (The deadlock problem discussed earlier, where a router has used up all of its buffer space to hold fragments and can no longer accept new ones.)
  (*iii*) Reassembling fragments uses precious router resources that could otherwise be used forwarding packets).
  (*iv*) May fragment/re-assemble several times along the route!

- **Non-Transparent Fragmentation:** As before, routers fragment packets when needed. Routers along the path do not reassemble. Destination hosts perform re-assembly (if needed).
  Drawbacks are:
  (*i*) Now every host must be prepared to do this job.
  (*ii*) Overhead of carrying along small segments lasts until destination.
- **Problems Associated with Fragmentation in General:**
  (*i*) Fragmenting increases waste; the sum of the bits of the individual fragments exceeds the number of bits in the original message.
  (*ii*) Loss of a single fragment requires an end-to-end retransmission; the loss of a single fragment has the same effect as losing the entire packet.
  (*iii*) More work to forward three small packets than one large one. The cost of forwarding packets includes a fixed per-packet cost, that includes doing the route lookup, fielding interrupts, etc.

***Example*-4.6** An IP datagram carrying 10000 bytes of data must be sent over a link (i.e. network) that has an MTU of 4468 bytes. Assume the datagram has no Options, and the Identification number is 218. How many fragments will be generated? State the values (in decimal numbers) of the following fields for each of the fragments: Identification, Total Length, D-bit, M-bit, Fragmentation Offset.

Solution:
The format of the IP header is shown on the subsequent page.
3 fragments
1st fragment : 4448 + 20 bytes
2nd fragment : 4448 + 20 bytes
3rd fragment : 1104 + 20 bytes

	First	Second	Third
Identification	218	215	218
Total length	4468	4468	1124
DNF	0	0	0
MF	1	1	0
Fragment offset	0	556	1112

4.10 Address Resolution Protocol

Address Resolution Protocol (ARP) is a **protocol** for mapping an Internet Protocol address **(IP address)** to a physical machine address that is recognized in the local network.

For *example*, in IP Version 4, the most common level of IP in use today, an address is 32 bits long. In an **Ethernet** local area network, however, addresses for attached devices are 48 bits long. (The physical machine address is also known as a Media Access Control or **MAC address**).

A table, usually called the ARP cache, is used to maintain a correlation between each MAC address and its corresponding IP address. ARP provides the protocol rules for making this correlation and providing address conversion in both directions.

How ARP Works

When an incoming packet destined for a host machine on a particular local area network arrives at a **gateway**, the **gateway** asks the ARP program to find a physical host or MAC address that matches the IP address.

The ARP program looks in the ARP cache and, if it finds the address, provides it so that the packet can be converted to the right packet length and format and sent to the machine.

If no entry is found for the IP address, ARP broadcasts a request packet in a special format to all the machines on the LAN to see if one machine knows that it has that IP address associated with it.

A machine that recognizes the IP address as its own returns a reply so indicating. ARP updates the ARP cache for future reference and then sends the packet to the MAC address that replied. Since protocol details differ for each type of local area network, there are separate ARP Requests for Comments (**RFC**) for Ethernet, **ATM**, Fiber Distributed-Data Interface, HIPPI, and other protocols.

4.11 Reverse ARP (RARP)

RARP (Reverse Address Resolution Protocol) is a **protocol** by which a physical machine in a local area network can request to **learn its IP address** from a gateway server's Address Resolution Protocol (ARP) table or cache. A network administrator creates a table in a local area network's gateway **router** that maps the physical machine (or Media Access Control - **MAC address**) addresses to corresponding Internet Protocol addresses.

When a new machine is set up, its RARP **client** program requests from the RARP **server** on the router to be sent its IP address. Assuming that an entry has been set up in the router table, the RARP server will return the IP address to the machine which can store it for future use. RARP is available for **Ethernet**, Fiber Distributed-Data Interface, and **token ring** LANs.

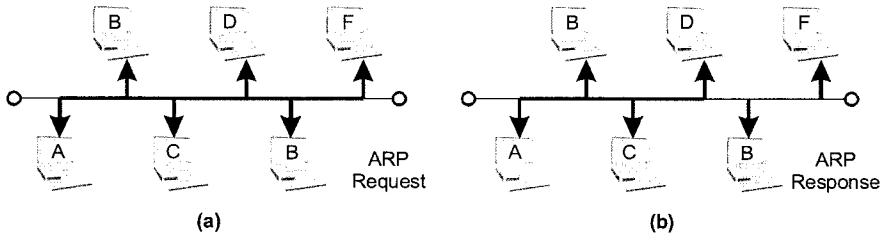


Figure: ***(a)*** *ARP request with a broadcast to all the stations and* ***(b)*** *ARP response is a unicast only to the requesting host*

The TCP/IP protocols include another related protocol known as reverse ARP, which can be used by a computer such as a diskless host to find out its own IP address. It involves the following steps:

- Diskless host A broadcasts a RARP request specifying itself as the target
- RARP server responds with the reply directly to host A
- Host A preserves the IP address in its main memory for future use until it reboots.
- The protocol that maps hardware addresses to Internet addresses is called Reverse ARP, or RARP.
- Necessary when a disk less machine first boots. It doesn't know its own IP address (and can't read it from a local disk). The booting client contacts a server to obtain its Internet address.
  - (*i*) The client communicates with a server by using a special protocol that requires only Ethernet frames. In essence it says "My ethernet address is 'aa.bb.cc.dd.ee.ff.'. Does anyone know my IP address?"
  - (*ii*) The broadcast goes to all nodes, including the RARP server. The RARP server maintains a database of physical address to Internet address mappings.
  - (*iii*) The actual format of RARP messages is similar to those of ARP.
    - (a) RARP uses two new message types; RARP request' and 'RARP response'.
    - (b) The remaining fields are the same as in ARP.
- We now see one of the primary benefits of broadcasting; locating servers.
- However, because broadcasting is resource intensive, (every machine on the local network must process the message, even if only to reject it) broadcasting should be used sparingly.
- It is used in diskless machines.

4.11.1 BOOTP

BOOTP is an application layer protocol. BOOTP message are encapsulated in a UDP packet, and the UDP packet itself is encapsulated in an IP packet.

Since a client send an IP datagram when does not knows its own IP address (the source address) or the server's IP address by simply uses all 0's as the source address and all 1's as the destination address.

The BOOTP request is broadcast because the client does not know the IP address of the server, so broadcast IP datagram cannot pass through any router. So, there is a need for an intermediary i.e., one of the hosts can be used as a relay (which is called a relay agent). The relay agent knows the unicast address of a BOOTP server. When it receives this kind of packet, it encapsulates the message in a unicast datagram and sends the request to the BOOTP server. The packet, carrying a unicast destination address, is routed by any router and reaches the BOOTP server. The BOOTP server knows the message comes from a relay agent because one of the fields in the request message defines the IP address of the relay agent. The relay agent, after receiving the reply, sends it to the BOOTP client.

BOOTP is not a dynamic configuration protocol. When a client requests its IP address, the BOOTP server consults a table (which has predetermined binding), that matches the existed physical address of the client with its IP address.

BOOTP is a static configuration protocol whereas for static and dynamic address allocation Dynamic Host Configuration (DHCP) has been used.

4.12 DHCP

The Dynamic Host Configuration Protocol (DHCP) service enables devices on a network to obtain IP addresses and other information from a DHCP server.

This service automates the assignment of IP addresses, subnet masks, gateway and other IP networking parameters. DHCP allows a host to obtain an IP address dynamically when it connects to the network.

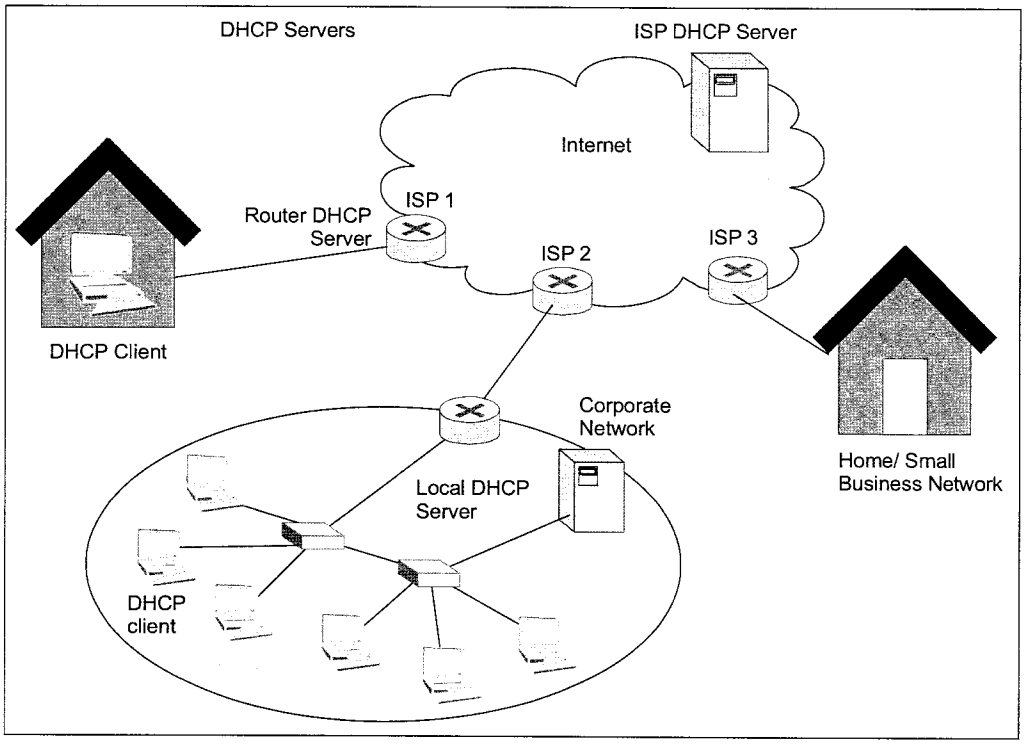
The DHCP server is contacted and an address requested. The DHCP server chooses an address from a configured range of addresses called a pool and assigns ("leases") it to the host for a set period. On larger local networks, or where the user population changes frequently, DHCP is preferred. New users may arrive with laptops and need a connection. Others have new workstations that need to be connected. Rather than have the network administrator assign IP addresses for each workstation, it is more efficient to have IP addresses assigned automatically using DHCP. DHCP distributed addresses are not permanently assigned to hosts but are only leased for a period of time.

If the host is powered down or taken off the network, the address is returned to the pool for reuse. This is especially helpful with mobile users that come and go on a network. Users can freely move from location to location and re-establish network connections. The host can obtain an IP address once the hardware connection is made, either via a wired or wireless LAN.

DHCP makes it possible for you to access the Internet using wireless hotspots at airports or coffee shops. As you enter the area, your laptop DHCP client contacts the local DHCP server via a wireless connection.

The DHCP server assigns an IP address to your laptop. As the figure shows, various types of devices can be DHCP servers when running DHCP service software. The DHCP server in most medium to large networks is usually a local dedicated PC-based server. With home networks the DHCP server is usually located at the ISP and a host on the home network receives its IP configuration directly from the ISP. DHCP can pose a security risk because any device connected to the network can receive an address. This risk makes physical security an important factor when determining whether to use dynamic or manual addressing. Dynamic and static addressing both has their places in network designs. Many networks use both DHCP and static addressing.

DHCP is used for general purpose hosts such as end user devices, and fixed addresses are used for network devices such as gateways, switches, servers and printers.



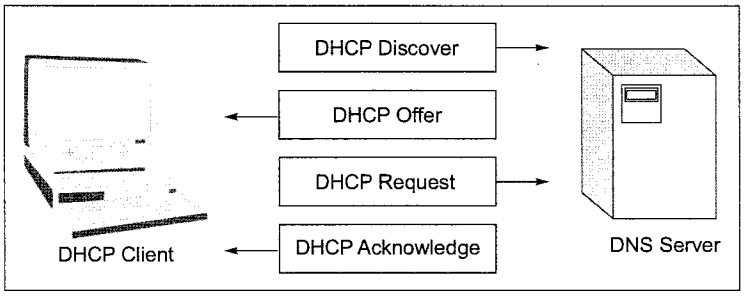
Without DHCP, users have to manually input the IP address, subnet mask and other network settings in order to join the network.

The DHCP server maintains a pool of IP addresses and leases an address to any DHCP-enabled client when the client is powered on. Because the IP addresses are dynamic (leased) rather than static (permanently assigned), addresses no longer in use are automatically returned to the pool for reallocation. When a DHCP-configured device boots up or connects to the network, the client broadcasts a DHCP DISCOVER packet to identify any available DHCP servers on the network. A DHCP server replies with a DHCP OFFER, which is a lease offer message with an assigned IP address, subnet mask, DNS server, and default gateway information as well as the duration of the lease.

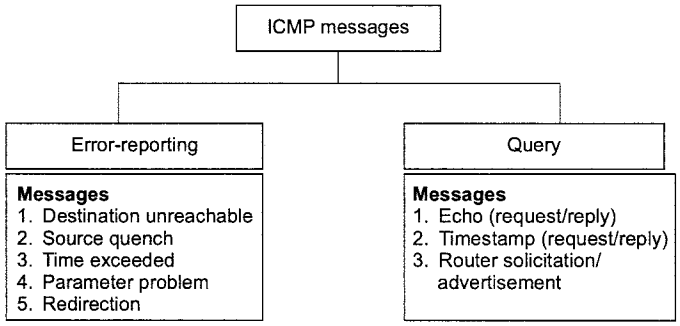
The client may receive multiple DHCP OFFER packets if there is more than one DHCP server on the local network, so it must choose between them, and broadcast a DHCP REQUEST packet that identifies the explicit server and lease offer that the client is accepting. A client may choose to request an address that it had previously been allocated by the server. Assuming that the IP address requested by the client, or offered by the server, is still valid, the server would return a DHCP ACK message that acknowledges to the client the lease is finalized.

If the offer is no longer valid - perhaps due to a time-out or another client allocating the lease - then the selected server will respond with a DHCP NAK message (Negative Acknowledgement). If a DHCP NAK message is returned, then the selection process must begin again with a new DHCP DISCOVER message being transmitted. Once the client has the lease, it must be renewed prior to the lease expiration through another DHCP REQUEST message. The DHCP server ensures that all IP addresses are unique (an IP address cannot be assigned to two different network devices simultaneously). Using DHCP enables network administrators to easily reconfigure client IP addresses without having to manually make changes to the clients.

Most Internet providers use DHCP to allocate addresses to their customers who do not require a static address. The fourth CCNA Exploration course will cover the operation of DHCP in greater detail.



4.13 ICMP



The IP provides unreliable and connectionless datagram delivery. It was designed to make efficient use of network resources.

- IP has no error-reporting or error correcting mechanism.
- IP has no mechanism for host and management queries
- ICMP has been designed to compensate for the above deficiencies.

To make efficient use of the network resources, IP was designed to provide unreliable and connectionless best-effort datagram delivery service. As a consequence, **IP has no error-control** mechanism and also **lacks mechanism for host and management queries.** A companion protocol known as *Internet Control Message Protocol* (ICMP), has been designed to compensate these two deficiencies. ICMP messages can be broadly divided into two broad categories: error reporting messages and query messages as follows.

- **Error reporting Messages:** Destination unreachable, Time exceeded, Source quench, Parameter problems, Redirect
- **Query:** Echo request and reply, Timestamp request and reply, Address mask request and reply.

NOTE

- No ICMP error message for a datagram carrying an ICMP error message.
- No ICMP error message for a fragmented datagram that is not the first fragment.
- No ICMP error message for a datagram having a multicast address.
- No ICMP error message for a datagram with a special address such as 127.0.0.0 or 0.0.0.0.

4.14 IPV6

Internet Protocol version 6 is a new addressing protocol designed to incorporate all the possible requirements of future Internet known to us as Internet version 2. This protocol as its predecessor IPv4, works on the Network Layer (Layer-3). Along with its offering of an enormous amount of logical address space, this protocol has ample features to address the shortcoming of IPv4.

Why New IP Version? So far, IPv4 has proven itself as a robust routable addressing protocol and has served us for decades on its best-effort-delivery mechanism. It was designed in the early 80s and did not get any major change afterward. At the time of its birth, Internet was limited only to a few universities for their research and to the Department of Defense. IPv4 is 32 bits long and offers around 4,294,967,296 (2^{32}) addresses. This address space was considered more than enough that time.

Given below are the major points that played a key role in the birth of IPv6:

- Internet has grown exponentially and the address space allowed by IPv4 is saturating. There is a requirement to have a protocol that can satisfy the needs of future Internet addresses that is expected to grow in an unexpected manner.
- IPv4 on its own does not provide any security features. Data has to be encrypted with some other security application before being sent on the Internet.
- Data prioritization in IPv4 is not up-to-date. Though IPv4 has a few bits reserved for Type of Service or Quality of Service, but they do not provide much functionality.
- IPv4 enabled clients can be configured manually or they need some address configuration mechanism. It does not have a mechanism to configure a device to have globally unique IP address.

4.14.1 Features

The successor of IPv4 is not designed to be backward compatible. Trying to keep the basic functionalities of IP addressing, IPv6 is redesigned entirely. It offers the following features:

Larger Address Space

In contrast to IPv4, IPv6 uses 4 times more bits to address a device on the Internet. This much of extra bits can provide different combinations of addresses other than IPv4. This address can accumulate the aggressive requirement of address allotment for almost everything in this world.

End-to-end Connectivity

Every system now has unique IP address and can traverse through the Internet without using NAT or other translating components. After IPv6 is fully implemented, every host can directly reach other hosts on the Internet, with some limitations involved like Firewall, organization policies, etc.

Auto-configuration

IPv6 supports both stateful and stateless auto-configuration mode of its host devices. So, the absence of a DHCP server does not create trouble.

Faster Forwarding/Routing

Simplified header puts all unnecessary information at the end of the header. The information contained in the first part of the header is adequate for a Router to take routing decisions, thus making routing decision as quickly as looking at the mandatory header.

IPSec

IPv6 must have IPSec security, making it more secure than IPv4.

No Broadcast

Though Ethernet/Token Ring are considered as broadcast network because they support Broadcasting, IPv6 does not have any broadcast support anymore. It uses multicast to communicate with multiple hosts.

Anycast Support

This is another characteristic of IPv6. IPv6 has introduced Anycast mode of packet routing. In this mode, multiple interfaces over the Internet are assigned same Anycast IP address. Routers, while routing, send the packet to the nearest destination.

Mobility

IPv6 was designed keeping mobility in mind. This feature enables hosts (such as mobile phone) to roam around in different geographical area and remain connected with the same IP address. The mobility feature of IPv6 takes advantage of auto IP configuration and Extension headers.

Enhanced Priority Support

IPv4 used 6 bits DSCP (Differential Service Code Point) and 2 bits ECN (Explicit Congestion Notification) to provide Quality of Service but it could only be used if the end-to-end devices support it, that is, the source and destination device and underlying network must support it whereas, in IPv6, Traffic class and Flow label are used to tell the underlying routers how to efficiently process the packet and route it.

Smooth Transition

Large IP address scheme in IPv6 enables to allocate devices with globally unique IP addresses. This mechanism saves IP addresses and NAT is not required. So devices can send/receive data among each other, for example, VoIP and/or any streaming media can be used much efficiently.

Other fact is, the header is less loaded, so routers can take forwarding decisions and forward them as quickly as they arrive.

Extensibility

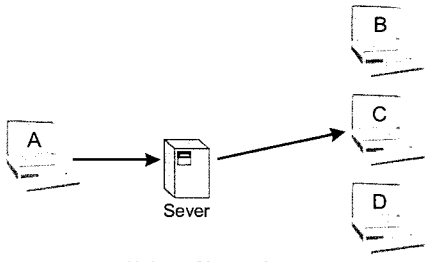
IPv4 provides only 40-bytes for options field, whereas options field in IPv6 can be as much as the size of IPv6 packet itself.

4.14.2 Addressing Modes

In computer networking, addressing mode refers to the mechanism of hosting an address on the network. IPv6 offers several types of modes by which a single host can be addressed. More than one host can be addressed at once or the host at the closest distance can be addressed.

Unicast

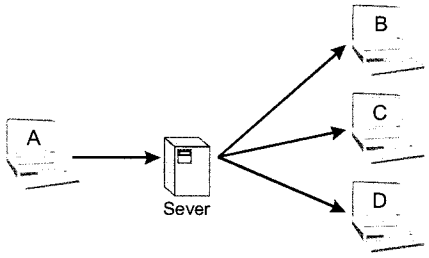
In unicast mode of addressing, an IPv6 interface (host) is uniquely identified in a network segment. The IPv6 packet contains both source and destination IP addresses. A host interface is equipped with an IP address which is unique in that network segment. When a network switch or a router receives a unicast IP packet, destined to a single host, it sends out one of its outgoing interface which connects to that particular host.



Unicast Messaging

Multicast

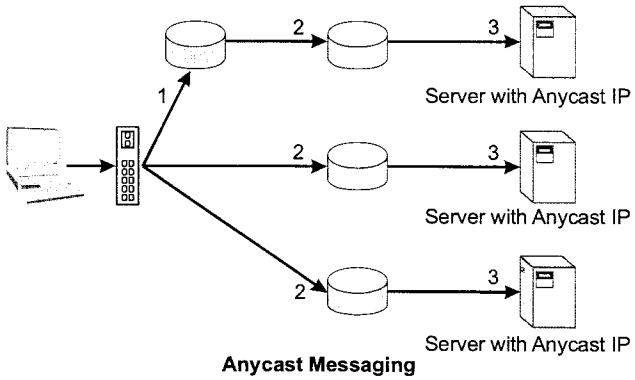
The IPv6 multicast mode is same as that of IPv4. The packet destined to multiple hosts is sent on a special multicast address. All the hosts interested in that multicast information need to join that multicast group first. All the interfaces that joined the group receive the multicast packet and process it, while other hosts not interested in multicast packets ignore the multicast information.



Multicast Messaging

Anycast

IPv6 has introduced a new type of addressing, which is called Anycast addressing. In this addressing mode, multiple interfaces (hosts) are assigned same Anycast IP address. When a host wishes to communicate with a host equipped with an Anycast IP address, it sends a Unicast message. With the help of complex routing mechanism, that Unicast message is delivered to the host closest to the Sender in terms of Routing cost.



Anycast Messaging

In the above picture, when a client computer tries to reach a server, the request is forwarded to the server with the lowest Routing Cost.

4.14.3 Address Types and Formats

Hexadecimal Number System

Before introducing IPv6 Address format, we shall look into Hexadecimal Number System. Hexadecimal is a positional number system that uses radix (base) of 16. To represent the values in readable format, this system uses 0-9 symbols to represent values from zero to nine and A-F to represent values from ten to fifteen. Every digit in Hexadecimal can represent values from 0 to 15.

Decimal	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3

4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Conversion Table

Address Structure

An IPv6 address is made of 128 bits divided into eight 16-bits blocks. Each block is then converted into 4-digit Hexadecimal numbers separated by colon symbols.

For example, given below is a 128-bit IPv6 address represented in binary format and divided into eight 16-bits blocks:

0010000000000001	0000000000000000	0011001000111000	1101111111100001
0000000001100011	0000000000000000	0000000000000000	1111111011111011

Each block is then converted into Hexadecimal and separated by ':' symbol:

2001:0000:3238:DFE1:0063:0000:0000:FEFB

Even after converting into Hexadecimal format, IPv6 address remains long. IPv6 provides some rules to shorten the address. The rules are as follows:

Rule 1: Discard leading Zero(es):

In Block 5, 0063, the leading two 0s can be omitted, such as (5th block):

2001:0000:3238:DFE1:63:0000:0000:FEFB

Rule 2: If two of more blocks contain consecutive zeroes, omit them all and replace with double colon sign ::, such as (6th and 7th block):

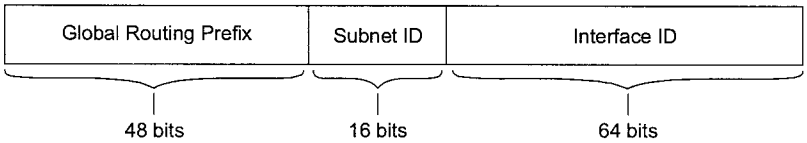
2001:0000:3238:DFE1:63::FEFB

Consecutive blocks of zeroes can be replaced only once by :: so if there are still blocks of zeroes in the address, they can be shrunk down to a single zero, such as (2nd block):

2001:0:3238:DFE1:63::FEFB

4.14.4 Global Unicast Address

This address type is equivalent to IPv4's public address. Global Unicast addresses in IPv6 are globally identifiable and uniquely addressable.



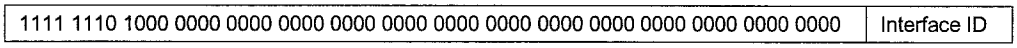
Global Unicast Address

Global Routing Prefix

The most significant 48-bits are designated as Global Routing Prefix which is assigned to specific autonomous systems. The three most significant bits of Global Routing Prefix is always set to 001.

4.14.5 Link-Local Address

Auto-configured IPv6 address is known as Link-Local Address. This address always starts with FE80. The first 16 bits of link-local address is always set to 1111 1110 1000 0000 (FE80). The next 48-bits are set to 0, thus:

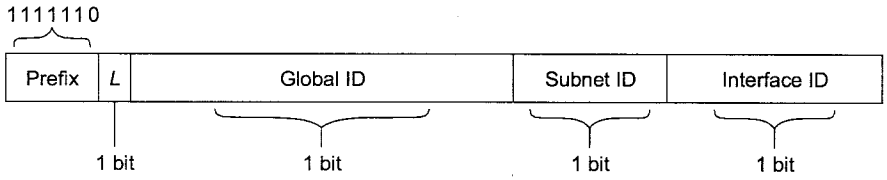


Link-Local Address

Link-local addresses are used for communication among IPv6 hosts on a link (broadcast segment) only. These addresses are not routable, so a Router never forwards these addresses outside the link.

4.14.6 Unique-Local Address

This type of IPv6 address is globally unique, but it should be used in local communication. The second half of this address contains Interface ID and the first half is divided among Prefix, Local Bit, Global ID, and Subnet ID.



Unique-Local Address

Prefix is always set to 1111 110. L bit is set to 1 if the address is locally assigned. So far, the meaning of L bit to 0 is not defined. Therefore, Unique Local IPv6 address always starts with 'FD'.

4.14.7 Special Addresses

Version 6 has slightly complex structure of IP address than that of IPv4. IPv6 has reserved a few addresses and address notations for special purposes. See the table below:

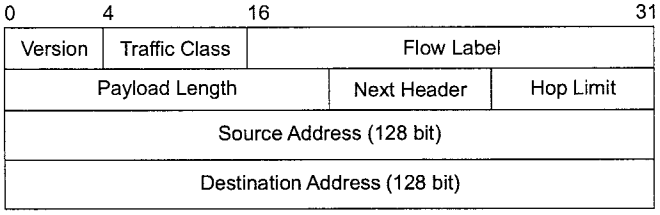
IPv6 Address	Meaning
: : / 128	Unspecified Address
: : / 0	Default Route
: : 1 / 128	Loopback Address

- As shown in the table, the address 0:0:0:0:0:0:0:0/128 does not specify anything and is said to be an unspecified address. After simplifying, all the 0s are compacted to : :/128.
- In IPv4, the address 0.0.0.0 with netmask 0.0.0.0 represents the default route. The same concept is also applied to IPv6, the address 0:0:0:0:0:0:0:0 with netmask all 0s represents the default route. After applying IPv6 rule, this address is compressed to ::/0.
- Loopback addresses in IPv4 are represented by 127.0.0.1 to 127.255.255.255 series. But in IPv6, only 0:0:0:0:0:0:0:1/128 represents the Loopback address. After loopback address, it can be represented as ::1/128.

4.14.8 Headers

The wonder of IPv6 lies in its header. An IPv6 address is 4 times larger than IPv4, but surprisingly, the header of an IPv6 address is only 2 times larger than that of IPv4. IPv6 headers have one Fixed Header and zero or more Optional (Extension) Headers. All the necessary information that is essential for a router is kept in the Fixed Header. The Extension Header contains optional information that helps routers to understand how to handle a packet/flow.

Fixed Header



IPv6 Header

IPv6 fixed header is 40 bytes long and contains the following information.

Defination of all the Field inside the IPv6 Header

1. **Version (4-bits):** It represents the version of Internet Protocol, i.e., 0110.
2. **Traffic Class (8-bits):** These 8 bits are divided into two parts. The most significant 6 bits are used for Type of Service to let the Router know what services should be provided to this packet. The least significant 2 bits are used for Explicit Congestion Notification (ECN).
3. **Flow Label (20-bits):** This label is used to maintain the sequential flow of the packets belonging to a communication. The source labels the sequence to help the router identify that a particular packet belongs to a specific flow of information. This field helps avoid re-ordering of data packets. It is designed for streaming/real-time media.
4. **Payload Length (16-bits):** This field is used to tell the routers how much information a particular packet contains in its payload. Payload is composed of Extension Headers and Upper Layer data. With 16 bits, up to 65535 bytes can be indicated; but if the Extension Headers contain Hop-by-Hop Extension Header, then the payload may exceed 65535 bytes and this field is set to 0.
5. **Next Header (8-bits):** This field is used to indicate either the type of Extension Header, or if the Extension Header is not present, then it indicates the Upper Layer PDU. The values for the type of Upper Layer PDU are same as IPv4's.
6. **Hop Limit (8-bits):** This field is used to stop packet to loop in the network infinitely. This is same as TTL in IPv4. The value of Hop Limit field is decremented by 1 as it passes a link (router/hop). When the field reaches 0, the packet is discarded.
7. **Source Address (128-bits):** This field indicates the address of originator of the packet.
8. **Destination Address (128-bits):** This field provides the address of intended recipient of the packet.

4.14.9 Extension Headers

In IPv6, the Fixed Header contains only that much information which is necessary, avoiding those information which is either not required or is rarely used. All such information is put between the Fixed Header and the Upper layer header in the form of Extension Headers. Each Extension Header is identified by a distinct value.

When Extension Headers are used, IPv6 Fixed Header's Next Header field points to the first Extension Header. If there is one more Extension Header, then the first Extension Header's 'Next-Header' field points to the second one, and so on. The last Extension Header's 'Next-Header' field points to the Upper Layer Header. Thus, all the headers point to the next one in a linked list manner.

If the Next Header field contains the value 59, it indicates that there are no headers after this header, not even Upper Layer Header. The following Extension Headers must be supported as per RFC 2460:

Extension Header	Next Header Value	Description
Hop-by-Hop Options header	0	read by all devices in transit network
Routing header	43	contains methods to support making routing decision
Fragment header	44	contains parameters of datagram fragmentation
Destination Options header	60	read by destination devices
Authentication header	51	information regarding authenticity
Encapsulating Security Payload header	50	encryption information

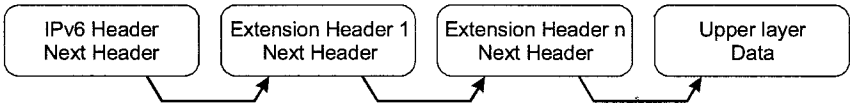
The sequence of Extension Headers should be:

IPv6 header
Hop-by-Hop Options header
Destination Options header[1]
Routing header
Fragment header
Authentication header
Encapsulating Security Payload header
Destination Options header[2]
Upper-layer header

These headers:

- should be processed by First and subsequent destinations.
- should be processed by Final Destination.

Extension Headers are arranged one after another in a linked list manner, as depicted in the following diagram:



Extension Headers Connected Format

4.14.10 Communication

In IPv4, a host that wants to communicate with another host on the network needs to have an IP address acquired either by means of DHCP or by manual configuration. As soon as a host is equipped with some valid IP address, it can speak to any host on the subnet.

To communicate on layer-3, a host must also know the IP address of the other host. Communication on a link is established by means of hardware-embedded MAC Addresses. To know the MAC address of a host whose IP address is known, a host sends ARP broadcast and in return, the intended host sends back its MAC address.

In IPv6, there are no broadcast mechanisms. It is not a must for an IPv6 enabled host to obtain an IP address from DHCP or manually configure one, but it can auto-configure its own IP.

ARP has been replaced by ICMPv6 Neighbor Discovery Protocol.

Neighbor Discovery Protocol

A host in IPv6 network is capable of auto-configuring itself with a unique link- local address. As soon as the host gets an IPv6 address, it joins a number of multicast groups. All communications related to that segment take place on those multicast addresses only. A host goes through a series of states in IPv6:

- **Neighbor Solicitation:** After configuring all IPv6's either manually or by DHCP Server or by auto-configuration, the host sends a Neighbor Solicitation message out to FF02::1/16 multicast address for all its IPv6 addresses in order to know that no one else occupies the same addresses.
- **DAD (Duplicate Address Detection):** When the host does not listen from anything from the segment regarding its Neighbor Solicitation message, it assumes that no duplicate address exists on the segment.
- **Neighbor Advertisement:** After assigning the addresses to its interfaces and making them up and running, the host once again sends out a Neighbor Advertisement message telling all other hosts on the segment that it has assigned those IPv6 addresses to its interfaces.

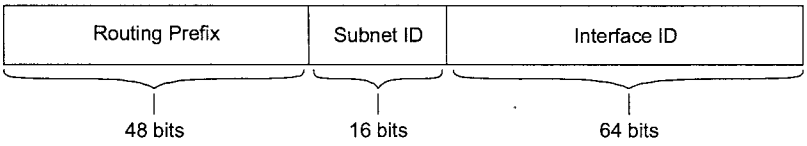
Once a host is done with the configuration of its IPv6 addresses, it does the following things:

- **Router Solicitation:** A host sends a Router Solicitation multicast packet (FF02::2/16) out on its segment to know the presence of any router on this segment. It helps the host to configure the router as its default gateway. If its default gateway router goes down, the host can shift to a new router and makes it the default gateway.
- **Router Advertisement:** When a router receives a Router Solicitation message, it responds back to the host, advertising its presence on that link.
- **Redirect:** This may be the situation where a Router receives a Router Solicitation request but it knows that it is not the best gateway for the host. In this situation, the router sends back a Redirect message telling the host that there is a better 'next-hop' router available. Next-hop is where the host will send its data destined to a host which does not belong to the same segment.

4.14.11 Subnetting

In IPv4, addresses were created in classes. Classful IPv4 addresses clearly define the bits used for network prefixes and the bits used for hosts on that network. To subnet in IPv4, we play with the default classful netmask which allows us to borrow host bits to be used as subnet bits. This results in multiple subnets but less hosts per subnet. That is, when we borrow host bits to create a subnet, it costs us in lesser bit to be used for host addresses.

IPv6 addresses use 128 bits to represent an address which includes bits to be used for subnetting. The second half of the address (least significant 64 bits) is always used for hosts only. Therefore, there is no compromise if we subnet the network.



IPv6 Subnetting

16 bits of subnet is equivalent to IPv4's Class B Network. Using these subnet bits, an organization can have another 65 thousands of subnets which is by far, more than enough.

Thus routing prefix is /64 and host portion is 64 bits. We can further subnet the network beyond 16 bits of Subnet ID, by borrowing host bits; but it is recommended that 64 bits should always be used for hosts addresses because auto-configuration requires 64 bits.

IPv6 subnetting works on the same concept as Variable Length Subnet Masking in IPv4.

/48 prefix can be allocated to an organization providing it the benefit of having up to /64 subnet prefixes, which is 65535 sub-networks, each having 264 hosts. A /64 prefix can be assigned to a point-to-point connection where there are only two hosts (or IPv6 enabled devices) on a link.

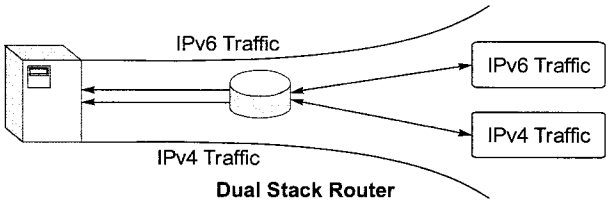
4.15 Transition from IPv4 to IPv6

Complete transition from IPv4 to IPv6 might not be possible because IPv6 is not backward compatible. This results in a situation where either a site is on IPv6 or it is not. It is unlike implementation of other new technologies where the newer one is backward compatible so the older system can still work with the newer version without any additional changes.

To overcome this shortcoming, we have a few technologies that can be used to ensure slow and smooth transition from IPv4 to IPv6.

Dual Stack Routers

A router can be installed with both IPv4 and IPv6 addresses configured on its interfaces pointing to the network of relevant IP scheme.

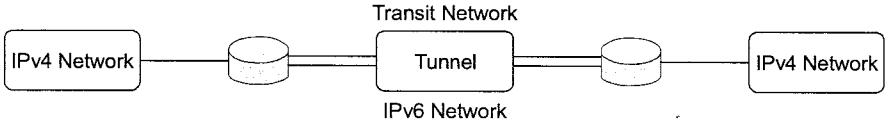


Dual Stack Router

In the above diagram, a server having IPv4 as well as IPv6 address configured for it can now speak with all the hosts on both the IPv4 as well as the IPv6 networks with the help of a Dual Stack Router. The Dual Stack Router can communicate with both the networks. It provides a medium for the hosts to access a server without changing their respective IP versions.

Tunneling

In a scenario where different IP versions exist on intermediate path or transit networks, tunneling provides a better solution where user's data can pass through a non-supported IP version.



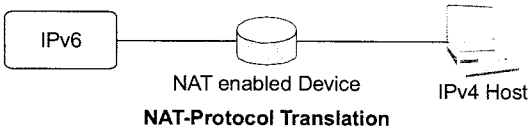
Tunneling

The above diagram depicts how two remote IPv4 networks can communicate via a Tunnel, where the transit network was on IPv6. Its reverse is also possible where the transit network is on IPv6 and the remote sites that intend to communicate are on IPv4.

NAT Protocol Translation

This is another important method of transition to IPv6 by means of a NAT-PT (Network Address Translation-Protocol Translation) enabled device. With the help of a NAT-PT device, actual conversion can take place between IPv4 and IPv6 packets and vice versa.

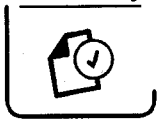
A host with IPv4 address sends a request to an IPv6 enabled server on Internet that does not understand IPv4 address.



NAT-Protocol Translation

In this scenario, the NAT-PT device can help them communicate. When the IPv4 host sends a request packet to the IPv6 server, the NAT-PT device/router strips down the IPv4 packet, removes IPv4 header, and adds IPv6 header and passes it through the Internet. When a response from the IPv6 server comes for the IPv4 host, the router does vice versa.

Summary



- The Address Resolution Protocol (ARP) is a dynamic mapping method that finds a physical address, given an IP address.
- An ARP request is broadcast to all devices on the network.
- An ARP reply is unicast to the host requesting the mapping.
- IP is an unreliable connectionless protocol responsible for source-to-destination delivery.
- Packets in the IP layer are called datagrams.
- A datagram consists of a header (20 to 60 bytes) and data.
- The MTU is the maximum number of bytes that a data link protocol can encapsulate. MTUs vary from protocol to protocol.
- Fragmentation is the division of a datagram into smaller units to accommodate the MTU of a data link protocol.
- The fields in the IP header that relate to fragmentation are the identification number, the fragmentation flags, and the fragmentation offset.
- The Internet Control Message Protocol (ICMP) sends five types of error-reporting messages and four pairs of query messages to support the unreliable and connectionless Internet Protocol (IP).
- ICMP messages are encapsulated in IP datagrams.
- The destination-unreachable error message is sent to the source host when a datagram is undeliverable.
- The source-quench error message is sent in an effort to alleviate congestion.
- The time-exceeded message notifies a source host that (1) the time-to-live field has reached zero or (2) fragments of a message have not arrived in a set amount of time.
- The parameter-problem message notifies a host that there is a problem in the header field of a datagram.
- The redirection message is sent to make the routing table of a host more effective.
- The echo-request and echo-reply messages test the connectivity between two systems.
- The time-stamp-request and time-stamp-reply messages can determine the roundtrip time between two systems or the difference in time between two systems.
- The address-mask request and address-mask reply messages are used to obtain the subnet mask.
- The router-solicitation and router-advertisement messages allow hosts to update their routing tables.
- IPv6, the latest version of the Internet Protocol, has a 128-bit address space, a resource allocation, and increased security measures.
- IPv6 uses hexadecimal colon notation with abbreviation methods available.
- Three strategies used to make the transition from version 4 to version 6 are dual stack, tunneling, and header translation.

Student's Assignment

Q.1 A router has the following CIDR entries in its routing table:

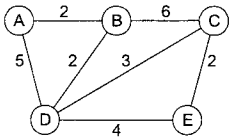
Address/mask	Next hop
135.46.56.0/22	I_0: Interface 0
135.46.60.0/22	I_1: Interface 1
192.53.40.0/23	R_1: Router 1
default	R_2: Router 2

For each of the following IP addresses, what does the router do if a packet with that address arrives?

(a) 135.46.63.10 (b) 135.46.57.14
(c) 135.46.52.2 (d) 192.53.40.7
(e) 192.53.56.7

Q.2 A computer on a 6 Mbps network is regulated by a token bucket. The token bucket is filled with a rate of 1 Mbps. The bucket is initially filled to capacity with 1 Mb. How long (in seconds) can the computer transmit at the full 6 Mbps?

Q.3 Consider the following subnet



Find the link state routing table for node A using OSPF.

(a)

	Cost	via
A	0	A
B	2	B
C	8	B
D	5	D
E	8	B

(b)

	Cost	via
A	0	A
B	2	B
C	7	D
D	4	B
E	8	B

(c)

	Cost	via
A	0	A
B	2	B
C	5	B
D	4	B
E	8	D

(d)

	Cost	via
A	0	A
B	2	B
C	7	B
D	4	B
E	8	B

Q.4 Suppose that x bits of user data are to be transmitted over K-hop path in a packet-switched network as a series of packets each containing p data bits and h header bits with $x >> (p + h)$. The bit rate of lines is b bps and Propagation delay is negligible. What is the time taken by the source to transmit total bits?

(a) $(p + h)\,x/b$ bits (b) $(p + h)\,x/pb$ bits
(c) $p\,x/b$ bits (d) hx/pb bits

Q.5 Router operate a layer _____ LAN switches operate a layer _____. Ethernet hubs operate atleast _____ ward processing operates a layer _____.

(a) 3, 3, 1, 7 (b) 3, 2, 1, none
(c) 3, 2, 1, 6 (d) 3, 3, 2, none

Q.6 Which of the following describe router functions

(a) Packet switching (b) Packet filtering
(c) Internetwork communication
(d) Path selection
(e) All of the above

Q.7 Match the following

List-I

A. Distance vector routing
B. Link-state routing
C. Flooding
D. Hierarchical routing

List-II

1. Shortest path first
2. Large topology
3. Split-horizon hack
4. Duplicate packets

Codes:

	A	B	C	D
(a)	1	2	3	4
(b)	2	3	1	4
(c)	3	1	4	2
(d)	4	2	1	3

Q.8 Which one of the following is not a function of network layer?

(a) Routing (b) Inter-networking
(c) Congestion control (d) None of these

Q.9 Which one of the following algorithm is not used for congestion control

(a) traffic aware routing (b) admission control
(c) load shedding (d) None of these

Q.10 ICMP is used in

(a) ping (b) trace route
(c) ipconfig (d) Both (a) and (b)

Q.11 The data field can carry which of the following?

(a) TCP segment (b) UDP segment
(c) ICMP message (d) None of these

Answer Key:

1. (a) Forward to I_1 (b) Forward to I_0
(c) Forward to R_2 (d) Forward to R_1
(e) Forward to R_2

2. 0.2 **3.** (d) **4.** (b) **5.** (b) **6.** (e)
7. (b) **8.** (d) **9.** (d) **10.** (d) **11.** (c)

CHAPTER 05

Transport Layer & Protocols

5.1 Introduction

- TCP provides logical communication between two processes located on two different hosts.
- At the data-link level, delivery of frames take place between two nodes connected by a point-to-point link or a LAN, by using the data-link layers address, say MAC address. (**Node-to-node delivery**)
- At the network level, delivery of datagram can take place between two hosts by using IP address. (**Host-to-host delivery**)
- At the transport level, communication can take place between processes or application programs by using port addresses (**Process-to-process delivery**)
- From the viewpoint of the transport layer, the underlying network layer has certain limitations in the level of service it can provide:
  (*i*) Limited packet size (MTU, Maximum Transmission Unit)
  (*ii*) Loss, reordering, and duplicate delivery of packets
  (*iii*) Arbitrary long delays
  (*iv*) Host to host communication
- Hence IP (Network layer protocol) is unreliable because it provides **best-effort** level of service but NOT Guaranteed service and we need TCP and UDP to extend the services provided by Network layer.

5.2 Transport Layer Services

The following list shows some of the common services that a transport protocol can be expected to provide:

1. **Reliable message delivery:**
   (*i*) Byte stream broken into small chunks called segments
   (*ii*) Receiver sends Ack's for segments
   (*iii*) TCP maintains a timer. If ACK is not received in time then it is retransmitted

2. **Byte stream service:**
   (*i*) To the lower layers, TCP handles data in blocks, the Segments.
   (*ii*) To the higher layers TCP handles data as a sequence of bytes and does not identify boundaries between bytes. So higher layers do not know about the beginning and end of segments.
   (*iii*) If a sender process sends a stream of bytes, the receiver process will be getting exactly the same stream of bytes.
3. **Synchronization between sender and receiver through flow control:**
   (*i*) Flow control is also sometimes necessary between two users for speed matching, that is, for ensuring that a fast transmitter does not overwhelm a slow receiver with more packets than the latter can handle.
   (*ii*) TCP allows the receiver to apply flow control to the sender (advertising the receiver widow size) and Prevents sender from overrunning the receiver.
4. **Support for multiple application processes on each host:**
   (*i*) Unique port assigned for each process both at sender and receiver (the range of port numbers is 0 to 65535).
   (*ii*) Ports can provide multiple endpoints on a single node (Multiplexing). For example, the name on a postal address is a kind of multiplexing, and distinguishes between different recipients of the same location.

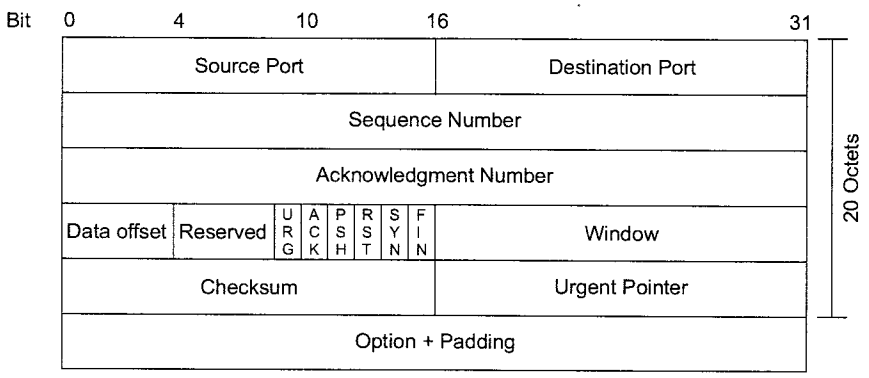
5.3 Transmission Control Protocol (TCP)

TCP provides a connection-oriented, full -duplex, reliable, streamed delivery service using IP to transport messages between two processes.

Reliability is ensured by:

- Connection-oriented service
- Flow control using sliding window protocol
- Error detection using checksum
- Error control using go-back-N ARQ technique
- Congestion avoidance algorithms; multiplicative decrease and slow-start

5.3.1 TCP Datagram



The TCP datagram format

Brief descriptions of various fields in the TCP header:

Field name	# of bits	description
Source port	16	It defines the port number of the application program in the host of the sender
Destination port	16	It defines the port number of the application program in the host of the receiver
Sequence number	32	It conveys the receiving host which octet in this sequence comprises the first byte in the segment
Acknowledgement number	32	This specifies the sequence number of the next octet that receiver expects to receive
HLEN	4	This field specifies the number of 32-bit words present in the TCP header
Control flag bits	6	**URG:** Urgent pointer URG =1 (urgent pointer is in use). ACK = 1 (acknowledgement number is valid). ACK = 0 (the segment does not contain acknowledgement) **PSH:** Push the data without buffering PSH = 1(request to forward the data to application layer without buffering it). **RST** =1 (abruptly reset the connection, whenever there is a host crash or sometimes used to reject a segment). **SYN:** Synchronize sequence numbers during connection establishment **Connection request: SYN =1,ACK=0** **Reply : SYN=1,ACK=1** **FIN:** Terminate the connection
Window size	16	It tells how many bytes may be sent, starting at the byte acknowledged.
Checksum	16	Checksum used for error detection. It checksums the data, header and pseudo header.
Urgent pointer	16	Used only when URG flag is valid
Options		40 bytes of information Some widely used options are: • MSS(maximum segment size) • window scale • Time Stamp

5.3.2 TCP Options

Window Size

- Window size (advertised window size) is used for synchronization between sender and receiver. Sender can send the data according to the requirement of the receiver.
- When the ACK segment containing the window size 0 comes to the sender, then sender is waiting for next acknowledgment.
- If the next ACK transmitted by receiver is lost in the network then receiver is also waiting for data.
- This situation is called DEADLOCK.

NOTE: Window size = 0 is valid and legal and specifies that bytes up to and including $ACK_{number}-1$ have been received , but was unable to consume the data and would like no more data.

Urgent Pointer

Urgent pointer in the header indicate byte offset from current sequence number at which urgent data is to be found.

Window Scaling Factor (Option in Header)

- To increase the window size, a window scale factor is used.
- The new window size is found by first raising 2 to the number specified in the window scale factor.
- Then this result is multiplied by the value of the window size in the header.
- New window size = (window size defined in the header) * 2^ (window scale factor)

***Example*-5.1** Suppose the value of the window scale factor is 3 and host receives an acknowledgment in which the window size is advertised as 32,768. What is the size of window this host can use?

Solution:

Size of window this host can use is 32,768 * 2^3 or 262,144 bytes. The same value can be obtained if we shift the number 32,768 three bits to the left.

Time Stamp

This is a **10-byte option.** Note that the end with the active open announces a timestamp in the connection request segment (SYN segment). If it receives a timestamp in the next segment (SYN + ACK) from the other end, it is allowed to use the timestamp; otherwise, it does not use it any more. The time-stamp option has two applications: it measures the round trip time and prevents wraparound sequence numbers.

Measuring RTT

- Timestamp can be used to measure the round trip time (RTT). TCP, when ready to send a segment, reads the value of the system clock and inserts this value, a 32 bit number, in the timestamp value field.
- The receiver, when sending an acknowledgment for this segment or an cumulative acknowledgment that covers the bytes in this segment, copies the timestamp received in the timestamp echo reply.
- The sender, upon receiving the acknowledgment, subtracts the value of the timestamp echo reply from the time shown by the clock to find RTT.
- RTT calculated is the time difference between sending the first segment and receiving the third segment.
- This is actually the meaning of RTT: the time difference between a packet sent and the acknowledgment received. The third segment carries the acknowledgment for the first and second segments.

Protection Against Wrapped Sequence Numbers (PAWS)

- The sequence number defined in the TCP protocol is only 32 bits long and it could be wrapped around in a high-speed connection.
- This implies that if a sequence number is ***n*** at one time, it could be ***n*** again during the lifetime of the same connection.
- Now if the first segment is duplicated and arrives during the second round of the sequence numbers, the segment belonging to the past is wrongly taken as the segment belonging to the new round
- Solution is to include the timestamp in the identification of a segment. In other words, the identity of a segment can be defined as the combination of timestamp and sequence number.
- Two segments 400:12,001 and 700:12,001 definitely belongs to different rounds definitely belong to different incarnations. The first was sent at time 400, the second at time 700.

EOP

The end-of-option (EOP) option is a 1-byte option used for padding at the end of the option section. It can only be used as the last option. Only one occurrence of this option is allowed. After this option, the receiver looks for the payload data.

RTO timer

- Retransmits after timeout
- It is also called general timer
- Generally static timers are used in DLL and dynamic timers are used in Transport layer

5.3.3 Port Numbers

Transport layer address is specified with the help of a 16-bit port number in the range of 0 to 65535. Internet assigned numbers authority (IANA) has divided the address range in to 3 ranges

Well known Ports: 0 to 1023

These port numbers are commonly used as universal port numbers in the servers for the convenience of many clients. Port numbers in this range are controlled and assigned by IANA.

Registered ports: 1024 to 49151

These port numbers are not controlled or assigned by IANA. However they can only be registered with IANA to avoid duplication (assigning same port number to different process).

Dynamic ports: 49152 to 65535

These ports are neither controlled by IANA nor need to be registered.

These are defined at the client side and chosen randomly by transport layer software.

Example diagram for port numbers

Table - 5.2 *Well-known ports used by TCP*

Port	Protocol	Description
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the data and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
20	FTP, Data	File Transfer Protocol (data connections)
21	FTP, control	File Transfer Protocol (control connections)
23	TELNET	Terminal Network
25	SMTP	Simple Mail Transfer protocol
53	DNS	Domain Name Server
67	BOOTP	BOOTP Protocol
79	Finger	Finger
80	HTTP	Hypertext Transfer Protocol
111	RPC	Remote Procedure Call

5.3.4 Connection-oriented Service

- TCP establishes a virtual path between the source and destination processes before any data communication by using two procedures, *connection establishment* to start reliably and *connection termination* to terminate gracefully.
- TCP performs data communication in full-duplex mode, that is both the sender and receiver processes can send segments simultaneously.
- TCP requires that each side select an initial *SN* at random (well, in practice *SN* is initialized by the value of an internal clock counter that increments once every 4 sec).
- The sequence numbers are then exchanged while establishing the connection using a three-way handshake

For **connection establishment** in full-duplex mode, a four-way protocol can be used (the second and third steps can be combined to form a three-way handshaking protocol).

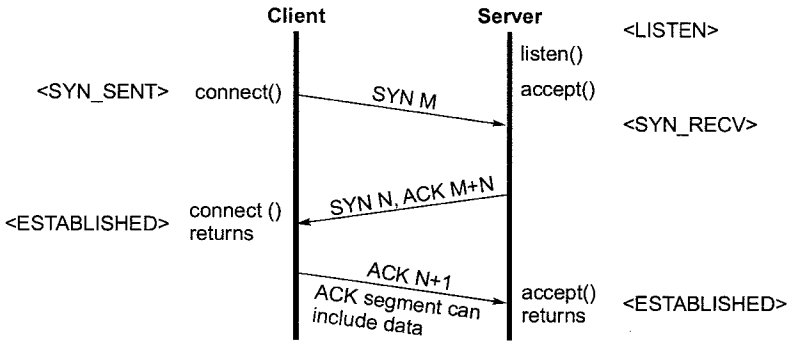


Figure: *Protocol for connection establishment*

Step -1: The client sends SYN segment, which includes, source and destination port numbers, and an *initialization sequence number* (ISN), which is essentially the byte number to be sent from the client to the server.

Step-2: The server sends a segment, which is a two-in-one segment. It acknowledges the receipt of the previous segment and it also acts as initialization segment for the server.

Step-3: The sends an ACK segment, which acknowledges the receipt of the second segment.

Connection termination: A four-way handshaking protocol is necessary for termination of connection in both directions as shown in Figure. The four steps are as follows:

Step-1: The client sends a FIN segment to the server.

Step-2: The server sends an ACK segment indicating the receipt of the FIN segment and the segment also acts as initialization segment for the server.

Step-3: The server can still continue to send data and when the data transfer is complete it sends a FIN segment to the client.

Step-4: The client sends an ACK segment, which acknowledges the receipt of the FIN segment sent by the server.

Both the connections are terminated after this four-way handshake protocol.

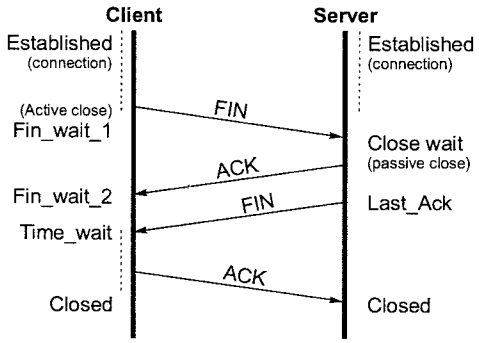


Figure: *Protocol for connection termination*

State	Description
CLOSED	No connection is active or pending
LISTEN	The server is waiting for an incoming call
SYN RCVD	A connection request has arrived; wait for ACK
SYN SENT	The application has started to open a connection
ESTABLISHED	The normal data transfer state
FIN WAIT 1	The application has said it is finished
FIN WAIT 2	The over side has agreed to release
TIME WAIT	Wait for all packets to die off
CLOSING	Both sides have tried to close simultaneously
CLOSE WAIT	The other side has initiated a release
LAST ACK	Wait for all packets to die off

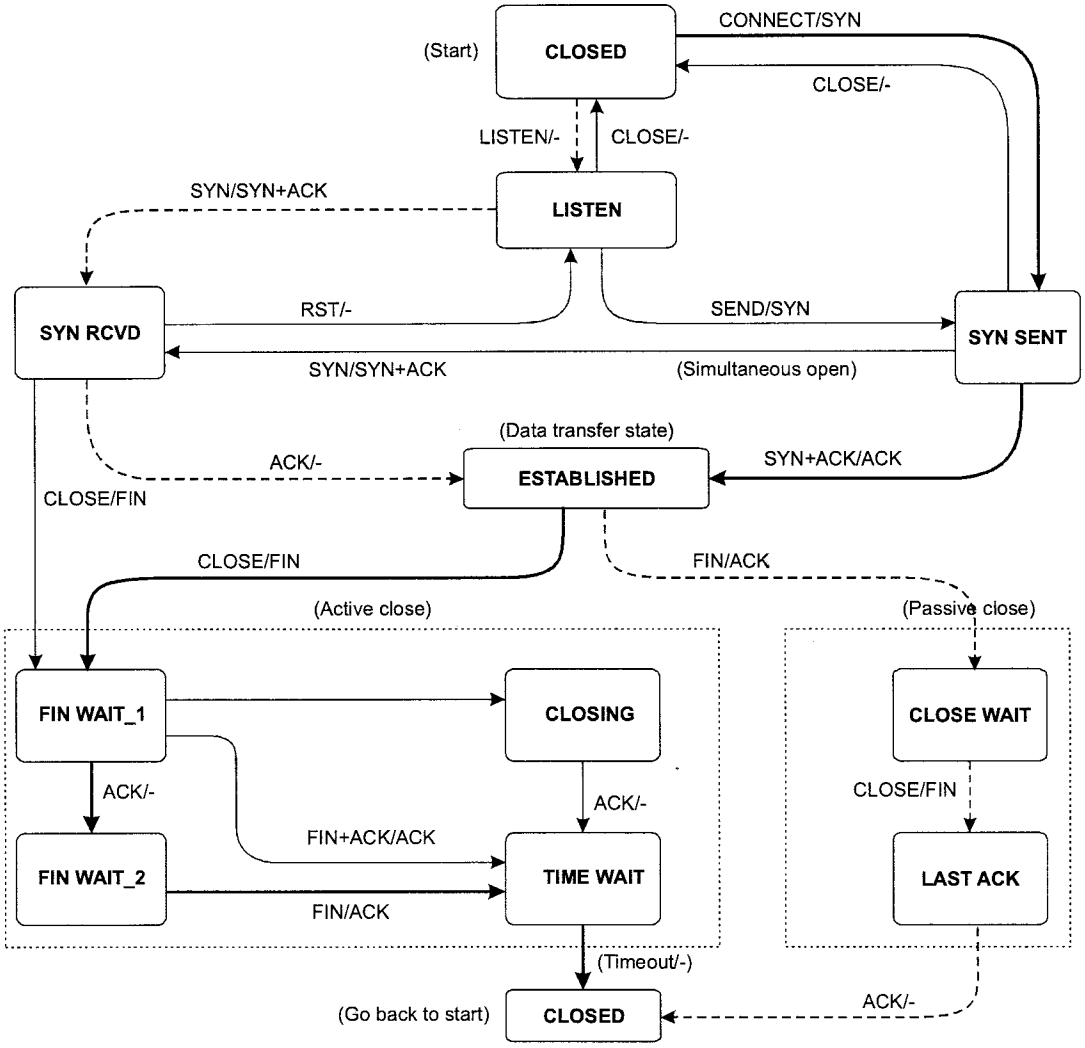


Figure: *TCP connection management finite state machine*

The heavy solid line is the normal path for a client. The heavy dashed line is the normal path for a server. The light lines are unusual events. Each transition is labeled with the event causing it and the action resulting from it, separated by a slash.

TCP is a Byte Oriented Protocol

The sender generates bytes into the TCP buffer (sliding window), and TCP collects enough bytes from this buffer to fill a reasonably sized packet called segment (one that is appropriate for the underlying network to prevent further segmentation at the network layer).

SN represents the sequence number of the first byte of data in the segment. Thus if a segment has a sequence number $SN = m$ and contains n bytes of data, then the next segment for that TCP session has $SN = m + n$.

TCP at the receiving side empties the content of the segment into the TCP buffer (sliding window), and the receiver read bytes from this buffer.

NOTE: When a segment is transmitted initially the initial sequence number will be a random number in the range 0 to $(2^{32} - 1)$.

ACK will always be the sequence number of the next expected segment

5.3.5 Sequence Numbers Wrap Around

Sequence numbers are given for every byte in the segment. During TCP connection, sequence numbers will eventually wrap around (*SN* is finite) if the connection remains active for a long time. Without the FIFO property, a delayed segment that suddenly shows up will cause the sliding window algorithm to make a mistake in accepting it.

The way TCP fights against this possibility is through an appropriate choice of MSL, maximum segment life (hence a segment cannot be delayed indefinitely).

Example-5.2 **Currently, MSL is 120 seconds. How much time is needed for the sequence numbers to wrap around? If 32 bits are used for generating sequence numbers On Ethernet of bandwidth 10 Mbps;**

Solution:

We need $2^{32} * 8 / (10 * 10^6)$ seconds to transmit 2^{32} bytes.(to wrap around) MSL.

On faster networks (e.g. 622 Mbps and 1.2 Gbps), the time to wrap around is much smaller than 120 seconds.

Example-5.3 **Imagine a TCP connection is transferring a file of 6000 bytes. The first byte is numbered 10010. What are the sequence numbers for each segment if data is sent in five segments with the first four segments carrying 1,000 bytes and the last segment carrying 2,000 bytes?**

Solution:

The following shows the sequence number for each segment:

Segment 1 → 10,010 (10,010 to 11,009)
Segment 2 → 11,010 (11,010 to 12,009)
Segment 3 → 12,010 (12,010 to 13,009)
Segment 4 → 13,010 (13,010 to 14,009)
Segment 5 → 14,010 (14,010 to 16,009)

TCP Segments

Since IP is a packet switching protocol the TCP stream of bytes has to be divided into groups called segments before sent to IP layer for encapsulation into IP datagram's and transmission. Segments are transparent to the sending and receiving processes, which see stream only.

The bytes of data being transferred in each connection are numbered by TCP. The numbering starts with a randomly generated number.

In order to keep track of segments and their ordering, they have two fields: sequence number and acknowledgment number. However, these numbers do not represent the number of segments (like frame numbers in link layer error flow/error control), they rather refer to the number of bytes.

The value of the sequence number field in a segment defines the number of the first data byte contained in that segment.

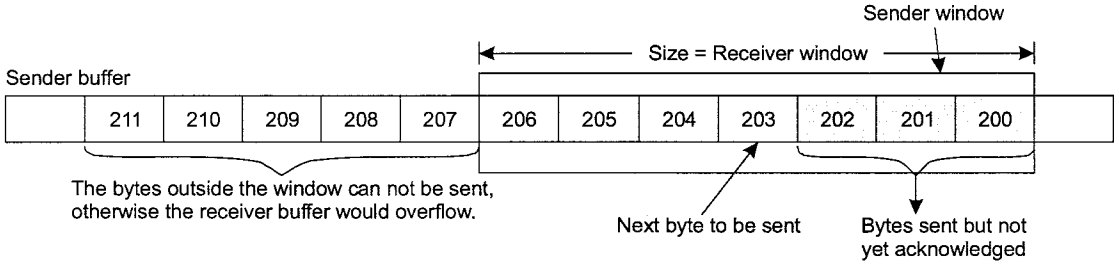
The value of the acknowledgment field in a segment defines the number of the next byte a party expects to receive. The acknowledgment number is cumulative. The sequence and acknowledge numbers are 32-bit numbers.

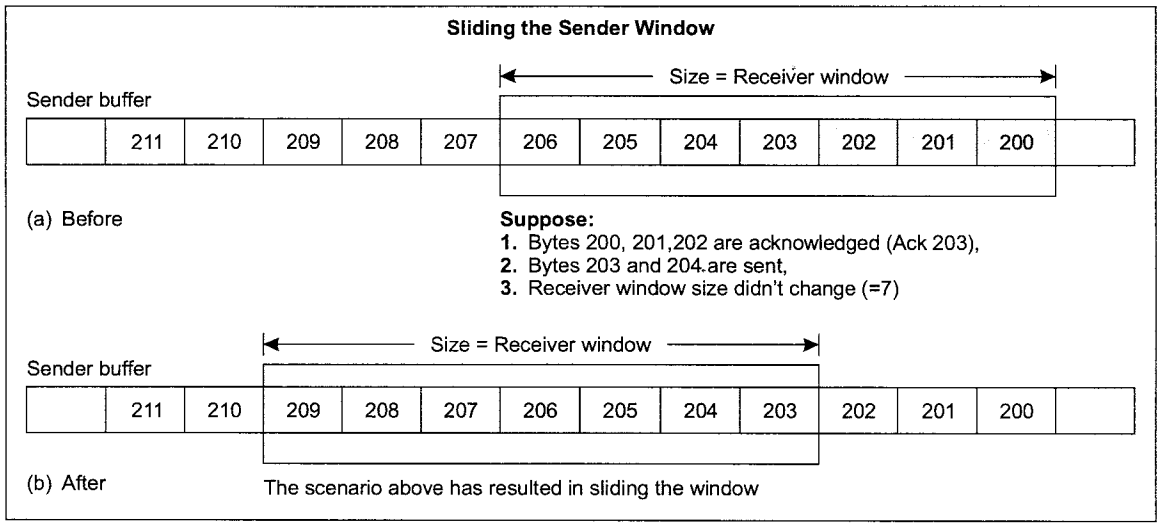
5.3.6 Reliable Communication

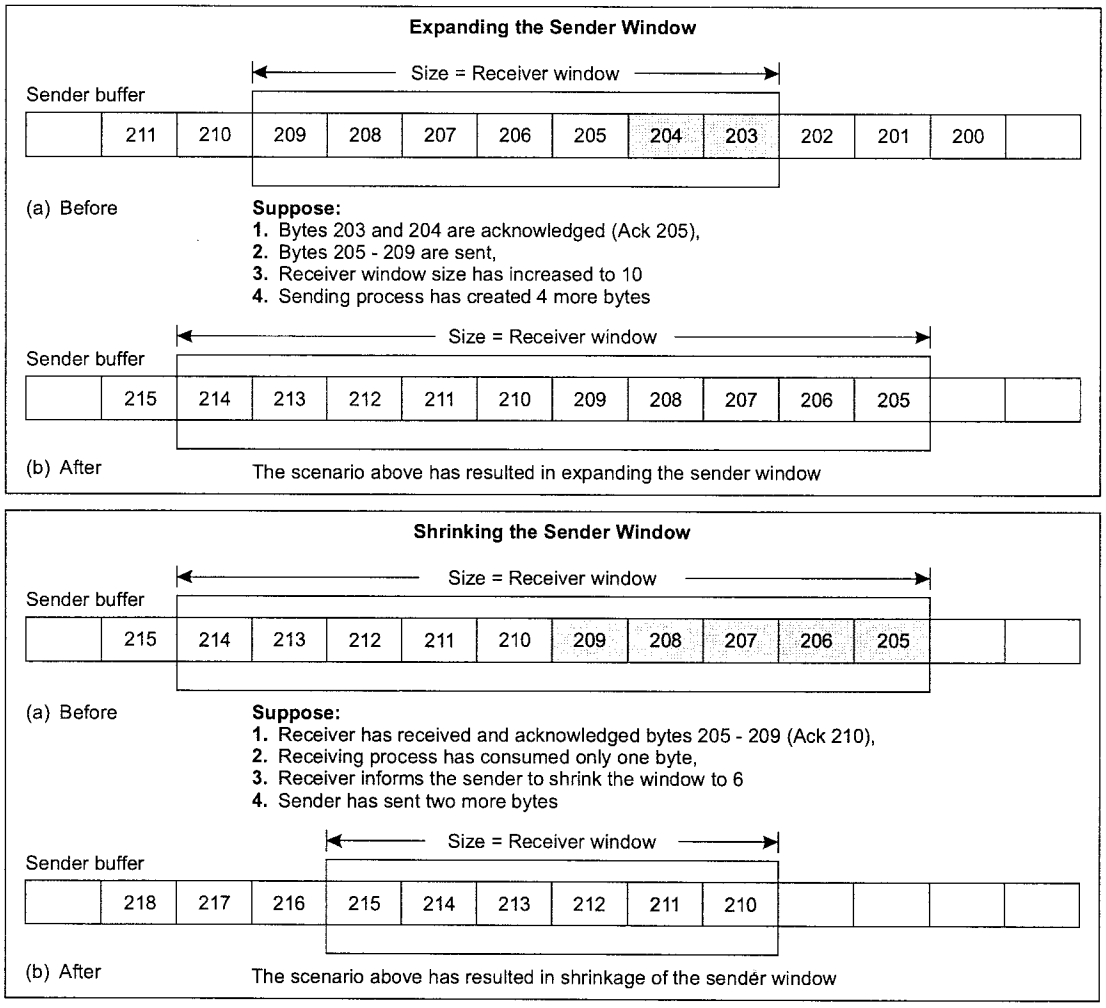
To ensure reliability, TCP performs flow control, error control and congestion control.

1. Flow Control

TCP uses byte-oriented sliding window protocol, which allows efficient transmission of data and at the same time the destination host is not overwhelmed with data.







Silly Window Syndrome

If either sender or receiver slow down significantly, results in very small segments. An extreme are one byte segments, causing IP packets of size 41 carrying only one byte of data (20 - IP header, 20-TCP header, and 1-payload). This is very inefficient use of network, called **silly syndrome**. Solution to that is not to send segments if the sender window has small opening, but to wait until the receiver opens the window and advertises that to the sender.

How much to wait? Too long wait causes interactive application (like TELNET) to suffer. Too short wait causes silly window syndrome. Nagle's self-clocking algorithm is a simple and effective solution to that

Nagle's Self-clocking Algorithm

```
if (available data and window > MSS)
Send a full segment;
else
        if (there is unacknowledged data in flight)
        Buffer the new data until ACK arrives;
        else
        Send all the new data now;
```

Effective Window Size

The window size W (in bits) must be large enough to use the full capacity of the network. Let MSS be the Maximum Segment Size. If the bandwidth is B, then the sender transmits W bits every RTT + MSS / B ≈ BRTT (usually RTT >> MSS / B). Of course one of the difficulties in determining W is to obtain accurate measurements of RTT and B. But assuming these are known, the sender is limited to min (B, W / RTT) bps.

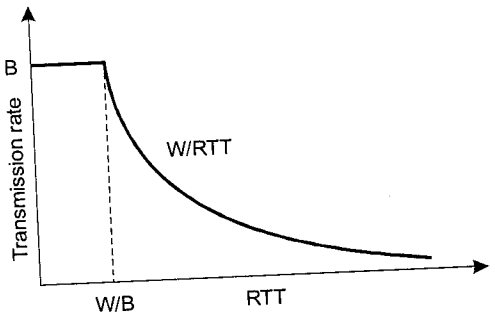


Figure: *Window size*

Therefore, we need W ≥ RTT × B. TCP can support windows up to $2^{16} - 1$ bytes (≈ 64 KB), as we will see next.

2. Error Control

Error control in TCP includes mechanism for detecting corrupted segments with the help of checksum field. Acknowledgment method is used to confirm the receipt of uncorrupted data. If the acknowledgment is not received before the time-out, it is assumed that the data or the acknowledgment has been corrupted or lost. It may be noted that there is no negative acknowledgment in TCP. To keep track of lost or discarded segments and to perform the operations smoothly, the following four timers are used by TCP:

- Retransmission; it is dynamically decided by the round trip delay time.
- Persistence; this is used to deal with window size advertisement.
- Keep-alive; commonly used in situations where there is long idle connection between two processes
- Time-waited; it is used during connection terminations

3. TCP Congestion Control

When the load offered to any network is more than it can handle, congestion builds up. The Internet is no exception. TCP congestion control is based on implementing this approach using a window and with packet loss as the binary signal. To do so, TCP maintains a congestion window (**cwnd**) in addition to the flow control window or receiver's window (**rwnd**), whose size is the number of bytes the sender may have in the network at any time.

Senders rate = cwnd/RTT

TCP adjusts the size of the window according to the AIMD rule. Both windows are tracked in parallel, and the number of bytes that may be sent is the smaller of the two windows.

Senders window = min {Cwnd, rwnd}

For *example*, if the receiver says "send 64 KB" but the sender knows that bursts of more than 32 KB clog the network, it will send 32 KB. On the other hand, if the receiver says "send 64 KB" and the sender knows that bursts of up to 128 KB get through effortlessly, it will send the full 64 KB requested.

TCP will stop sending data if either the congestion or the flow control window is temporarily full. Given a good retransmission timeout, the TCP sender can track the outstanding number of bytes, which are loading the network. It simply looks at the difference between the sequence numbers that are transmitted and acknowledged.

All we need to do is to track the congestion window, using sequence and acknowledgement numbers, and adjust the congestion window using an AIMD rule.

Considerations for Designing Congestion Algorithm

A **first consideration** is that the way packets are sent into the network, even over short periods of time, must be matched to the network path. Otherwise the traffic will cause congestion.

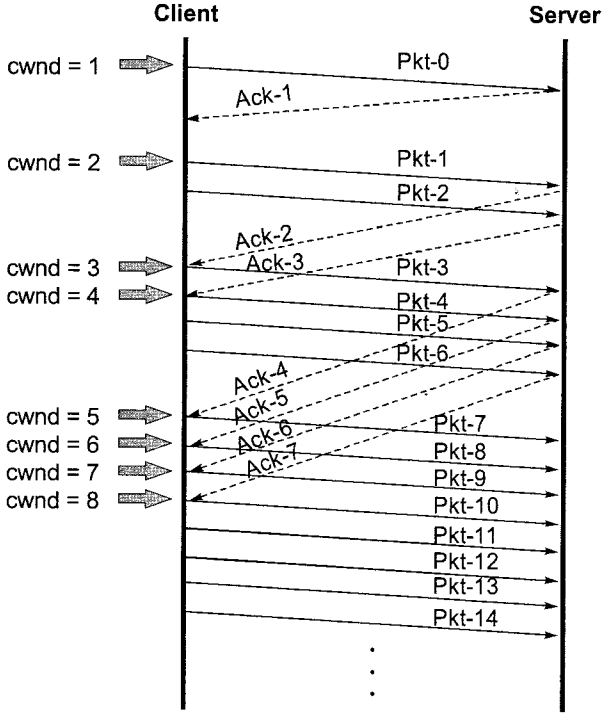
For *example*, consider a host with a congestion window of 64 KB attached to a 1-Gbps switched Ethernet. If the host sends the entire window at once, this burst of traffic may travel over a slow 1-Mbps ADSL line further along the path. The burst that took only half a millisecond on the 1-Gbps line will clog the 1-Mbps line for half a second, completely disrupting protocols such as voice over IP.

A **second consideration** is that the AIMD rule will take a very long time to reach a good operating point on fast networks if the congestion window is started from a small size.

Example, Consider a modest network path that can support 10 Mbps with an RTT of 100 msec. The appropriate congestion window is the bandwidth-delay product, which is 1 Mbit or 100 packets of 1250 bytes each. If the congestion window starts at 1 packet and increases by 1 packet every RTT, it will be 100 RTTs or 10 seconds before the connection is running at about the right rate.

SLOW START (Jacobson's Solution)

- The congestion window is doubling every round trip time. TCP congestion window extends in the following scheme: 1, 2, 4, 8, 16, ...
- Slow-start works well over a range of link speeds and round trip times, and uses an ack clock to match the rate of sender transmissions to the network path.



- To keep slow start under control, the sender keeps a threshold for the connection called the ***slow start threshold***.

  Cwnd <= slow start threshold

- Initially this value is set arbitrarily high, to the size of the **flow control window (receiver window)**, so that it will not limit the connection.
- TCP keeps increasing the congestion window in slow start until **a timeout occurs** or the congestion window exceeds the threshold (or the receiver's window is filled).
- Whenever a packet loss is detected, for example, by a timeout, the slow start threshold is set to be half of the congestion window and the entire process is restarted. **(Cwnd = 1 MSS)**

  Timeout == > threshold=cwnd/2
- The idea is that the current window is too large because it caused congestion previously that is only now detected by a timeout. Half of the window, which was used successfully at an earlier time, is probably a better estimate for a congestion window that is close to the path capacity but will not cause loss.

Congestion Avoidance

Whenever the slow start threshold is crossed, TCP switches from slow start to additive increase. In this mode, the congestion window is increased by one segment every round trip time

If (cwnd >= SSTH)

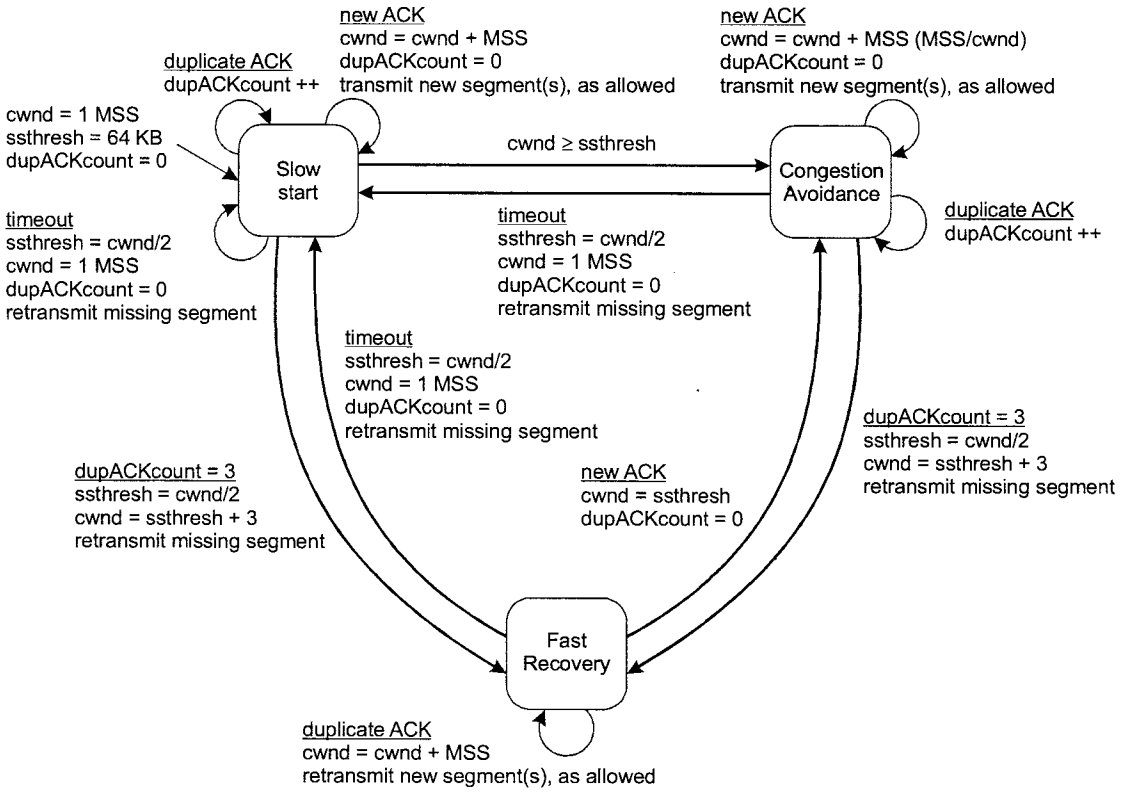
Congestion avoidance: {*cwnd*=cwnd+1/cwnd , for every ACK}

Else if (timeout)

SSTH = cwnd/2 and start slow start again (cwnd=1MSS)

Else

Cwnd = cwnd + 1MSS (for every ACK)



- The sender can often detect packet loss well before the timeout event occurs by noting so-called duplicate ACKs.
- A duplicate ACK is an ACK that reacknowledges a segment for which the sender has already received an earlier acknowledgment.
- Since TCP does not use negative acknowledgments, the receiver cannot send an explicit negative acknowledgment back to the sender. Instead, it simply reacknowledges (that is, generates a duplicate ACK for) the last in-order byte of data it has received.
- In the case that three duplicate ACKs are received, the TCP sender performs a fast retransmit retransmitting the missing segment before that segment's timer expires.
- SSTH = cwnd/2, Cwnd = 1MSS, dACK count = 0 and Retransmit missing segment

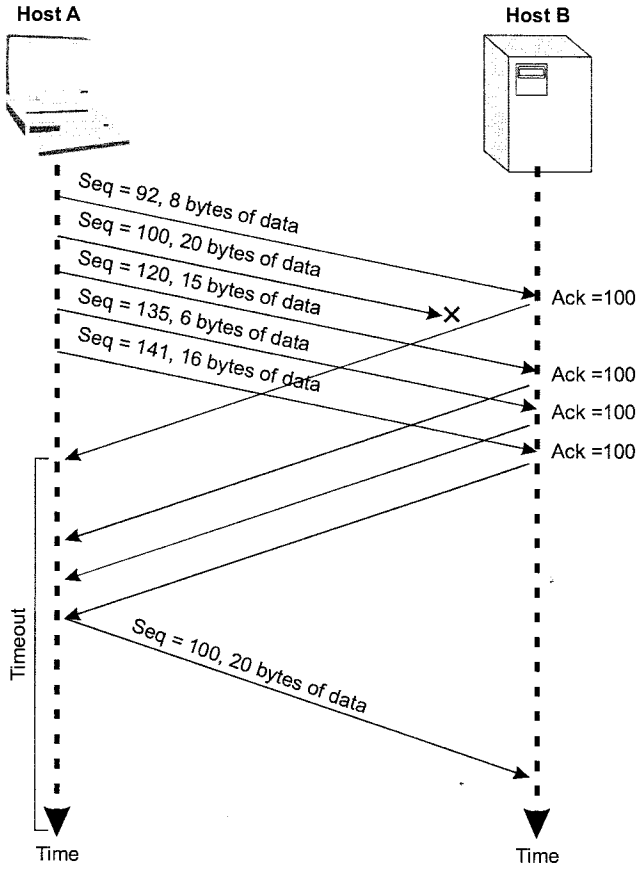


Figure: *Fast retransmit: retransmitting the missing segment before*

Example - 5.4 Let the window size be W at the beginning of RTT. Assuming there are no losses in the RTT time, what are the respective window sizes for slow start and congestion. Avoidance after completion of RTT?

Solution:

During slow start, each Ack increases the window size by 1. The source will receive W acknowledgments, thus increasing it's window size by W + W = 2 W.

During congestion avoidance, the window size increases by 1/W. So the size at the end of the RTT will be approximately $W + \left(\frac{1}{W}\right) \times W = W + 1$.

Example-5.5 Let the size of the congestion window of a TCP connection be 64 KB when a timeout occurs. The RTT of the connection is 50 m sec and maximum segment size is 2 KB. The time taken by the TCP to get back to 64 KB congestion window is _____.

Solution:

Timeout occurs new threshold value is 32 KB.

and Cwnd = 1 MSS = 2 KB

2 KB, 4 KB, 8 KB, 16 KB, **32 KB.** This is threshold value

Now TCP move into congestion avoidance.

34 KB, 36 KB, 38 KB, ...,62 KB, 64 KB.

Upto 32 KB it takes 5 RTT's

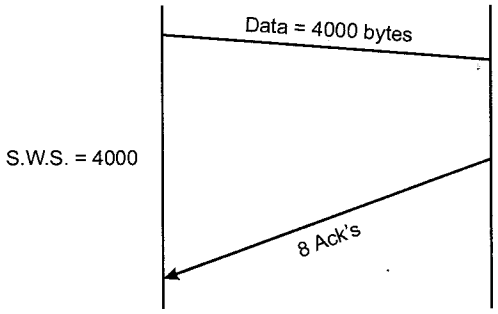
From 34 KB to 64 KB takes another 15 RTT's

$\therefore$ The total time taken to get back to 64 KB is

$$20 \times 50 \text{ ms} = 1000 \text{ ms}$$

Example-5.6 The sender window size is 4000 bytes and MSS = 100 bytes. When data is transmitted 8 Ack's came. Then what is the present sender window size.

Solution:



In slow start algorithm, the sender window size is based on the number of Ack's. Window size increases by 1 MSS for every 1 Ack received.

Present window size = 4000 + 8 × 100 = 4800 bytes.

Example-5.7 The initial congestion window size over the TCP is 1. If slow start algorithm is used and the size of congestion window changes by 1 whenever an Ack is received i.e., after first round trip time congestion window size is 2 segments. Assume connection never leaves slow start. Find the number of RTT's to send 3999 segments.

Solution:

Window size [WS = 1] initially

After 1st RTT, window size = 2 and 1 segment is sent.

After 2nd RTT, window size = 4 and 3 segments

After 3rd RTT, window size = 8 and 7 segment sent in total.

⋮

After 'x' RTT, window size = 2^x and $2^x - 1$ segments send.

Now $2^x - 1 \geq 3999$

$2^x \geq 4000$

$x \log 2 \geq \log(4000)$

$x \geq 1196$

$\therefore$ $x \geq 12$ RTT's

FAST Recovery

- In fast recovery, the value of cwnd is increased by 1 MSS for every duplicate ACK received for the missing segment that caused TCP to enter the fast-recovery state.
- Eventually, when an ACK arrives for the missing segment, TCP enters the congestion-avoidance state after deflating cwnd.
- If a timeout event occurs, fast recovery transitions to the slow-start state after performing the following actions (same as in slow start and congestion avoidance)
- The value of cwnd is set to 1 MSS, and the value of ssthresh is set to half the value of cwnd when the loss event occurred.

5.3.7 TCP Timers

TCP maintains four (4) timers for each connection

Retransmission Timer: The timer is started during a transmission. A timeout causes a retransmission.

Setting the RT timer:

- When a segment is sent and RT timer is not running, start RT timer with RTO value
- Turn off RT timer, when all data is acknowledged
- When an ACK is received for new data, reset the RT timer to RTO value

RT timer expires:

- Retransmit the earliest segment that has not been acknowledged
- Double value of RTO (see Karn's rule)
- Start the RT timer with RTO value

Persistent timer:

- Assume the window size goes down to zero and the ACK that opens the window gets lost
- Ensures that window size information is transmitted even if no data is transmitted.
- Forces that the sender periodically queries the receiver about its window size (window probes)
- The persist timer is started by the sender when the sliding window is zero
  Persist timer uses exponential back off (initial value is 1.5 seconds) rounded to the range [5 sec, 60sec] so the time intervals between timeouts are at:

  5, 5, 6, 12, 24, 48, 60, 60...

The window probe packet contains one byte of data (TCP can do this even if the window size is zero).

Keep Alive Timer

Used to Detects crashes on the other end of the connection

- When a TCP connection has been idle for a long time (1 min – 2 hours), a Keep alive timer reminds a station to check if the other side is still there.
- A segment without data is sent if the connection has been idle for 2 hours
- Assume a probe has been sent from A to B:
  (*i*) *B is up and running:* B responds with an ACK
  (*ii*) *B has crashed and is down:* A will send 10 more probes, each 75 seconds apart. If A does not get a response, it will close the connection
  (*iii*) *B has rebooted:* B will send a RST segment
  (*iv*) *B is up, but unreachable:* Looks to A the same as (2)

2MSL Timer (TIME_WAIT)

Measures the time that a connection has been in the TIME_WAIT state. It runs for the twice the maximum packet life time to make sure when a connection is closed, all packets created by it have died off.

5.3.8 TCP: Retransmission and Timeouts

- If the timeout is set too short, unnecessary retransmissions will occur, clogging the internet with useless packets. If it is set too long performance will suffer due to the long retransmission delay whenever a packet is lost.
- TCP implementations attempt to predict future round trip times by sampling the behavior of packets sent over a connection and averaging those samples into a "smoothed" round trip time estimate, SRTT.
- The dynamic algorithm generally used by TCP is due to Jacobson (1988) and works as follows.
- For each connection, TCP maintains a variable, SRTT (Smoothed Round trip Time) that is the best current estimate of the round trip time to the destination.
- When a segment is sent, a timer is started, both to see how long the acknowledgement takes and also to trigger a retransmission if it takes too long.
- If the acknowledgement gets back before the timer expires, TCP measures how long the acknowledgement took, say, R.
- It then updates SRTT according to the formula

$$SRTT = \alpha\, SRTT + (1 - \alpha)\, R$$

  Where α is a smoothing factor that determines how quickly the old values are forgotten.
- Typically, $\alpha = 7/8$. This kind of formula is an EWMA (Exponentially Weighted Moving Average) or low-pass filter that discards noise in the samples.
- To fix this problem, Jacobson proposed making the timeout value sensitive to variance in round trip times as well as the smoothed round trip time. This change requires keeping track of another smoothed variable, RTTVAR (Round trip Time Variation) that is updated using the formula

$$RTTVAR = \beta\, RTTVAR + (1 - \beta)\, |\, SRTT - R\, |$$

  This is an EWMA as before, and typically $\beta = 3/4$. The retransmission timeout, RTO, is set to be

$$RTO = SRTT + 4 \times RTTVAR$$

Example-5.8 Assume that a TCP process A first measures the actual round trip time to another TCP process to be 30 ms, and A thus sets its estimated round trim time to be 30 ms. The next actual round trip time that A sees is 60 ms. In response, A increases its estimated round trip to 50 ms. The next actual round trip time that A sees is 40 ms. What is the next estimated round trip computed by A?

Solution:

Based on the provided results, it is clear that in this case TCP does not use the simple average, but weighted/smoothed average to estimate RTTs.

Thus

$$\text{New-estimated-RTT} = x \times \text{Old-estimated-RTT} + (1-x) \times \text{New-observed-RTT}$$

$$50 = x \times 30 + (1-x) \times 60$$

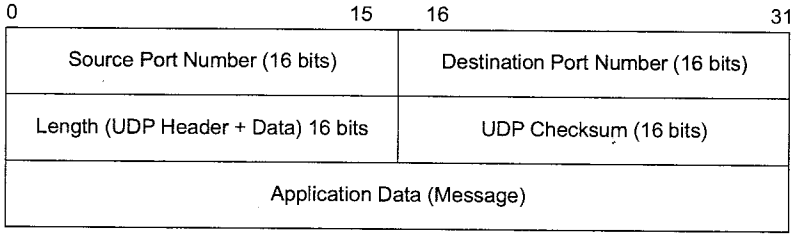
$$\Rightarrow \quad 50 = -30 \times x + 60$$

So, $x = 1/3$

and $\text{New-estimated-RTT} = 1/3 \times 50 + 2/3 \times 40 = 130/3 = 43.33$ [msec]

5.4 Introduction to UDP

- The Internet protocol suite supports a connectionless transport protocol called UDP (User Datagram Protocol).
- UDP provides a way for applications to send encapsulated IP datagrams without having to establish a connection.
- UDP transmits segments consisting of an 8-byte header followed by the payload.



- The two ports serve to identify the endpoints within the source and destination machines.
- When a UDP packet arrives, its payload is handed to the process attached to the destination port. This attachment occurs when the BIND primitive or something similar is used.
- Think of ports as mailboxes that applications can rent to receive packets.
- The source port is primarily needed when a reply must be sent back to the source.
- The UDP length field includes the 8-byte header and the data.
- The minimum length is 8 bytes, to cover the header. The maximum length is 65,515 bytes, which is lower than the largest number that will fit in 16 bits because of the size limit on IP packets.
- An optional Checksum is also provided for extra reliability.
- It checksums the header, the data, and a conceptual IP pseudoheader. When performing this computation, the Checksum field is set to zero and the data field is padded out with an additional zero byte if its length is an odd number.

NOTE: The checksum algorithm is simply to add up all the 16-bit words in one's complement and to take the one's complement of the sum. As a consequence, when the receiver performs the calculation

- On the entire segment, including the Checksum field, the result should be 0. If the checksum is not computed, it is stored as a 0.
- The pseudoheader for the case of IPv4 contains the 32-bit IPv4 addresses of the source and destination machines, the protocol number for UDP (17), and the byte count for the UDP segment (including the header).
- Including the pseudo header in the UDP checksum *computation helps detect misdelivered packets*, but including it also violates the protocol hierarchy since the IP addresses in it belong to the IP layer, not to the UDP layer. TCP uses the same pseudoheader for its checksum. 32 Bits Source address Destination address 0 0 0 0 0 0 0 0 Protocol = 17 UDP length
- The IPv4 pseudoheader included in the UDP checksum.

NOTE

- It does not do flow control, congestion control, or retransmission upon receipt of a bad segment.
- All of that is up to the user processes. What it does do is *provide an interface to the IP protocol* with the added feature of demultiplexing multiple processes using the ports and optional end-to-end error detection.

- One area where it is especially useful is in client-server situations. Often, the client sends a short request to the server and expects a short reply back.
- If either the request or the reply is lost, the client can just time out and try again. Not only is the code simple, but fewer messages are required (one in each direction) than with a protocol requiring an initial setup like TCP.
- An application that uses UDP this way is DNS (Domain Name System), which we will study in Application layer.

5.5 Congestion Control

Too many packets present in (a part of) the network causes packet delay and loss that degrades performance. This situation is called **congestion**. The network and transport layers share the responsibility for handling congestion. Since congestion occurs within the network, it is the network layer that directly experiences it and must ultimately determine what to do with the excess packets The nature of a ***Packet switching network*** can be summarized in following points:

- At each node, there is a queue of packets for each outgoing channel.
- If packet arrival rate exceeds the packet transmission rate, the queue size grows without bound.
- When the line for which packets are queuing becomes more than 80% utilized, the queue length grows alarmingly.

At very high traffic, performance collapse completely, and almost no packet is delivered.

5.5.1 Causes of Congestion

If there is *insufficient memory* to hold these packets, then packets will be lost (dropped). Adding more memory also may not help in certain situations.

If router have an infinite amount of memory even then instead of congestion being reduced, it gets worse; because by the time packets gets at the head of the queue, to be dispatched out to the output line, they have already timed-out (repeatedly), and duplicates may also be present. All the packets will be forwarded to next router up to the destination, all the way only increasing the load to the network more and more.

Finally when it arrives at the destination, the packet will be discarded, due to time out, so instead of been dropped at any intermediate router (in case memory is restricted) such a packet goes all the way up to the destination, increasing the network load throughout and then finally gets dropped there.

Slow processors also cause Congestion. If the router CPU is slow at performing the task required for them (Queuing buffers, updating tables, reporting any exceptions etc.), queue can build up even if there is excess of line capacity. Similarly, *Low- Bandwidth* lines can also cause congestion. Upgrading lines but not changing slow processors, or vice-versa, often helps a little; these can just shift the bottleneck to some other point. The real problem is the mismatch between different parts of the system.

Congestion tends to feed upon itself to get even worse. Routers respond to overloading by dropping packets. When these packets contain TCP segments, the segments don't reach their destination, and they are therefore left unacknowledged, which eventually leads to timeout and retransmission.

So, the major cause of congestion is often the *bursty* nature of traffic.

5.5.2 Effects of Congestion

Congestion affects two vital parameters of the network performance, namely *throughput* and *delay*. Initially throughput increases linearly with offered load, because utilization of the network increases. However, as the offered load increases ***beyond certain limit***, say 60% of the capacity of the network, the ***throughput drops***. If the offered load increases further, a point is reached when not a single packet is delivered to any destination, which is commonly known as *deadlock* situation.

The ***delay also increases with offered load***, as shown in Figure (b). And no matter what technique is used for congestion control, the delay grows without bound as the load approaches the capacity of the system. It may be noted that initially there is longer delay when congestion control policy is applied. However, the network without any congestion control will saturate at a lower offered load.

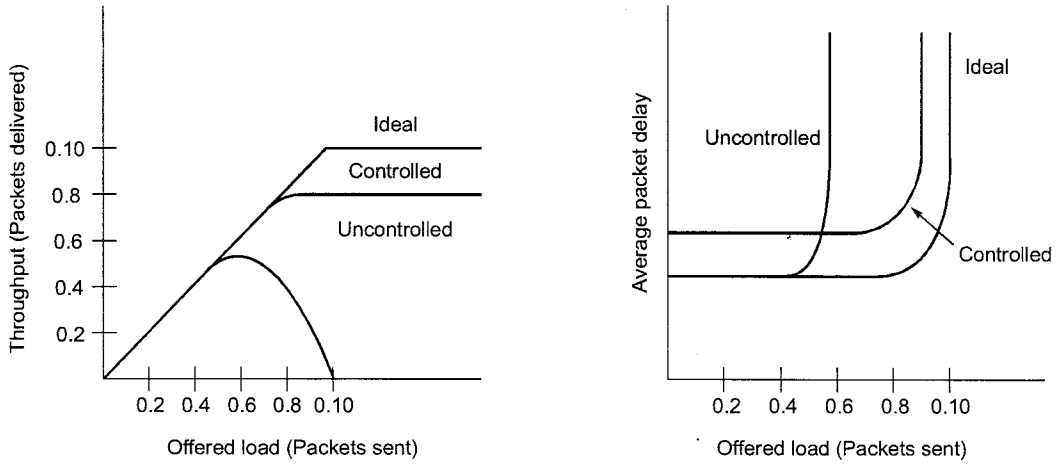


Figure: *(a) Effect of congestion on throughput (b) Effect of congestion on delay*

There are three curves in Figure (a), the ideal one corresponds to the situation when all the packets introduced are delivered to their destination up to the maximum capacity of the network. The second one corresponds to the situation when there is no congestion control. The third one is the case when some congestion control technique is used. This prevents the throughput collapse, but provides lesser throughput than the ideal condition due to overhead of the congestion control technique.

5.5.3 Congestion Control Techniques

Congestion control refers to the mechanisms and techniques used to control congestion and keep the traffic below the capacity of the network. As shown in Figure, the congestion control techniques can be broadly classified two broad categories:

- **Open loop:** Protocols to prevent or avoid congestion, ensuring that the system (or network under consideration) never enters a Congested State.
- **Close loop:** Protocols that allow system to enter congested state, detect it, and remove it.

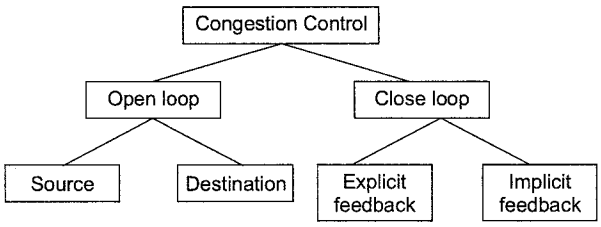


Figure: *Congestion Control Categories*

The first category of solutions or protocols attempt to solve the problem by a good design, at first, to make sure that it doesn't occur at all. Once system is up and running midcourse corrections are not made. These solutions are somewhat static in nature, as the policies to control congestion don't change much according to the current state of the system. Such Protocols are also known as *Open Loop* solutions. These rules or policies include deciding upon when to accept traffic, when to discard it, making scheduling decisions and so on. Main point here is that they make decision without taking into consideration the current state of the network. The open loop algorithms are further divided on the basis of whether these acts on source versus that act upon destination.

The second category is based on the concept of feedback. During operation, some system parameters are measured and feed back to portions of the subnet that can take action to reduce the congestion. This approach can be divided into 3 steps:

- Monitor the system (network) to detect whether the network is congested or not and what's the actual location and devices involved.
- To pass this information to the places where actions can be taken
- Adjust the system operation to correct the problem.

These solutions are known as *Closed Loop* solutions. Various Metrics can be used to monitor the network for congestion. Some of them are: the average queue length, number of packets that are timed-out, average packet delay, number of packets discarded due to lack of buffer space, etc.

A general feedback step would be, say a router, which detects the congestion send special packets to the source (responsible for the congestion) announcing the problem.

These extra packets increase the load at that moment of time, but are necessary to bring down the congestion at a later time.

Other approaches are also used at times to curtail down the congestion. For example, hosts or routers send out probe packets at regular intervals to explicitly ask about the congestion and source itself regulate its transmission rate, if congestion is detected in the network.

This kind of approach is a *pro-active* one, as source tries to get knowledge about congestion in the network and act accordingly.

The closed loop algorithms can also be divided into two categories, namely *explicit feedback* and *implicit feedback* algorithms.

In the explicit approach, special packets are sent back to the sources to curtail down the congestion. While in implicit approach, the source itself acts pro-actively and tries to deduce the existence of congestion by making local observations.

In the following sections we shall discuss about some of the popular algorithms from the above categories.

5.5.4 Leaky Bucket Algorithm

Consider a Bucket with a small hole at the bottom, whatever may be the rate of water pouring into the bucket; the rate at which water comes out from that small hole is constant.

This scenario is depicted in Figure (a). Once the bucket is full, any additional water entering it spills over the sides and is lost (i.e. it doesn't appear in the output stream through the hole underneath).

The same idea of leaky bucket can be applied to packets, as shown in Figure (b). Conceptually each network interface contains a *leaky bucket*. And the following steps are performed:

- When the host has to send a packet, the packet is thrown into the bucket.
- The bucket leaks at a constant rate, meaning the network interface transmits packets at a constant rate.
- Bursty traffic is converted to a uniform traffic by the leaky bucket.
- In practice the bucket is a finite queue that outputs at a finite rate.

This arrangement can be simulated in the operating system or can be built into the hardware.

Implementation of this algorithm is easy and consists of a finite queue. Whenever a packet arrives, if there is room in the queue it is queued up and if there is no room then the packet is discarded.

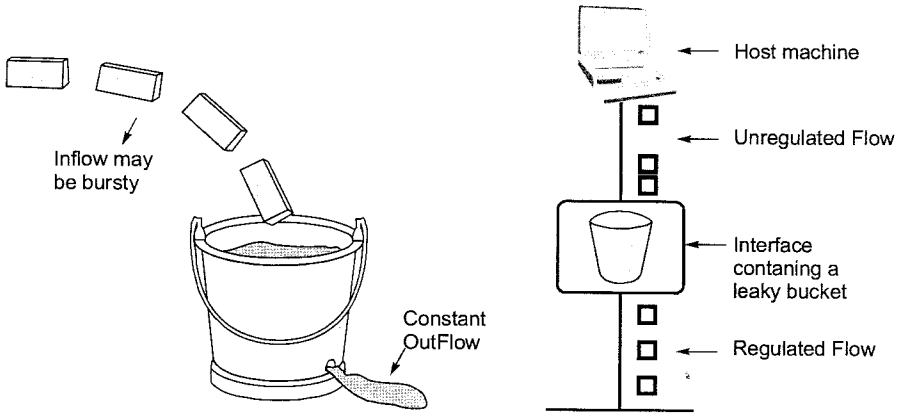


Figure: ***(a)*** *Leaky bucket* ***(b)*** *Leaky bucket implementation*

5.5.5 Token Bucket Algorithm

The leaky bucket algorithm described above, enforces a rigid pattern at the output stream, irrespective of the pattern of the input.

For many applications it is better to allow the output to speed up somewhat when a larger burst arrives than to loose the data. Token Bucket algorithm provides such a solution. In this algorithm leaky bucket holds token, generated at regular intervals. Main steps of this algorithm can be described as follows:

- In regular intervals tokens are thrown into the bucket.
- The bucket has a maximum capacity.
- If there is a ready packet, a token is removed from the bucket, and the packet is send.
- If there is no token in the bucket, the packet cannot be send.

Figure shows the two scenarios before and after the tokens present in the bucket have been consumed. In Figure (a) the bucket holds two tokens, and three packets are waiting to be sent out of the interface, in Figure (b) two packets have been sent out by consuming two tokens, and 1 packet is still left.

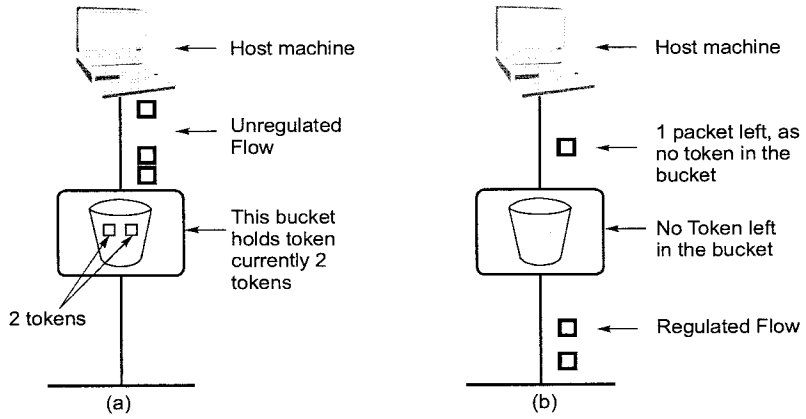


Figure (a) Token bucket holding two tokens, before packets are send out, (b) Token bucket after two packets are sending, one packet still remains as no token is left.

The token bucket algorithm is less restrictive than the leaky bucket algorithm, in a sense that it allows bursty traffic. However, the limit of burst is restricted by the number of tokens available in the bucket at a particular instant of time.

The implementation of basic token bucket algorithm is simple; a variable is used just to count the tokens. This counter is incremented every *t seconds* and is decremented whenever a packet is sent. Whenever this counter reaches zero, no further packet is sent out as shown in Figure.

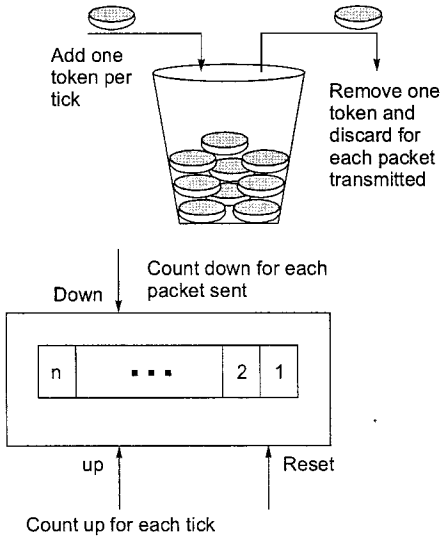


Figure: *Implementation of the Token bucket algorithm*

5.6 Congestion Control in Virtual Circuit

Till now we have discussed two open loop algorithms, where the policy decisions are made in the beginning, irrespective of the current state. Both leaky bucket algorithm and token bucket algorithm are open loop algorithms. In this section we shall have a look at how the congestion is tackled in a virtual- circuit network.

Admission control is one such closed-loop technique, where action is taken once congestion is detected in the network. Different approaches can be followed:

- **Simpler one being:** Do not set-up new connections, once the congestion is signaled. This type of approach is often used in normal telephone networks. When the exchange is overloaded, then no new calls are established.
- **Another approach, which can be followed is:** To allow new virtual connections, but route these carefully so that none of the congested router (or none of the problem area) is a part of this route.
- **Yet another approach can be:** To negotiate different parameters between the host and the network, when the connection is setup. During the setup time itself, Host specifies the volume and shape of traffic, quality of service, maximum delay and other parameters, related to the traffic it would be offering to the network. Once the host specifies its requirement, the resources needed are reserved along the path, before the actual packet follows.

5.6.1 Choke Packet Technique

The *choke packet* technique, a closed loop control technique, can be applied in both virtual circuit and datagram subnets. Each router monitors its resources and the utilization at each of its output line. There is a threshold set by the administrator, and whenever any of the resource utilization crosses this threshold and action is taken to curtail down this. (enters into warning state)

The router sends a *choke packet* back to the source, giving it a feedback to reduce the traffic. And the original packet is tagged (a bit is manipulated in the header field) so that it will not generate other choke packets by other intermediate router, which comes in place and is forwarded in usual way. It means that the first router (along the way of a packet), which detects any kind of congestion, is the only one that sends the choke packets.

When the source host gets the choke packet, it is required to reduce down the traffic send out to that particular destination (choke packet contains the destination to which the original packet was send out).

After receiving the choke packet the source reduces the traffic by a particular fixed percentage, and this percentage decreases as the subsequent choke packets are received. Figure depicts the functioning of choke packets.

For Example, when source A receives a choke packet with destination B at first, it will curtail down the traffic to destination B by 50%, and if again after affixed duration of time interval it receives the choke packet again for the same destination, it will further curtail down the traffic by 25% more and so on.

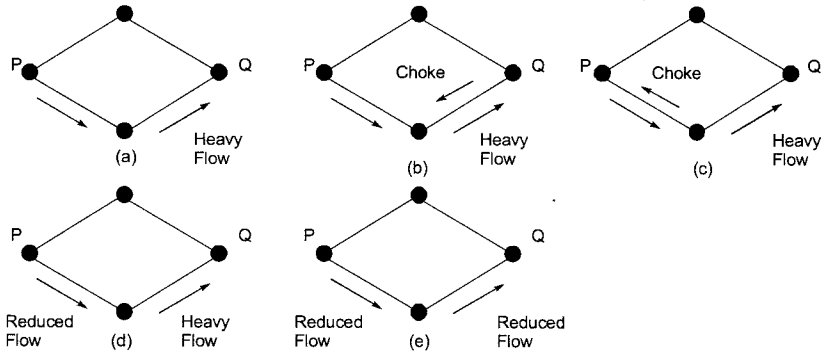


Figure: *Depicts the functioning of choke packets, (a) Heavy traffic between nodes P and Q, (b) Node Q sends the Choke packet to P, (c) Choke packet reaches P, (d) P reduces the flow and send a reduced flow out, (e) Reduced flow reaches node Q.*

Hop-by Hop Choke Packets

This technique is an advancement over Choked packet method.

At high speed over long distances, sending a packet all the way back to the source doesn't help much, because by the time choke packet reach the source, already a lot of packets destined to the same original destination would be out from the source. So to help this, Hop-by-Hop Choke packets are used.

In this approach, the choke packet affects each and every intermediate router through which it passes by. Here, as soon as choke packet reaches a router back to its path to the source, it curtails down the traffic between those intermediate routers.

In this scenario, intermediate nodes must dedicate few more buffers for the incoming traffic as the outflow through that node will be curtailed down immediately as choke packet arrives it, but the input traffic flow will only be curtailed down when choke packet reaches the node which is before it in the original path.

As compared to choke packet technique, hop-by-hop choke packet algorithm is able to restrict the flow rapidly. Hence, in a more complicated network, one can achieve a significant advantage by using hop-by-hop choke packet method.

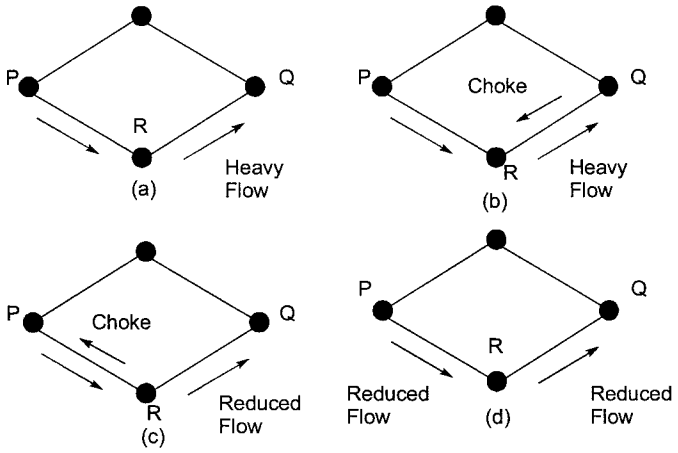


Figure: *Hop-by-Hop Choke Packets*

Figure depicts the functioning of Hop-by-Hop choke packets, (a) Heavy traffic between nodes P and Q, (b) Node Q sends the Choke packet to P, (c) Choke packet reaches R, and the flow between R and Q is curtail down, Choke packer reaches P, and P reduces the flow out.

5.6.2 Load Shedding

Another simple closed loop technique is *Load Shedding*; it is one of the simplest and more effective techniques. In this method, whenever a router finds that there is congestion in the network, it simply starts dropping out the packets. There are different methods by which a host can find out which packets to drop. Simplest way can be just choose the packets randomly which has to be dropped.

More effective ways are there but they require some kind of cooperation from the sender too. For many applications, some packets are more important than others. So, sender can mark the packets in priority classes to indicate how important they are. If such a priority policy is implemented than intermediate nodes can drop packets from the lower priority classes and use the available bandwidth for the more important packets.

Slow Start – a Pro-active technique

This is one of the pro-active techniques, which is used to avoid congestion. A special algorithm is used that allows the device to drop the rate at which segments are sent quickly when congestion occurs.

The device then uses the *Slow Start* algorithm just above to gradually increase the transmission rate back up again to try to maximize throughput without congestion occurring again. We will study slow start algorithm in transport layer.

Flow Control Versus Congestion control

Flow control is a very important part of regulating the transmission of data between devices, but it is limited in a way that it only considers what is going on within each of the devices on the connection, and not what is happening in devices between them. It relates to the point-point traffic between a given sender and a receiver. Flow control always involves some kind of feedback from receiver to sender to tell sender how things are at other end of the network.

Since we are dealing with how TCP works between a typical server and client at layer four, we don't worry about how data gets between them; that's the job of the Internet Protocol at layer three.

These networks and routers are also carrying data from many other connections and higher-layer protocols. If the internet becomes very busy, the speed at which segments are carried between the endpoints of our connection will be reduced, and they could even be dropped. This is called *congestion control*. Congestion control has to do with making sure that subnet carry the offered traffic.

Summary

- Internet is a network of different types of network. TCP/IP is a set of rules and procedures that govern the exchange of messages between hosts linked to different networks. TCP/IP creates an environment as if all hosts are connected to a single logical network.
- Both TCP and UDP belong to transport layer. The UDP is simpler with much less overhead. UDP provides unreliable connectionless service. On the other hand, TCP provides connection oriented reliable service with the help of suitable flow control and error control protocols. As a consequence, TCP has much more overhead.
- UDP protocol provides user programs the ability to communicate using unreliable connectionless packet delivery service with minimum overhead.
- As the UDP datagram does not contain source and destination address information, a pseudo-header is added with these information to verify that the UDP datagram has reached its correct destination.
- TCP uses sliding window protocol for flow control.
- Instead of sending a separate packet for positive/negative acknowledgement, piggybacking technique utilizes the full-duplex communication environment of TCP. The positive/negative acknowledgement information is added to a normal packet sent by the receiving side. It helps to save precious network bandwidth.
- TCP establishes connection using a three-way handshaking protocol and connection is terminated by a 2-way/4-way handshaking protocol.
- Client-Server paradigm is used by all the application layer protocols.

Student's Assignment

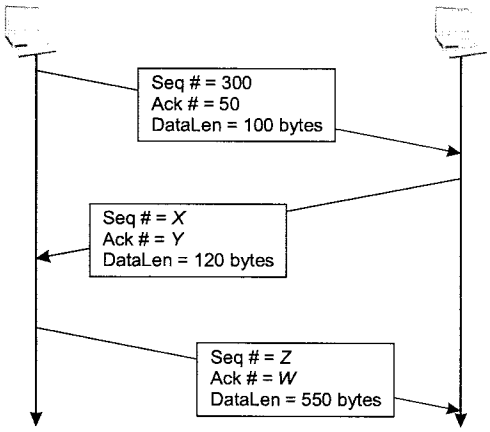
Q.1 Consider a TCP connection between two machines (A and B) in an environment with 0% packet loss. Assume the round trip time (RTT) between the two machines is 4 [seconds], and the segment size is 3 [Kbytes]. The bandwidth of the connection is 500 [kbps]. What is the smallest TCP window size for which there will be no stalling? (We say a TCP connection experiences no stalling if the acknowledgments arrive back to the sending machine before the sliding window over the send buffer close to zero i.e., TCP packets are continuously, back-to-back, sent out of the sending machine.)

Q.2 A TCP connection has just been established between two computers (A and B). Assume:

- Round trip time (RTT) = 100 [ms],
- Congestion window threshold = 4
- Receiver's advertised window = 12

What is the largest window size that the sender is allowed to have? What is the least amount of time before the sender reaches this window size?

Q.3 For the TCP segments indicated below, specify the omitted values (X,Y,Z and W). Assume the packets are transmitted over a reliable link with no packet loss or corruption.



Q.4 Suppose host 'A' sending a large file to host 'B' over a TCP connection. The two end hosts are 10 ms apart (20 ms RTT) connected by a 1 Gbps link. Assume that they are using 1000 bytes packets to transmit the file. For simplicity ignore Ack packets. Atleast how big would the window size (in packets) have to be for the channel utilization to be greater than 80%.

Q.5 On a TCP connection, current congestion window size is 4 KB. The window advertised by the receiver is 6 KB. The last byte sent acknowledged by the receiver is 8192. The current window size at the sender is _____?

Q.6 If the TCP round trip time, is currently 30 μsec and the following acknowledgments come in after 26, 32 and 24 msec, respectively. What is the new RTT estimate? Use $\alpha = 0.9$. Also calculate the problem by using Jacobson's's algorithm. Assume initial deviation as '4'.

Q.7 Consider the effect of using slow start on a line with a 10 msec RTT and no congestion. The receive window is 24 KB and the maximum segment size is 2 KB. How long does it take before the first full window can be sent?

Q.8 Suppose that the TCP congestion window is set to 18 KB and a time out occurs. How big will be window if the next four transmission bursts are all successful? Assume that the MSS is 1 KB.

Q.9 Which of the following services use TCP?

1. DHCP
2. SMTP
3. HTTP
4. TFTP
5. FTP

(a) 1 and 2 (b) 2, 3 and 5
(c) 1, 2 and 4 (d) 1, 3 and 4

Q.10 What layer is the TCP/IP stack is equivalent to the transport layer of the OSI model?

(a) Application (b) Host to host
(c) Internet (d) Network access

Q.11 Which of the following describe the DHCP discover message?

1. It uses FF: FF: FF: FF: FF: FF: as a layer 2 broadcast
2. It uses UDP as the transport layer protocol
3. It uses TCP as the transport layer protocol
4. It does not use a layer 2 destination address

(a) 1 only (b) 1 and 2
(c) 3 and 4 (d) 4 only

Q.12 You want to complement a mechanism that automata the IP configuration, including IP address, subnet mask, default gateway, and DNS information, which protocol will you see to accomplish this?

(a) SMTP (b) SNMP
(c) DHCP (d) ARP

Q.13 Which of the following allows a router to respond to an ARP request that is intended for a remote host?

(a) gateway DP (b) Reverse ARP (RARP)
(c) Proxy ARP (d) Inverse ARP (IARP)

Q.14 Which of the following services use UDP?

1. DHCP 2. SMTP
3. SNMP 4. FTP
5. HTTP 6. TFTP

(a) 1, 3 and 6 (b) 2 and 4
(c) 1, 2 and 4 (d) All of these

Q.15 If you use either Telnet or FTP, which is the highest layer you are using to transmit data?

(a) Application (b) Presentation
(c) Transport (d) Both (a) and (b)

Q.16 Which of he following layers in the TCP/IP model?

1. Application 2. Senior
3. Transport 4. Internet
5. Data link 6. Physical

(a) 1 and 2 (b) 1, 3 and 4
(c) 2 and 3 (d) 3, 4 and 5

Q.17 Which of the following are TCP/IP protocol used at the application layer of the OSI model?

1. IP 2. TCP
3. Telnet 4. FTP
5. TFTP

(a) 1 and 3 (b) 1, 3 and 5
(c) 3, 4 and 5 (d) All of these

Q.18 What protocol is used to find the hardware address of a local device

(a) RARP (b) ARP
(c) IP (d) ICMP

Answer Key:

1. $W \leq 84$
2. Window size is 12, Time is 1 second
3. X = 50, Y = 400, Z = 400 and W = 170
4. 2000 **5.** 4 KB
6. 29.6, 29.84, 29.256 **7.** 40 ms
8. 8 KB
9. (b) **10.** (b) **11.** (b) **12.** (c) **13.** (c)
14. (c) **15.** (a) **16.** (b) **17.** (b) **18.** (c)

CHAPTER 06

Application Layer and Protocols

6.1 Inroduction

The most widely-known TCP/IP Application layer ISO OSI protocols specify the format and control information necessary for many of the common Internet communication functions. Among these TCP/IP protocols are:

- Domain Name Service Protocol (DNS) is used to resolve Internet names to IP addresses.
- Hypertext Transfer Protocol (HTTP) is used to transfer files that make up the Web pages of the World Wide Web.
- Simple Mail Transfer Protocol (SMTP) is used for the transfer of mail messages and attachments.
- Telnet, a terminal emulation protocol, is used to provide remote access to servers and networking devices.
- File Transfer Protocol (FTP) is used for interactive file transfer between systems.

Application Layer ISO OSI Software

The functions associated with the Application layer protocols enable our human network to interface with the underlying data network. When we open a web browser or an instant message window, an application is started, and the program is put into the device's memory where it is executed.

Application Layer Services

Other programs may need the assistance of Application layer services to use network resources, like file transfer or network print spooling.

Though transparent to the user, these services are the programs that interface with the network and prepare the data for transfer.

Different types of data - whether it is text, graphics, or video - require different network services to ensure that it is properly prepared for processing by the functions occurring at the lower layers of OSI model.

Each application or network service uses protocols which define the standards and data formats to be used.

Without protocols, the data network would not have a common way to format and direct data.

6.2 Application Layer Protocols

Application layer **uses protocols that are implemented within applications and services.**

While applications provide people with a way to create messages and application layer services establish an interface to the network, protocols provide the rules and formats that govern how data is treated.

All three components may be used by a single executable program and may even use the same name. For example, when discussing "Telnet" we could be referring to the application, the service, or the protocol.

Within the Application layer, protocols specify what messages are exchanged between the source and destination hosts, the syntax of the control commands, the type and format of the data being transmitted, and the appropriate methods for error notification and recovery.

6.2.1 Application Layer Protocol Functions

Application layer ISO OSI protocols are used by both the source and destination devices during a communication session.

Protocols specify how data inside the messages is structured and the types of messages that are sent between source and destination. Protocols establish consistent rules for exchanging data between applications and services loaded on the participating devices.

In order for the communications to be successful, the application layer protocols implemented on the source and destination host must match. The messages exchanged can be requests for services, acknowledgments, data messages, status messages, or error messages. Protocols also define message dialogues, ensuring that a message being sent is met by the expected response and the correct services are invoked when data transfer occurs.

Applications and services may also use multiple protocols in the course of a single conversation. One protocol may specify how to establish the network connection and another describe the process for the data transfer when the message is passed to the next lower layer. Application layer protocols define:

1. Types of messages
2. Syntax of messages
3. Meaning of any informational fields
4. How messages are sent and the expected response
5. Interaction with next lower layer

6.2.2 Application Layer-Making Provisions for Applications and Services

The Client-Server Model

When people attempt to access information on their device, whether it is a PC, laptop, PDA, cell phone, or some other device connected to a network, the data may not be physically stored on their device. If that is the case, a request to access that information must be made to the device where the data resides.

The Client/Server Model

In the client/server model, the device requesting the information is called a client and the device responding to the request is called a server.

Client and server processes are considered to be in the Application layer. The client begins the exchange by requesting data from the server, which responds by sending one or more streams of data to the client.

Application layer protocols describe the format of the requests and responses between clients and servers. In addition to the actual data transfer, this exchange may also require control information, such as user authentication and the identification of a data file to be transferred.

One example of a client/server network is a corporate environment where employees use a company e-mail server to send, receive and store e-mail. The e-mail client on an employee computer issues a request to the e-mail server for any unread mail. The server responds by sending the requested e-mail to the client. Although data is typically described as flowing from the server to the client, some data always flows from the client to the server.

Data flow may be equal in both directions, or may even be greater in the direction going from the client to the server. For example, a client may transfer a file to the server for storage purposes. Data transfer from a client to a server is referred to as an upload and data from a server to a client as a download.

Servers

In a general networking context, any device that responds to requests from client applications is functioning as a server. A server is usually a computer that contains information to be shared with many client systems.

For example, web pages, documents, databases, pictures, video, and audio files can all be stored on a server and delivered to requesting clients.

In a client/server network, the server runs a service, or process, sometimes called a server daemon. Like most services, daemons typically run in the background and are not under an end user's direct control.

Daemons are described as "listening" for a request from a client, because they are programmed to respond whenever the server receives a request for the service provided by the daemon.

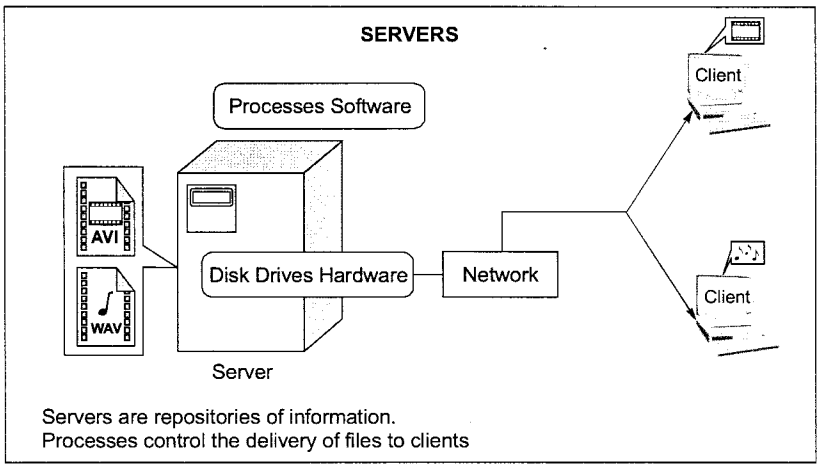
When a daemon "hears" a request from a client, it exchanges appropriate messages with the client, as required by its protocol, and proceeds to send the requested data to the client in the proper format.

Application Layer Services and Protocols

A single application may employ many different supporting Application layer services; thus what appears to the user as one request for a web page may, in fact, amount to dozens of individual requests. And for each request, multiple processes may be executed.

For example, a client may require several individual processes to formulate just one request to a server. Additionally, servers typically have multiple clients requesting information at the same time. For example, a Telnet server may have many clients requesting connections to it.

Individual client requests must be handled simultaneously and separately for the network to succeed. The Application layer processes and services rely on support from lower layer functions to successfully manage the multiple conversations.



6.3 Application Layer Protocols and Services Examples

DNS Services and Protocol

Transport layer uses an addressing scheme called a port number. Port numbers identify applications and Application layer services that are the source and destination of data.

For TCP/IP Application layer protocols and services, we will be referring to the TCP and UDP port numbers normally associated with these services. Some of these services are:

- Domain Name System (DNS): TCP/UDP Port 53
- Hypertext Transfer Protocol (HTTP): TCP Port 80
- Simple Mail Transfer Protocol (SMTP): TCP Port 25
- Post Office Protocol (POP): UDP Port 110
- Telnet: TCP Port 23
- Dynamic Host Configuration Protocol: UDP Port 67
- File Transfer Protocol (FTP): TCP Ports 20 and 21

DNS (Domain Name Service)

In data networks, devices are labelled with numeric IP addresses, so that they can participate in sending and receiving messages over the network. However, most people have a hard time remembering this numeric address.

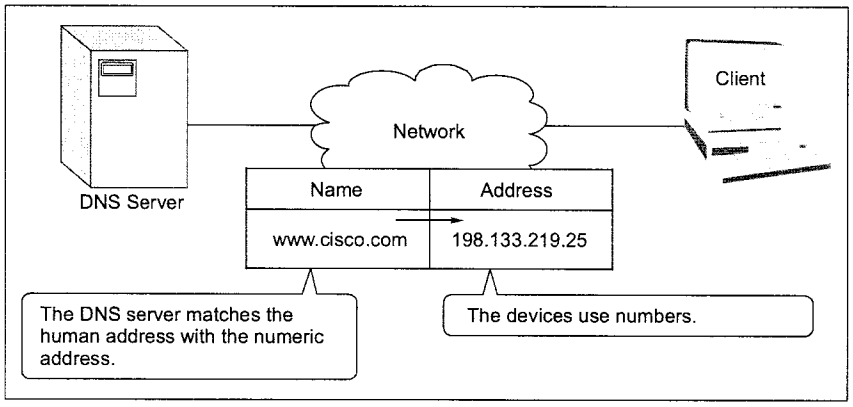
Hence, domain names were created to convert the numeric IP address into a simple, recognizable name.

On the Internet these domain names, such as www.cisco.com, are much easier for people to remember than 198.133.219.25, which is the actual numeric address for this server. Also, if Cisco decides to change the numeric address, it is transparent to the user, since the domain name will remain www.cisco.com. The new address will simply be linked to the existing domain name and connectivity is maintained.

When networks were small, it was a simple task to maintain the mapping between domain names and the addresses they represented. However, as networks began to grow and the number of devices increased, this manual system became unworkable.

The Domain Name System (DNS) was created for domain name to address resolution for these networks. DNS uses a distributed set of servers to resolve the names associated with these numbered addresses.

The DNS protocol defines an automated service that matches resource names with the required numeric network address. It includes the format for queries, responses, and data formats. DNS protocol communications use a single format called a message.



This message format is used for all types of client queries and server responses, error messages, and the transfer of resource record information between servers.

DNS is a client/server service; however, it differs from the other client/server services that we are examining. While other services use a client that is an application (such as web browser, e-mail client), the DNS client runs as a service itself.

The DNS client, sometimes called the DNS resolver, supports name resolution for our other network applications and other services that need it.

When configuring a network device, we generally provide one or more DNS Server addresses that the DNS client can use for name resolution.

Usually the Internet service provider provides the addresses to use for the DNS servers. When a user's application requests to connect to a remote device by name, the requesting DNS client queries one of these name servers to resolve the name to a numeric address.

Computer operating systems also have a utility called nslookup that allows the user to manually query the name servers to resolve a given host name. This utility can also be used to troubleshoot name resolution issues and to verify the current status of the name servers.

In the figure, when the nslookup is issued, the default DNS server configured for your host is displayed. In this example, the DNS server is dns-sjk.cisco.com which has an address of 171.68.226.120. We then can type the name of a host or domain for which we wish to get the address. In the first query in the figure, a query is made for www.cisco.com.

The responding name server provides the address of 198.133.219.25. The queries shown in the figure are only simple tests. The nslookup has many options available for extensive testing and verification of the DNS process. A DNS server provides the name resolution using the name daemon, which is often called named, (pronounced name-dee). The DNS server stores different types of resource records used to resolve names.

These records contain the name, address, and type of record. Some of these record types are:

- **A:** An end device address
- **NS:** An authoritative name server
- **CNAME:** The canonical name (or Fully Qualified Domain Name) for an alias; used when multiple services have the single network address but each service has its own entry in DNS
- **MX:** Mail exchange record; maps a domain name to a list of mail exchange servers for that domain.

When a client makes a query, the server's "named" process first looks at its own records to see if it can resolve the name. If it is unable to resolve the name using its stored records, it contacts other servers in order to resolve the name. The request may be passed along to a number of servers, which can take extra time and consume bandwidth. Once a match is found and returned to the original requesting server, the server temporarily stores the numbered address that matches the name in cache. If that same name is requested again, the first server can return the address by using the value stored in its name cache. Caching reduces both the DNS query data network traffic and the workloads of servers higher up the hierarchy. The DNS Client service on Windows PCs optimizes the performance of DNS name resolution by storing previously resolved names in memory, as well. The **ipconfig/displaydns** command displays all of the cached DNS entries on a Windows XP or 2000 computer system.

DNS Message Format

DNS uses the same message format for

- All types of client queries and server responses
- error messages
- the transfer of resource record information between servers

Header	
Question	The question for the name server
Answer	Resource Records answering the question
Authority	Resource Records pointing toward an authority
Additional	Resource Records holding additional information

The Domain Name System uses a hierarchical system to create a name database to provide name resolution.

The hierarchy looks like an inverted tree with the root at the top and branches below. At the top of the hierarchy, the root servers maintain records about how to reach the top-level domain servers, which in turn have records that point to the secondary level domain servers and so on. The different top-level domains represent either the type of organization or the country of origin.

Examples of top-level domains are:

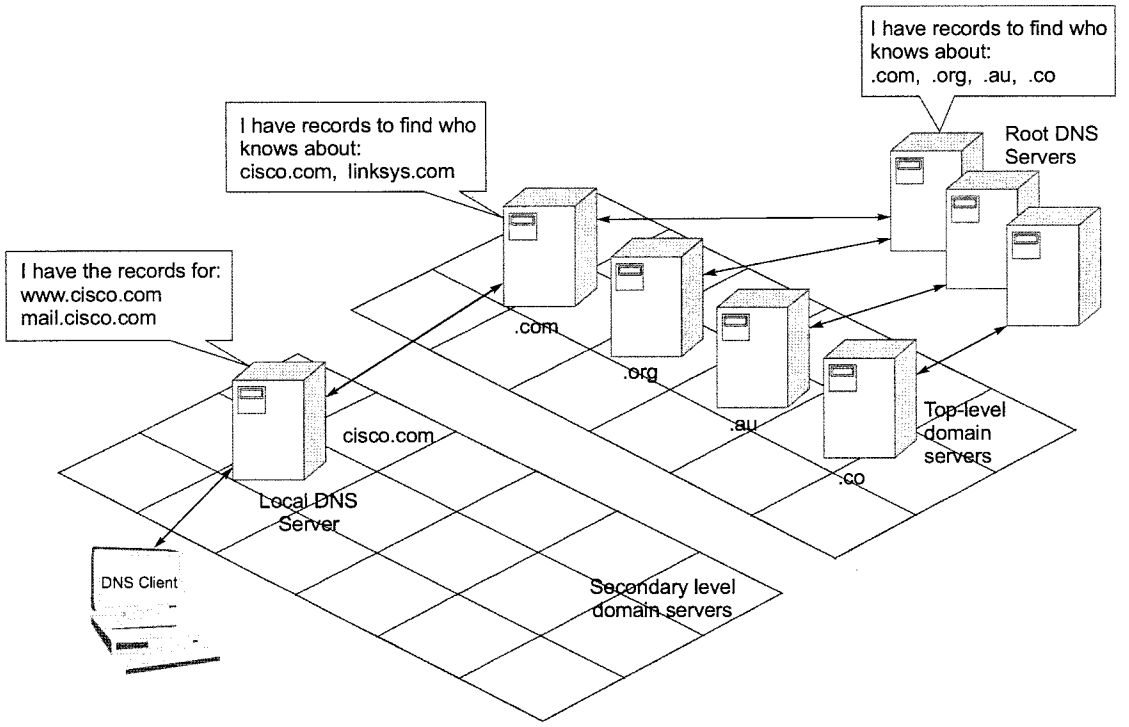
- .au - Australia
- .co - Colombia
- .com - a business or industry
- .jp - Japan
- .org - a non-profit organization

After top-level domains are second-level domain names, and below them are other lower level domains. Each domain name is a path down this inverted tree starting from the root. For example, as shown in the figure, the root DNS server may not know exactly where the e-mail server mail.cisco.com is located, but it maintains a record for the "com" domain within the top-level domain.

Likewise, the servers within the "com" domain may not have a record for mail.cisco.com, but they do have a record for the "cisco.com" domain.

The servers within the cisco.com domain have a record (a MX record to be precise) for mail.cisco.com. The Domain Name System relies on this hierarchy of decentralized servers to store and maintain these resource records. The resource records list domain names that the server can resolve and alternative servers that can also process requests. If a given server has resource records that correspond to its level in the domain hierarchy, it is said to be authoritative for those records.

For example, a name server in the cisco.netacad.net domain would not be authoritative for the mail.cisco.com record because that record is held at a higher domain level server, specifically the name server in the cisco.com domain.



A hierarchy of DNS servers contains the resource records that match names with addresses.

6.4 WWW Service and HTTP

When a web address (or URL) is typed into a web browser, the web browser establishes a connection to the web service running on the server using the HTTP protocol. URLs (or Uniform Resource Locator) and URIs (Uniform Resource Identifier) are the names most people associate with web addresses.

The URL http://www.mysite.com/index.html is an example of a URL that refers to a specific resource - a web page namedindex.html on a server identified as mysite.com (click the tabs in the figure to see the steps used by HTTP). Web browsers are the client applications our computers use to connect to the World Wide Web and access resources stored on a web server.

As with most server processes, the web server runs as a background service and makes different types of files available. In order to access the content, web clients make connections to the server and request the desired resources. The server replies with the resources and, upon receipt, the browser interprets the data and presents it to the user.

Browsers can interpret and present many data types, such as plain text or Hypertext Markup Language (HTML, the language in which web pages are constructed). Other types of data, however, may require another service or program, typically referred to as plug-ins or add-ons.

To help the browser determine what type of file it is receiving, the server specifies what kind of data the file contains. To better understand how the web browser and web client interact, we can examine how a web page is opened in a browser.

For this example, we will use the URL: http://www.mysite.com/web-server.html.

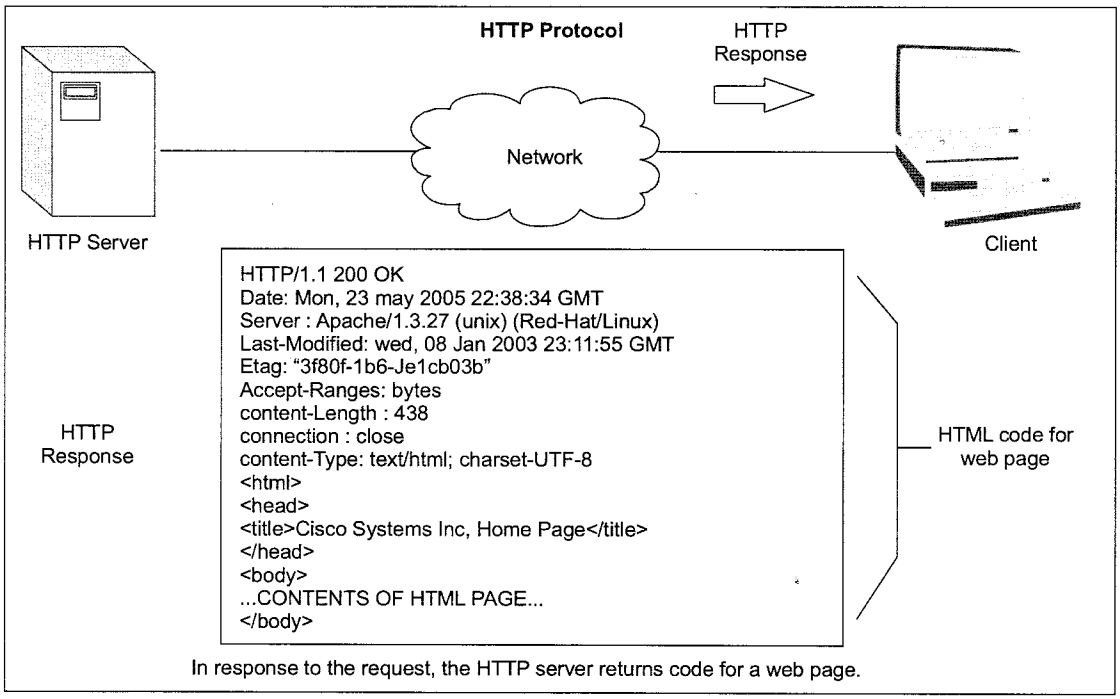
First, the browser interprets the three parts of the URL:

1. http (the protocol or scheme)
2. www.mysite.com (the server name)
3. web-server.htm (the specific file name requested).

The browser then checks with a name server to convert www.cisco.com into a numeric address, which it uses to connect to the server.

Using the HTTP protocol requirements, the browser sends a GET request to the server and asks for the file web-server.htm.

The server in turn sends the HTML code for this web page to the browser. Finally, the browser deciphers the HTML code and formats the page for the browser window.



In response to the request, the HTTP server returns code for a web page.

The Hypertext Transfer Protocol (HTTP), one of the protocols in the TCP/IP suite, was originally developed to publish and retrieve HTML pages and is now used for distributed, collaborative information systems.

HTTP is used across the World Wide Web for data transfer and is one of the most used application protocols. HTTP specifies a request/response protocol. When a client, typically a web browser, sends a request message to a server, the HTTP protocol defines the message types the client uses to request the web page and also the message types the server uses to respond.

The three common message types are GET, POST, and PUT. GET is a client request for data. A web browser sends the GET message to request pages from a web server.

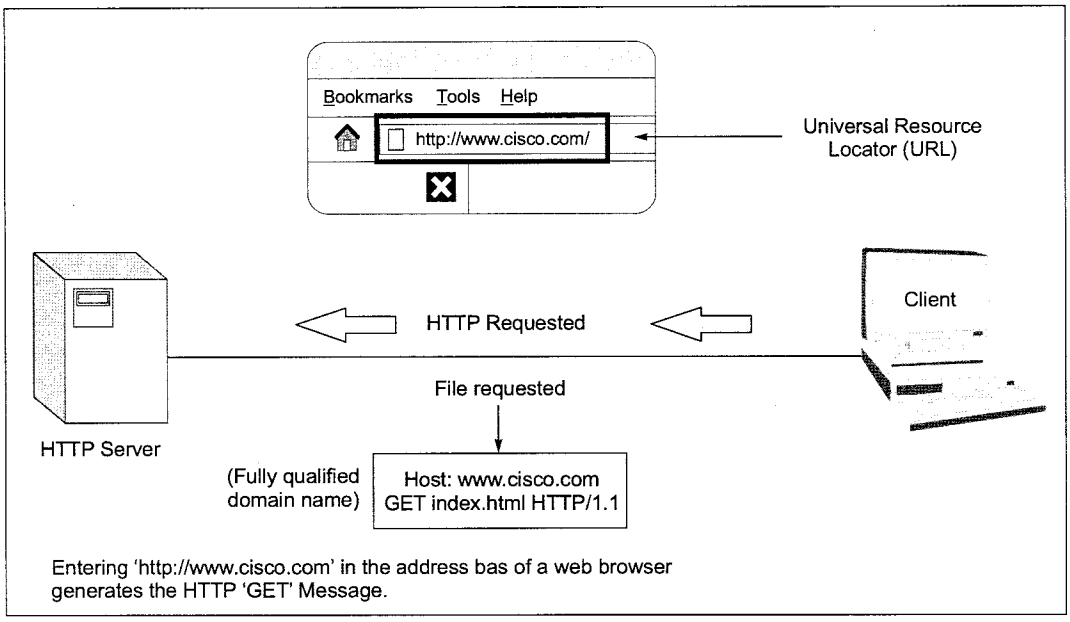
As shown in the figure, once the server receives the GET request, it responds with a status line, such as HTTP/1.1 200 OK, and a message of its own, the body of which may be the requested file, an error message, or some other information.

POST and PUT are used to send messages that upload data to the web server. For example, when the user enters data into a form embedded in a web page, POST includes the data in the message sent to the server. PUT uploads resources or content to the web server.

Although it is remarkably flexible, HTTP is not a secure protocol. The POST messages upload information to the server in plain text that can be intercepted and read.

Similarly, the server responses, typically HTML pages, are also unencrypted. For secure communication across the Internet, the HTTP Secure (HTTPS) protocol is used for accessing or posting web server information.

HTTPS can use authentication and encryption to secure data as it travels between the client and server. HTTPS specifies additional rules for passing data between the Application layer and the Transport Layer.



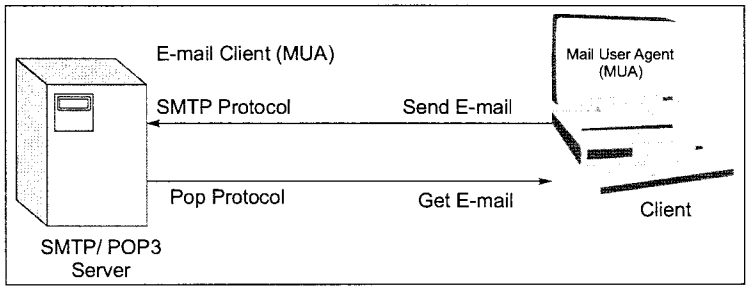
6.5 E-mail Services and SMTP/POP Protocols

E-mail, the most popular network service, has revolutionized how people communicate through its simplicity and speed. Yet to run on a computer or other end device, e-mail requires several applications and services.

Two example Application layer protocols are Post Office Protocol (POP) and Simple Mail Transfer Protocol (SMTP), shown in the Figure.

As with HTTP, these protocols define client/server processes. When people compose e-mail messages, they typically use an application called a Mail User Agent (MUA), or e-mail client.

The MUA allows messages to be sent and places received messages into the client's mailbox, both of which are distinct processes. In order to receive e-mail messages from an e-mail server, the e-mail client can use POP. Sending e-mail from either a client or a server uses message formats and command strings defined by the SMTP protocol. Usually an e-mail client provides the functionality of both protocols within one application.



E-mail Server Processes – MTA and MDA

The e-mail server operates two separate processes:

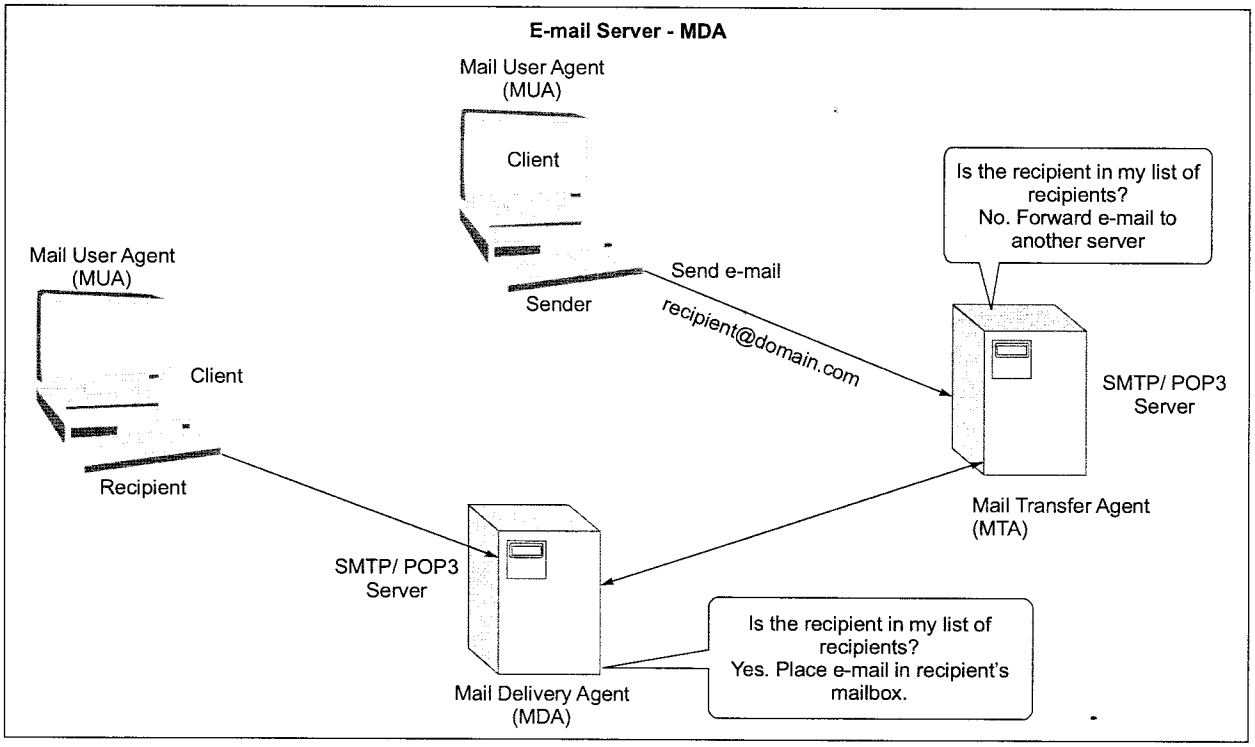
- Mail Transfer Agent (MTA)
- Mail Delivery Agent (MDA)

The Mail Transfer Agent (MTA) process is used to forward e-mail. As shown in the figure, the MTA receives messages from the MUA or from another MTA on another e-mail server. Based on the message header, it determines how a message has to be forwarded to reach its destination.

If the mail is addressed to a user whose mailbox is on the local server, the mail is passed to the MDA. If the mail is for a user not on the local server, the MTA routes the e-mail to the MTA on the appropriate server. In the figure, we see that the Mail Delivery Agent (MDA) accepts a piece of e-mail from a Mail Transfer Agent (MTA) and performs the actual delivery. The MDA receives all the inbound mail from the MTA and places it into the appropriate users' mailboxes. The MDA can also resolve final delivery issues, such as virus scanning, spam filtering, and return-receipt handling. Most e-mail communications use the MUA, MTA, and MDA applications. However, there are other alternatives for e-mail delivery.

A client may be connected to a corporate e-mail system, such as IBM's Lotus Notes, Novell's Groupwise, or Microsoft's Exchange. These systems often have their own internal e-mail format, and their clients typically communicate with the e-mail server using a proprietary protocol.

The server sends or receives e-mail via the Internet through the product's Internet mail gateway, which performs any necessary reformatting. If, for example, two people who work for the same company exchange e-mail with each other using a proprietary protocol, their messages may stay completely within the company's corporate e-mail system. As another alternative, computers that do not have an MUA can still connect to a mail service on a web browser in order to retrieve and send messages in this manner. Some computers may run their own MTA and manage inter-domain e-mail themselves.



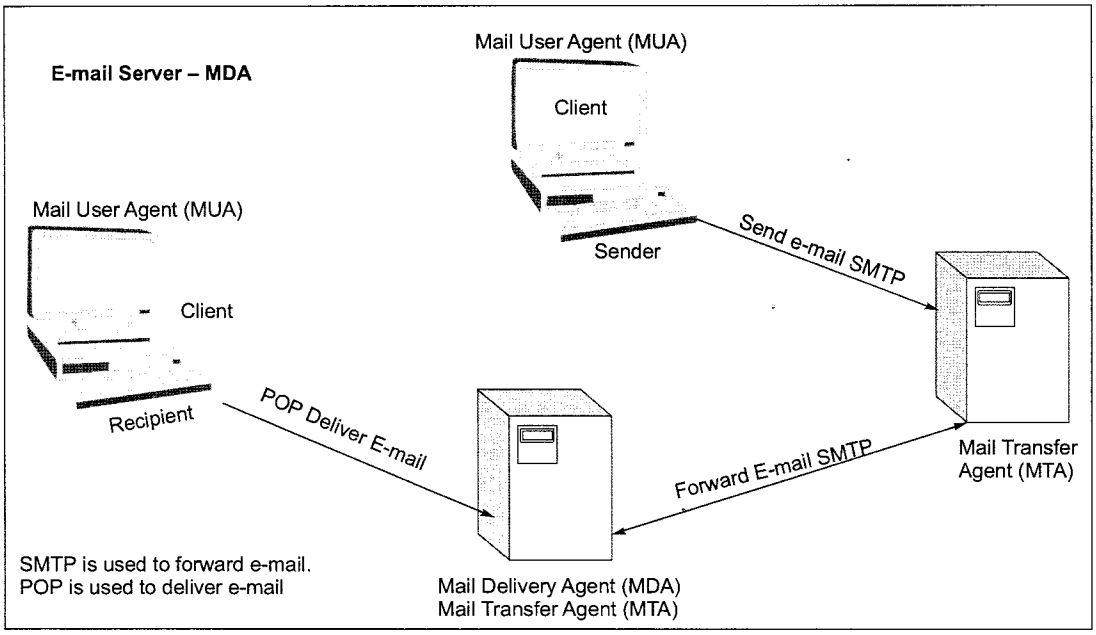
As mentioned earlier, e-mail can use the protocols, POP and SMTP (see the figure for an explanation of how they each work). POP and POP3 (Post Office Protocol, version 3) are inbound mail delivery protocols and are typical client/server protocols.

They deliver e-mail from the e-mail server to the client (MUA). The MDA listens for when a client connects to a server. Once a connection is established, the server can deliver the e-mail to the client. The Simple Mail Transfer Protocol (SMTP), on the other hand, governs the transfer of outbound e-mail from the sending client to the e-mail server (MDA), as well as the transport of e-mail between e-mail servers (MTA).

SMTP enables e-mail to be transported across data networks between different types of server and client software and makes e-mail exchange over the Internet possible. The SMTP protocol message format uses a rigid set of commands and replies.

These commands support the procedures used in SMTP, such as session initiation, mail transaction, forwarding mail, verifying mailbox names, expanding mailing lists, and the opening and closing exchanges. Some of the commands specified in the SMTP protocol are:

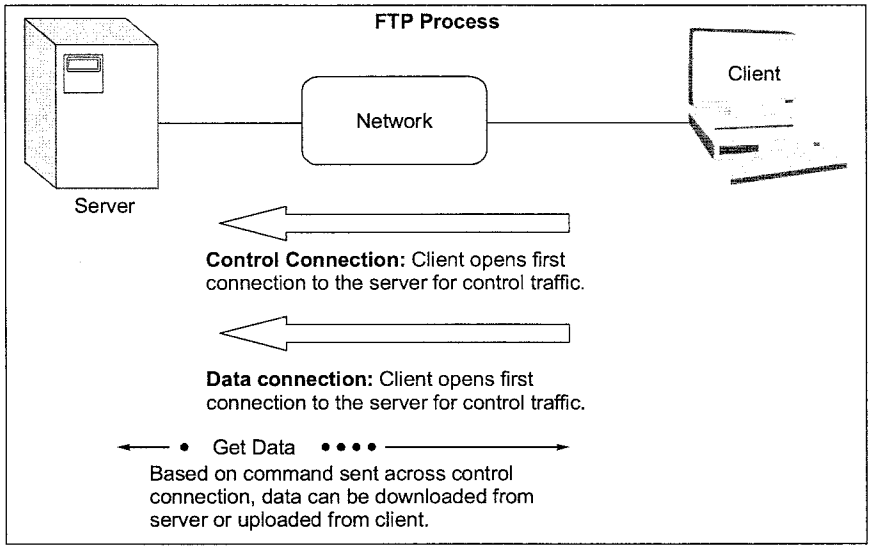
- **HELO:** Identifies the SMTP client process to the SMTP server process
- **EHLO:** Is a newer version of HELO, which includes services extensions
- **MAIL FROM:** Identifies the sender
- **RCPT TO:** Identifies the recipient
- **DATA:** Identifies the body of the message



6.6 FTP (File Transfer Protocol)

The File Transfer Protocol (FTP) is another commonly used Application layer protocol. FTP was developed to allow for file transfers between a client and a server. An FTP client is an application that runs on a computer that is used to push and pull files from a server running the FTP daemon (FTPd). To successfully transfer files, FTP requires two connections between the client and the server: one for commands and replies, the other for the actual file transfer. The client establishes the first connection to the server on TCP port 21. This connection is used for control traffic, consisting of client commands and server replies. The client establishes the second connection

to the server over TCP port 20. This connection is for the actual file transfer and is created every time there is a file transferred. The file transfer can happen in either direction. The client can download (pull) a file from the server or, the client can upload (push) a file to the server.



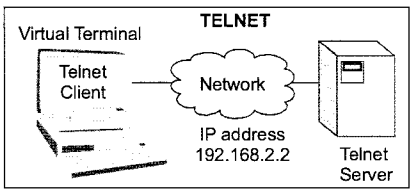
6.7 Telnet Services and Protocol

Long before desktop computers with sophisticated graphical interfaces existed, people used text-based systems which were often just display terminals physically attached to a central computer.

Once networks were available, people needed a way to remotely access the computer systems in the same manner that they did with the directly attached terminals.

Telnet was developed to meet that need. Telnet dates back to the early 1970s and is among the oldest of the Application layer protocols and services in the TCP/IP suite. Telnet provides a standard method of emulating text-based terminal devices over the data network. Both the protocol itself and the client software that implements the protocol are commonly referred to as Telnet. Appropriately enough, a connection using Telnet is called a Virtual Terminal (VTY) session, or connection.

Rather than using a physical device to connect to the server, Telnet uses software to create a virtual device that provides the same features of a terminal session with access to the server command line interface (CLI). To support Telnet client connections, the server runs a service called the Telnet daemon. A virtual terminal connection is established from an end device using a Telnet client application.



Most operating systems include an Application layer Telnet client. On a Microsoft Windows PC, Telnet can be run from the command prompt. Other common terminal applications that run as Telnet clients are HyperTerminal, Minicom, and TeraTerm. Once a Telnet connection is established, users can perform any authorized function on the server, just as if they were using a command line session on the server itself. If authorized, they can start and stop processes, configure the device, and even shut down the system.

Telnet is a client/server protocol and it specifies how a VTY session is established and terminated. It also provides the syntax and order of the commands used to initiate the Telnet session, as well as control commands that can be issued during a session. Each Telnet command consists of at least two bytes. The first byte is a special character called the Interpret as Command (IAC) character. As its name implies, the IAC defines the next byte as a command rather than text.

Some sample Telnet protocol commands include:

- **Are You There (AYT):** Lets the user request that something appear on the terminal screen to indicate that the VTY session is active.
- **Erase Line (EL):** Deletes all text from the current line.
- **Interrupt Process (IP):** Suspends, interrupts, aborts, or terminates the process to which the Virtual Terminal is connected. For example, if a user started a program on the Telnet server via the VTY, he or she could send an IP command to stop the program.

While the Telnet protocol supports user authentication, it does not support the transport of encrypted data. All data exchanged during a Telnet sessions is transported as plain text across the network. This means that the data can be intercepted and easily understood.

If security is a concern, the Secure Shell (SSH) protocol offers an alternate and secure method for server access. SSH provides the structure for secure remote login and other secure network services.

Summary

- Domain Name Service Protocol (DNS) is used to resolve Internet names to IP addresses.
- Hypertext Transfer Protocol (HTTP) is used to transfer files that make up the Web pages of the World Wide Web.
- Simple Mail Transfer Protocol (SMTP) is used for the transfer of mail messages and attachments.
- Telnet, a terminal emulation protocol, is used to provide remote access to servers and networking devices.
- File Transfer Protocol (FTP) is used for interactive file transfer between systems.
- Application layer **uses protocols that are implemented within applications and services.**
- Application layer ISO OSI protocols are used by both the source and destination devices during a communication session.
- Data transfer from a client to a server is referred to as an upload and data from a server to a client as a download.
- In a general networking context, any device that responds to requests from client applications is functioning as a server.
- The DNS protocol defines an automated service that matches resource names with the required numeric network address. It includes the format for queries, responses, and data formats.
- The e-mail server operates two separate processes: (a) Mail Transfer Agent (MTA) and (b) Mail Delivery Agent (MDA).
- The Mail Transfer Agent (MTA) process is used to forward e-mail. As shown in the figure, the MTA receives messages from the MUA or from another MTA on another e-mail server.
- SMTP enables e-mail to be transported across data networks between different types of server and client software and makes e-mail exchange over the Internet possible. The SMTP protocol message format uses a rigid set of commands and replies.
- An FTP client is an application that runs on a computer that is used to push and pull files from a server running the FTP daemon (FTPd). To successfully transfer files, FTP requires two connections between the client and the server: one for commands and replies, the other for the actual file transfer.
- The Dynamic Host Configuration Protocol (DHCP) service enables devices on a network to obtain IP addresses and other information from a DHCP server.

Student's Assignment

Q.1 Which of the following statements are true?

S_1 : UDP of transport layer protocol uses FTP

S_2 : TCP of transport layer protocol uses TFTP

S_3 : TCP uses both FTP and TFTP.

(a) Only S_1 (b) Only S_3

(c) Only S_1 and S_2 (d) None of these

Q.2 Which of the following is true about Flow Control in FTP and TFTP respectively?

(a) YES, YES (b) NO, NO

(c) YES, NO (d) NO, YES

Q.3 Match the following control commands and their respective port numbers

List-I	List-II
A. FTP control connection	1. 23
B. FTP data connection	2. 21
C. TELNET	3. 20

Codes:

	A	B	C
(a)	1	2	3
(b)	2	3	1
(c)	3	1	2
(d)	1	3	2

Q.4 Which of the following is true about SMTP protocol?

S_1 : Can send image files with the help of POP_3

S_2 : Can send image files with the help of $IMAP_4$

S_3 : $IMAP_4$ is more secure than POP_3

(a) Only S_1 (b) Only S_2

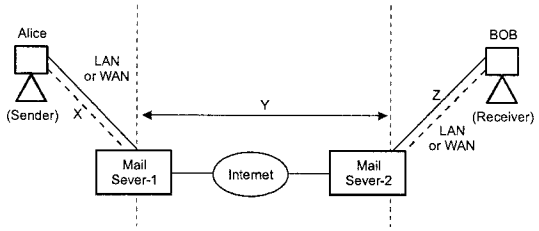
(c) S_2 and S_3 only (d) S_3 Only

Q.5 Which of the following protocol allows non-ASCII data to be sent through e-mail?

(a) POP_3 (b) $IMAP_4$

(c) TELNET (d) MIME

Q.6 Identify the protocols X, Y and Z respectively used for mail transfer (Assume mail sent by Alice is received by Bob).



(a) POP_3, SMTP, $IMAP_4$

(b) $IMAP_4$, SMTP, SMIP

(c) SMTP, POP_3, SMTP

(d) SMIP, SMTP, $IMAP_4$

Q.7 Which of the following statement is incorrect?

(a) A reliable data transfer protocol may send multiple packets without waiting for acknowledgments, rather than operating in a stop and wait manner. This technique is called "Pipelining'.

(b) A process sends/receives messages to/ from the network through a software interface called a "Socket".

(c) Because an HTTP server maintains no information about the clients, an HTTP server said to be "Statefull".

(d) The "Traceroute" can be used to determine the number of hops to a destination and the round trip time for each hop.

Q.8 Match List-I with List-II

List-I	List-II
A. DNS	1. port - 20
B. POP3	2. port - 21
C. FTP (Data)	3. port - 53
D. FTP (Control)	4. port - 110
	5. port - 69

Codes:

	A	B	C	D
(a)	3	4	2	1
(b)	3	4	1	2
(c)	3	5	1	2
(d)	3	5	2	1

Q.9 The packet of information at the application layer is called

(a) packet (b) message

(c) segment (d) frame

Q.10 This is one of the architecture paradigm
(a) Peer to peer (b) Client-server
(c) HTTP (d) Both (a) and (b)

Q.11 Application developer has permission to decide the following on transport layer side
(a) Transport layer protocol
(b) Maximum buffer size
(c) Both (a) and (b)
(d) None of these

Q.12 e-mail is
(a) Loss-tolerant application
(b) Bandwidth-sensitive application
(c) Elastic application
(d) None of these

Q.13 To deliver a message to the correct application program running on a host, the address must be consulted
(a) IP (b) MAC
(c) Port (d) IP and MAC

Q.14 This is a Time-sensitive service
(a) File transfer (b) File download
(c) e-mail (d) Internet telephony

Q.15 The _____ layer is the top most layer in the subnet
(a) Network (b) Application
(c) Transport (d) Physical

Q.16 An application process is assigned a process identifier number (Process ID), which is likely to be _____ each time that process is started
(a) Similar (b) Smaller
(c) Different (d) Larger

Q.17 The post office protocol is an _____ protocol with both client (sender/receiver) and _____ functions
(a) Electronic mail, server (storage)
(b) Three layer, server
(c) UDP, transfer
(d) TCP, server

Q.18 The purpose of a proxy server is to control exchange of data between the two networks at _____ instead of _____.
(a) an application layer, an IP layer
(b) network layer, physical layer
(c) an application layer, physical layer
(d) network layer, an application layer

Q.19 PGP is one of the protocol used to provide security at the _____ it is designed to create authenticated and confidential _____.
(a) application layer, e-mail
(b) network layer, packets
(c) application layer, packets
(d) network layer, e-mail

Answer Key:

1. (d)	**2.** (d)	**3.** (b)	**4.** (d)	**5.** (d)
6. (d)	**7.** (c)	**8.** (b)	**9.** (b)	**10.** (d)
11. (c)	**12.** (c)	**13.** (c)	**14.** (d)	**15.** (a)
16. (c)	**17.** (a)	**18.** (a)	**19.** (a)	

CHAPTER 07

Network Security

7.1 Cryptography

The word **cryptography** refers to the tools and techniques used to make messages secure for communication between the participants and make messages immune to attacks by hackers.

7.1.1 Symmetric Key Cryptography

Symmetric key cryptography is also called as Private Key cryptography.

We use the same cryptographic **keys** for both encryption of plaintext and decryption of cipher text. The **keys** may be identical or there may be a simple transformation to go between the two **keys.**

The algorithm used to decrypt is just the inverse of the algorithm used for encryption. For example, if addition and division is used for encryption, multiplication and subtraction are to be used for decryption.

Symmetric key cryptography algorithms are simple requiring lesser execution time. As a consequence, these are commonly used for long messages. However, these algorithms suffer from the following limitations:

- Requirement of large number of unique keys. For example for n users the number of keys required is n(n-1)/2.
- Distribution of keys among the users in a secured manner is difficult.

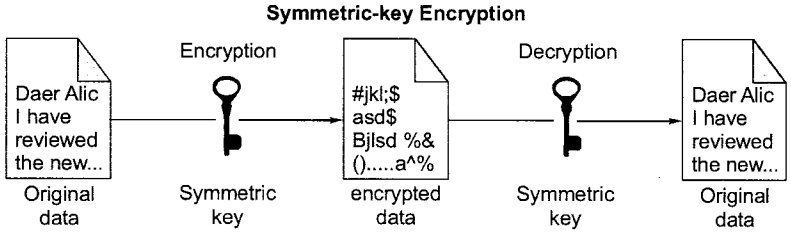


Figure: *A simple symmetric key cryptography model*

Monoalphabetic Substitution

One simple example of symmetric key cryptography is the *Monoalphabetic substitution.* In this case, the relationship between a character in the plaintext and a character in the cipher text is always one-to-one.

ROT13 is a Caesar cipher, a type of substitution cipher. In ROT13, the alphabet is rotated 13 steps.

Character in the cipher text is substituted by another character shifted by 13 places. Key feature of this approach is that it is very simple but the code can be attacked very easily.

ROT13 replaces each letter by its partner 13 characters further along the alphabet. For example, HELLO becomes URYYB (or, reversing, URYYB becomes HELLO again).

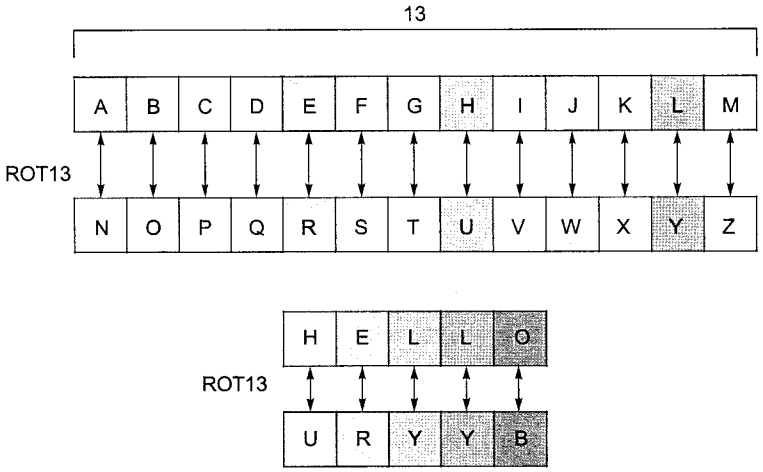


Figure: *ROT13 is a Caesar cipher, a type of substitution cipher*

Polyalphabetic Substitution

This is an improvement over the Caesar cipher. Here the relationship between a character in the plaintext and a character in the cipher text is always one-to-many.

Example of Polyalphabetic substitution is the Vigenere cipher. In this case, a particular character is substituted by different characters in the cipher text depending on its position in the plaintext. Figure 7.3 explains the Polyalphabetic substitution. Here the top row shows different characters in the plaintext and the characters in different bottom rows show the characters by which a particular character is to be replaced depending upon its position in different rows from row-0 to row-25.

Key feature of this approach is that it is more complex and the code is harder to attack successfully.

Character in plaintext

	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
0	W R K D O V C A S B Y Q M L H I T U F E Z N G J P X
1	H Q B G W E R K F C O A Z J M S L V N I P U D T X Y
2	P I D Z X V S T O C M J N L B Q R U W K H G E F A Y
⋮	⋮
25	M C I D A X V S T O N L K U R E W Z H F P G Y J B Q

Character in plaintext

Figure: *Polyalphabetic substitution*

Transpositional Cipher

The transpositional cipher, the characters remain unchanged but their **positions are changed** to create the cipher text. Figure illustrates how five lines of a text get modified using transpositional cipher.

The characters are arranged in two-dimensional matrix and columns are interchanged according to a key is shown in the middle portion of the diagram.

The key defines which columns are to be swapped. Decryption can be done by swapping in the reverse order using the same key.

Transpositional cipher is also not a very secure approach. The attacker can find the plaintext by trial and error utilizing the idea of the frequency of occurrence of characters.

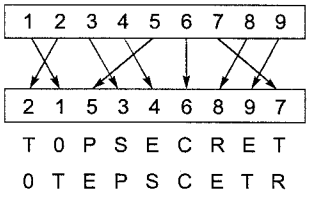


Figure: *Operation of a transpositional cipher*

Block ciphers

Block ciphers use a block of bits as the unit of encryption and decryption.

To encrypt a 64-bit block, one has to take each of the 264 input values and map it to one of the 264 output values. The **mapping should be one-to-one**.

Some operations, such as permutation and substitution, are performed on the block of bits based on a key (a secret number) to produce another block of bits.

In the decryption process, operations are performed in the reverse order based on the same key to get back the original block of bits.

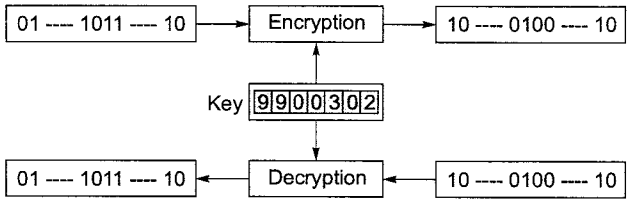


Figure: *Transformations in Block Ciphers*

Permutation: As shown in Figure, the permutation is performed by a permutation box at the bit-level, which keeps the number of 0s and 1s same at the input and output. Although it can be implemented either by hardware or software, the hardware implementation is faster.

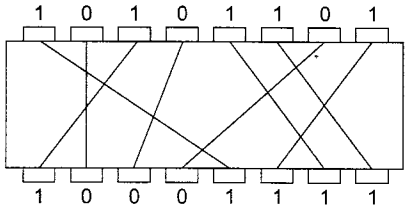


Figure: *Permutation operation used in Block Ciphers*

Substitution: As shown in Figure, the substitution is implemented with the help of three building blocks – a decoder, one p-box and an encoder. For an n-bit input, the decoder produces a 2^n bit output having only one 1, which is applied to the P-box. The P-box permutes the output of the decoder and it is applied to the encoder. The encoder, in turn, produces an n-bit output. For example, if the input to the decoder is 011, the output of the decoder is 00001000. Let the permuted output is 01000000, the output of the encoder is 011.

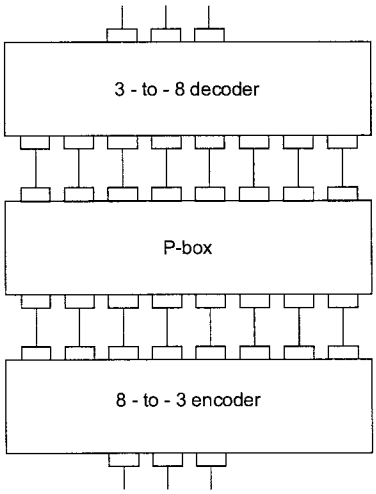


Figure: *Substitution operation used in Block Ciphers*

A block Cipher: A block cipher realized by using substitution and permutation operations is shown in Figure 7.8. It performs the following steps:

Step-1: Divide input into 8-bit pieces

Step-2: Substitute each 8-bit based on functions derived from the key

Step-3: Permute the bits based on the key

All the above three steps are repeated for an optimal number of rounds.

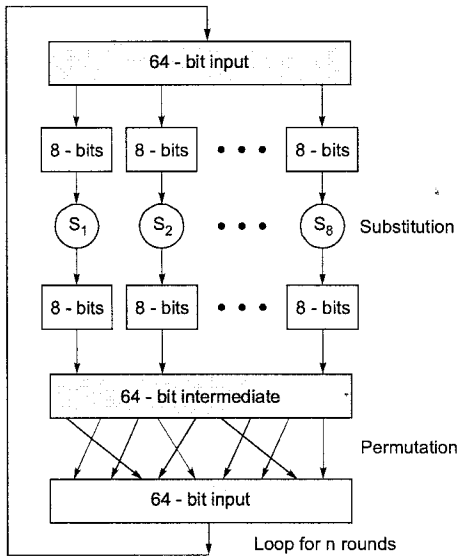


Figure: *Encryption by using substitution and permutation*

7.2 Public Key Cryptography

Asymmetric key cryptography is also called as Public Key cryptography. In public key cryptography, there are two keys: a private key and a public key. The public key is announced to the public; where as the private key is kept by the receiver. The **sender uses the public key of the receiver for encryption** and the receiver uses his private key for decryption.

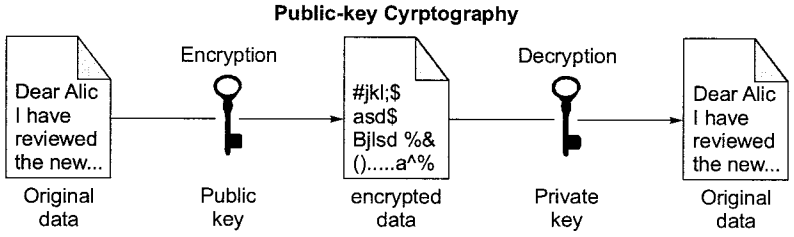


Figure: *Public key encryption technique*

Advantages

- The pair of keys can be used with any other entity
- The number of keys required is small

Disadvantages

- It is not efficient for long messages
- Association between an entity and its public key must be verified

7.2.1 RSA Algorithm

The most popular public-key algorithm is the RSA (named after their inventors Rivest, Shamir and Adelman). Key features of the RSA algorithm are given below:

- Public key algorithm that performs encryption as well as decryption based on number theory
- Variable key length; long for enhanced security and short for efficiency (typical 512 bytes)
- Variable block size, smaller than the key length
- The private key is a pair of numbers (d, n) and the public key is also a pair of numbers (e, n)
- Choose two large primes p and q (typically around 256 bits)
- Compute $n = p \times q$ and $z = (p - 1) \times (q - 1)$
- Choose a number d relatively prime to z
- Find e such that $e \times d \bmod (p - 1) \times (q - 1) = 1$
- For encryption: $C = Pe \pmod n$ For decryption: $P = Cd \pmod n$

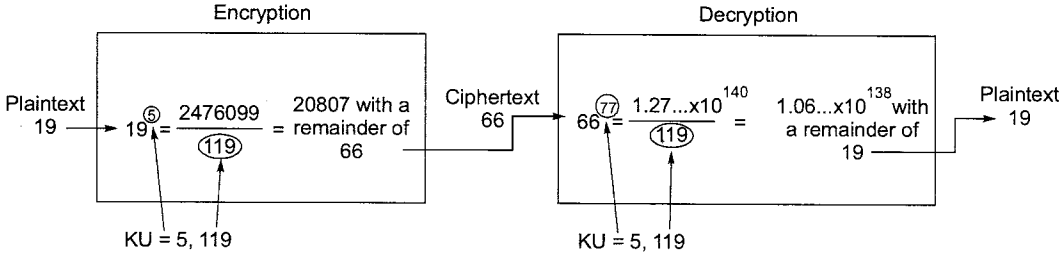


Figure: *The RSA public key encryption technique*

7.3 Secured Communication

7.3.1 Introduction

The basic objective is to communicate securely over an insecure medium. Any action that compromises the security of information can be considered as attack on security. Possible type of attacks mentioned below:

- **Interruption:** It is an attack on the availability of information by cutting wires, jamming wireless signals or dropping of packets by a switch.

- **Interception:** As a message is communicated through a network, eavesdroppers can listen in use it for his/her own benefit and try to tamper it.
- **Modification:** As a message is communicated through a network, eavesdroppers can intercept it and send a modified message in place of the original one.
- **Fabrication:** A message may be sent by a stranger by posing as a friend. This is also known as impersonation.

These attacks can be prevented with the help of several services implemented with the help of cryptography, as mentioned in the following section.

7.3.2 Security Services

Secured communication requires the following four basic services:

- **Privacy:** A person (say Sita) should be able to send a message to another person (say Ram) privately. It implies that to all others the message should be unintelligible.
- **Authentication:** After the message is received by Ram, he should be sure that the message has been sent by nobody else but by Sita.
- **Integrity:** Ram should be sure that the message has not been tampered on transit.
- **Nonrepudiation:** Ram should be able to prove at a later stage that the message was indeed received from Sita.

Privacy

Privacy can be achieved using symmetric key cryptography. In this case, the key is shared between the sender (Sita) and the receiver (Ram). Privacy can also be achieved by using public-key cryptography. However, in this case the owner should be verified. Digital Signature provides the remaining three security services; Authentication, Integrity and No repudiation.

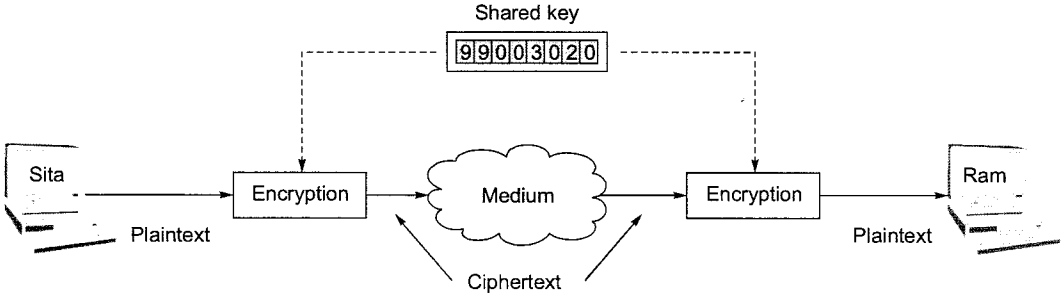


Figure: *Privacy using private-key cryptography*

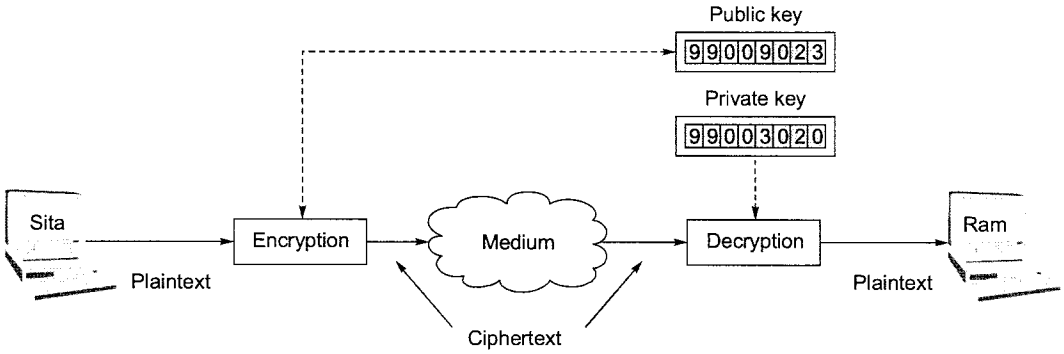


Figure: *Privacy using public-key cryptography*

Authentication, Integrity and Nonrepudiation Using DIGITAL SIGNATURE

By message authentication we mean that the receiver should be sure about sender's identity. One approach to provide authentication is with the help of digital signature. The idea is similar to signing a document.

7.3.3 Digital Signature

There are two alternatives for Digital Signature:

- Signing the entire document
- Signing the digest

In the first case the entire document is encrypted using private key of the sender and at the receiving end it is decrypted using the public key of the sender. For a large message this approach is very inefficient. In the second case a miniature version of the message, known as *digest*, is encrypted using the private key of the sender and then the signed digest along with the message is sent to the receiver.

The receiver decrypts the signed digest using the public key of the sender and the digest created using the received message is compared with the decrypted digest. If the two are identical, it is assumed that the sender is authenticated. This is somewhat similar to error detection using parity bit.

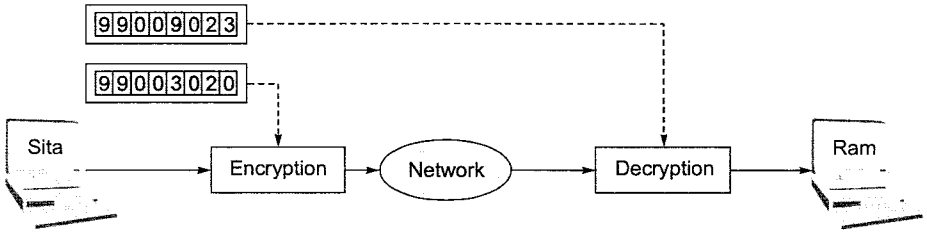


Figure: *Authentication by signing the whole document*

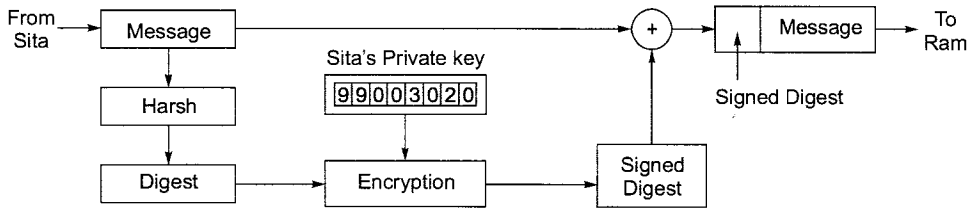


Figure: *Sender site for authentication by signed digest*

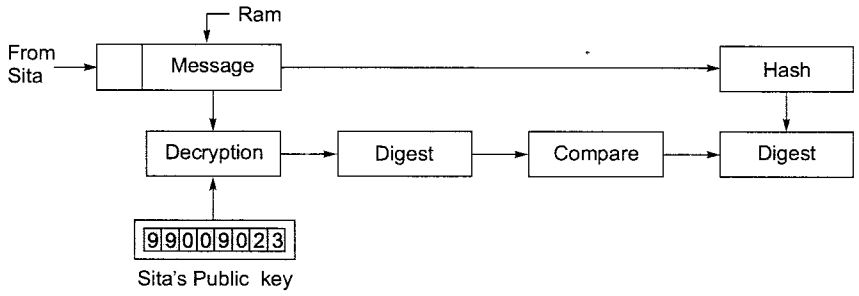


Figure: *Receiver site for authentication by signed digest*

Some key features of this approach are mentioned below:

- Digital signature does not provide privacy
- Hash function is used to create a message digest

- It creates a fixed-length digest from a variable-length message
- Most common Hash functions:
  (a) MD5 (Message Digest 5): 120-bit
  (b) SHA-1 (Secure Hash algorithm 1): 160-bit
- Important properties:
  (a) One-to-One
  (b) One-way

7.3.4 User Authentication Using Symmetric Key Cryptography

User authentication is different from message authentication. In case of message authentication, the identity of the sender is verified for each and every message. On the other hand, in user authentication, the user authentication is performed once for the duration of system access.

In the first approach, the sender (Sita) sends her identity and password in an encrypted message using the symmetric-key KSR and then sends the message as shown in Figure. However, an intruder (say Ravana) can cause damage without accessing it. He can also intercept both the authentication message and the data message, store them and then resends them, which is known as *replay attack*.

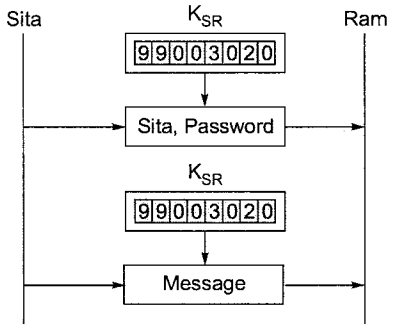


Figure: *User authentication using symmetric key cryptography*

Using nonce, a large random number used only once

To prevent the replay attack, the receiver (Ram) sends *nonce*, a large random number that is used only once to the sender (Sita) to challenge Sita.

In response Sita sends an encrypted version of the random number using the symmetric key. The procedure is shown in Figure.

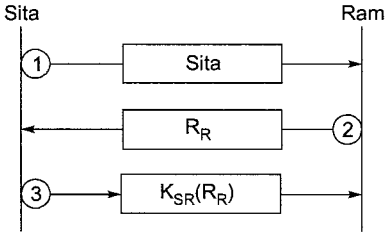


Figure: *User authentication using a nonce*

Bidirectional Authentication

In the bidirectional authentication approach, Ram sends *nonce* to challenge Sita and Sita in turn sends nonce to challenge Ram as shown in Figure (a). This protocol uses extra messages for user authentication. Protocol with lesser number of messages is possible as shown in Figure (b).

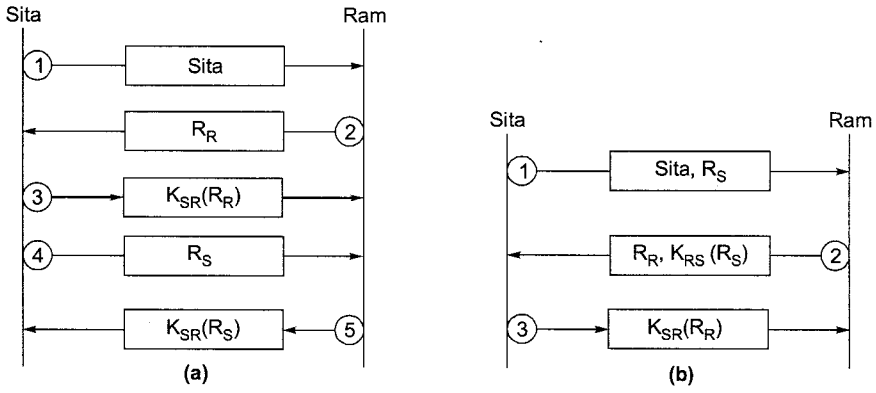


Figure: ***(a)*** *Bidirectional authentication using a nonce* ***(b)*** *Bidirectional authentication using lesser number of messages*

7.3.5 User Authentication Using Public Key Cryptography

Public key cryptography can also be used to authenticate a user.

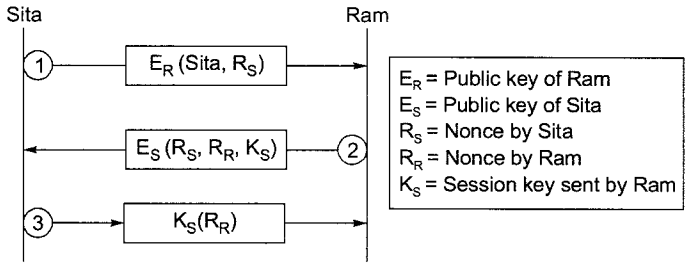


Figure: *User authentication using public key cryptography*

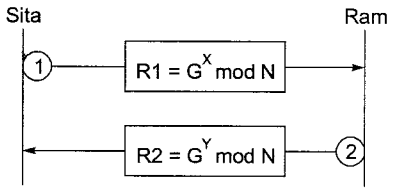
7.4 Key Management

Although symmetric-key and public-key cryptography can be used for privacy and user authentication, question arises about the techniques used for the distribution of keys. Particularly, symmetric-key distribution involves the following three problems:

- For n people to communicate with each other requires n(n-1)/2 keys. The problem is aggravated as n becomes very large.
- Each person needs to remember (n-1) keys to communicate with the remaining (n-1) persons.
- How the two parties will acquire the shared key in a secured manner?

In view of the above problems, the concept of *session key* has emerged. A session key is created for each session and destroyed when the session is over. The **Diffie-Hellman** protocol is one of the most popular approach for providing one-time session key for both the parties.

7.4.1 Diffie-Hellman Protocol



- Used to establish a shared secret key
- **Prerequisite:** N is a large prime number such that (N-1)/2 is also a prime number. G is also a prime number. Both N and G are known to Ram and Sita..
- Sita chooses a large random number X and calculates $R1 = G^X \bmod N$ and sends it to Ram
- Ram chooses another large random number y and calculates $R2 = G^Y \bmod N$ and sends it to Sita
- Ram calculates $K = (R1)^Y \bmod N$
- Sita calculates $K = (R2)^X \bmod N$

7.4.2 Key Management using KDC

It may be noted that both **Ya** and **yb** are sent as plaintext, which may be intercepted by an intruder. This is a serious flaw of the Diffie-Hellman Protocol. Another approach is to use a trusted third party to assign a symmetric key to both the parties. This is the basic idea behind the use of *key distribution center (KDC).*

7.4.3 Key Management using Kerberos

Another popular authentication protocol known as *Kerberos.* It uses an authentication server (AS), which performs the role of KDC and a ticket-granting server (TGS), which provides the session key (KAB) between the sender and receiver parties.

Apart from these servers, there is the real data server say Ram that provides services to the user Sita. The operation of Kerberos is depicted with the help of Figure. The client process (Sita) can get a service from a process running in the real server Ram after six steps as shown in the Figure. The steps are as follows:

Step-1: Sita uses her registered identity to send her message in plaintext.

Step-2: The AS server sends a message encrypted with Sita's symmetric key KS. The message contains a session key Kse, which is used by Sita to contact the TGS and a ticket for TGS that is encrypted with the TGS symmetric key KTG.

Step-3: Sita sends three items to the TGS; the ticket received from the AS, the name of the real server, and a timestamp encrypted by Kse. The timestamp prevents replay by Ram.

Step-4: The TGS sends two tickets to Sita. The ticket for Sita encrypted with Kse and the ticket for Ram encrypted with Ram's key. Each of the tickets contains the session key KSR between Sita and Ram.

Step-5: Sita sends Ram's ticket encrypted by KSR.

Step-6: Ram sends a message to Sita by adding 1 to the timestamp confirming the receipt of the message using KSR as the key for encryption. Following this Sita can request and get services from Ram using KSR as the shared key.

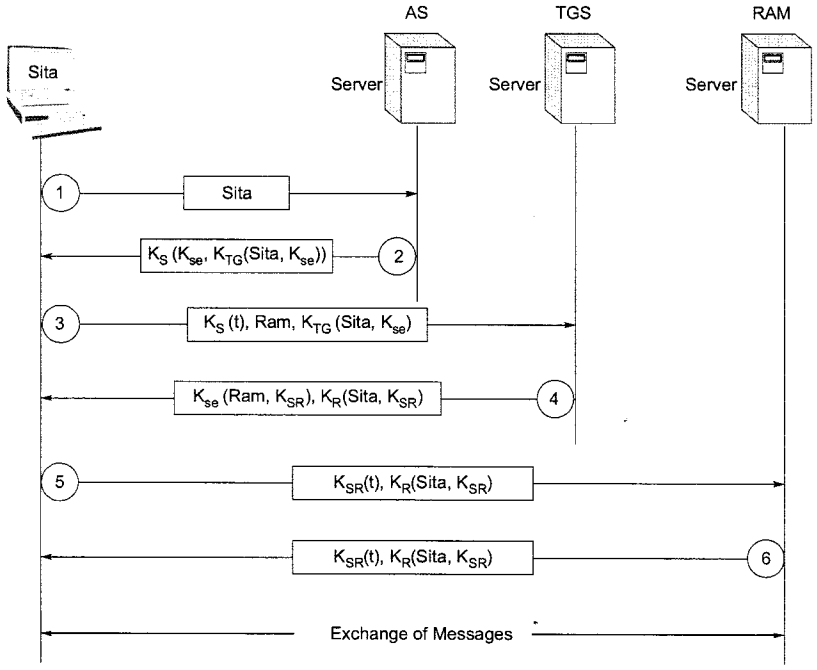


Figure: *The Kerberos Protocol*

7.5 Application Layer Security

Based on the encryption techniques we have discussed so far, security measures can be applied to different layers such as network, transport or application layers. However, implementation of security features in the application layer is far simpler and feasible compared to implementing at the other two lower layers. In this subsection, a protocol known as Pretty Good Privacy (PGP), invented by Phil Zinmermann, that is used in the application layer to provide all the four aspects of security for sending an email is briefly discussed. PGP uses a combination of private-key and public key for privacy. For integrity, authentication and nonrepudiation, it uses a combination of hashing to create digital signature and public-key encryption as shown in figure.

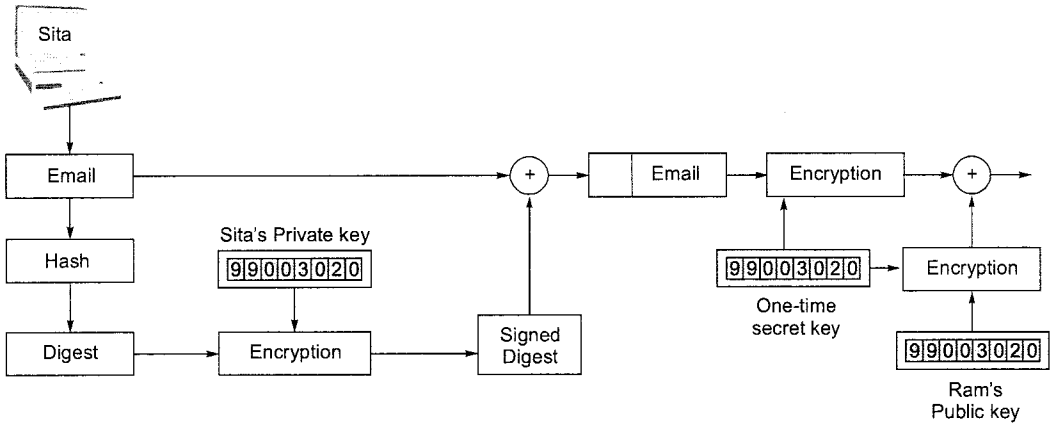


Figure: ***(a)*** *The sender's site of the PGP*

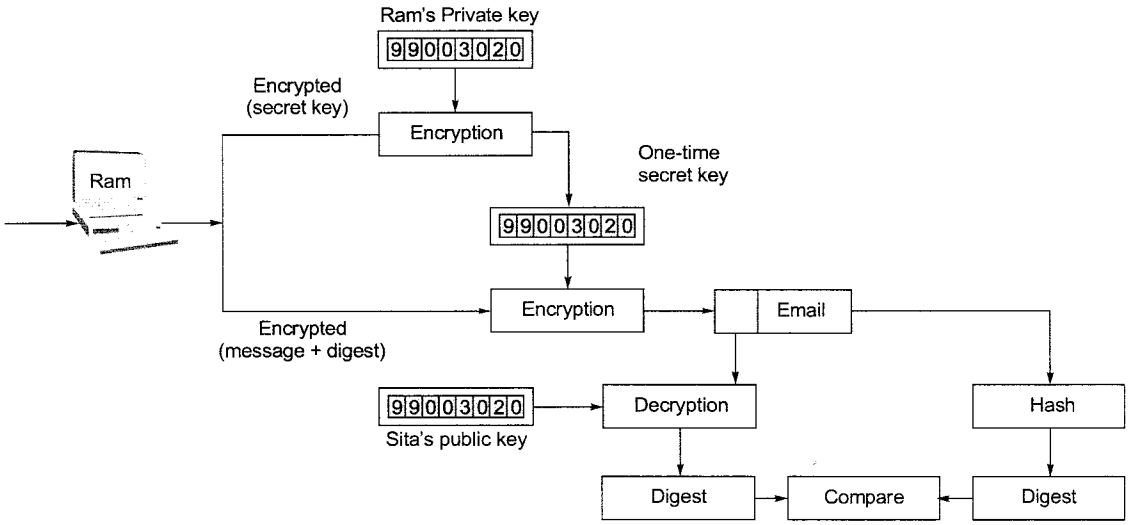


Figure: ***(b)*** *Sender site of the PGP*

7.6 Firewalls, Tunnels, and Network Intrusion Detection

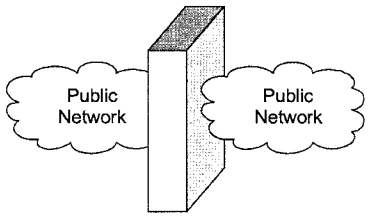
7.6.1 Firewalls

Why Use Firewalls?

1. Most hosts have security holes. Proof: Most software is buggy. Therefore, most security software has security bugs.
2. Firewalls run much less code, and hence have few bugs (and holes).
3. Firewalls can be professionally (and hence better) administered.
4. Firewalls run less software, with more logging and monitoring.
5. They enforce the partition of a network into separate security domains.
6. Without such a partition, a network acts as a giant virtual machine, with an unknown set of privileged and ordinary users.

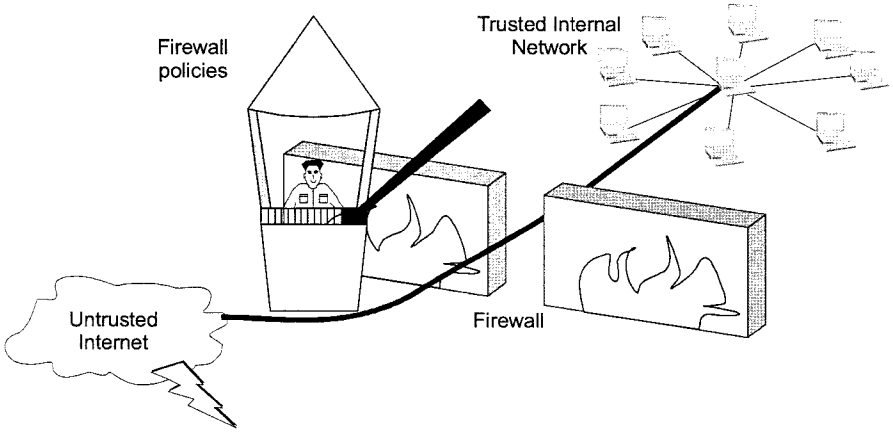
A firewall is an integrated collection of security measures **designed to prevent unauthorized electronic access** to a networked computer system.

A network firewall is similar to firewalls in building construction, because in both cases they are intended to isolate one "network" or "compartment" from another.



Firewall Policies

To protect private networks and individual machines from the dangers of the greater Internet, a firewall can be employed **to filter incoming or outgoing traffic** based on a predefined set of rules called **firewall policies.**



Policy Actions

- Packets flowing through a firewall can have one of three outcomes:
  (a) **Accepted:** permitted through the firewall
  (b) **Dropped:** not allowed through with no indication of failure
  (c) **Rejected:** not allowed through, accompanied by an attempt to inform the source that the packet was rejected
- Policies used by the firewall to handle packets are based on several properties of the packets being inspected, including the protocol used, such as:
  (a) TCP or UDP
  (b) The source and destination IP addresses
  (c) The source and destination ports
  (d) The application-level payload of the packet (e.g., whether it contains a virus).

Blacklists and White Lists

- There are two fundamental approaches to creating firewall policies (or rule sets) to effectively minimize vulnerability to the outside world while maintaining the desired functionality for the machines in the trusted internal network (or individual computer).
- **Blacklist** approach
  (a) All packets are allowed through except those that fit the rules defined specifically in a blacklist.

(b) This type of configuration is more flexible in ensuring that service to the internal network is not disrupted by the firewall, but is naïve from a security perspective in that it assumes the network administrator can enumerate all of the properties of malicious traffic.

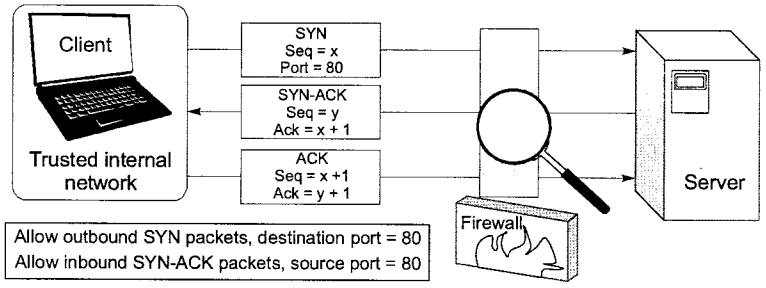
- **White list** approach: A safer approach to defining a firewall rule set is the default-deny policy, in which packets are dropped or rejected unless they are specifically allowed by the firewall.

Firewall Types

1. **Packet filters (stateless):** If a packet matches the packet filter's set of rules, the packet filter will drop or accept it.
2. **State full filters:** It maintains records of all connections passing through it and can determine if a packet is either the start of a new connection, a part of an existing connection, or is an invalid packet.
3. **Application layer:** It works like a **proxy** it can "understand" certain applications and protocols.
4. It may inspect the contents of the traffic, blocking what it views as inappropriate content (i.e. websites, viruses, vulnerabilities).

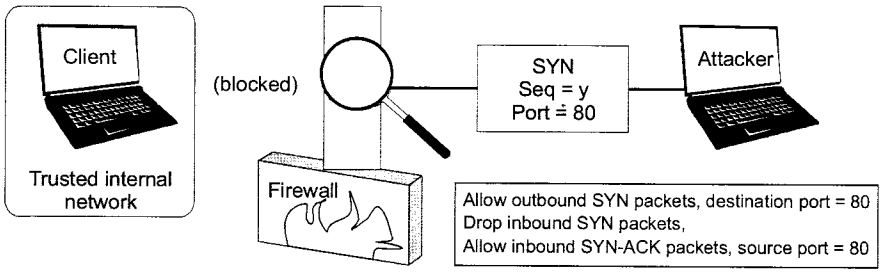
Stateless Firewalls

A stateless firewall doesn't maintain any remembered context (or "state") with respect to the packets it is processing. Instead, it treats each packet attempting to travel through it in isolation without considering packets that it has processed previously.



Stateless Restrictions

Stateless firewalls may have to be fairly restrictive in order to prevent most attacks.

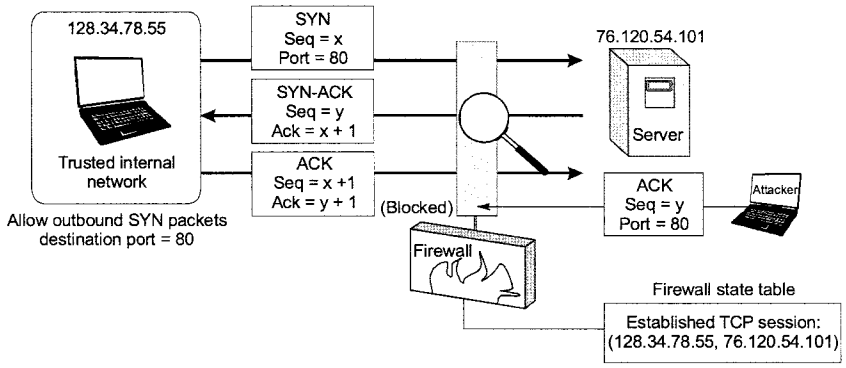


State Full Firewalls

- **Stateful firewalls** can tell when packets are part of legitimate sessions originating within a trusted network.
- Stateful firewalls maintain tables containing information on each active connection, including the IP addresses, ports, and sequence numbers of packets.
- Using these tables, Stateful firewalls can allow only inbound TCP packets that are in response to a connection initiated from within the internal network.

State Full Firewall Example

Allow only requested TCP connections:

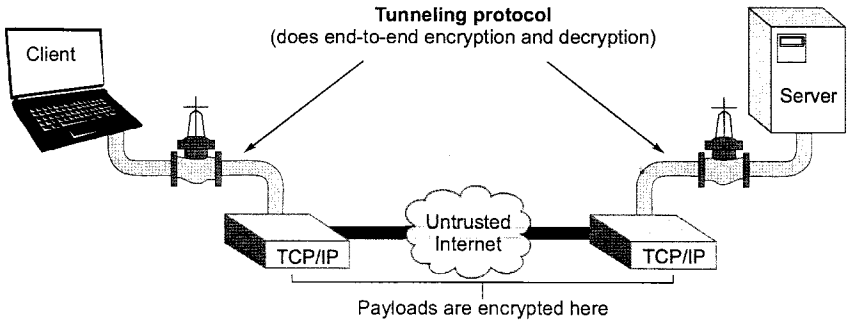


Tunnels

- The contents of TCP packets are not normally encrypted, so if someone is eavesdropping on a TCP connection, he can often see the complete contents of the payloads in this session.
- One way to prevent such eavesdropping without changing the software performing the communication is to use a **tunneling protocol.**
- In such a protocol, the communication between a client and server is automatically encrypted, so that useful eavesdropping is infeasible.

Tunneling Prevents Eaves Dropping

- Packets sent over the Internet are automatically encrypted.



Secure Shell (SSH)

A secure interactive command session:

1. The client connects to the server via a TCP session.
2. The client and server exchange information on administrative details, such as supported encryption methods and their protocol version, each choosing a set of protocols that the other supports.
3. The client and server initiate a secret-key exchange to establish a shared secret session key, which is used to encrypt their communication (but not for authentication). This session key is used in conjunction with a chosen block cipher (typically AES, 3DES) to encrypt all further communications.
4. The server sends the client a list of acceptable forms of authentication, which the client will try in sequence. The most common mechanism is to use a password or the following public-key authentication method:
   (a) If public-key authentication is the selected mechanism, the client sends the server its public key.

(b) The server then checks if this key is stored in its list of authorized keys. If so, the server encrypts a challenge using the client's public key and sends it to the client.

(c) The client decrypts the challenge with its private key and responds to the server, proving its identity.

5. Once authentication has been successfully completed, the server lets the client access appropriate resources, such as a command prompt.

IPSec

- IPSec defines a set of protocols to provide confidentiality and authenticity for IP packets
- Each protocol can operate in one of two modes, **transport mode** or **tunnel mode.**
  (a) In **transport mode,** additional IPsec header information is inserted before the data of the original packet, and only the payload of the packet is encrypted or authenticated.
  (b) In **tunnel mode**, a new packet is constructed with IPsec header information, and the entire original packet, including its header, is encapsulated as the payload of the new packet.

Example-7.1 Symmetric encryption algorithm is same as

(a) RSA algorithm
(b) Secure Hash Algorithm
(c) Secret key encryption algorithm
(d) Public key encryption algorithm

Solution: (c)

The secret key encryption alogrithm are often referred to as symmetric encryption alogrithms as the same key can be used in bidirectional communication between sender and receiver.

Example-7.2 Which of the following is true regarding message digest?

(a) It converts small data into large fixed-length string
(b) Given P, No one can find P' such that MD(P') = MD(P) where P and P' are small numbers.
(c) It is used to provide Confidentiality.
(d) None of the above

Solution: (d)

Converts large data into fixed size small data. P and P' should be large numbers it is used for authentication of data.

Example-7.3 In the RSA public key cryptosystem, the private and public keys are (e, n) and (d, n) respectively, where $n = p \times q$ and p and q are large primes. Besides, n is public and p and q are private. Let M be an integer such that $0 < M < n$ and $\phi(n) = (p-1)(q-1)$. Now consider the following equations.

I. $M' = M^e \bmod n$, $M = (M')^d \bmod n$
II. $ed \equiv 1 \bmod n$
III. $ed \equiv 1 \bmod \phi(n)$
IV. $M' = M^e \bmod \phi(n)$, $M = (M')^d \bmod \phi(n)$

Which of the above equations correctly represent RSA cryptosystem?

Solution:

I and III equations correctly represent RSA cryptosystem.

Example-7.4 Using public key cryptography, X adds a digital signature σ to message M, encrypts <M, σ >, and sends it to Y, where it is decrypted. Which one of the following sequence of keys is used for the operations?

(a) Encryption : X's private key followed by Y's private key; Decryption : X's public key followed by Y's public key

(b) Encryption : X's private key followed by Y's private key; Decryption : X's public key followed by Y's private key

(c) Encryption : X's public key followed by Y's private key; Decryption : Y's public key followed by X's private key

(d) Encryption : X's private key followed by Y's public key; Decryption : Y's private key followed by X's public key

Solution: (d)

The message over the network should be encrypted by y's public key.

So order of encryption is x's private key and y's public key.

On receiving the encrypted message, y will decrypt it using its private key and x's public key for signature. So order of decrypting is y's private key followed by x's public key.

Example-7.5 In RSA algorithm the value of p, q and e are 7, 17 and 5 respectively then find the value of d. Assume value of d is a whole number and not a fraction.

Solution:

Given $\boxed{p = 7, q = 17, e = 5}$

$$\text{Now } z = (7-1)(17-1) = 96$$

$$\therefore \quad \text{Now } (e \times d) \bmod z = 1$$

$$\text{i.e. } (5\,d) \bmod 96 = 1$$

5 d is ("multiple of 96" + 1)

$96 \times 1 + 1 = 97 \Rightarrow$ 'd' value as fraction (Reject)

$96 \times 2 + 1 = 193 \Rightarrow$ 'd' value as fraction (Reject)

$96 \times 3 + 1 = 289 \Rightarrow$ 'd' value as fraction (Reject)

$96 \times 4 + 1 = 385 \Rightarrow$ 'd' = 75 (Accepted)

$\therefore \quad d = 75$

Example-7.6 Find the value of e in RSA algorithm, given that $p = 13$, $q = 31$ and $d = 7$.

Solution:

$$z = 12 \times 30 = 360$$

$$\text{Now, } (e\,d) \bmod z = 1$$

$$(e\,7) \bmod z = 1$$

$\therefore$ "e 7" is multiple 360 + 1

$360 \times 1 + 1 = 362 \Rightarrow$ e is fraction (Reject)

$306 \times 2 + 2 = 721 \Rightarrow$ e = 103 (Accept)

Example-7.7 Ram and Sita uses the Diffie-Hellman protocol for generating session key. Ram chooses $y = 3$ and Sita chooses $x = 5$. Identify session key value if G = 7 and N = 23

Solution:

Given N = 23, G = 7

$R_1 = G^x \bmod N$
$= 7^3 \bmod 23 = 21$

$R_2 = G^y \bmod N$
$= 7^5 \bmod 23 = 17$

$\therefore \quad K = (R_1)^y \bmod N$
$= (21)^5 \bmod 23$
$= 4084101 \bmod 23$
$= 14$

$K = (R_2)^x \bmod N$
$= (17)^3 \bmod 23$
$= 4913 \bmod 23$
$= 14$

$\therefore \quad K = 14$

Note: We can directly compute $G^{xy} \bmod n = 14$

Example-7.8 Two gate aspirants talking to each other use the RSA algorithm to encrypt their messages. They encrypt the message character by character. The value of p, q and d are 5, 17 and 13 respectively, where p, q and d are their integers having usual meaning in the RSA algorithm. Identify the sum of integers in cipher text for corresponding characters in plain text: "IIT". Assume that corresponding cipher characters are placed in their corresponding plain text character places. Also each character is converted to ASCII value before applying RSA (ASCII value of A, B, C,... and so on are 1, 2, 3,..., respectively).

Solution:

Given p = 5, q = 17, d = 13, $n = 85$

$z = (5 - 1)(17 - 1) = 64$

Here $d = 13$ is relatively prime to z

Now, $(e \times d) \bmod 64 = 1 \Rightarrow e = 5$

P	$P^e \bmod n$	Cipher Character
I	$9^5 \bmod 85$	59
I	$9^5 \bmod 85$	59
T	$20^5 \bmod 85$	1

$\therefore$ Sum of integers in cipher text message: 59 + 59 + 1 = 119

7.7 Basics of WiFi

- **Wireless Hosts:** As in the case of wired networks, hosts are the end-system devices that run applications. A wireless host might be a laptop, palmtop, PDA, phone, or desktop computer. The hosts themeselves may or may not be mobile.
- **Wireless Links:** A host connects to a base station (defined below) or to another wireless host through a wireless communication link. Different wireless link technologies have different transmission rates and can transmit over different distance.
- **Base Station:** The base station is a key part of the wireless network infrastructure. A base station is responsible for sending and receiving packets to and from a wireless host that is associated with that base station. A base station is also responsible for managing the transmission of multiple wireless hosts with which it is associated.

If we say a wireless host is associated with a base station, we mean that:

1. Host is within the wireless communication distance of the base station.
2. Host uses that base station to relay data between it (the host) and the larger network.

Wireless LANs 802.11

Wireless communication is one of the fastest-growing technologies. The demand for connecting devices without the use of cables is increasing everywhere. Wireless LANs can be found on college campuses, in office buildings, and in many public areas. IEEE has defined the specifications for a wireless LAN, called IEEE 802.11, which covers the physical and data link layers.

7.7.1 Architecture

The standard defines two kinds of services:

1. Basic Service Set (BSS)
2. Extended Service Set (ESS).

Basic Service Set

IEEE 802.11 defines the basic service set (BSS) as the building block of a wireless LAN. A basic service set is made of stationary or mobile wireless stations and an optional central base station, known as the access point (AP). There are two types of network with BSS:

- **Adhoc network:** The BSS without an AP is a stand-alone network and cannot send data to other BSSs. It is called an *adhoc architecture*. In this architecture, stations can form a network without the need of an AP; they can locate one another and agree to be part of a BSS.
- **Infrastructure network:** A BSS with an AP is sometimes referred to as an *infrastructure* network.

NOTE: A BSS without an AP is called an adhoc network; a BSS with an AP is called an infrastructure network.

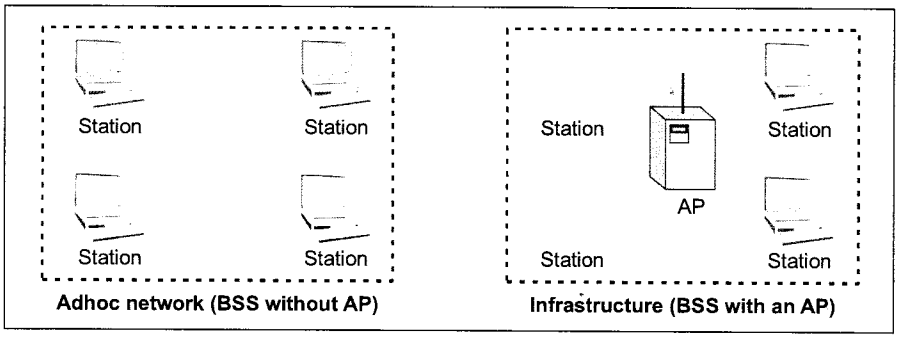


Figure : *Basic service sets (BSSs)*

Extended Service Set

An extended service set (ESS) is made up of two or more BSSs with APs. In this, the BSSs are connected through a *distribution system,* which is usually a wired LAN. The distribution system connects the APs in the BSSs. IEEE 802.11 does not restrict the distribution system; it can be any IEEE LAN such as an Ethernet. Extended service set uses two types of stations:

- **Mobile:** The mobile stations are normal stations inside a BSS.
- **Stationary:** The stationary stations are AP stations that are part of a wired LAN.

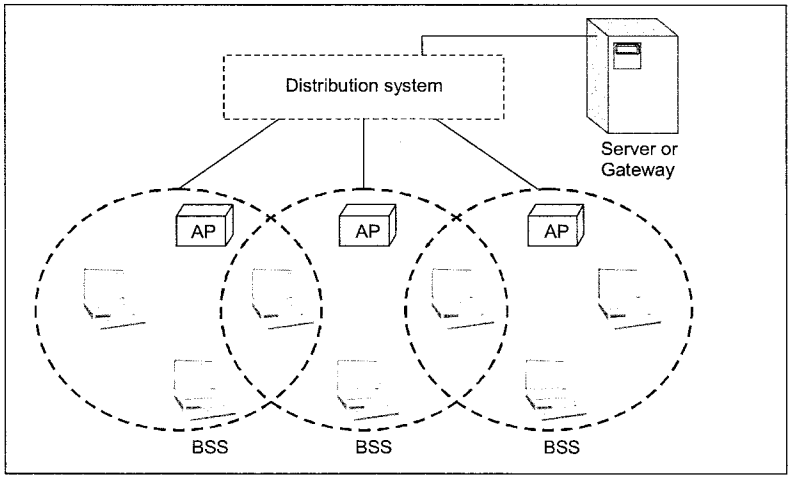


Figure : *Extended service sets (ESSs)*

When BSSs are connected, the stations within reach of one another can communicate without the use of an AP. However, communication between two stations in two different BSSs usually occurs via two APs. A mobile station can belong to more than one BSS at the same time.

Frame Format

The MAC layer frame consists of nine fields, as shown in Figure.

- **Frame Control (FC).** The FC field is 2 bytes long and defines the type of frame and some control information. Table describes the subfields.

2 bytes	2 bytes	6 bytes	6 bytes	6 bytes	2 bytes	6 bytes	0 to 2312 bytes	2 bytes
FC	D	Address-1	Address-2	Address-3	SC	Address-4	Frame body	FCS

Frame Format

Frame Control

Protocol version	Type	Subtype	To DS	From DS	More flag	Retry	Power management	More data	WEP	Rsvd
2 bits	2 bits	4 bits	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit

Subfield of Frame Control

Field	Explanation
Version	Current version is 0
Type	Type of information: management (00), control (01), or data (10)
Subtype	Subtype of each type
To DS	Defined later
From DS	Defined later
More flag	When set to 1, means more fragments
Retry	When set to 1, means retransmitted frame
Power management	When set to 1, means station is in power management mode
More data	When set to 1, means station has more data to send
WEP	Wired equivalent privacy (encryption implemented)
Rsvd	Reserved

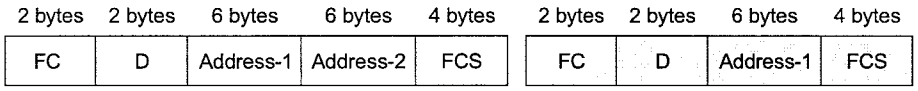
Table: *Subfields in FC Field*

- **Destination:** In all frame types except one, this field defines the duration of the transmission that is used to set the value of NAY. In one control frame, this field defines the ID of the frame.
- **Addresses:** There are four address fields, each 6 bytes long. The meaning of each address field depends on the value of the *To* DS and *From* DS subfields.
- **Sequence control:** This field defines the sequence number of the frame to be used in flow control.
- **Frame body:** This field, which can be between 0 and 2312 bytes, contains information based on the type and the subtype defined in the FC field.
- **FCS:** The FCS field is 4 bytes long and contains a CRC-32 error detection sequence.

7.7.2 Types of Frames

A wireless LAN defined by IEEE 802.11 has three categories of frames:

1. **Management frames:** Management frames are used for the initial communication between stations and access points.
2. **Control frames:** Control frames are used for accessing the channel and acknowledging frames.



Control Frames

Table: *Values of Subfields in Control Frames*

Subtype	Meaning
1011	Request to send (RTS)
1100	Clear to send (CTS)
1101	Acknowledgment (ACK)

- **Data Frames:** Data frames are used for carrying data and control information.

7.7.3 Addressing Mechanism

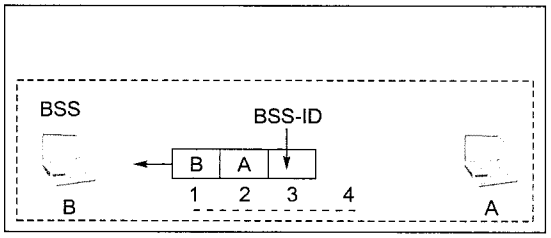
The IEEE 802.11 addressing mechanism specifies four cases, defined by the value of the two flags in the FC field, *To DS* and *From DS*. Each flag can be either 0 or 1, resulting in four different situations. The interpretation of the four addresses (address 1 to address 4) in the MAC frame depends on the value of these flags, as shown below.

Table: *Addresses*

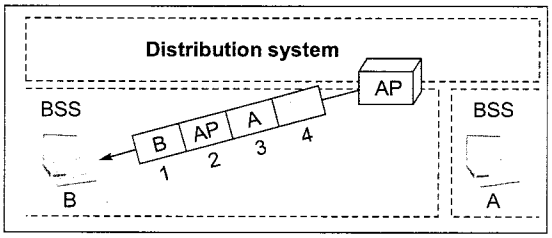
To DS	From DS	Address 1	Address 2	Address 3	Address 4
0	0	Destination	Source	BSS ID	N/A
0	1	Destination	SendingAP	Source	N/A
1	0	Receiving AP	Source	Destination	N/A
1	1	Receiving AP	SendingAP	Destination	Source

Note that address 1 is always the address of the next device. Address 2 is always the address of the previous device. Address 3 is the address of the final destination station if it is not defined by address 1. Address 4 is the address of the original source station if it is not the same as address 2.

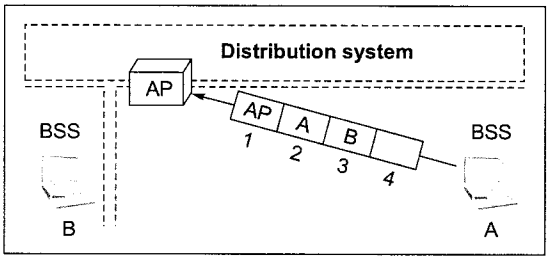
- **Case 1:** 00 In this case, *To DS* = 0 and *From DS* = O. This means that the frame is not going to a distribution system *(To DS* = 0) and is not coming from a distribution system (From DS =0). The frame is going from one station in a BSS to another without passing through the distribution system. The ACK frame should be sent to the original sender.



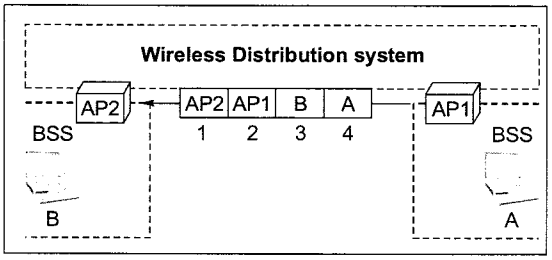
(a): Case-1



(a): Case-2



(a): Case-3



(a): Case-4

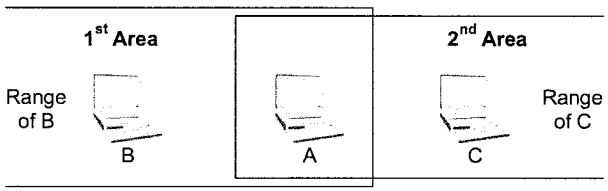
Addressing mechanisms

- **Case 2:** 01 In this case, To DS = 0 and From DS = 1. This means that the frame is coming from a distribution system (From DS = 1). The frame is coming from an AP and going to a station. The ACK should be sent to the AP.
- **Case 3:** 10 In this case, To DS = 1 and From DS = O. This means that the frame is going to a distribution system (To DS = 1). The frame is going from a station to an AP. The ACK is sent to the original station.
- **Case 4:** 11 In this case, To DS = 1 and From DS = 1. This is the case in which the distribution system is also wireless. The frame is going from one AP to another AP in a wire-less distribution system. We do not need to define addresses if the distribution system is a wired LAN because the frame in these cases has the format of a wired LAN frame (Ethernet, for example). Here, we need four addresses to define the original sender, the final destination, and two intermediate APs.

7.7.4 Problem in Wireless LAN

Hidden Station Problem

Station B has a transmission range shown by the 1st area range; every station in this range can hear any signal transmitted by station B. Station C has a transmission range shown by the 2nd area range; every station located in this range can hear any signal transmitted by C. Station C is outside the transmission range of B; same as, station B is outside the transmission range of C. Station A, however, is in the area covered by both Band C; it can hear any signal transmitted by B or C.

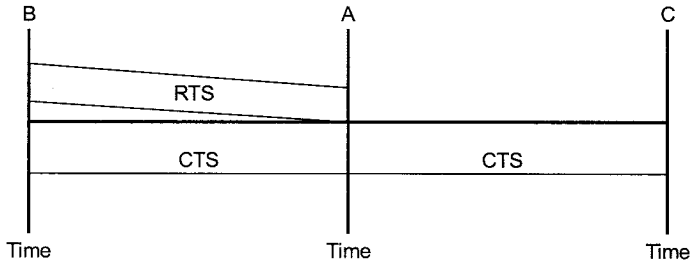


Hidden Station Problem

Assume that station B is sending data to station A. In the middle of this transmission, station C also has data to send to station A. However, station C is out of B's range and transmissions from B cannot reach C. Therefore C thinks the medium is free. Station C sends its data to A, which results in a collision at A because this station is receiving data from both B and C. In this case, we say that stations Band C are hidden from each other with respect to A. Hidden stations can reduce the capacity of the network because of the possibility of collision.

Solution: The solution to the hidden station problem is the use of the handshake frames (RTS and CTS) that we discussed earlier. Below figure shows that the RTS message from B reaches A, but not C. However, because both Band C are within the range of A, the CTS message, which contains the duration of data transmission from B to A reaches C. Station C knows that some hidden station is using the channel and refrains from transmitting until that duration is over.

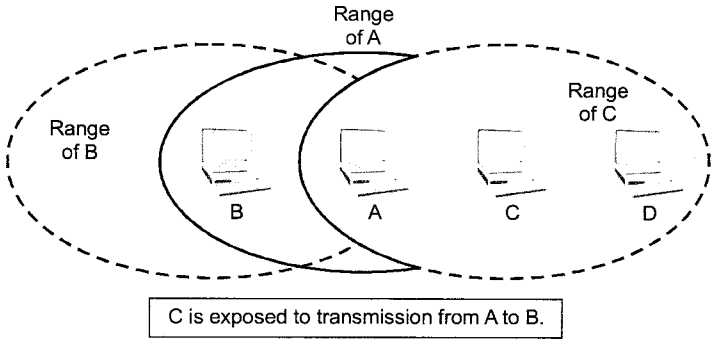
NOTE: The CTS frame in CSMAICA handshake can prevent collision from a hidden station.



Use of handshaking to prevent hidden station problem

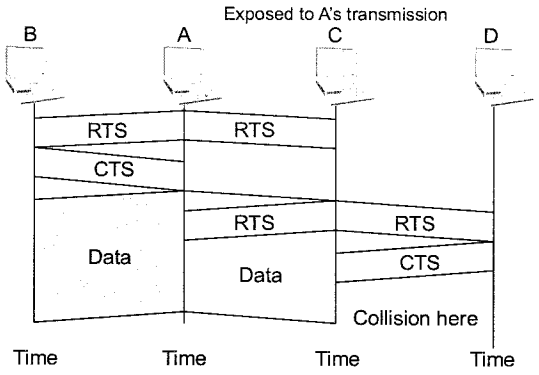
Exposed Station Problem

Now consider a situation that is the inverse of the previous one: the exposed station problem. In this problem a station refrains from using a channel when it is, in fact, available. In below figure, station A is transmitting to station B. Station C has some data to send to station D, which can be sent without interfering with the transmission from A to B. However, station C is exposed to transmission from A; it hears what A is sending and thus refrains from sending. In other words, C is too conservative and wastes the capacity of the channel.



Exposed station problem

The handshaking messages RTS and CTS cannot help in this case, despite what you might think. Station C hears the RTS from A, but does not hear the CTS from B. Station C, after hearing the RTS from A, can wait for a time so that the CTS from B reaches A; it then sends an RTS to D to show that it needs to communicate with D. Both stations B and A may hear this RTS, but station A is in the sending state, not the receiving state. Station B, however, responds with a CTS. The problem is here. If station A has started sending its data, station C cannot hear the CTS from station D because of the collision; it cannot send its data to D. It remains exposed until A finishes sending its data as shown below figure.



Use of handshaking in exposed station problem

Summary

- Symmetric key cryptography is also called as Private Key cryptography.
- Block ciphers use a block of bits as the unit of encryption and decryption.
- To encrypt a 64-bit block, one has to take each of the 264 input values and map it to one of the 264 output values. The **mapping should be one-to-one**.
- Asymmetric key cryptography is also called as Public Key cryptography. In public key cryptography, there are two keys: a private key and a public key. The public key is announced to the public; where as the private key is kept by the receiver. The **sender uses the public key of the receiver for encryption** and the receiver uses his private key for decryption.
- The types of attack that compromises secure communication are:
  (a) Interruption (b) Interception
  (c) Modification (d) Fabrication
- Secured communication requires the following four basic services:
  (a) Privacy (b) Authentication
  (c) Integrity (d) Nonrepudiation
- Diffie-Hellman protocol is used to establish a shared secret key.
- Implementation of security features in the application layer is far simpler and feasible compared to implementing at the other two lower layers.
- A firewall is an integrated collection of security measures **designed to prevent unauthorized electronic access** to a networked computer system.
- To prevent eavesdropping without changing the software performing the communication is to use a **tunneling protocol.**
- **Remote access** VPNs allow authorized clients to access a private network that is referred to as an **intranet.**

Student's Assignment

Q.1 Consider a secured environment making use of symmetric key Cryptography. Every host connects every other host. Calculate the number unique keys required (Symmetric Keys) if there 5 hosts in the network

(a) 5 (b) 10
(c) 15 (d) 25

Q.2 Consider a secured environment making use of symmetric key Cryptography. Every host connects every other host. Calculate the number of unique keys required (Symmetric Keys) if there are 5 hosts in the network.

Q.3 Which of the following is a DES mode of operation?

(a) Cipher Block Chaining (CBC)
(b) Cipher Feedback Mode (CFB)
(c) Output Feedback Mode (OFB)
(d) All of the above

Q.4 Which of the following statements are true?

S_1 : Same key is used at sender and receiver for symmetric key cryptography.

S_2 : In public key cryptography sender transmits with private key.

(a) Only S_2
(b) Both S_1 and S_2
(c) Only S_2
(d) Neither of them

Q.5 For 'n' number of users, how many total keys are needed by using private key cryptography and public key cryptography schemes respectively?

(a) $2n, 2n$ (b) $\frac{n(n-1)}{2}, \frac{n(n+1)}{2}$
(c) $2n+2, 2n$ (d) $\frac{n(n-1)}{2}, 2n$

Q.6 Suppose consider a closed network consisting of 14 users. Let P be the total number of unique keys required when we use symmetric key cryptography.

Let Q be the total number of unique keys required when we use asymmetric key cryptography. Compute the value of P+2Q.

Q.7 Which of the following security services is/are Not provided by digital signature?

1. Authentication of message
2. Integrity
3. Privacy
4. Non repudiation

(a) Only 1 and 2 (b) Only 4
(c) Only 3 (d) Only 3 and 4

Q.8 Ram and Sita uses the Diffie-Hellman protocol for generating session key. Ram chooses $y = 3$ and Sita chooses $x = 5$. Identify session key value if G = 7 and N = 23

Q.9 Which of the following is/are best candidate for value of 'n' in Diffie-Hellman key exchange algorithm

1. 107 3. 37
2. 47 4. 109

(a) Only 4
(b) Only 1 and 2
(c) Only 1 and 3
(d) Only 2 and 3

Q.10 In RSA algorithm the value of p, q and e are 7, 17 and 5 respectively then find the value of d. Assume value of d is a whole number and not a fraction.

(a) 45 (b) 75
(c) 35 (d) 25

Q.11 Find the value of e in RSA algorithm, given that $p = 13$, $q = 31$ and $d = 7$.

(a) 109 (b) 103
(c) 111 (d) 113

Q.12 In order to achieve confidentially the sender is employing public key cryptography. Which of the following statements is true?

(a) Sender uses his private key to encrypt
(b) Sender uses receiver's public key to encrypt
(c) Sender uses his public key to encrypt
(d) Sender uses receivers private key to encrypt

Q.13 Which of the following is/are true about digital signature?
1. Symmetric key cryptography is safe to use.
2. Asymmetric key cryptography can be used.
3. If public key cryptography is used then the sender will encrypt using his public key.

(a) Only 1 (b) Only 1 and 2
(c) Only 3 and 2 (d) Only 2

Q.14 Caesar Cipher is an example of
(a) Data Encryption Standard (DES) algorithm
(b) Monoalphabetic substitution
(c) Polyalphabetic substitution
(d) Block Cipher

Q.15 Cryptography, the order of the letters in a message is rearranged by
(a) transpositional ciphers
(b) substitution ciphers
(c) both (a) and (b)
(d) None of these

Q.16 What is data encryption standard (DES)?
(a) Block cipher (b) Stream cipher
(c) Bit cipher (d) All of these

Q.17 Which one of the following is a cryptographic protocol used to server HTTP connection?
(a) Stream control transmission protocol (SCTP)
(b) Transport layer security (TSL)
(c) Explicit congestion notification (ECN)
(d) Resource reservation protocol (RRP)

Q.18 Network layer firewall works as a
(a) Frame filter
(b) Packet filter
(c) Both (a) and (b)
(d) None of these

Q.19 Pretty good privacy (PGP) is used in
(a) browser security
(b) e-mail security
(c) FTP security
(d) All of these

Q.20 PGP encrypts data by using a block cipher called
(a) International data encryption algorithm
(b) Internet data encryption algorithm
(c) Private data encryption algorithm
(d) None of these

Answer Key:

1. (b)	**2.** (10)	**3.** (d)	**4.** (d)	**5.** (d)
6. (147)	**7.** (a)	**8.** (14)	**9.** (c)	**10.** (b)
11. (b)	**12.** (b)	**13.** (d)	**14.** (b)	**15.** (a)
16. (a)	**17.** (b)	**18.** (b)	**19.** (b)	**20.** (a)

◆■◆■◆■◆