

**JAIPUR ENGINEERING COLLEGE & RESEARCH CENTRE, JAIPUR**



**Lecture Notes**

**Data Communication and Computer Network**

**Prepared By: Archana Gupta**

**Dept.: Computer Science & Engineering**

## **Contents**

<b>S.No</b>	<b>Topic</b>
<b>1</b>	<b>Vision and Mission of Department</b>
<b>2</b>	<b>Program Outcomes</b>
<b>3</b>	<b>Course Outcomes</b>
<b>4</b>	<b>Program Educational Objectives</b>
<b>5</b>	<b>Syllabus</b>
<b>6</b>	<b>Lecture Notes</b>

## Vision & Mission

### Vision

To become renowned Centre of excellence in computer science and engineering and make competent engineers & professionals with high ethical values prepared for lifelong learning.

### Mission

M1-To impart outcome based education for emerging technologies in the field of computer science and engineering.

M2-To provide opportunities for interaction between academia and industry.

M3- To provide platform for lifelong learning by accepting the change in technologies

M4- To develop aptitude of fulfilling social responsibilities.

### Program Outcomes (PO)

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

## **Course Outcomes (CO)**

**Upon successful completion of this course students will be able to:**

**CO1:** Understand the principles of Network Protocols and OSI and TCP/IP model.

**CO2:** Analyze and implement the concepts of various protocols of Error Detection and Correction.

**CO3:** Analyze and apply the concept of various Routing algorithms and principles of reliable data transfers along with transactional TCP and associated congestion control.

**CO4:** Classify role of application layer, its various elements like WWW, DNS FTP and network security

## **Program Educational Objectives (PEO)**

**PEO1:** To provide students with the fundamentals of Engineering Sciences with more Emphasis in Computer Science & Engineering by way of analyzing and exploiting Engineering challenges.

**PEO2:** To train students with good scientific and engineering knowledge so as to

Comprehend, analyze, design, and create novel products and solutions for the real life Problems.

**PEO3:** To inculcate professional and ethical attitude, effective communication skills, Teamwork skills, multidisciplinary approach, entrepreneurial thinking and an ability to Relate engineering issues with social issues.

**PEO4:** To provide students with an academic environment aware of excellence, leadership, Written ethical codes and guidelines, and the self-motivated life-long learning needed For a successful professional career.

**PEO5:** To prepare students to excel in Industry and Higher education by Educating Students Along with High moral values and Knowledge.

## Syllabus



### RAJASTHAN TECHNICAL UNIVERSITY, KOTA

#### Syllabus

II Year-IV Semester: B.Tech. Computer Science and Engineering

4CS4-07: Data Communication and Computer Networks

Credit: 3

Max. Marks: 150(IA:30, ETE:120)

3L+0T+0P

End Term Exam: 3 Hours

SN	Contents	Hours
1	<b>Introduction:</b> Objective, scope and outcome of the course.	1
2	Introductory Concepts: Network hardware, Network software, topologies, Protocols and standards, OSI model, TCP model, TCP/IP model, Physical Layer: Digital and Analog Signals, Periodic Analog Signals, Signal Transmission, Limitations of Data Rate, Digital Data Transmission, Performance Measures, Line Coding, Digital Modulation, Media and Digital Transmission System	7
3	Data Link Layer: Error Detection and Correction, Types of Errors, Two dimensional parity check, Detection verses correction, Block Coding, Linear Block Coding, Cyclic Codes, Checksum, Standardized Polynomial Code, Error Correction Methods, Forward Error Correction, Protocols: Stop and wait, Go-back-N ARQ, Selective Repeat ARQ, Sliding window, Piggy backing, Pure ALOHA, Slotted ALOHA, CSMA/CD, CSMA/CA	9
4	Network Layer: Design issues, Routing algorithms: IPV4, IPV6, Address mapping: ARQ, RARQ, Congestion control, Unicast, Multicast, Broadcast routing protocols, Quality of Service, Internetworking	8
5	Transport Layer: Transport service, Elements of transport protocols, User Datagram Protocol, Transmission Control Protocol, Quality of service, Leaky Bucket and Token Bucket algorithm	8
6	Application Layer: WWW, DNS, Multimedia, Electronic mail, FTP, HTTP, SMTP, Introduction to network security	7
<b>Total</b>		<b>40</b>

# **Unit-I**

## **Introduction**

### **Overview:**

A system of interconnected computers and computerized peripherals such as printers is called computer network. This interconnection among computers facilitates information sharing among them. Computers may connect to each other by either wired or wireless media.

### **Classification of Computer Networks:**

Computer networks are classified based on various factors.

They include:

- Geographical span
- Inter-connectivity
- Administration
- Architecture

### **Geographical Span:**

Geographically a network can be seen in one of the following categories:

- It may be spanned across your table, among Bluetooth enabled devices, Ranging not more than few meters.
- It may be spanned across a whole building, including intermediate devices to connect all floors.
- It may be spanned across a whole city.
- It may be spanned across multiple cities or provinces.
- It may be one network covering whole world

### **Inter-Connectivity:**

Components of a network can be connected to each other differently in some fashion. By connectedness we mean either logically, physically, or both ways.

- Every single device can be connected to every other device on network, making the network mesh
- All devices can be connected to a single medium but geographically disconnected, created bus-like structure.
- Each device is connected to its left and right peers only, creating linear structure.
- All devices connected together with a single device, creating star-like structure.
- All devices connected arbitrarily using all previous ways to connect each other, resulting in a hybrid structure.

### **Administration:**

From an administrator's point of view, a network can be private network which belongs a single autonomous system and cannot be accessed outside its physical or logical domain. A network can be public, which is accessed by all.

### **Network Architecture:**

Computer networks can be discriminated into various types such as Client-Server, peer-to-peer or hybrid, depending upon its architecture.

- There can be one or more systems acting as Server. Other being Client, requests the Server to serve requests. Server takes and processes request on behalf of Clients.
- Two systems can be connected Point-to-Point, or in back-to-back fashion. They both reside at the same level and called peers.
- There can be hybrid network which involves network architecture of both the above types.

### **Network Applications:**

Computer systems and peripherals are connected to form a network. They provide numerous advantages:

- Resource sharing such as printers and storage devices
- Exchange of information by means of e-Mails and FTP
- Information sharing by using Web or Internet
- Interaction with other users using dynamic web pages
- IP phones

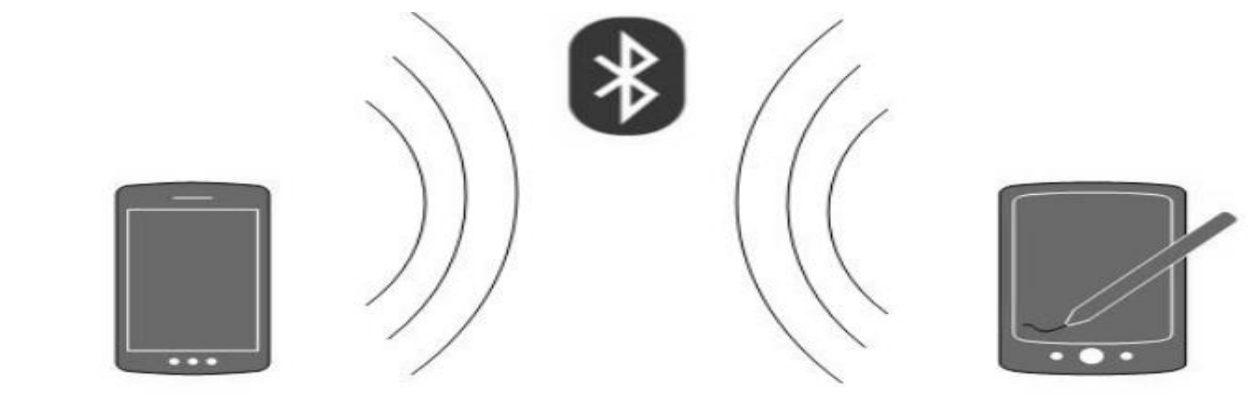
- Video conferences
- Parallel computing
- Instant messaging

## **TYPES OF COMPUTER NETWORKS:**

Generally, networks are distinguished based on their geographical span. A network can be as small as distance between your mobile phone and its Bluetooth headphone and as large as the internet itself, covering the whole geographical world.

### **Personal Area Network:**

A Personal Area Network (PAN) is smallest network which is very personal to a user. This may include Bluetooth enabled devices or infra-red enabled devices. PAN has connectivity range up to 10 meters. PAN may include wireless computer keyboard and mouse, Bluetooth enabled headphones, wireless printers, and TV remotes.



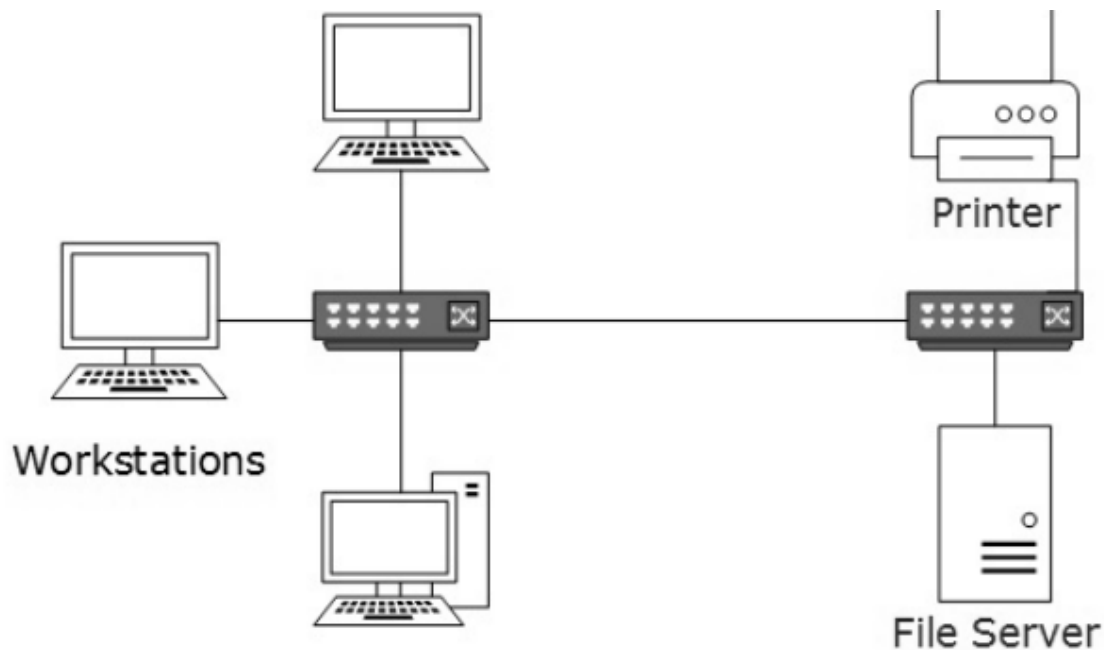
For example, Piconet is Bluetooth-enabled Personal Area Network which may contain up to 8 devices connected together in a master-slave fashion.

### **Local Area Network:**

A computer network spanned inside a building and operated under single administrative system is generally termed as Local Area Network (LAN). Usually, LAN covers an organization offices, schools, colleges or universities. Number of systems connected in LAN may vary from as least



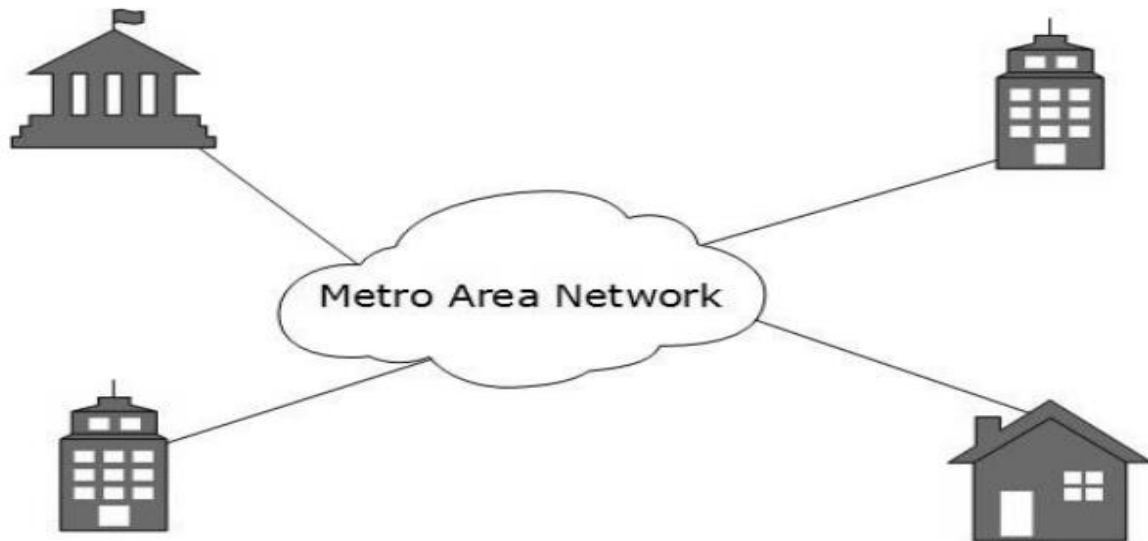
as two to as much as 16 million. LAN provides a useful way of sharing the resources between end users. The resources such as printers, file servers, scanners, and internet are easily sharable among computers.



LANs are composed of inexpensive networking and routing equipment. It may contain local servers serving file storage and other locally shared applications. It mostly operates on private IP addresses and does not involve heavy routing. LAN works under its own local domain and is controlled centrally. LAN uses either Ethernet or Token-ring technology. Ethernet is most widely employed LAN technology and uses Star topology, while Token-ring is rarely seen. LAN can be wired, wireless, or in both forms at once.

### **Metropolitan Area Network:**

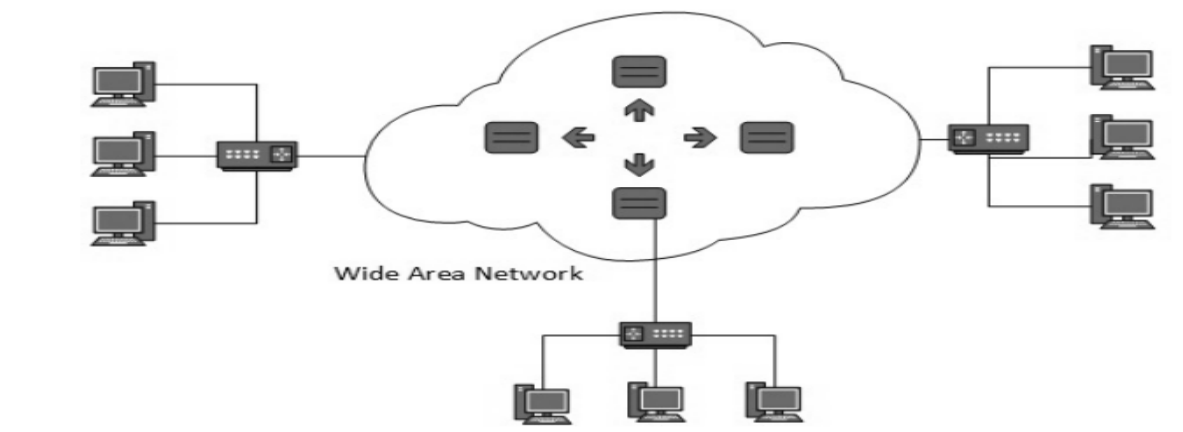
The Metropolitan Area Network (MAN) generally expands throughout a city such as cable TV network. It can be in the form of Ethernet, Token-ring, ATM, or Fiber Distributed Data Interface (FDDI). Metro Ethernet is a service which is provided by ISPs. This service enables its users to expand their Local Area Networks. For example, MAN can help an organization to connect all of its offices in a city.



Backbone of MAN is high-capacity and high-speed fiber optics. MAN works in between Local Area Network and Wide Area Network. MAN provides uplink for LANs to WANs or internet.

### **Wide Area Network:**

As the name suggests, the Wide Area Network (WAN) covers a wide area which may span across provinces and even a whole country. Generally, telecommunication networks are Wide Area Network. These networks provide connectivity to MANs and LANs. Since they are equipped with very high speed backbone, WANs use very expensive network equipment.



WAN may use advanced technologies such as Asynchronous Transfer Mode (ATM), Frame Relay, and Synchronous Optical Network (SONET). WAN may be managed by multiple administration.

## **Internetwork**

A network of networks is called an internetwork, or simply the internet. It is the largest network in existence on this planet. The internet hugely connects all WANs and it can have connection to LANs and Home networks. Internet uses TCP/IP protocol suite and uses IP as its addressing protocol. Present day, Internet is widely implemented using IPv4. Because of shortage of address spaces, it is gradually migrating from IPv4 to IPv6.

Internet enables its users to share and access enormous amount of information worldwide. It uses WWW, FTP, email services, audio, and video streaming etc. At huge level, internet works on Client-Server model.

Internet uses very high speed backbone of fiber optics. To inter-connect various continents, fibers are laid under sea known to us as submarine communication cable.

Internet is widely deployed on World Wide Web services using HTML linked pages and is accessible by client software known as Web Browsers. When a user requests a page using some web browser located on some Web Server anywhere in the world, the Web Server responds with the proper HTML page. The communication delay is very low.

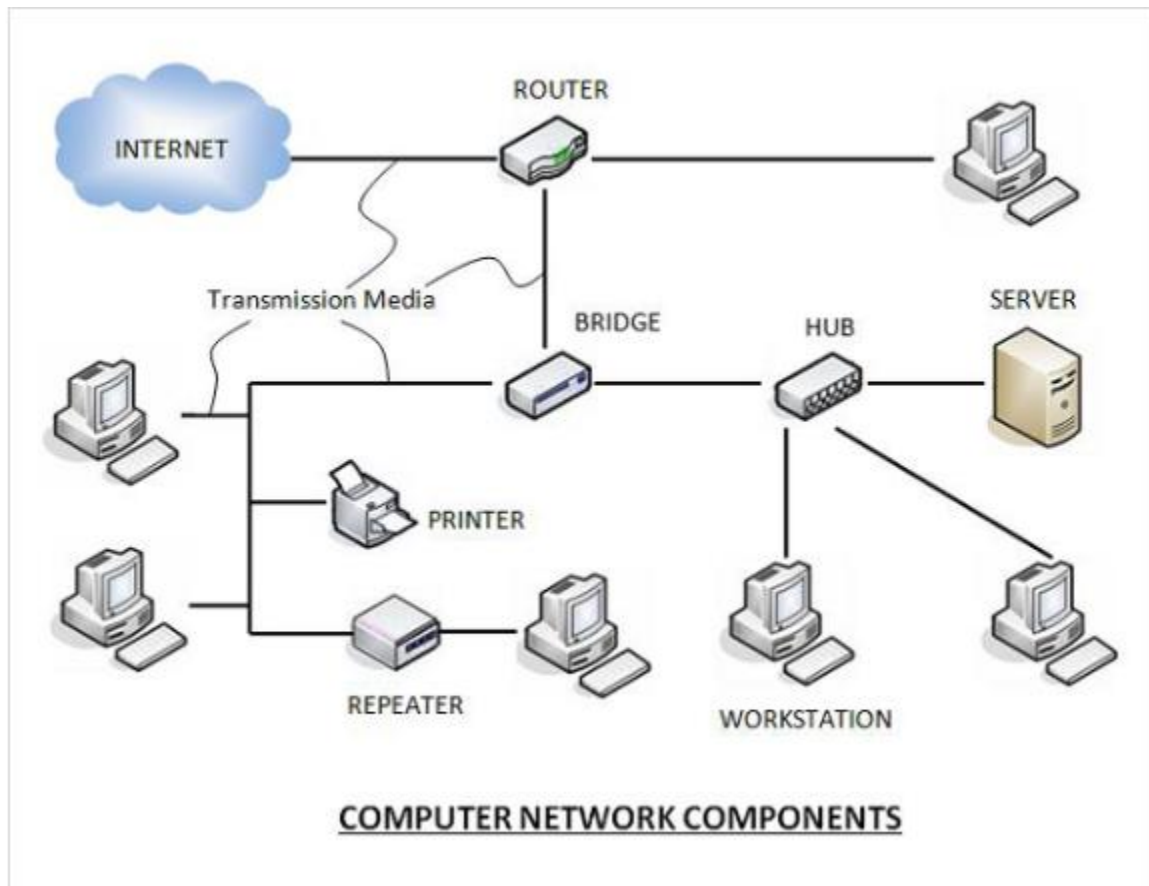
Internet is serving many proposes and is involved in many aspects of life. Some of them are:

- Web sites
- E-mail
- Instant Messaging
- Blogging
- Social Media
- Marketing
- Networking
- Resource Sharing
- Audio and Video Streaming

## Unit-II

### Network hardware and software

Networks are composed of hardware and software components. Each hardware and software component has a different function to perform. However, network operating system (NOS) software makes all the different components work together as a single network. NetWare, Solaris, Linux, AIX, and Windows NT are examples of NOS software. The figure below is a diagram of a simple network.



## **Hardware Components:**

- **Server** – Servers are high-configuration computers that manage the resources of the network. The network operating system is typically installed in the server and so they give user accesses to the network resources. Servers can be of various kinds: file servers, database servers, print servers etc.
- **Peers** – Peers are computers that provide as well as receive services from other peers in a workgroup network.
- **Clients** – Clients are computers that request and receive service from the servers to access and use the network resources.
- **Transmission Media** – Transmission media are the channels through which data is transferred from one device to another in a network. Transmission media may be guided media like coaxial cable, fiber optic cables etc.; or maybe unguided media like microwaves, infra-red waves etc.
- **Connecting Devices** – Connecting devices act as middleware between networks or computers, by binding the network media together. Some of the common connecting devices are:
  - a. Routers
  - b. Bridges
  - c. Hubs
  - d. Repeaters
  - e. Gateways
  - f. Switches

## **Software Components:**

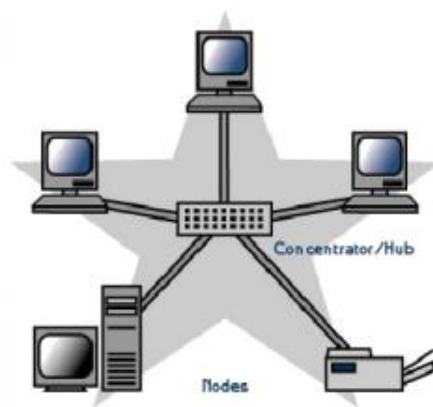
- **Networking Operating System** – Network Operating Systems is typically installed in the server and facilitate workstations in a network to share files, database, applications, printers etc.
- **Protocol Suite** – A protocol is a rule or guideline followed by each computer for data communication. Protocol suite is a set of related protocols that are laid down for computer networks. The two popular protocol suites are –
  - a. OSI Model (Open System Interconnections)
  - b. TCP / IP Model

## **Network Topologies**

In computer networking, topology refers to the layout of connected devices, i.e. how the computers, cables, and other components within a data communications network are interconnected, both physically and logically. The physical topology describes how the network is actually laid out, and the logical topology describes how the data actually flow through the network. Two most basic topologies are point-to-point and multipoint. A point to-point topology usually connects two mainframe computers for high-speed digital information. A multipoint topology connects three or more stations through a single transmission medium and some examples are star, bus, ring, mesh and hybrid.

### **Star topology:**

A star topology is designed with each node (file server, workstations, and peripherals) connected directly to a central network hub, switch, or concentrator. Data on a star network passes through the hub, switch, or concentrator before continuing to its destination. The hub, switch, or concentrator manages and controls all functions of the network. It also acts as a repeater for the data flow.

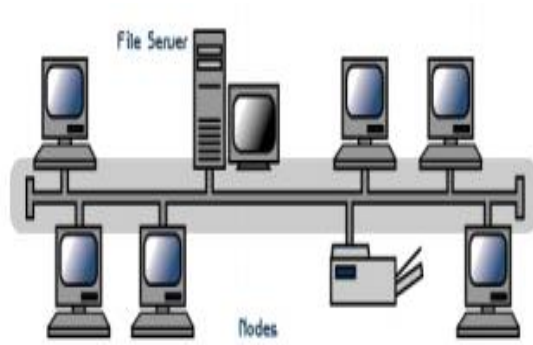


Advantages	Disadvantages
Easily expanded without disruption to the network	Requires more cable
Cable failure affects only a single user	A central connecting device allows for a single point of failure
Easy to troubleshoot and isolate problems	More difficult to implement

### **Bus topology:**

Bus networks use a common backbone to connect all devices. A single cable, (the backbone) functions as a shared communication medium that devices attach or tap into with an interface connector. A device wanting to communicate with another device on the network sends a broadcast message onto the wire that all other devices see, but only the

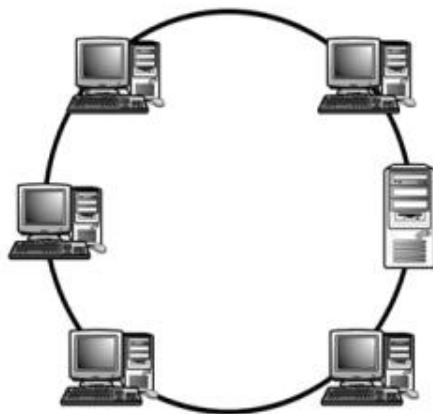
intended recipient actually accepts and processes the message. The bus topology is the simplest and most common method of interconnecting computers. The two ends of the transmission line never touch to form a complete loop. A bus topology is also known as multi drop or linear bus or a horizontal bus.



Advantages	Disadvantages
Cheap and easy to implement	Network disruption when computers are added or removed
Require less cable	A break in the cable will prevent all systems from accessing the network.
Does not use any specialized network equipment.	Difficult to troubleshoot.

### Ring topology:

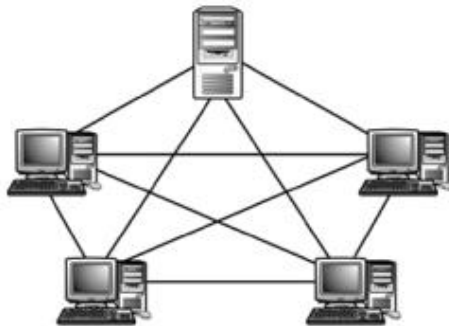
In a ring network (sometimes called a loop), every device has exactly two neighbors for communication purposes. All messages travel through a ring in the same direction (either "clockwise" or "counter clockwise"). All the stations are interconnected in tandem (series) to form a closed loop or circle. Transmissions are unidirectional and must propagate through all the stations in the loop. Each computer acts like a repeater and the ring topology is similar to bus or star topologies.



Advantages	Disadvantages
Cable faults are easily located, making troubleshooting easier	Expansion to the network can cause network disruption
Ring networks are moderately easy to install	A single break in the cable can disrupt the entire network.

## Mesh topology:

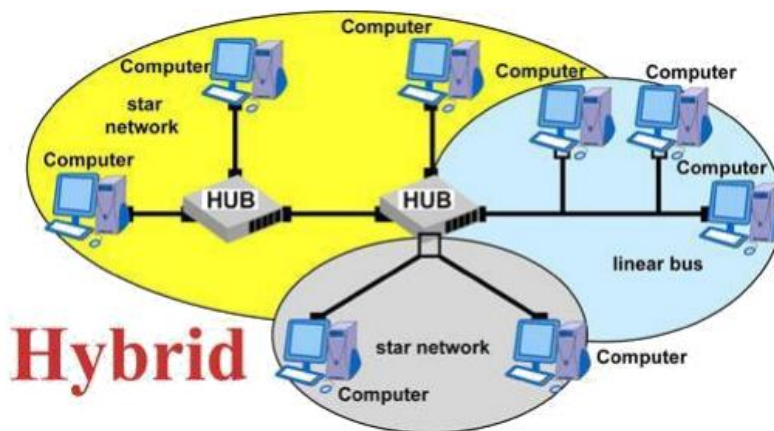
The mesh topology incorporates a unique network design in which each computer on the network connects to every other, creating a point-to-point connection between every device on the network. Unlike each of the previous topologies, messages sent on a mesh network can take any of several possible paths from source to destination. A mesh network in which every device connects to every other is called a full mesh. A disadvantage is that, a mesh network with  $n$  nodes must have  $n(n-1)/2$  links and each node must have  $n-1$  I/O ports (links).



Advantages	Disadvantages
Provides redundant paths between devices	Requires more cable than the other LAN topologies
The network can be expanded without disruption to current uses	Complicated implementation

## Hybrid topology:

This topology (sometimes called mixed topology) is simply combining two or more of the traditional topologies to form a larger, more complex topology. Main aim is being able to share the advantages of different topologies.





## **Protocols & Standards**

**Protocol:** A protocol is a set of rules that govern data communications. It represents an agreement between the communicating devices.

- Syntax
- Semantics
- Timing

An association of organizations, governments, manufacturers and users form the standards organizations and are responsible for developing, coordinating and maintaining the standards. The intent is that all data communications equipment manufacturers and users comply with these standards. The primary standards organizations for data communication are

- 1. International Standard Organization (ISO):** ISO is the international organization for standardization on a wide range of subjects. It is comprised mainly of members from the standards committee of various governments throughout the world. It is even responsible for developing models which provides high level of system compatibility, quality enhancement, improved productivity and reduced costs. The ISO is also responsible for endorsing and coordinating the work of the other standards organizations.
- 2. International Telecommunications Union-Telecommunication Sector (ITU-T):** ITU-T is one of the four permanent parts of the International Telecommunications Union based in Geneva, Switzerland. It has developed three sets of specifications: the V series for modem interfacing and data transmission over telephone lines, the X series for data transmission over public digital networks, email and directory services; the I and Q series for Integrated Services Digital Network (ISDN) and its extension Broadband ISDN. ITU-T membership consists of government authorities and representatives from many countries and it is the present standards organization for the United Nations.
- 3. Institute of Electrical and Electronics Engineers (IEEE):** IEEE is an international professional organization founded in United States and is comprised of electronics, computer and communications engineers. It is currently the world's largest professional society with over 200,000 members. It develops communication and information processing standards with the underlying goal of advancing theory, creativity, and product quality in any field related to electrical engineering.
- 4. American National Standards Institute (ANSI)** ANSI is the official standards agency for the United States and is the U.S voting representative for the ISO. ANSI

is a completely private, non-profit organization comprised of equipment manufacturers and users of data processing equipment and services. ANSI membership is comprised of people from professional societies, industry associations, governmental and regulatory bodies, and consumer goods.

5. **Electronics Industry Association (EIA)** EIA is a non-profit U.S. trade association that establishes and recommends industrial standards. EIA activities include standards development, increasing public awareness, and lobbying and it is responsible for developing the RS (recommended standard) series of standards for data and communications.

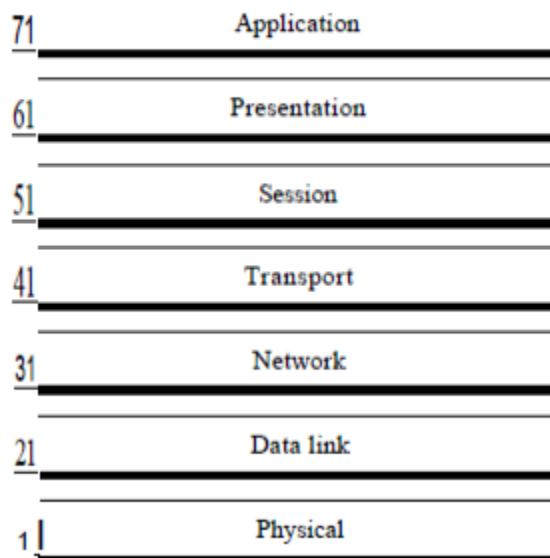
## **OSI model**

The OSI model is a layered framework for the design of network systems that allows communication between all types of computer systems. It consists of seven separate but related layers, each of which defines a part of the process of moving information across a network.

---

### *Seven layers of the OSI model*

---



## **1. Physical Layer**

The physical layer coordinates the functions required to carry a bit stream over a physical medium. It deals with the mechanical and electrical specifications of the interface and transmission medium. It also defines the procedures and functions that physical devices and interfaces have to perform for transmission to occur.

The physical layer is also concerned with the following:

- Physical characteristics of interfaces and medium.
- Representation of bits.
- Data rate.
- Synchronization of bits.
- Line configuration.
- Physical topology.
- Transmission mode.

## **2. Data Link Layer**

The data link layer transforms the physical layer, a raw transmission facility, to a reliable link. It makes the physical layer appear error-free to the upper layer (network layer). Other responsibilities of the data link layer include the following:

- Framing.
- Physical addressing
- Flow control
- Access control.

## **3. Network Layer**

The network layer is responsible for the source-to-destination delivery of a packet, possibly across multiple networks (links). Whereas the data link layer oversees the delivery of the packet between two systems on the same network (links), the network layer ensures that each packet gets from its point of origin to its final destination. If two systems are connected to the same link, there is usually no need for a network layer. However, if the two systems are attached to different networks (links) with connecting devices between the networks (links), there is often a need for the network layer to accomplish source-to-destination delivery. Other responsibilities of the network layer include the following:

- Logical addressing.
- Routing

## **4. Transport Layer**

The transport layer is responsible for process-to-process delivery of the entire message. A process is an application program running on a host. Whereas the network layer oversees

source to-destination delivery of individual packets, it does not recognize any relationship between those packets. It treats each one independently, as though each piece belonged to a separate message, whether or not it does. The transport layer, on the other hand, ensures that the whole message arrives intact and in order, overseeing both error control and flow control at the source to-destination level. Other responsibilities of the transport layer include the following:

- Service-point addressing
- o Segmentation and reassembly
- Connection control
- Flow control
- Error control

## **5. Session Layer**

The services provided by the first three layers (physical, data link, and network) are not sufficient for some processes. The session layer is the network dialog controller. It establishes, maintains, and synchronizes the interaction among communicating systems. Specific responsibilities of the session layer include the following:

- Dialog control
- Synchronization.

## **6. Presentation Layer**

The presentation layer is concerned with the syntax and semantics of the information exchanged between two systems. Specific responsibilities of the presentation layer include the following:

- Translation
- Encryption
- Compression

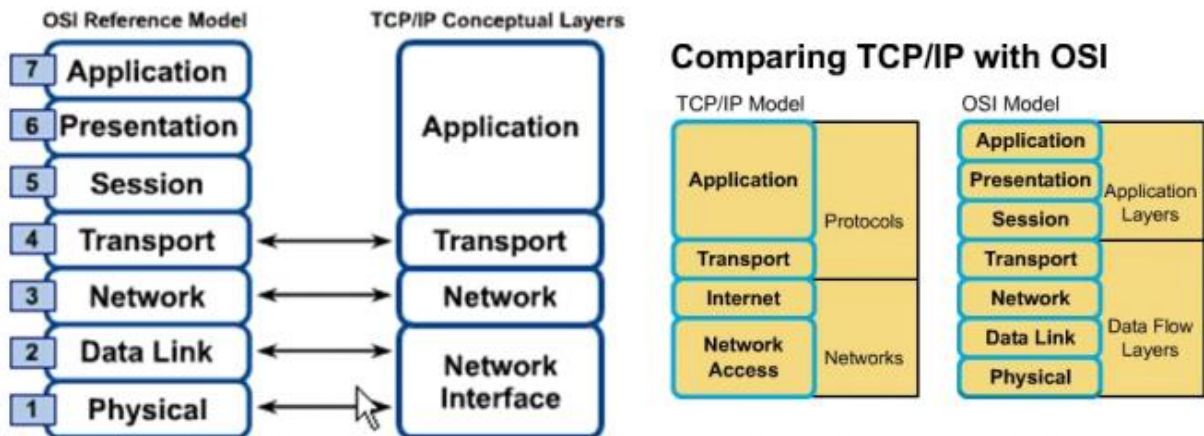
## **7. Application Layer**

The application layer enables the user, whether human or software, to access the network. It provides user interfaces and support for services such as electronic mail, remote file access and transfer, shared database management, and other types of distributed information services. Specific services provided by the application layer include the following:

- Network virtual terminal
- File transfer, access, and management
- Mail services
- Directory services

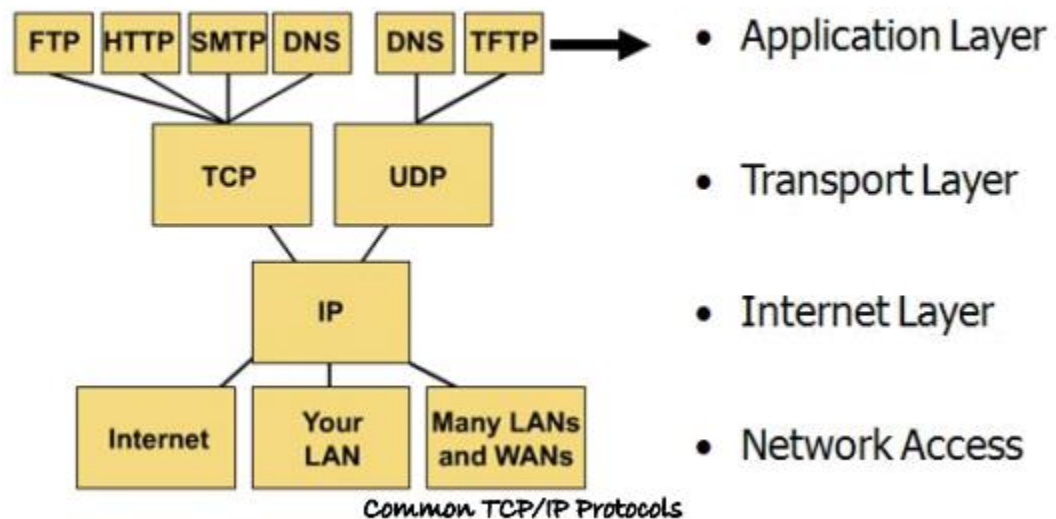
## TCP/IP model

TCP/IP model is a set of communication protocols that allow communication across multiple diverse networks. TCP/IP is a hierarchical protocol comprised of either three or four layers. The three-layer version of TCP/IP contains the network, transport and application layers. Four layer version specifies the host to network layer.



The TCP/IP transport layer deals with the quality-of-service issues of reliability, flow control, and error correction. One of its protocols, the transmission control protocol (TCP), provides excellent and flexible ways to create reliable, well-flowing, low-error network communications. TCP is a connection-oriented protocol. The other protocol is User Datagram Protocol (UDP) which is a connection less protocol.

### Protocol Graph: TCP/IP



## **Differences between OSI and TCP/IP**

- TCP/IP combines the presentation and session layer issues into its application layer
- TCP/IP combines the OSI data link and physical layers into one layer
- TCP/IP appears simpler because it has fewer layers
- TCP/IP protocols are the standards around which the Internet developed, so the TCP/IP model gains credibility just because of its protocols. In contrast, typically networks aren't built on the OSI protocol, even though the OSI model is used as a guide.

## **Physical Layer: Digital and Analog Signals**

Physical layer in the OSI model plays the role of interacting with actual hardware and signaling mechanism. Physical layer is the only layer of OSI network model which actually deals with the physical connectivity of two different stations. This layer defines the hardware equipment, cabling, wiring, frequencies, pulses used to represent binary signals etc.

Physical layer provides its services to Data-link layer. Data-link layer hands over frames to physical layer. Physical layer converts them to electrical pulses, which represent binary data. The binary data is then sent over the wired or wireless media.

## **Signals**

When data is sent over physical medium, it needs to be first converted into electromagnetic signals. Data itself can be analog such as human voice, or digital such as file on the disk. Both analog and digital data can be represented in digital or analog signals.

## **Digital Signals**

Digital signals are discrete in nature and represent sequence of voltage pulses. Digital signals are used within the circuitry of a computer system.

## **Analog Signals**

Analog signals are in continuous wave form in nature and represented by continuous electromagnetic waves.

**Transmission Impairment:** When signals travel through the medium, they tend to deteriorate. This may have many reasons as given:

- Attenuation

- Dispersion
- Delay distortion
- Noise
- Thermal Noise
- Intermodulation
- Crosstalk
- Impulse

## **Transmission Media**

The media over which the information between two computer systems is sent, called transmission media. Transmission media comes in two forms.

### **Guided Media**

All communication wires/cables are guided media, such as UTP, coaxial cables, and fiber Optics. In this media, the sender and receiver are directly connected and the information is send (guided) through it.

### **Unguided Media**

Wireless or open air space is said to be unguided media, because there is no connectivity between the sender and receiver. Information is spread over the air, and anyone including the actual recipient may collect the information.

## **DIGITAL TRANSMISSION**

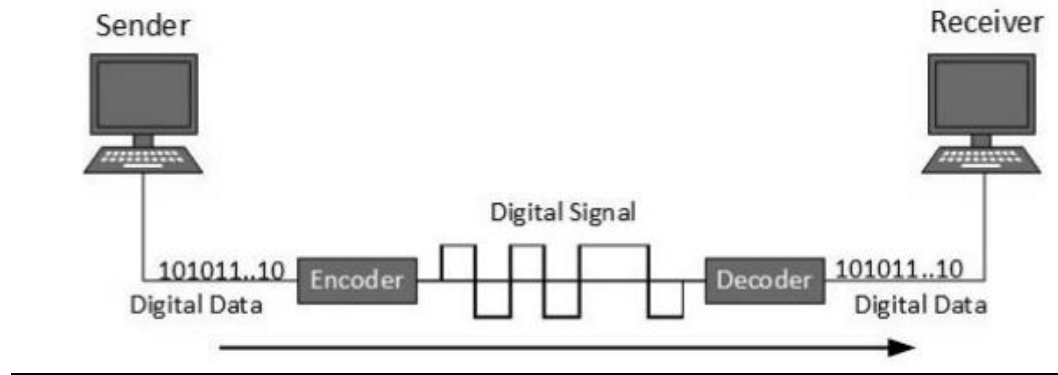
Data or information can be stored in two ways, analog and digital. For a computer to use the data, it must be in discrete digital form. Similar to data, signals can also be in analog and digital form. To transmit data digitally, it needs to be first converted to digital form.

### **Digital-to-Digital Conversion**

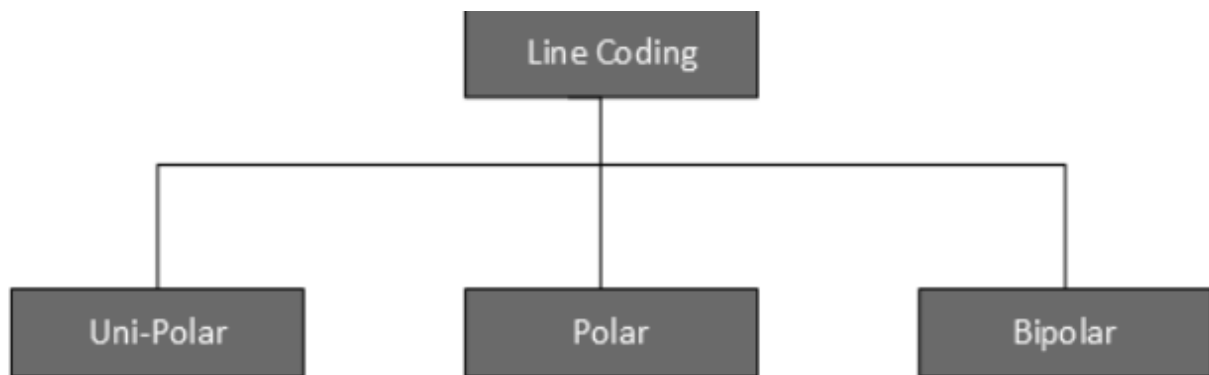
This section explains how to convert digital data into digital signals. It can be done in two ways, line coding and block coding. For all communications, line coding is necessary whereas block coding is optional.

### **Line Coding**

The process for converting digital data into digital signal is said to be Line Coding. Digital data is found in binary format. It is represented (stored) internally as series of 1s and 0s.



Digital signal is denoted by discrete signal, which represents digital data. There are three types of line coding schemes available:



### Unipolar Encoding

Unipolar encoding schemes use single voltage level to represent data. In this case, to represent binary 1, high voltage is transmitted and to represent 0, no voltage is transmitted. It is also called Unipolar-Non-return-to-zero, because there is no rest condition i.e. it either represents 1 or 0.

### Polar Encoding

Polar encoding scheme uses multiple voltage levels to represent binary values. Polar encodings are available in four types:

- Polar Non Return to Zero (Polar NRZ)
- Return to Zero (RZ)
- Manchester
- Differential Manchester



## Bipolar Encoding

Bipolar encoding uses three voltage levels, positive, negative, and zero. Zero voltage represents binary 0 and bit 1 is represented by altering positive and negative voltages.

## Analog-to-Digital Conversion

Microphones create analog voice and camera creates analog videos, which are treated as analog data. To transmit this analog data over digital signals, we need analog to digital conversion.

Analog data is a continuous stream of data in the wave form whereas digital data is discrete. To convert analog wave into digital data, we use Pulse Code Modulation (PCM). PCM is one of the most commonly used methods to convert analog data into digital form. It involves three steps:

- Sampling
- Quantization
- Encoding

## Transmission Modes

The transmission mode decides how data is transmitted between two computers. The binary data in the form of 1s and 0s can be sent in two different modes: **Parallel and Serial**.

## ANALOG TRANSMISSION

To send the digital data over an analog media, it needs to be converted into an analog signal. There can be two cases according to data formatting.

**Bandpass:** The filters are used to filter and pass frequencies of interest. A bandpass is a band of frequencies which can pass the filter.

**Low-pass:** Low-pass is a filter that passes low frequency signals.

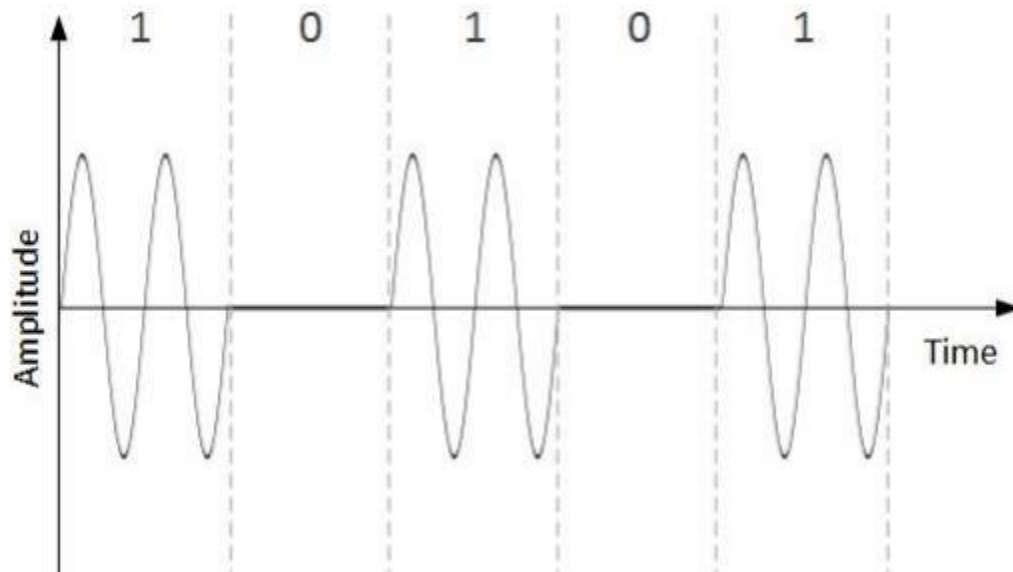
When digital data is converted into a bandpass analog signal, it is called digital-to-analog conversion. When low-pass analog signal is converted into bandpass analog signal, it is called analog-to-analog conversion.

## Digital-to-Analog Conversion

When data from one computer is sent to another via some analog carrier, it is first converted into analog signals. Analog signals are modified to reflect digital data. An analog signal is characterized by its amplitude, frequency, and phase. There are three kinds of digital-to-analog conversions:

## Amplitude Shift Keying

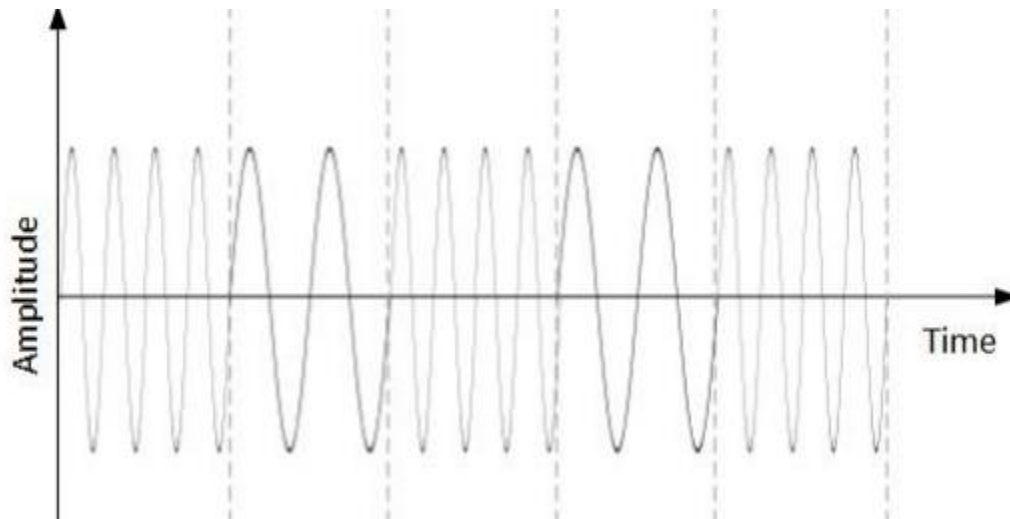
In this conversion technique, the amplitude of analog carrier signal is modified to reflect binary data.



When binary data represents digit 1, the amplitude is held; otherwise it is set to 0. Both frequency and phase remain same as in the original carrier signal.

## Frequency Shift Keying

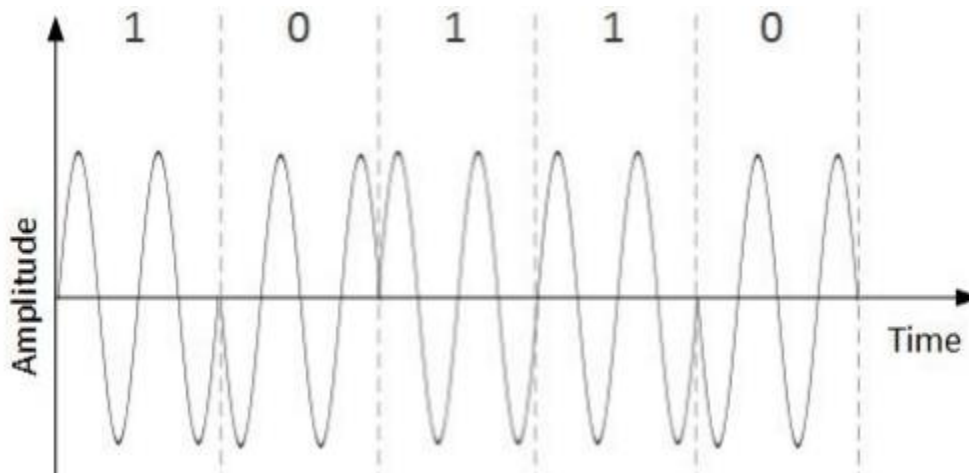
In this conversion technique, the frequency of the analog carrier signal is modified to reflect binary data.



This technique uses two frequencies,  $f_1$  and  $f_2$ . One of them, for example  $f_1$ , is chosen to represent binary digit 1 and the other one is used to represent binary digit 0. Both amplitude and phase of the carrier wave are kept intact.

### Phase Shift Keying

In this conversion scheme, the phase of the original carrier signal is altered to reflect the binary data.



When a new binary symbol is encountered, the phase of the signal is altered. Amplitude and frequency of the original carrier signal is kept intact.

### Quadrature Phase Shift Keying

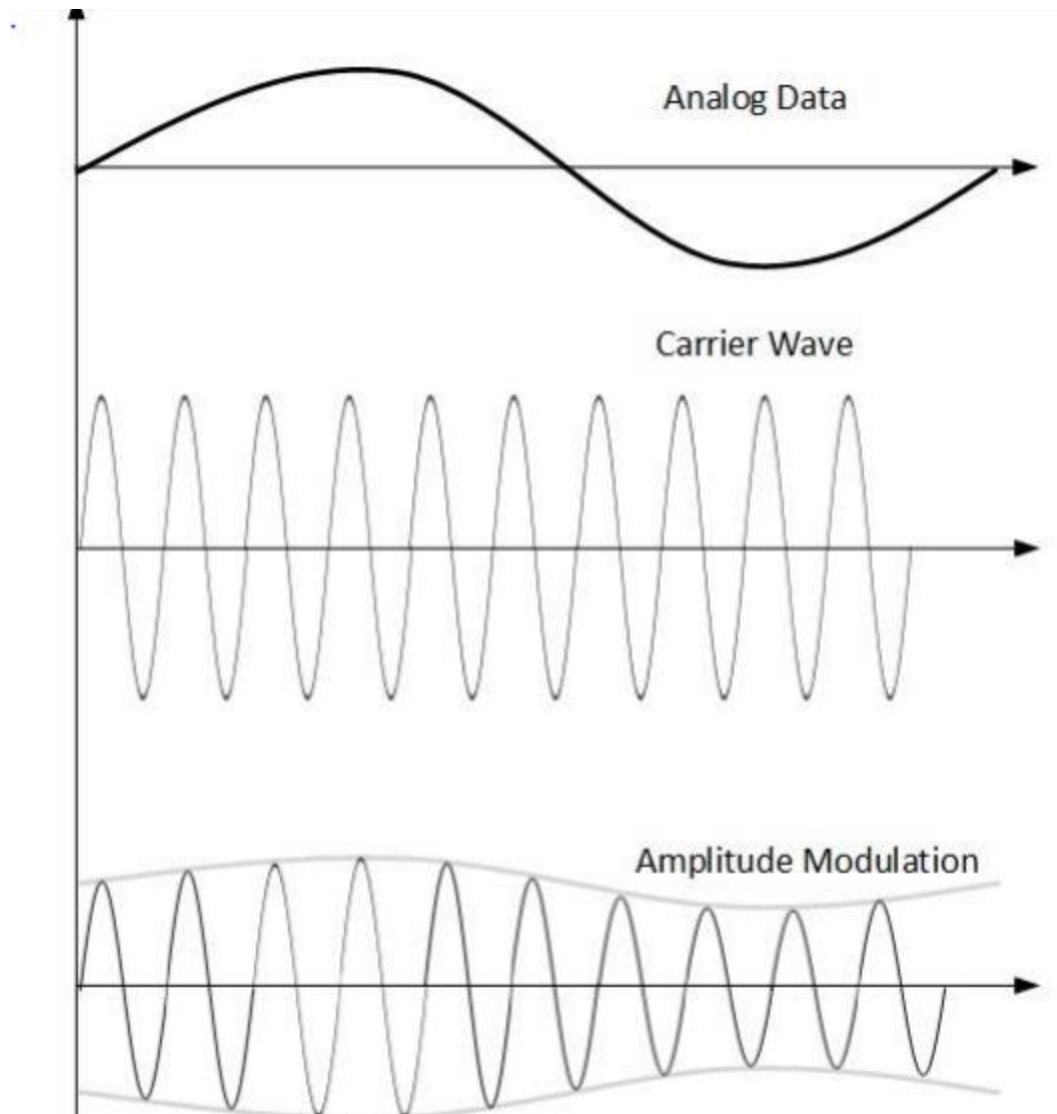
QPSK alters the phase to reflect two binary digits at once. This is done in two different phases. The main stream of binary data is divided equally into two sub-streams. The serial data is converted in to parallel in both sub-streams and then each stream is converted to digital signal using NRZ technique. Later, both the digital signals are merged together.

## Analog-to-Analog Conversion

Analog signals are modified to represent analog data. This conversion is also known as Analog Modulation. Analog modulation is required when bandpass is used. Analog to analog conversion can be done in three ways:

### 1. Amplitude Modulation

In this modulation, the amplitude of the carrier signal is modified to reflect the analog data.



Amplitude modulation is implemented by means of a multiplier. The amplitude of modulating signal (analog data) is multiplied by the amplitude of carrier frequency, which then reflects analog data. The frequency and phase of carrier signal remain unchanged.

### **Frequency Modulation**

In this modulation technique, the frequency of the carrier signal is modified to reflect the change in the voltage levels of the modulating signal (analog data).

### **Phase Modulation**

In the modulation technique, the phase of carrier signal is modulated in order to reflect the change in voltage (amplitude) of analog data signal.

Phase modulation is practically similar to Frequency Modulation, but in Phase modulation frequency of the carrier signal is not increased. Frequency of carrier is signal is changed (made dense and sparse) to reflect voltage change in the amplitude of modulating signal.

## **TRANSMISSION MEDIA**

The transmission media is nothing but the physical media over which communication takes place in computer networks.

### **Magnetic Media**

One of the most convenient way to transfer data from one computer to another, even before the birth of networking, was to save it on some storage media and transfer physical from one station to another. Though it may seem old-fashion way in today's world of high speed internet, but when the size of data is huge, the magnetic media comes into play.

### **Twisted Pair Cable**

A twisted pair cable is made of two plastic insulated copper wires twisted together to form a single media. Out of these two wires, only one carries actual signal and another is used for ground reference. The twists between wires are helpful in reducing noise (electromagnetic interference) and crosstalk.

There are two types of twisted pair cables:

- Shielded Twisted Pair (STP) Cable
- Unshielded Twisted Pair (UTP) Cable

### **Coaxial Cable**

Coaxial cable has two wires of copper. The core wire lies in the center and it is made of solid conductor. The core is enclosed in an insulating sheath. The second wire is wrapped around over the sheath and that too in turn encased by insulator sheath. This all is covered by plastic cover.

Because of its structure, the coax cable is capable of carrying high frequency signals than that of twisted pair cable. The wrapped structure provides it a good shield against noise and cross talk. Coaxial cables provide high bandwidth rates of up to 450 mbps.

There are three categories of coax cables namely, RG-59 (Cable TV), RG-58 (Thin Ethernet), and RG-11 (Thick Ethernet). RG stands for Radio Government. Cables are connected using BNC connector and BNC-T. BNC terminator is used to terminate the wire at the far ends.

### **Power Lines**

Power Line communication (PLC) is Layer-1 (Physical Layer) technology which uses power cables to transmit data signals. In PLC, modulated data is sent over the cables. The receiver on the other end de-modulates and interprets the data.

There are two types of PLCs:

- Narrow band PLC
- Broad band PLC

## **Fiber Optics**

Fiber Optic works on the properties of light. When light ray hits at critical angle, it tends to refract at 90 degree. This property has been used in fiber optic. The core of fiber optic cable is made of high quality glass or plastic. From one end of it light is emitted, it travels through it and at the other end light detector detects light stream and converts it to electric data.

Fiber Optic provides the highest mode of speed. It comes in two modes, one is single mode fiber and second is multimode fiber. Single mode fiber can carry a single ray of light whereas multimode is capable of carrying multiple beams of light.

Fiber Optic also comes in unidirectional and bidirectional capabilities. To connect and access fiber optic special type of connectors are used. These can be Subscriber Channel (SC), Straight Tip (ST), or MT-RJ.

## Unit-III

### Data Link Layer:

#### Services

1. **Providing services to the network layer:**
  - **Unacknowledged connectionless service.**  
Appropriate for low error rate and real-time traffic. Ex: Ethernet
  - **Acknowledged connectionless service.**  
Useful in unreliable channels, Wi-Fi. Ack /Timer/Resend
  - **Acknowledged connection-oriented service.**  
Guarantee frames are received exactly once and in the right order. Appropriate over long, unreliable links such as a satellite channel or a long-distance telephone circuit.
2. **Framing:** Frames are the streams of bits received from the network layer into manageable data units. This division of stream of bits is done by Data Link Layer.
3. **Physical Addressing:** The Data Link layer adds a header to the frame in order to define physical address of the sender or receiver of the frame, if the frames are to be distributed to different systems on the network.
4. **Flow Control:** A receiving node can receive the frames at a faster rate than it can process the frame. Without flow control, the receiver's buffer can overflow, and frames can get lost. To overcome this problem, the data link layer uses the flow control to prevent the sending node on one side of the link from overwhelming the receiving node on another side of the link.
5. **Error Control:** Error control is achieved by adding a trailer at the end of the frame. Duplication of frames are also prevented by using this mechanism. Data Link Layers adds mechanism to prevent duplication of frames.

**Error detection:** Errors can be introduced by signal attenuation and noise. Data Link Layer protocol provides a mechanism to detect one or more errors. This is achieved by adding error detection bits in the frame and then receiving node can perform an error check.

**Error correction:** Error correction is similar to the Error detection, except that receiving node not only detects the errors but also determine where the errors have occurred in the frame.

6. **Access Control:** Protocols of this layer determine which of the devices has control over the link at any given time, when two or more devices are connected to the same link.

**7. Reliable delivery:** Data Link Layer provides a reliable delivery service, i.e., transmits the network layer datagram without any error. A reliable delivery service is accomplished with transmissions and acknowledgements. A data link layer mainly provides the reliable delivery service over the links as they have higher error rates and they can be corrected locally, link at which an error occurs rather than forcing to retransmit the data.

**8. Half-Duplex & Full-Duplex:** In a Full-Duplex mode, both the nodes can transmit the data at the same time. In a Half-Duplex mode, only one node can transmit the data at the same time.

## **FRAMING:**

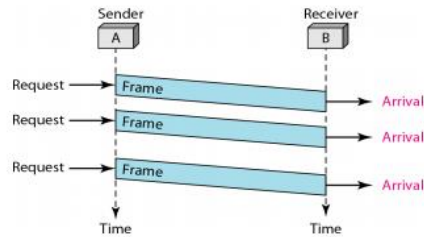
To provide service to the network layer, the data link layer must use the service provided to it by the physical layer. What the physical layer does is accept a raw bit stream and attempt to deliver it to the destination. This bit stream is not guaranteed to be error free. The number of bits received may be less than, equal to, or more than the number of bits transmitted, and they may have different values. It is up to the data link layer to detect and, if necessary, correct errors. The usual approach is for the data link layer to break the bit stream up into discrete frames and compute the checksum for each frame (framing). When a frame arrives at the destination, the checksum is recomputed. If the newly computed checksum is different from the one contained in the frame, the data link layer knows that an error has occurred and takes steps to deal with it (e.g., discarding the bad frame and possibly also sending back an error report). We will look at four framing methods:

1. Character count.
2. Flag bytes with byte stuffing.
3. Starting and ending flags, with bit stuffing.
4. Physical layer coding violations.

## **ELEMENTARY DATA LINK PROTOCOLS**

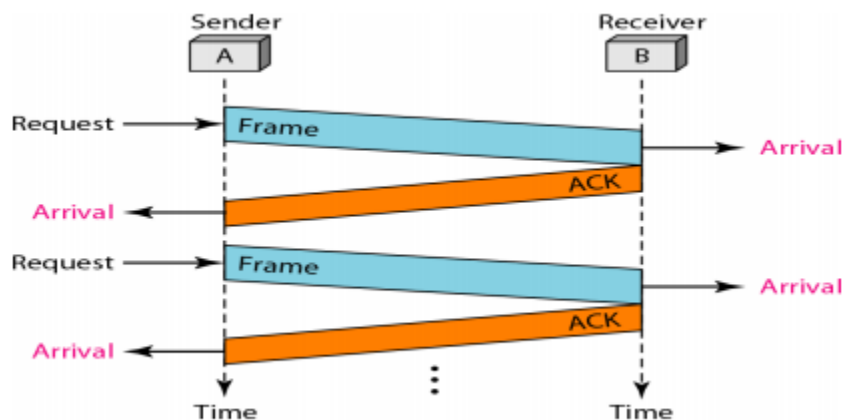
### **Simplest Protocol**





It is very simple. The sender sends a sequence of frames without even thinking about the receiver. Data are transmitted in one direction only. Both sender & receiver always ready. Processing time can be ignored. Infinite buffer space is available. And best of all, the communication channel between the data link layers never damages or loses frames. This thoroughly unrealistic protocol, which we will nickname “Utopia,”. The utopia protocol is unrealistic because it does not handle either flow control or error correction.

### Stop-and-wait Protocol



It is still very simple. The sender sends one frame and waits for feedback from the receiver. When the ACK arrives, the sender sends the next frame. It is Stop-and-Wait Protocol because the sender sends one frame, stops until it receives confirmation from the receiver (okay to go ahead), and then sends the next frame. We still have unidirectional communication for data frames, but auxiliary ACK frames (simple tokens of acknowledgment) travel from the other direction. We add flow control to our previous protocol.

### NOISY CHANNELS

Although the Stop-and-Wait Protocol gives us an idea of how to add flow control to its predecessor, noiseless channels are nonexistent. We can ignore the error (as we

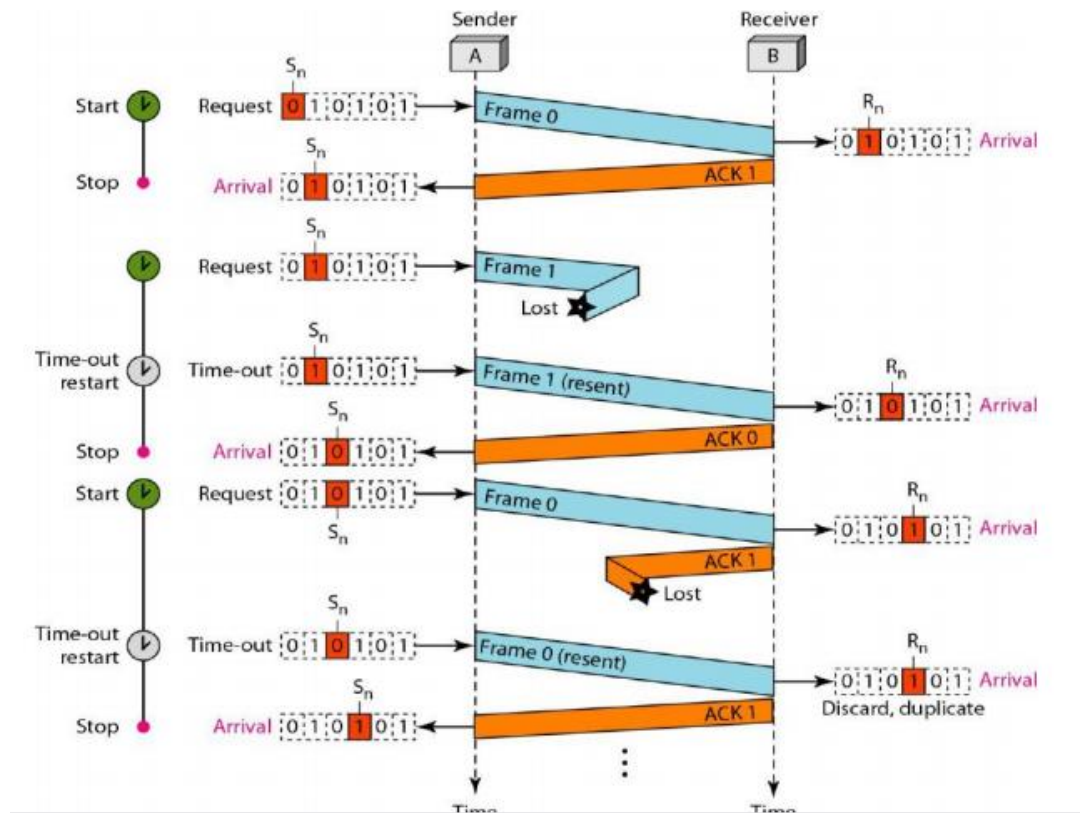
sometimes do), or we need to add error control to our protocols. We discuss three protocols in this section that use error control.

### **Sliding Window Protocols:**

- 1 Stop-and-Wait Automatic Repeat Request
- 2 Go-Back-N Automatic Repeat Request
- 3 Selective Repeat Automatic Repeat Request

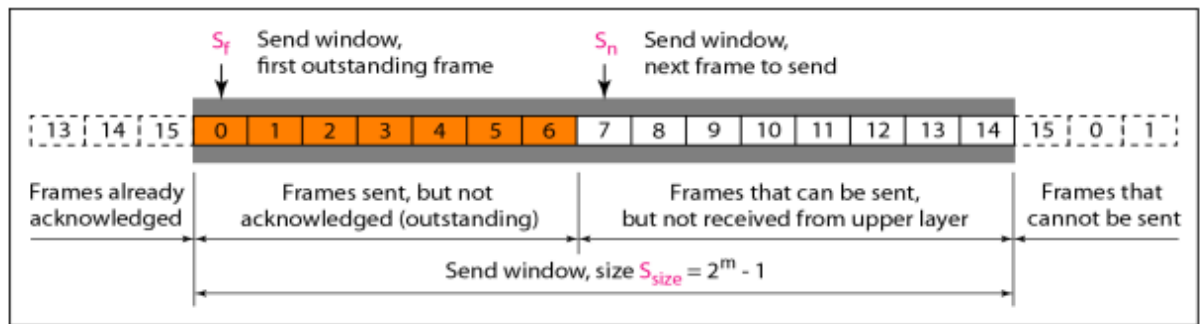
#### **1 Stop-and-Wait Automatic Repeat Request**

To detect and correct corrupted frames, we need to add redundancy bits to our data frame. When the frame arrives at the receiver site, it is checked and if it is corrupted, it is silently discarded. The detection of errors in this protocol is manifested by the silence of the receiver. Lost frames are more difficult to handle than corrupted ones. In our previous protocols, there was no way to identify a frame. The received frame could be the correct one, or a duplicate, or a frame out of order. The solution is to number the frames. When the receiver receives a data frame that is out of order, this means that frames were either lost or duplicated. The lost frames need to be resent in this protocol. If the receiver does not respond when there is an error, how can the sender know which frame to resend? To remedy this problem, the sender keeps a copy of the sent frame. At the same time, it starts a timer. If the timer expires and there is no ACK for the sent frame, the frame is resent, the copy is held, and the timer is restarted. Since the protocol uses the stop-and-wait mechanism, there is only one specific frame that needs an ACK. Error correction in Stop-and-Wait ARQ is done by keeping a copy of the sent frame and retransmitting of the frame when the timer expires. In Stop-and-Wait ARQ, we use sequence numbers to number the frames. The sequence numbers are based on modulo-2 arithmetic. In Stop-and-Wait ARQ, the acknowledgment number always announces in modulo-2 arithmetic the sequence number of the next frame expected.

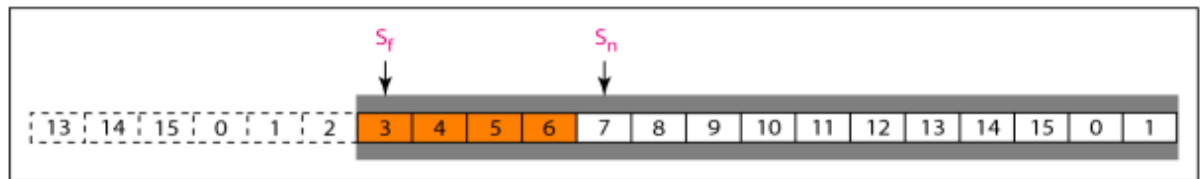


## 2. Go-Back-N Automatic Repeat Request

In this protocol we can send several frames before receiving acknowledgments; we keep a copy of these frames until the acknowledgments arrive. In the Go-Back-N Protocol, the sequence numbers are modulo  $2m$ , where  $m$  is the size of the sequence number field in bits. The sequence numbers range from 0 to  $2^m - 1$ . For example, if  $m$  is 4, the only sequence numbers are 0 through 15 inclusive.



a. Send window before sliding



b. Send window after sliding

The sender window at any time divides the possible sequence numbers into four regions. The first region, from the far left to the left wall of the window, defines the sequence numbers belonging to frames that are already acknowledged. The sender does not worry about these frames and keeps no copies of them. The second region, colored in Figure (a), defines the range of sequence numbers belonging to the frames that are sent and have an unknown status. The sender needs to wait to find out if these frames have been received or were lost. We call these outstanding frames. The third range, white in the figure, defines the range of sequence numbers for frames that can be sent; however, the corresponding data packets have not yet been received from the network layer. Finally, the fourth region defines sequence numbers that cannot be used until the window slides the send window is an abstract concept defining an imaginary box of size  $2m - 1$  with three variables:  $S_f$ ,  $S_n$ , and  $S_{size}$ . The variable  $S_f$  defines the sequence number of the first (oldest) outstanding frame. The variable  $S_n$  holds the sequence number that will be assigned to the next frame to be sent. Finally, the variable  $S_{size}$  defines the size of the window. Figure (b) shows how a send window can slide one or more slots to the right when an acknowledgment arrives from the other end. The acknowledgments in this protocol are cumulative, meaning that more than one frame can be acknowledged by an ACK frame. In Figure, frames 0, 1, and 2 are acknowledged, so the window has slide to the right three slots. Note that the value of  $S_f$  is 3 because frame 3 is now the first outstanding frame. The send window can slide one or more slots when a valid acknowledgment arrives.

**Receiver window:** variable  $R_n$  (receive window, next frame expected) the sequence numbers to the left of the window belong to the frames already received and acknowledged; the sequence numbers to the right of this window define the frames that cannot be received. Any received frame with a sequence number in these two regions is

discarded. Only a frame with a sequence number matching the value of  $R_n$  is accepted and acknowledged. The receive window also slides, but only one slot at a time. When a correct frame is received (and a frame is received only one at a time), the window slide (see below figure for receiving window) The receive window is an abstract concept defining an imaginary box of size 1 with one single variable  $R_n$ . The window slides when a correct frame has arrived; sliding occurs one slot at a time.

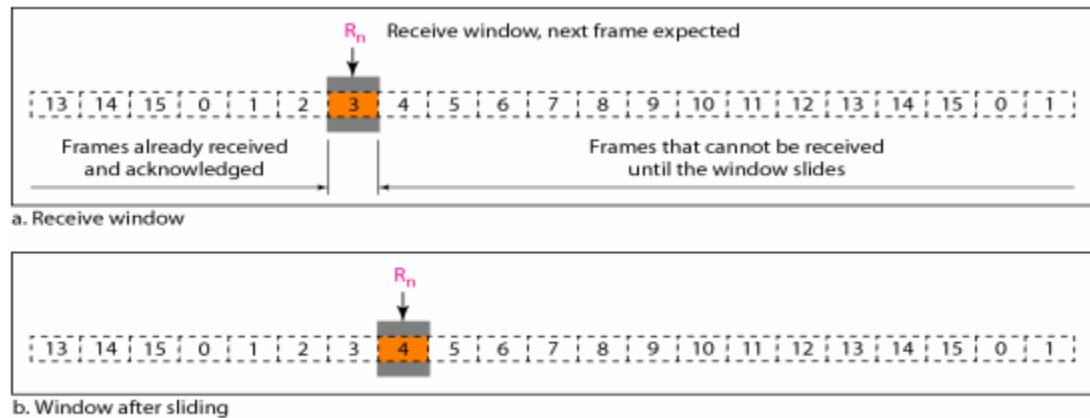
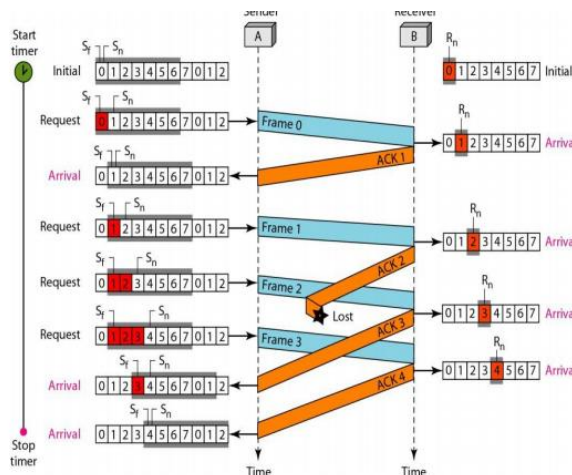
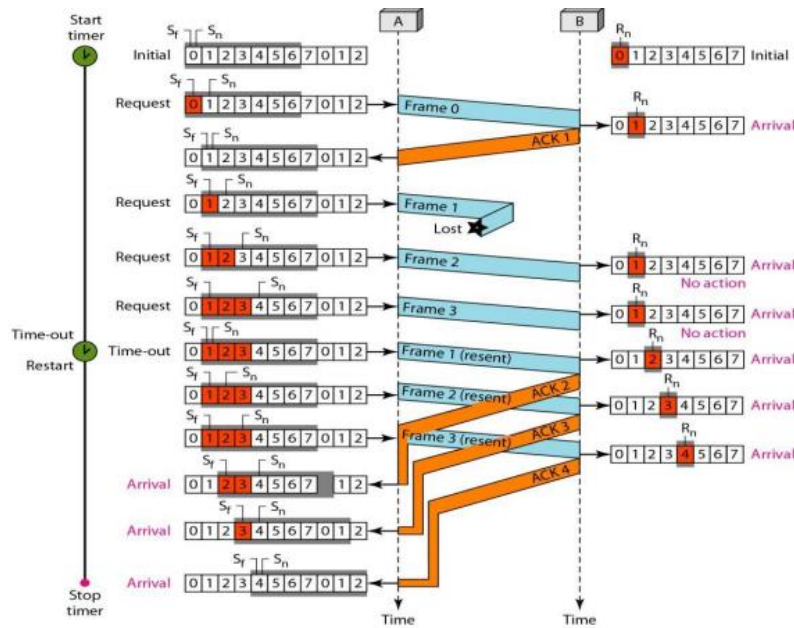


Fig: Receiver window (before sliding (a), After sliding (b))

Below figure is an example (if ack lost) of a case where the forward channel is reliable, but the reverse is not. No data frames are lost, but some ACKs are delayed and one is lost. The example also shows how cumulative acknowledgments can help if acknowledgments are delayed or lost



Below figure is an example (if frame lost)

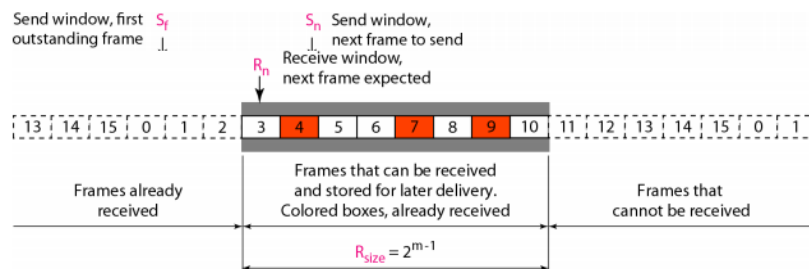


### 3 Selective Repeat Automatic Repeat Request

In Go-Back-N ARQ, The receiver keeps track of only one variable, and there is no need to buffer out-of-order frames; they are simply discarded. However, this protocol is very inefficient for a noisy link. In a noisy link a frame has a higher probability of damage, which means the resending of multiple frames. This resending uses up the bandwidth and slows down the transmission. For noisy links, there is another mechanism that does not resend N frames when just one frame is damaged; only the damaged frame is resent. This mechanism is called Selective Repeat ARQ. It is more efficient for noisy links, but the processing at the receiver is more complex.

**Sender Window** (explain go-back N sender window concept (before & after sliding.)

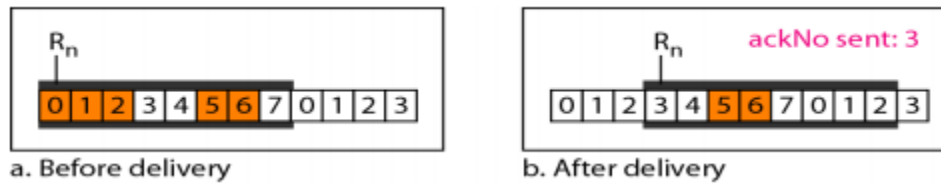
The only difference in sender window between Go-back N and Selective Repeat is Window size)



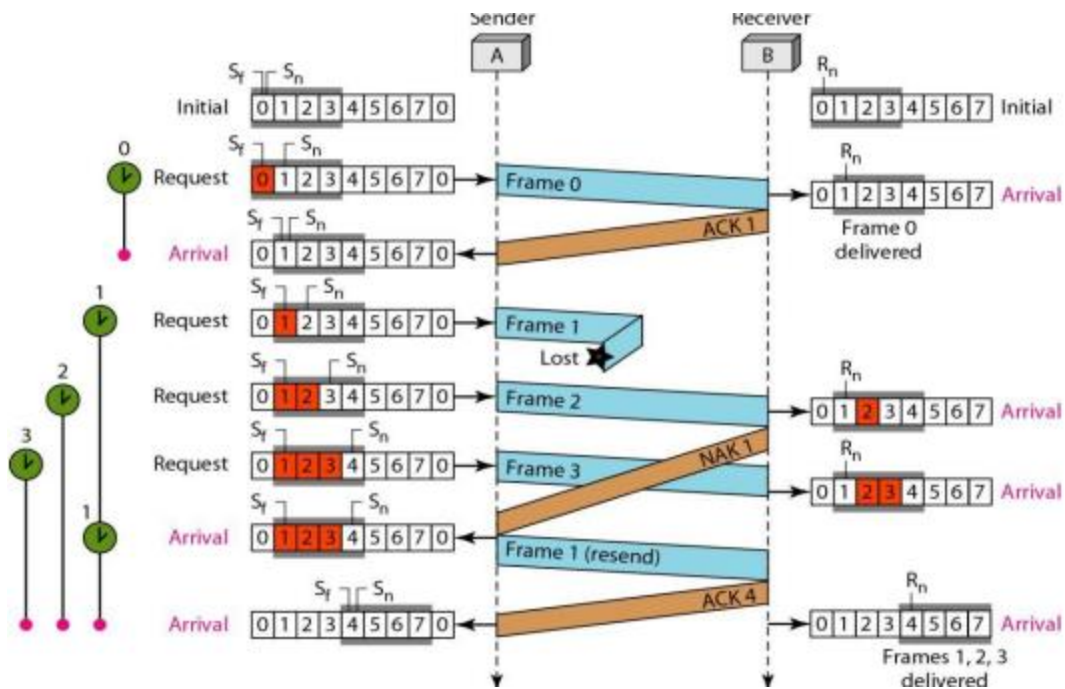
### Receiver window

The receiver window in Selective Repeat is totally different from the one in Go Back-N. First, the size of the receive window is the same as the size of the send window ( $2m-1$ ). The Selective Repeat Protocol allows as many frames as the size of the receiver window to arrive out of order and be kept until there is a set of in order frames to be delivered to the network layer. Because the sizes of the send window and receive window are the same, all the frames in the send window can arrive out of order and be stored until they can be delivered. However the receiver never delivers packets out of order to the network layer. Above Figure shows the receive window. Those slots inside the window that are colored define frames that have arrived out of order and are waiting for their neighbors to arrive before delivery to the network layer. In Selective Repeat ARQ, the size of the sender and receiver window must be at most one-half of  $2m$ .

### Delivery of Data in Selective Repeat ARQ:



### Flow Diagram



### Differences between Go-Back N & Selective Repeat

One main difference is the number of timers. Here, each frame sent or resent needs a timer, which means that the timers need to be numbered (0, 1, 2, and 3). The timer for frame 0 starts at the first request, but stops when the ACK for this frame arrives.

There are two conditions for the delivery of frames to the network layer: First, a set of consecutive frames must have arrived. Second, the set starts from the beginning of the window. After the first arrival, there was only one frame and it started from the beginning of the window. After the last arrival, there are three frames and the first one starts from the beginning of the window.

Another important point is that a NAK is sent.

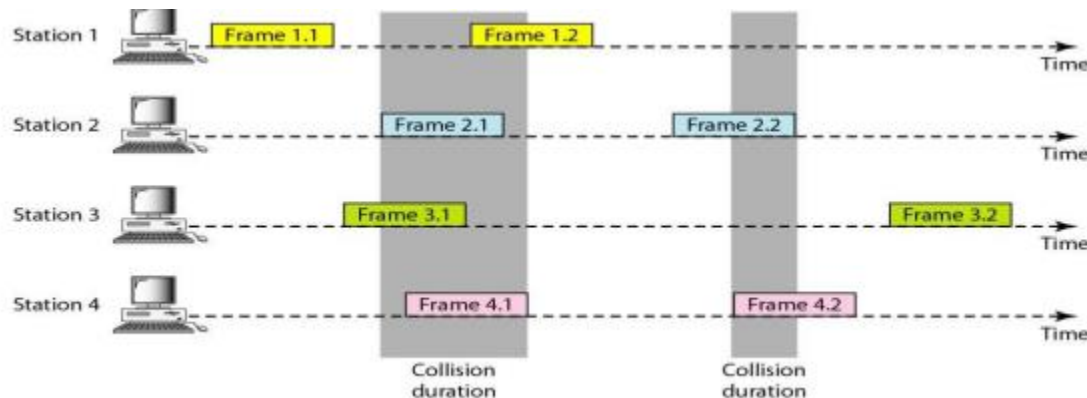
The next point is about the ACKs. Notice that only two ACKs are sent here. The first one acknowledges only the first frame; the second one acknowledges three frames. In Selective Repeat, ACKs are sent when data are delivered to the network layer. If the data belonging to  $n$  frames are delivered in one shot, only one ACK is sent for all of them.

## ALOHA

### 1 Pure ALOHA



The original ALOHA protocol is called pure ALOHA. This is a simple, but elegant protocol. The idea is that each station sends a frame whenever it has a frame to send. However, since there is only one channel to share, there is the possibility of collision between frames from different stations. Below Figure shows an example of frame collisions in pure ALOHA.

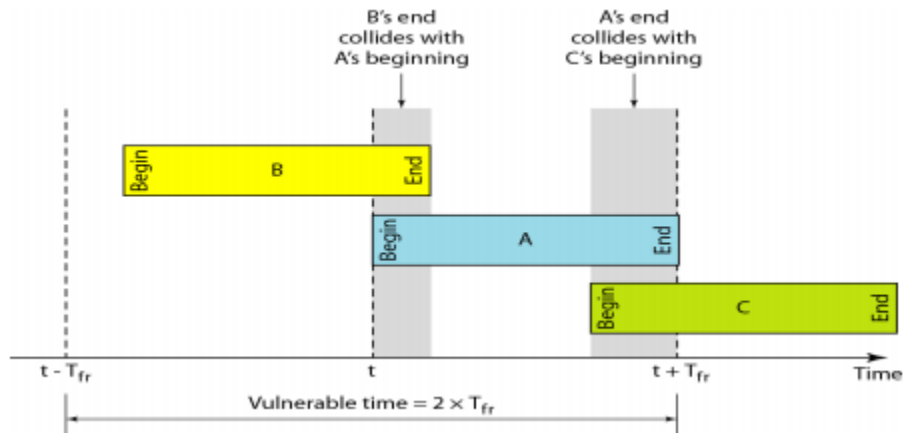


In pure ALOHA, the stations transmit frames whenever they have data to send.

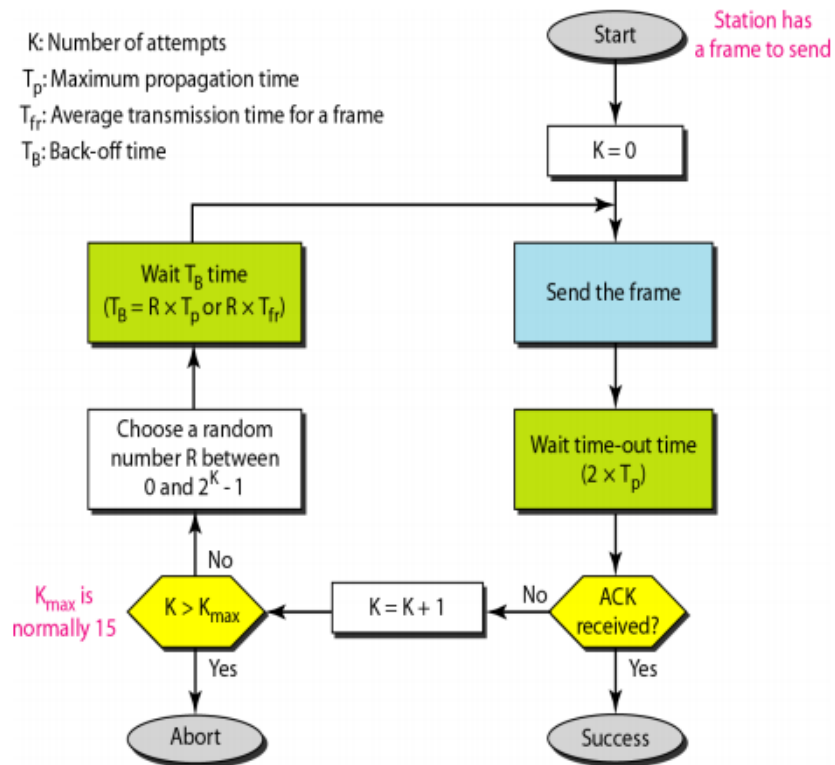
When two or more stations transmit simultaneously, there is collision and the frames are destroyed.

- In pure ALOHA, whenever any station transmits a frame, it expects the acknowledgement from the receiver.
- If acknowledgement is not received within specified time, the station assumes that the frame (or acknowledgement) has been destroyed.
- If the frame is destroyed because of collision the station waits for a random amount of time and sends it again. This waiting time must be random otherwise same frames will collide again and again.
- Therefore pure ALOHA dictates that when time-out period passes, each station must wait for a random amount of time before resending its frame. This randomness will help avoid more collisions.

**Vulnerable time** let us find the length of time, the vulnerable time, in which there is a possibility of collision. We assume that the stations send fixed length frames with each frame taking  $T_{fr}$  S to send. Below Figure shows the vulnerable time for station A.



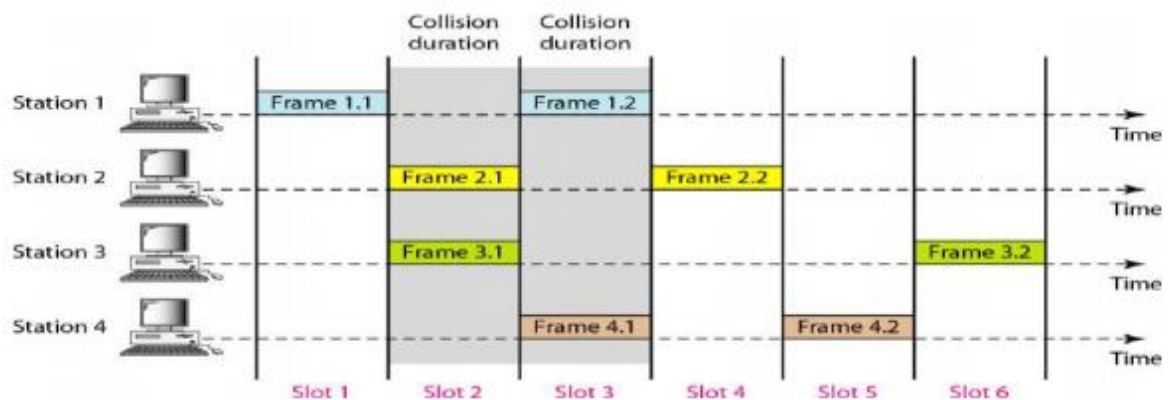
Station A sends a frame at time  $t$ . Now imagine station B has already sent a frame between  $t - T_{fr}$  and  $t$ . This leads to a collision between the frames from station A and station B. The end of B's frame collides with the beginning of A's frame. On the other hand, suppose that station C sends a frame between  $t$  and  $t + T_{fr}$ . Here, there is a collision between frames from station A and station C. The beginning of C's frame collides with the end of A's frame. Looking at Figure, we see that the vulnerable time, during which a collision may occur in pure ALOHA, is 2 times the frame transmission time. Pure ALOHA vulnerable time =  $2 \times T_{fr}$



**Procedure for pure ALOHA protocol**

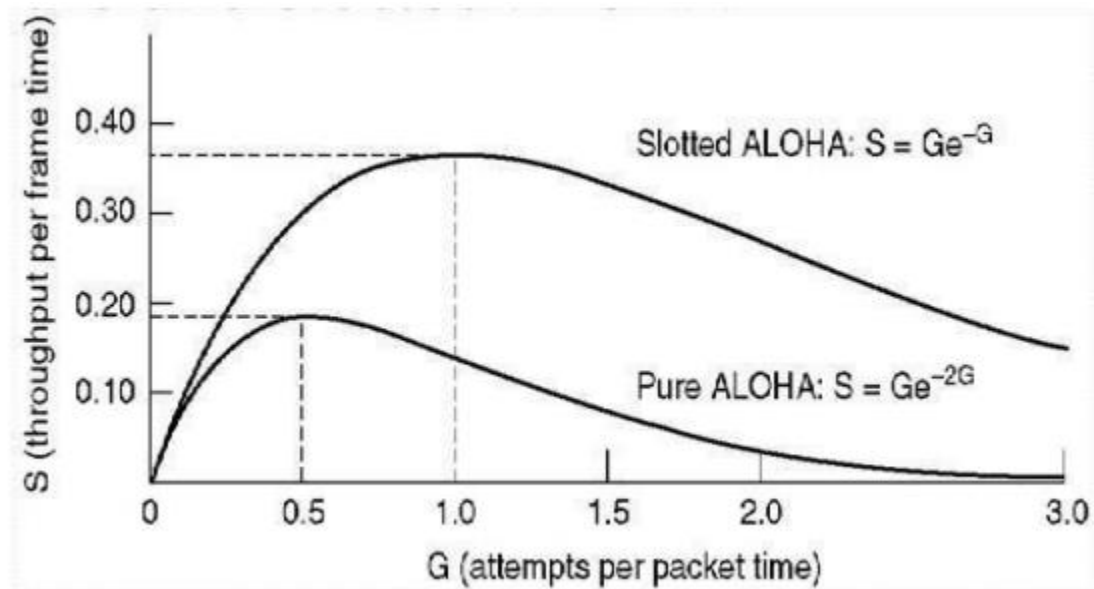
## 2 Slotted ALOHA

In slotted ALOHA we divide the time into slots of  $T_{fr}$  s and force the station to send only at the beginning of the time slot. Figure 3 shows an example of frame collisions in slotted ALOHA



Because a station is allowed to send only at the beginning of the synchronized time slot, if a station misses this moment, it must wait until the beginning of the next time slot. This means that the station which started at the beginning of this slot has already finished sending its frame. Of course, there is still the possibility of collision if two stations try to send at the beginning of the same time slot. However, the vulnerable time is now reduced to one-half, equal to  $T_{fr}$

### Comparison between Pure Aloha & Slotted Aloha



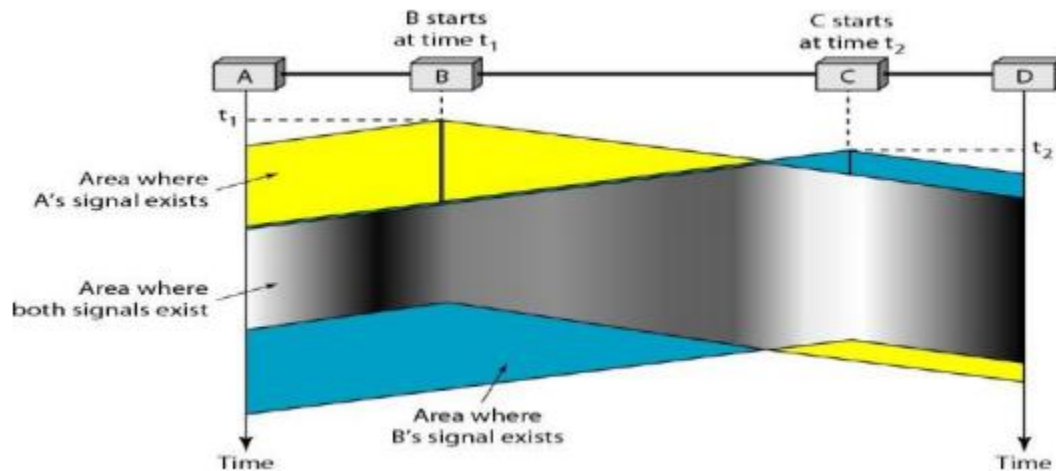
### Carrier Sense Multiple Access (CSMA)

To minimize the chance of collision and, therefore, increase the performance, the CSMA method was developed. The chance of collision can be reduced if a station senses the medium before trying to use it. Carrier sense multiple access (CSMA) requires that each station first listen to the medium (or check the state of the medium) before sending. In other words, CSMA is based on the principle "sense before transmit" or "listen before talk."

CSMA can reduce the possibility of collision, but it cannot eliminate it. The reason for this is shown in below Figure. Stations are connected to a shared channel (usually a dedicated medium).

The possibility of collision still exists because of propagation delay; station may sense the medium and find it idle, only because the first bit sent by another station has not yet been received.

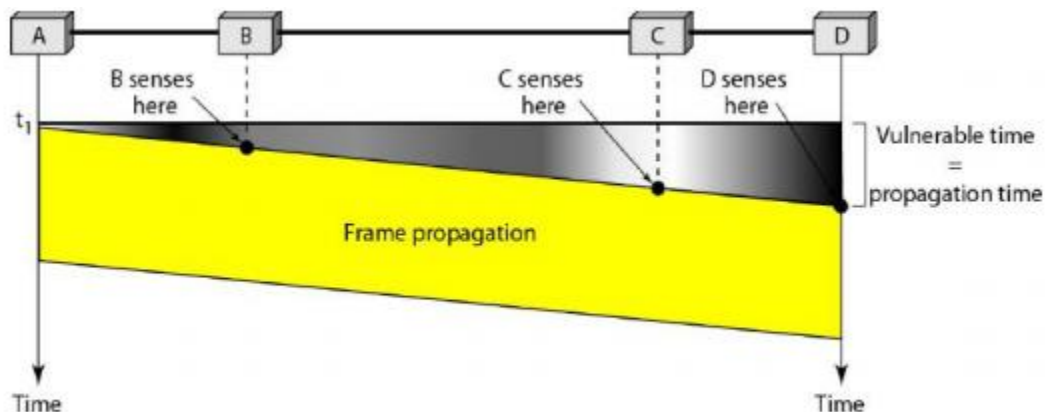
At time  $t_1$  station B senses the medium and finds it idle, so it sends a frame. At time  $t_2$  ( $t_2 > t_1$ ) station C senses the medium and finds it idle because, at this time, the first bits from station B have not reached station C. Station C also sends a frame. The two signals collide and both frames are destroyed.



Space/time model of the collision in CSMA

### Vulnerable Time

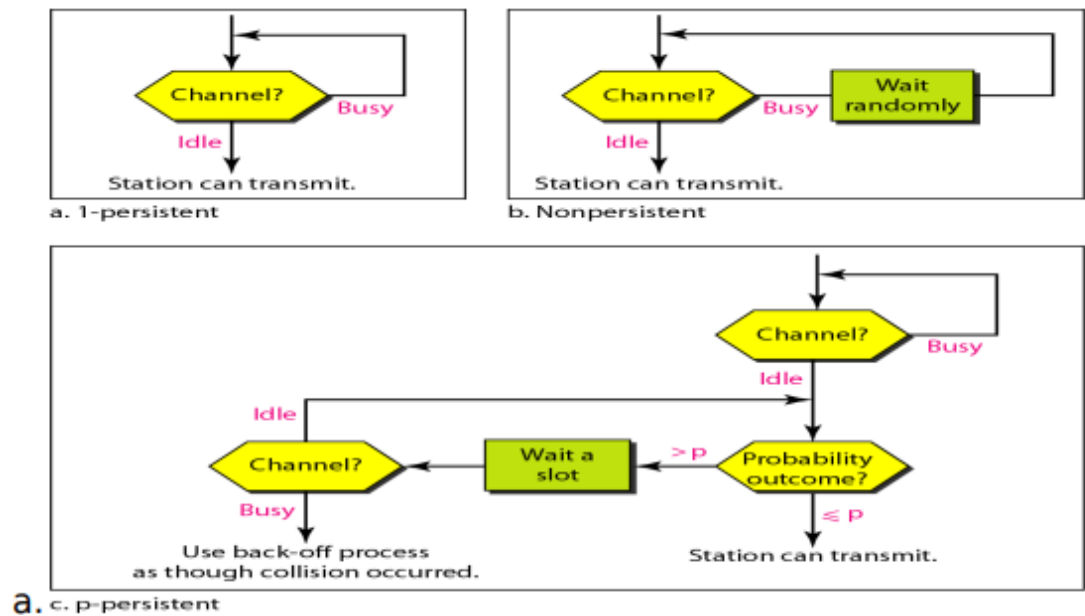
The vulnerable time for CSMA is the propagation time  $T_p$ . This is the time needed for a signal to propagate from one end of the medium to the other. When a station sends a frame, and any other station tries to send a frame during this time, a collision will result. But if the first bit of the frame reaches the end of the medium, every station will already have heard the bit and will refrain from sending.



Vulnerable time in CSMA

### Persistence Methods:

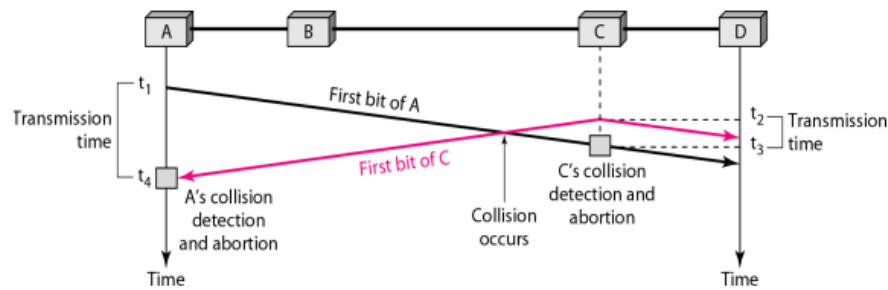
1. Persistent
2. Non-Persistent
3. P-Persistent



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

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

To better understand CSMA/CD, let us look at the first bits transmitted by the two stations involved in the collision. Although each station continues to send bits in the frame until it detects the collision, we show what happens as the first bits collide. In below Figure, stations A and C are involved in the collision.

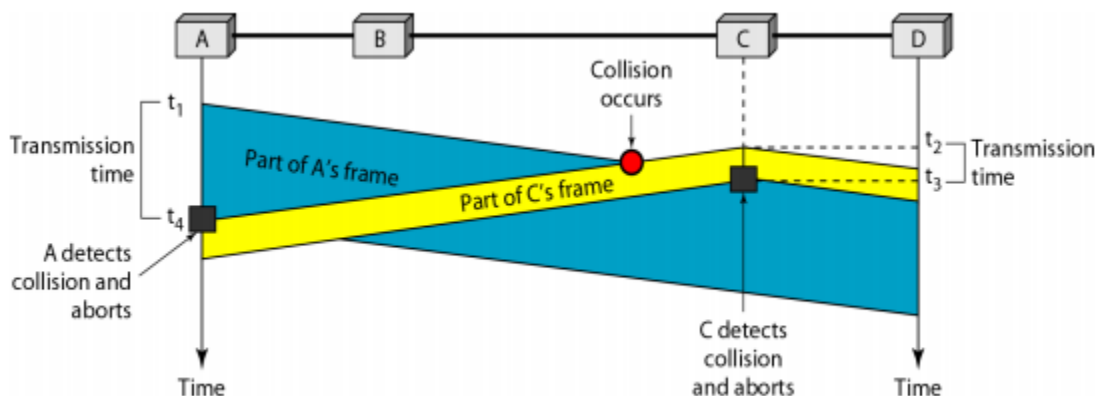


### Collision of the first bit in CSMA/CD

At time  $t_1$ , station A has executed its persistence procedure and starts sending the bits of its frame. At time  $t_2$ , station C has not yet sensed the first bit sent by A. Station C executes its persistence procedure and starts sending the bits in its frame, which propagate both to the left and to the right. The collision occurs sometime after time  $t_2$ . Station C detects a collision at time  $t_3$  when it receives the first bit of A's frame. Station C immediately (or after a short time, but we assume immediately) aborts transmission. Station A detects collision at time  $t_4$  when it receives the first bit of C's frame; it also immediately aborts transmission. Looking at the figure, we see that A transmits for the duration  $t_4 - t_1$ ; C transmits for the duration  $t_3 - t_2$ .

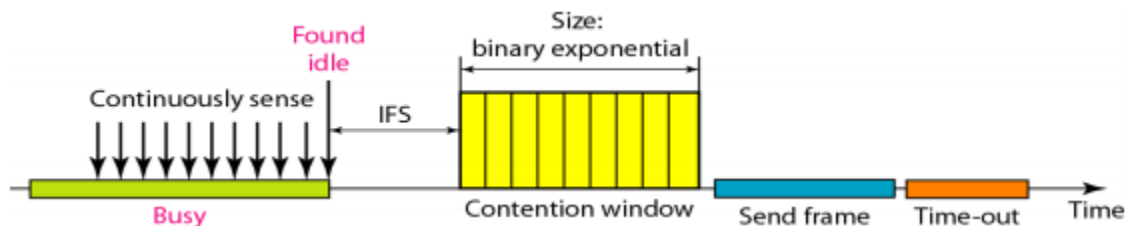
### Minimum Frame Size

For CSMA/CD to work, we need a restriction on the frame size. Before sending the last bit of the frame, the sending station must detect a collision, if any, and abort the transmission. This is so because the station, once the entire frame is sent, does not keep a copy of the frame and does not monitor the line for collision detection. Therefore, the frame transmission time  $T_{fr}$  must be at least two times the maximum propagation time  $T_p$ . To understand the reason, let us think about the worst-case scenario. If the two stations involved in a collision are the maximum distance apart, the signal from the first takes time  $T_p$  to reach the second, and the effect of the collision takes another time  $T_p$  to reach the first. So the requirement is that the first station must still be transmitting after  $2T_p$ .



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

We need to avoid collisions on wireless networks because they cannot be detected. Carrier sense multiple access with collision avoidance (CSMA/CA) was invented for wireless network. Collisions are avoided through the use of CSMA/CA's three strategies: the inter frame space, the contention window, and acknowledgments.



## Error Detection

### Error

A condition when the receiver's information does not match with the sender's information. During transmission, digital signals suffer from noise that can introduce errors in the binary bits travelling from sender to receiver. That means a 0 bit may change to 1 or a 1 bit may change to 0.

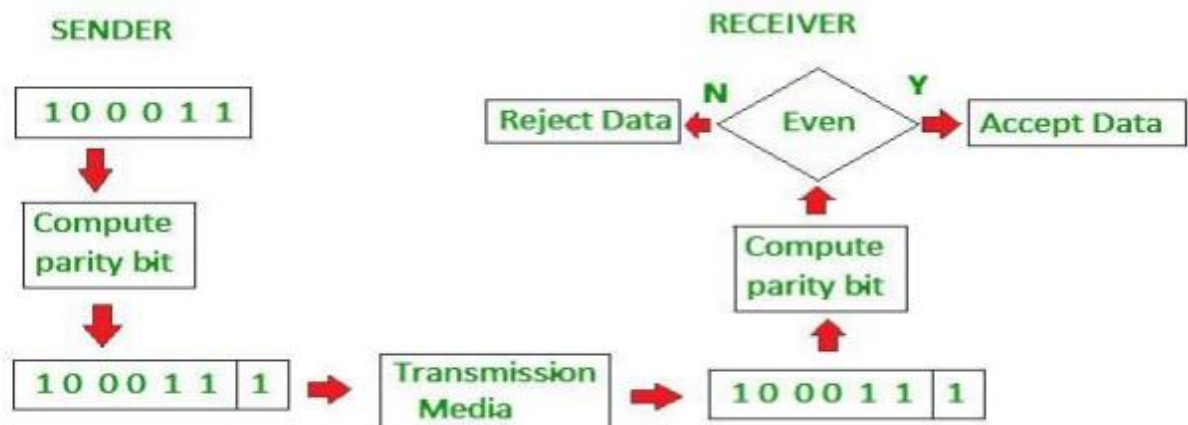
**Error Detecting Codes:** Whenever a message is transmitted, it may get scrambled by noise or data may get corrupted. To avoid this, we use error-detecting codes which are additional data added to a given digital message to help us detect if any error has occurred during transmission of the message. Basic approach used for error detection is the use of redundancy bits, where additional bits are added to facilitate detection of errors. Some popular techniques for error detection are:

1. Simple Parity check
2. Two-dimensional Parity check
3. Checksum
4. Cyclic redundancy check

### Simple Parity check

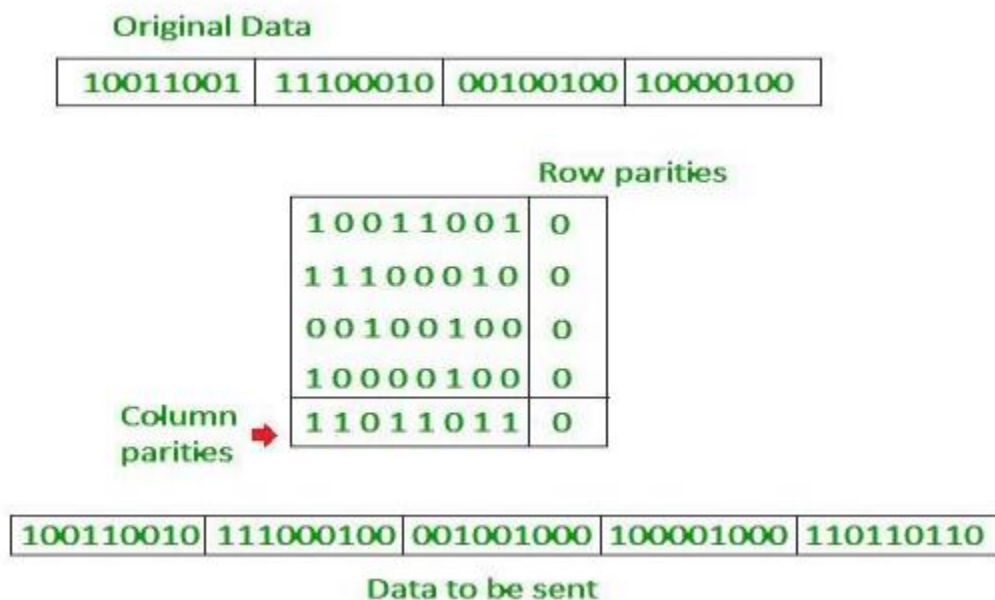


Blocks of data from the source are subjected to a check bit or parity bit generator form, where a parity of: 1 is added to the block if it contains odd number of 1's, and 0 is added if it contains even number of 1's. This scheme makes the total number of 1's even, that is why it is called even parity checking.



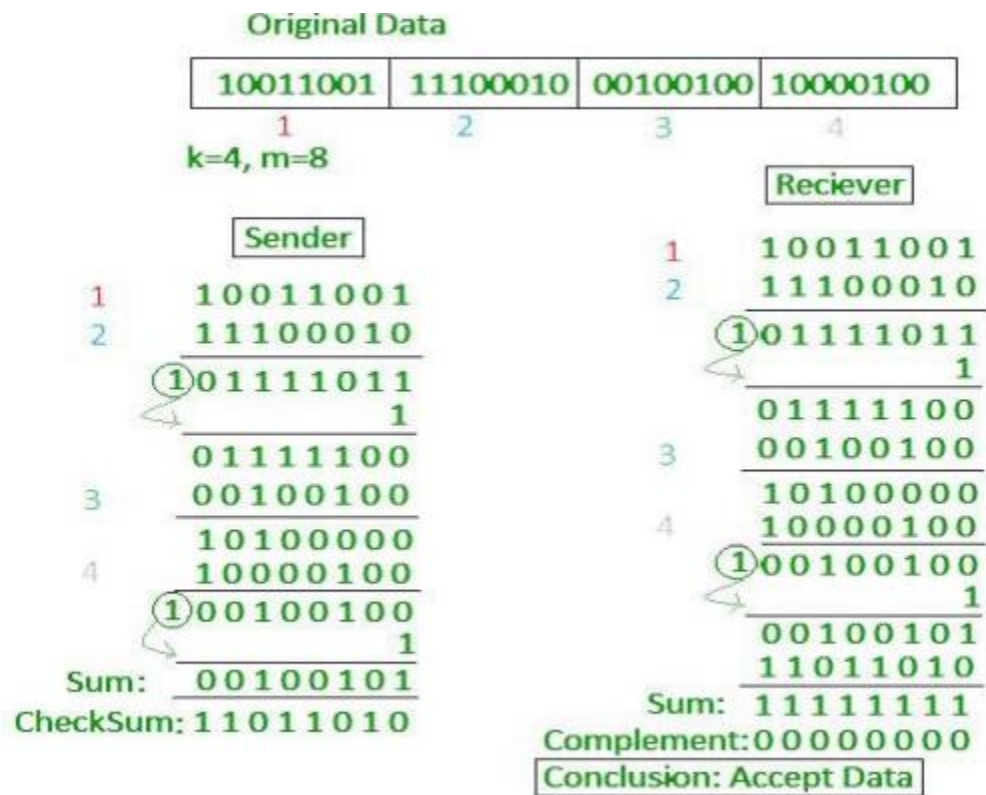
### Two-dimensional Parity check

Parity check bits are calculated for each row, which is equivalent to a simple parity check bit. Parity check bits are also calculated for all columns, then both are sent along with the data. At the receiving end these are compared with the parity bits calculated on the received data.

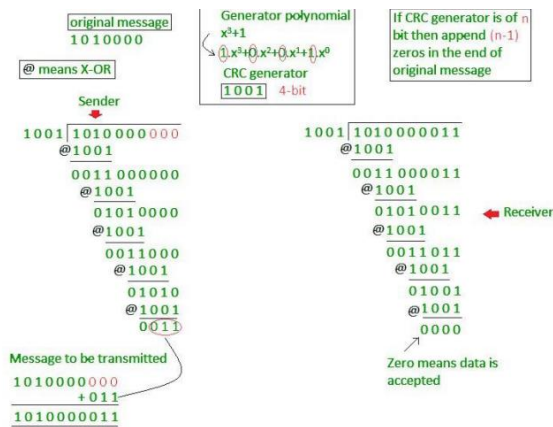


## Checksum

- In checksum error detection scheme, the data is divided into  $k$  segments each of  $m$  bits.
- In the sender's end the segments are added using 1's complement arithmetic to get the sum. The sum is complemented to get the checksum.
- The checksum segment is sent along with the data segments.
- At the receiver's end, all received segments are added using 1's complement arithmetic to get the sum. The sum is complemented.
- If the result is zero, the received data is accepted; otherwise discarded.



## Cyclic redundancy check (CRC)



Unlike checksum scheme, which is based on addition, CRC is based on binary division.

- In CRC, a sequence of redundant bits, called cyclic redundancy check bits, are appended to the end of data unit so that the resulting data unit becomes exactly divisible by a second, predetermined binary number.
- At the destination, the incoming data unit is divided by the same number. If at this step there is no remainder, the data unit is assumed to be correct and is therefore accepted.
- A remainder indicates that the data unit has been damaged in transit and therefore must be rejected.

## Error Correction

Error Correction codes are used to detect and correct the errors when data is transmitted from the sender to the receiver

Error Correction can be handled in two ways:

**Backward error correction:** Once the error is discovered, the receiver requests the sender to retransmit the entire data unit.

**Forward error correction:** In this case, the receiver uses the error-correcting code which automatically corrects the errors. A single additional bit can detect the error, but cannot correct it.

For correcting the errors, one has to know the exact position of the error. For example, if we want to calculate a single-bit error, the error correction code will determine which one of seven bits is in error. To achieve this, we have to add some additional redundant bits.

Suppose  $r$  is the number of redundant bits and  $d$  is the total number of the data bits. The number of redundant bits  $r$  can be calculated by using the formula:  $2^r \geq d + r + 1$  the value of  $r$  is calculated by using the above formula. For example, if the value of  $d$  is 4, then the possible smallest value that satisfies the above relation would be 3.

To determine the position of the bit which is in error, a technique developed by R.W Hamming is Hamming code which can be applied to any length of the data unit and uses the relationship between data units and redundant units.

**Hamming Code Parity bits:** The bit which is appended to the original data of binary bits so that the total number of 1s is even or odd.

**Even parity:** To check for even parity, if the total number of 1s is even, then the value of the parity bit is 0. If the total number of 1s occurrences is odd, then the value of the parity bit is 1.

**Odd Parity:** To check for odd parity, if the total number of 1s is even, then the value of parity bit is 1. If the total number of 1s is odd, then the value of parity bit is 0.

### Algorithm of hamming code:

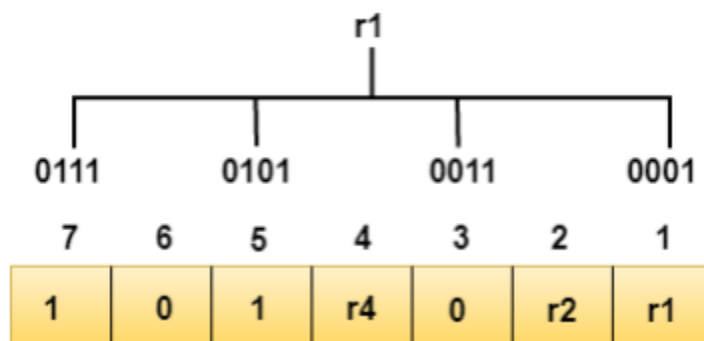
An information of 'd' bits are added to the redundant bits 'r' to form d+r. The location of each of the (d+r) digits is assigned a decimal value. The 'r' bits are placed in the positions 1,2,.....2k-1 At the receiving end, the parity bits are recalculated. The decimal value of the parity bits determines the position of an error.

### Relationship b/w Error position & binary number

Error Position	Binary Number
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

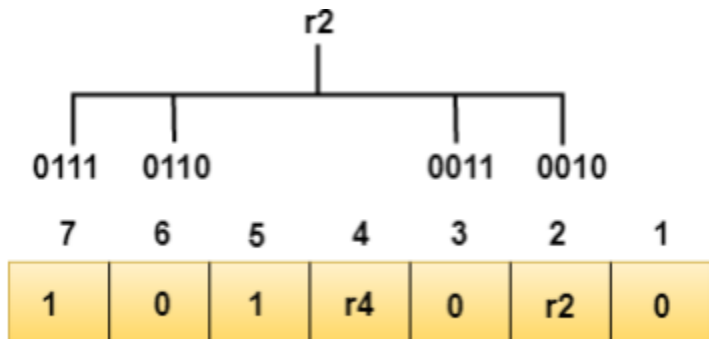
### Determining the Parity bits

**Determining the r1 bit:** The r1 bit is calculated by performing a parity check on the bit positions whose binary representation includes 1 in the first position.

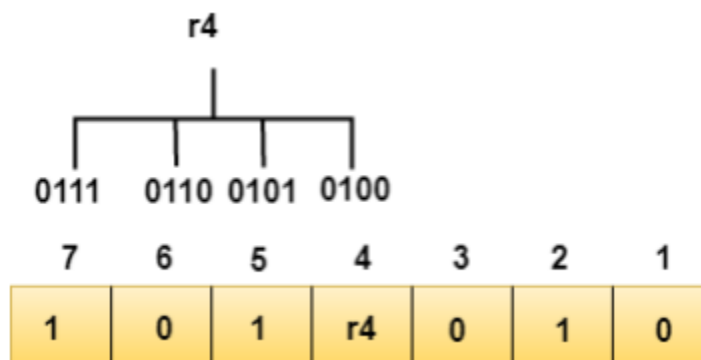


We observe from the above figure that the bit position that includes 1 in the first position are 1, 3, 5, and 7. Now, we perform the even-parity check at these bit positions. The total number of 1 at these bit positions corresponding to r1 is even, therefore, the value of the r1 bit is 0.

**Determining r2 bit:** The r2 bit is calculated by performing a parity check on the bit positions whose binary representation includes 1 in the second position.



**Determining r4 bit:** The r4 bit is calculated by performing a parity check on the bit positions whose binary representation includes 1 in the third position.



## Piggybacking

A technique called piggybacking is used to improve the efficiency of the bidirectional protocols. When a frame is carrying data from A to B, it can also carry control information about arrived (or lost) frames from B; when a frame is carrying data from B to A, it can also carry control information about the arrived (or lost) frames from

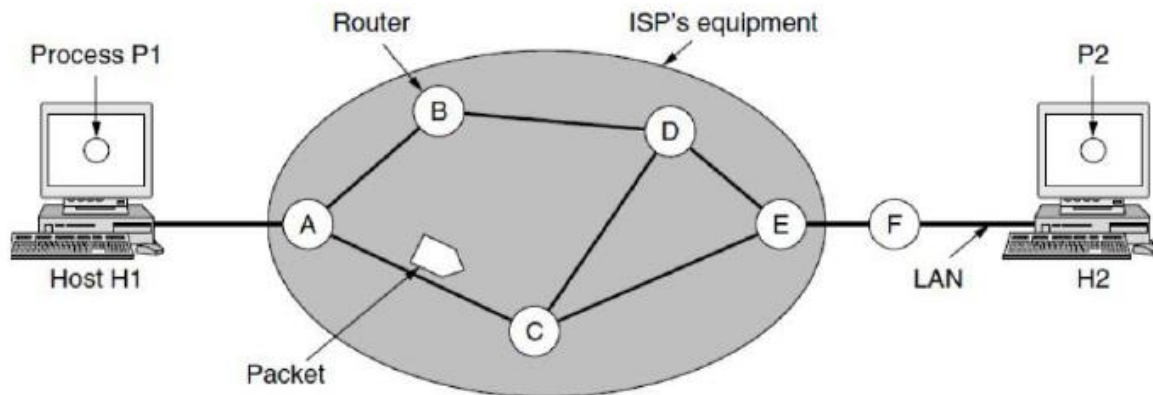
## Unit-IV

### Network Layer

#### **DESIGN ISSUES**

In the following sections, we will give an introduction to some of the issues that the designers of the network layer must grapple with. These issues include the service provided to the transport layer and the internal design of the network.

#### **Store-and-Forward Packet Switching**



**The environment of the network layer protocols.**

This equipment is used as follows. A host with a packet to send transmits it to the nearest router, either on its own LAN or over a point-to-point link to the ISP. The packet is stored there until it has fully arrived and the link has finished its processing by verifying the checksum. Then it is forwarded to the next router along the path until it reaches the destination host, where it is delivered. This mechanism is store-and-forward packet switching.

#### **Services Provided to the Transport Layer**

The network layer provides services to the transport layer at the network layer/transport layer interface. An important question is precisely what kind of services the network layer provides to the transport layer. The services need to be carefully designed with the following goals in mind:

- The services should be independent of the router technology.
- The transport layer should be shielded from the number, type, and topology of the routers present.
- The network addresses made available to the transport layer should use a uniform numbering plan, even across LANs and WANs.

## **NETWORK ROUTING**

When a device has multiple paths to reach a destination, it always selects one path by preferring it over others. This selection process is termed as Routing. Routing is done by special network devices called routers or it can be done by means of software processes. The software based routers have limited functionality and limited scope.

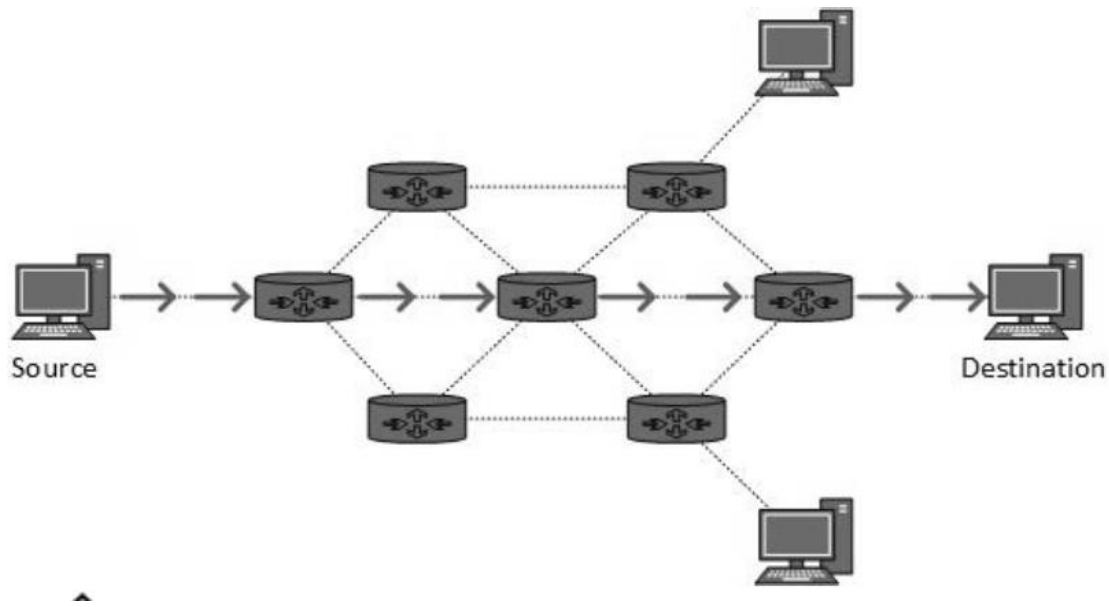
A router is always configured with some default route. A default route tells the router where to forward a packet if there is no route found for specific destination. In case there are multiple path existing to reach the same destination, router can make decision based on the following information:

- Hop Count
- Bandwidth
- Metric
- Prefix-length
- Delay

Routes can be statically configured or dynamically learnt. One route can be configured to be preferred over others.

### **Unicast routing**

Most of the traffic on the internet and intranets known as unicast data or unicast traffic is sent with specified destination. Routing unicast data over the internet is called unicast routing. It is the simplest form of routing because the destination is already known. Hence the router just has to look up the routing table and forward the packet to next hop.



## Broadcast routing

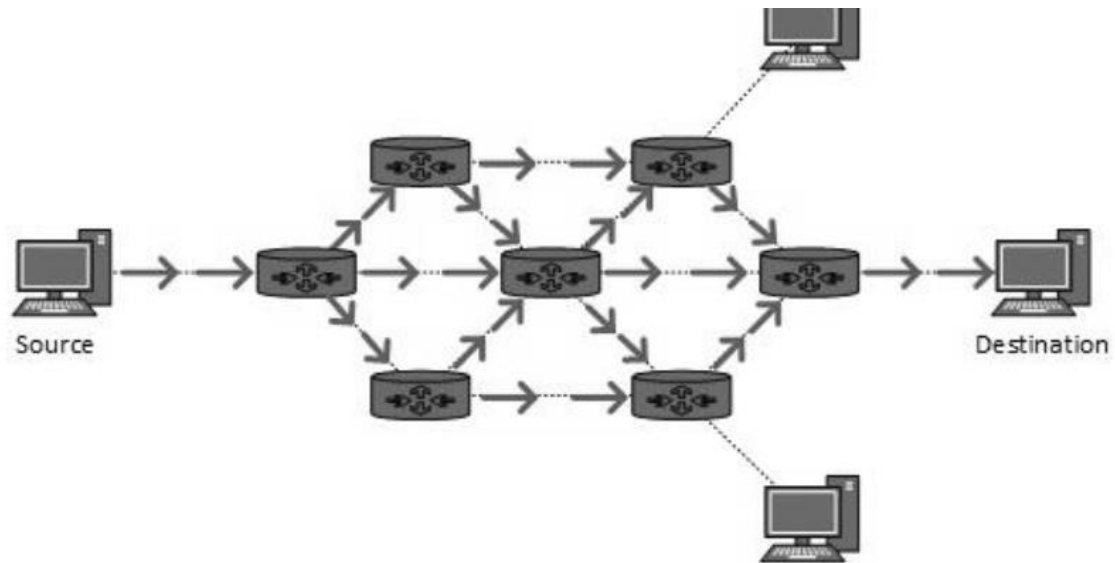
By default, the broadcast packets are not routed and forwarded by the routers on any network. Routers create broadcast domains. But it can be configured to forward broadcasts in some special cases. A broadcast message is destined to all network devices.

Broadcast routing can be done in two ways (algorithm):

A router creates a data packet and then sends it to each host one by one. In this case, the router creates multiple copies of single data packet with different destination addresses. All packets are sent as unicast but because they are sent to all, it simulates as if router is broadcasting.

- This method consumes lots of bandwidth and router must destination address of each node.
- Secondly, when router receives a packet that is to be broadcasted, it simply floods those packets out of all interfaces. All routers are configured in the same way.

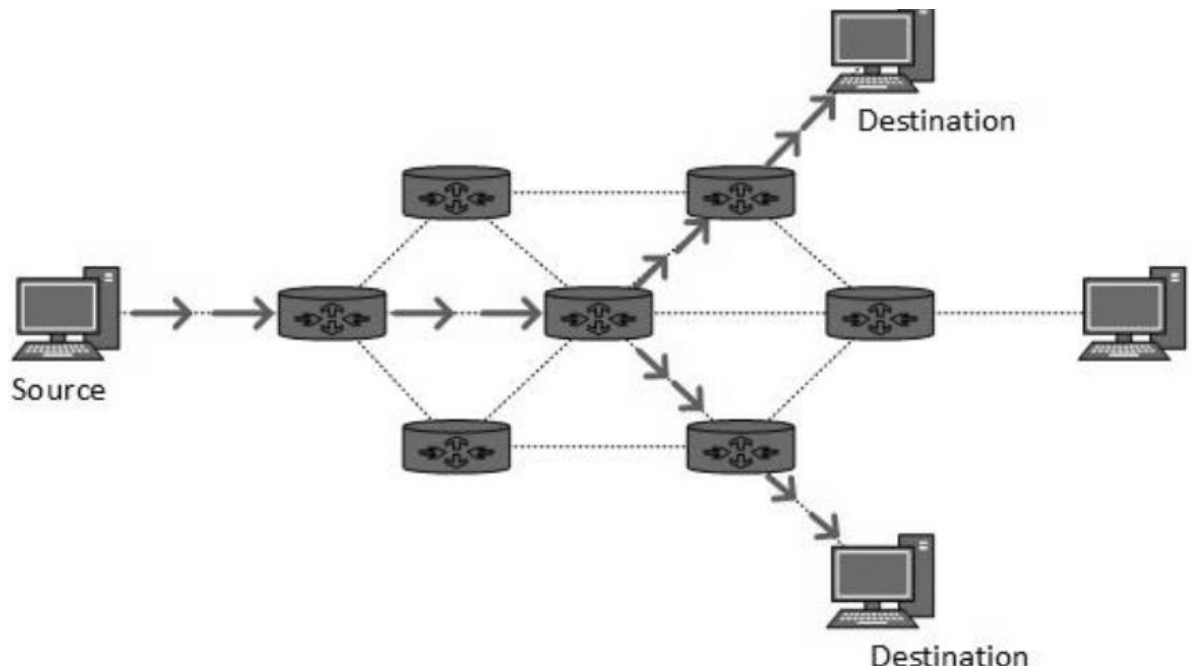




- This method is easy on router's CPU but may cause the problem of duplicate packets received from peer routers.
- Reverse path forwarding is a technique, in which router knows in advance about its predecessor from where it should receive broadcast. This technique is used to detect and discard duplicates.

### **Multicast Routing**

Multicast routing is special case of broadcast routing with significance difference and challenges. In broadcast routing, packets are sent to all nodes even if they do not want it. But in Multicast routing, the data is sent to only nodes which wants to receive the packets.



The router must know that there are nodes, which wish to receive multicast packets (or stream) then only it should forward. Multicast routing works spanning tree protocol to avoid looping. Multicast routing also uses reverse path Forwarding technique, to detect and discard duplicates and loops.

## Unicast Routing Protocols

There are two kinds of routing protocols available to route unicast packets:

### Distance Vector Routing Protocol

Distance Vector is simple routing protocol which takes routing decision on the number of hops between source and destination. A route with less number of hops is considered as the best route. Every router advertises its set best routes to other routers. Ultimately, all routers build up their network topology based on the advertisements of their peer routers, for example, Routing Information Protocol (RIP).

### Link State Routing Protocol

Link State protocol is slightly complicated protocol than Distance Vector. It takes into account the states of links of all the routers in a network. This technique helps routes build a common graph of the entire network. All routers then calculate their best path for routing purposes, for example, Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (ISIS).

## **Multicast Routing Protocols**

Unicast routing protocols use graphs while Multicast routing protocols use trees, i.e. spanning tree to avoid loops. The optimal tree is called shortest path spanning tree.

- **DVMRP:** Distance Vector Multicast Routing Protocol
- **MOSPF:** Multicast Open Shortest Path First
- **CBT:** Core Based Tree
- **PIM:** Protocol independent Multicast

Protocol Independent Multicast is commonly used now. It has two flavors:

- **PIM Dense Mode**

This mode uses source-based trees. It is used in dense environment such as LAN.

- **PIM Sparse Mode**

This mode uses shared trees. It is used in sparse environment such as WAN.

## **Routing Algorithms**

The routing algorithms are as follows:

### **Flooding**

Flooding is simplest method packet forwarding. When a packet is received, the routers send it to all the interfaces except the one on which it was received. This creates too much burden on the network and lots of duplicate packets wandering in the network.

Time to Live (TTL) can be used to avoid infinite looping of packets. There exists another approach for flooding, which is called Selective Flooding to reduce the overhead on the network. In this method, the router does not flood out on all the interfaces, but selective ones.

### **Shortest Path**

Routing decision in networks, are mostly taken on the basis of cost between source and destination. Hop count plays major role here. Shortest path is a technique which uses various algorithms to decide a path with minimum number of hops.

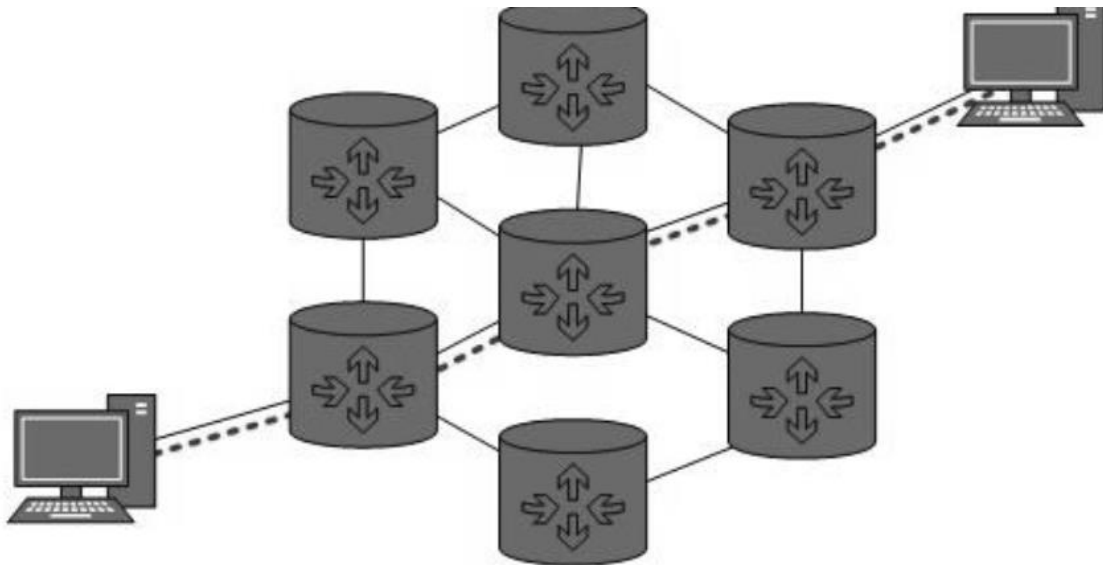
Common shortest path algorithms are:

- Dijkstra's algorithm
- Bellman Ford algorithm
- Floyd Warshall algorithm

## INTERNETWORKING

In real world scenario, networks under same administration are generally scattered geographically. There may exist requirement of connecting two different networks of same kind as well as of different kinds. Routing between two networks is called internetworking. Networks can be considered different based on various parameters such as, Protocol, topology, Layer-2 network and addressing scheme.

In internetworking, routers have knowledge of each other's address and addresses beyond them. They can be statically configured go on different network or they can learn by using internetworking routing protocol.

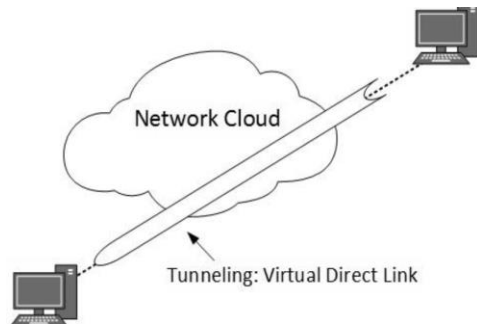


Routing protocols which are used within an organization or administration are called Interior Gateway Protocols or IGP. RIP, OSPF are examples of IGP. Routing between different organizations or administrations may have Exterior Gateway Protocol, and there is only one EGP i.e. Border Gateway Protocol.

### Tunneling

If they are two geographically separate networks, which want to communicate with each other, they may deploy a dedicated line between or they have to pass their data through intermediate networks.

Tunneling is a mechanism by which two or more same networks communicate with each other, by passing intermediate networking complexities. Tunneling is configured at both ends.



When the data enters from one end of Tunnel, it is tagged. This tagged data is then routed inside the intermediate or transit network to reach the other end of Tunnel. When data exists the Tunnel its tag is removed and delivered to the other part of the network.

## **Packet Fragmentation**

Most Ethernet segments have their maximum transmission unit (MTU) fixed to 1500 bytes. A data packet can have more or less packet length depending upon the application. Devices in the transit path also have their hardware and software capabilities which tell what amount of data that device can handle and what size of packet it can process.

If the data packet size is less than or equal to the size of packet the transit network can handle, it is processed neutrally. If the packet is larger, it is broken into smaller pieces and then forwarded. This is called packet fragmentation. Each fragment contains the same destination and source address and routed through transit path easily. At the receiving end it is assembled again.

If a packet with DF (do not fragment) bit set to 1 comes to a router which cannot handle the packet because of its length, the packet is dropped.

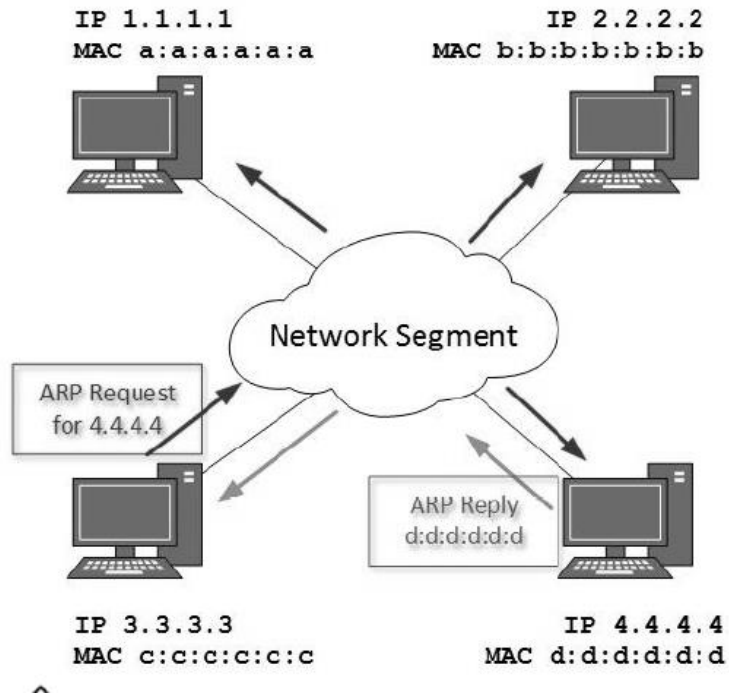
When a packet is received by a router has its MF (more fragments) bit set to 1, the router then knows that it is a fragmented packet and parts of the original packet is on the way.

If packet is fragmented too small, the overhead is increases. If the packet is fragmented too large, intermediate router may not be able to process it and it might get dropped.

## **Address Resolution Protocol (ARP)**

While communicating, a host needs Layer-2 (MAC) address of the destination machine which belongs to the same broadcast domain or network. A MAC address is physically burnt into the Network Interface Card (NIC) of a machine and it never changes.

On the other hand, IP address on the public domain is rarely changed. If the NIC is changed in case of some fault, the MAC address also changes. This way, for Layer-2 communication to take place, a mapping between the two is required.



To know the MAC address of remote host on a broadcast domain, a computer wishing to initiate communication sends out an ARP broadcast message asking, “Who has this IP address?” Because it is a broadcast, all hosts on the network segment (broadcast domain) receive this packet and process it. ARP packet contains the IP address of destination host, the sending host wishes to talk to. When a host receives an ARP packet destined to it, it replies back with its own MAC address.

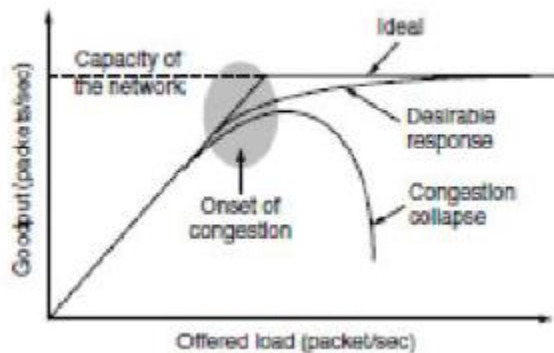
Once the host gets destination MAC address, it can communicate with remote host using Layer-2 link protocol. This MAC to IP mapping is saved into ARP cache of both sending and receiving hosts. Next time, if they require to communicate, they can directly refer to their respective ARP cache.

Reverse ARP is a mechanism where host knows the MAC address of remote host but requires to know IP address to communicate.

## CONGESTION CONTROL ALGORITHMS

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. However, the most effective way to control congestion is to reduce the load that the transport layer is placing on the network. This requires the network and transport layers to work together. In this chapter we will look at the network aspects of congestion. In Chap. 6, we

will complete the topic by covering the transport aspects of congestion. Figure depicts the onset of congestion. When the number of packets hosts send into the network is well within its carrying capacity, the number delivered is proportional to the number sent. If twice as many are sent, twice as many are delivered. However, as the offered load approaches the carrying capacity, bursts of traffic occasionally fill up the buffers inside routers and some packets are lost. These lost packets consume some of the capacity, so the number of delivered packets falls below the ideal curve. The network is now congested.



With too much traffic, performance drops sharply.

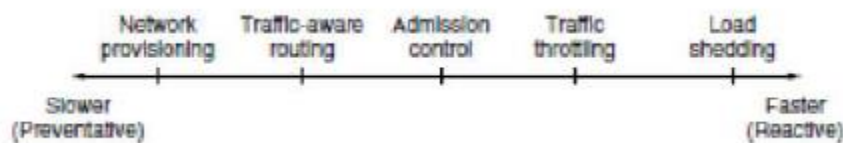
Unless the network is well designed, it may experience a **congestion collapse**, in which performance plummets as the offered load increases beyond the capacity. This can happen because packets can be sufficiently delayed inside the network that they are no longer useful when they leave the network. For example, in the early Internet, the time a packet spent waiting for a backlog of packets ahead of it to be sent over a slow 56-kbps link could reach the maximum time it was allowed to remain in the network. It then had to be thrown away. A different failure mode occurs when senders retransmit packets that are greatly delayed, thinking that they have been lost. In this case, copies of the same packet will be delivered by the network, again wasting its capacity. To capture these factors, the y-axis of Fig. is given as **good put**, which is the rate at which *useful* packets are delivered by the network. We would like to design networks that avoid congestion where possible and do not suffer from congestion collapse if they do become congested. Unfortunately, congestion cannot wholly be avoided. If all of a sudden, streams of packets begin arriving on three or four input lines and all need the same output line, a queue will build up. If there is insufficient memory to hold all of them, packets will be lost. Adding more memory may help up to a point, but Nagle (1987) realized that if routers have an infinite amount of memory, congestion gets worse, not better. This is because by the time packets get to the front of the queue, they have already timed out (repeatedly) and duplicates have been sent. This makes matters worse, not better—it leads to congestion collapse. Low-bandwidth links or routers that process packets more slowly than the line rate can also become congested. In this case, the situation can be improved by directing some of the traffic away from the bottleneck to other parts of the network. Eventually, however, all regions of the network will be congested. In this situation, there is no alternative but to shed load or build a faster network. It is worth pointing out the difference between congestion control and flow control, as the relationship is a very subtle one. Congestion control has to do with making sure the network is able to carry the offered traffic. It is a global issue, involving the behavior of all the hosts and routers. Flow control, in

contrast, relates to the traffic between a particular sender and a particular receiver. Its job is to make sure that a fast sender cannot continually transmit data faster than the receiver is able to absorb it. To see the difference between these two concepts, consider a network made up of 100-Gbps fiber optic links on which a supercomputer is trying to force feed a large file to a personal computer that is capable of handling only 1 Gbps. Although there is no congestion (the network itself is not in trouble), flow control is needed to force the supercomputer to stop frequently to give the personal computer chance to breathe. At the other extreme, consider a network with 1-Mbps lines and 1000 large computers, half of which are trying to transfer files at 100 kbps to the Other half. Here, the problem is not that of fast senders overpowering slow receivers, but that the total offered traffic exceeds what the network can handle.

The reason congestion control and flow control are often confused is that the best way to handle both problems is to get the host to slow down. Thus, a host can get a “slow down” message either because the receiver cannot handle the load or because the network cannot handle it. We will come back to this point in Chap. 6. We will start our study of congestion control by looking at the approaches that can be used at different time scales. Then we will look at approaches to preventing congestion from occurring in the first place, followed by approaches for coping with it once it has set in.

### Approaches to Congestion Control

The presence of congestion means that the load is (temporarily) greater than the resources (in a part of the network) can handle. Two solutions come to mind: increase the resources or decrease the load. As shown in Fig., these solutions are usually applied on different time scales to either prevent congestion or react to it once it has occurred.



Timescales of approaches to congestion control.

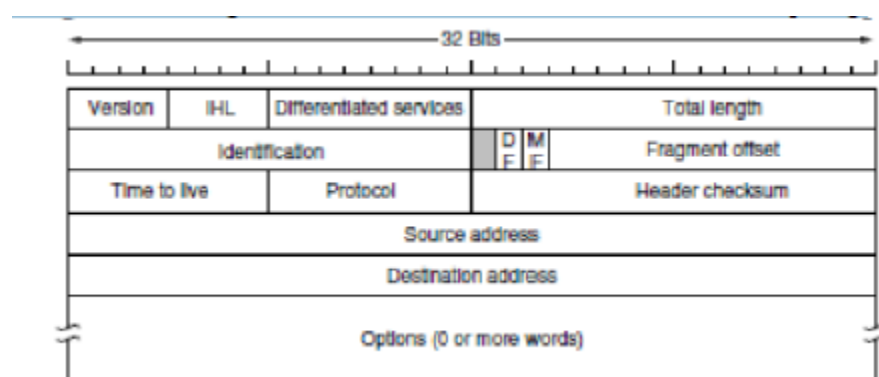
The most basic way to avoid congestion is to build a network that is well matched to the traffic that it carries. If there is a low-bandwidth link on the path along which most traffic is directed, congestion is likely. Sometimes resources on spare routers or enabling lines that are normally used only as backups (to make the system fault tolerant) or purchasing bandwidth on the open market. More often, links and routers that are regularly heavily utilized are upgraded at the earliest opportunity. This is called **provisioning** and happens on a time scale of months, driven by long-term traffic trends. To make the most of the existing network capacity, routes can be tailored to traffic patterns that change during the day as network user's wake and sleep in different time zones. For example, routes may be changed to shift traffic away from heavily used paths by changing the shortest path weights. Some local radio stations have helicopters flying around their cities to report on road congestion to make it possible for their mobile listeners to route their packets (cars) around hotspots. This is called **traffic-aware routing**. Splitting traffic across multiple paths is also helpful. However, sometimes it is not possible to increase capacity. The only way then to beat back the congestion is to decrease the load. In a virtual-circuit



network, new connections can be refused if they would cause the network to become congested. This is called **admission control**. At a finer granularity, when congestion is imminent the network can deliver feedback to the sources whose traffic flows are responsible for the problem. The network can request these sources to throttle their traffic, or it can slow down the traffic itself. Two difficulties with this approach are how to identify the onset of congestion, and how to inform the source that needs to slow down. To tackle the first issue, routers can monitor the average load, queuing delay, or packet loss. In all cases, rising numbers indicate growing congestion. To tackle the second issue, routers must participate in a feedback loop with the sources. For a scheme to work correctly, the time scale must be adjusted carefully. If every time two packets arrive in a row, a router yells STOP and every time a router is idle for 20 sec, it yells GO, the system will oscillate wildly and never converge. On the other hand, if it waits 30 minutes to make sure before saying anything, the congestion-control mechanism will react too sluggishly to be of any use. Delivering timely feedback is a nontrivial matter. An added concern is having routers send more messages when the network is already congested. Finally, when all else fails, the network is forced to discard packets that it cannot deliver. The general name for this is **load shedding**. A good policy for choosing which packets to discard can help to prevent congestion collapse.

## The IP Version 4 Protocol

An appropriate place to start our study of the network layer in the Internet is with the format of the IP datagrams themselves. An IPv4 datagram consists of a header part and a body or payload part. The header has a 20-byte fixed part and a variable-length optional part. The header format is shown in Fig. 5-46. The bits are transmitted from left to right and top to bottom, with the high-order bit of the *Version* field going first. (This is a “big-endian” network byte order. On little endian machines, such as Intel x86 computers, a software conversion is required on both transmission and reception.) In retrospect, little endian would have been a better choice, but at the time IP was designed, no one knew it would come to dominate computing.



The IPv4 (Internet Protocol) header.

The *Version* field keeps track of which version of the protocol the datagram belongs to. Version 4 dominates the Internet today, and that is where we have started our discussion. By including the version at the start of each datagram, it becomes possible to have a transition between versions over a long period of time. In fact, IPv6, the next version of IP, was defined more than a decade ago, yet is only just beginning to be deployed. We will describe it later in this section. Its use will eventually be forced when each of China's almost 231 people has a desktop PC, a laptop, and an IP phone. As an aside on numbering, IPv5 was an experimental real-time stream protocol that was never widely used.

Since the header length is not constant, a field in the header, *IHL*, is provided to tell how long the header is, in 32-bit words. The minimum value is 5, which applies when no options are present. The maximum value of this 4-bit field is 15, which limits the header to 60 bytes, and thus the *Options* field to 40 bytes. For some options, such as one that records the route a packet has taken, 40 bytes is far too small, making those options useless. The *Differentiated services* field is one of the few fields that has changed its meaning (slightly) over the years. Originally, it was called the *Type of service* field. It was and still is intended to distinguish between different classes of service.

Various combinations of reliability and speed are possible. For digitized voice, fast delivery beats accurate delivery. For file transfer, error-free transmission is more important than fast transmission. The *Type of service* field provided 3 bits to signal priority and 3 bits to signal whether a host cared more about delay, throughput, or reliability. However, no one really knew what to do with these bits at routers, so they were left unused for many years. When differentiated services were designed, IETF threw in the towel and reused this field. Now, the top 6 bits are used to mark the packet with its service class; we described the expedited and assured services earlier in this chapter. The bottom 2 bits are used to carry explicit congestion notification information, such as whether the packet has experienced congestion; we described explicit congestion notification as part of congestion control earlier in this chapter. The *Total length* includes everything in the datagram—both header and data. The maximum length is 65,535 bytes. At present, this upper limit is tolerable, but with future networks, larger datagrams may be needed. The *Identification* field is needed to allow the destination host to determine Which packet a newly arrived fragment belongs to. All the fragments of a packet contain the same *Identification* value.

Next comes an unused bit, which is surprising, as available real estate in the IP header is extremely scarce. As an April fool's joke, Bellovin (2003) proposed using this bit to detect malicious traffic. This would greatly simplify security, as packets with the "evil" bit set would be known to have been sent by attackers and could just be discarded. Unfortunately, network security is not this simple. Then come two 1-bit fields related to fragmentation. *DF* stands for Don't Fragment. It is an order to the routers not to fragment the packet. Originally, it was intended to support hosts incapable of putting the pieces back together again. Now it is used as part of the process to discover the path MTU, which is the largest packet that can travel along a path without being fragmented. By marking the datagram with the *DF* bit, the sender knows it will either arrive in one piece, or an error message will be returned to the sender. *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 *Fragment offset* tells where in the current packet this fragment belongs.

All fragments except the last one in a datagram must be a multiple of 8 bytes, the elementary fragment unit. Since 13 bits are provided, there is a maximum of 8192 fragments per datagram,

supporting a maximum packet length up to the limit of the *Total length* field. Working together, the *Identification*, *MF*, and *Fragment offset* fields are used to implement fragmentation as described in Sec. 5.5.5. The *TtL (Time to live)* field is a counter used to limit packet lifetimes. It was originally supposed to count time in seconds, allowing a maximum lifetime of 255 sec. It must be decremented on each hop and is supposed to be decremented multiple times when a packet is queued for a long time in a router. In practice, it just counts hops. When it hits zero, the packet is discarded and a warning packet is sent back to the source host. This feature prevents packets from wandering around forever, something that otherwise might happen if the routing tables ever become corrupted.

When the network layer has assembled a complete packet, it needs to know what to do with it. The *Protocol* field tells it which transport process to give the packet to. TCP is one possibility, but so are UDP and some others. The numbering of protocols is global across the entire Internet. Protocols and other assigned numbers were formerly listed in RFC 1700, but nowadays they are contained in an online database located at [www.iana.org](http://www.iana.org). Since the header carries vital information such as addresses, it rates its own checksum for protection, the *Header checksum*. The algorithm is to add up all the 16-bit half words of the header as they arrive, using one's complement arithmetic, and then take the one's complement of the result. For purposes of this algorithm, the *Header checksum* is assumed to be zero upon arrival. Such a checksum is useful for detecting errors while the packet travels through the network. Note that it must be recomputed at each hop because at least one field always changes (the *Time to live* field), but tricks can be used to speed up the computation. The *Source address* and *Destination address* indicate the IP address of the source and destination network interfaces. The *Options* field was designed to provide an escape to allow subsequent versions of the protocol to include information not present in the original design, to permit experimenters to try out new ideas, and to avoid allocating header bits to information that is rarely needed. The options are of variable length. Each begins with a 1-byte code identifying the option. Some options are followed by a 1-byte option length field, and then one or more data bytes. The *Options* field is padded out to a multiple of 4 bytes. Originally, the five options listed in Fig. were defined.

The *Security* option tells how secret the information is. In theory, a military router might use this field to specify not to route packets through certain countries the military considers to be "bad guys." In practice, all routers ignore it, so its only practical function is to help spies find the good stuff more easily. The *Strict source routing* option gives the complete path from source to destination as a sequence of IP addresses. The datagram is required to follow that

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

## **IP Version 6**

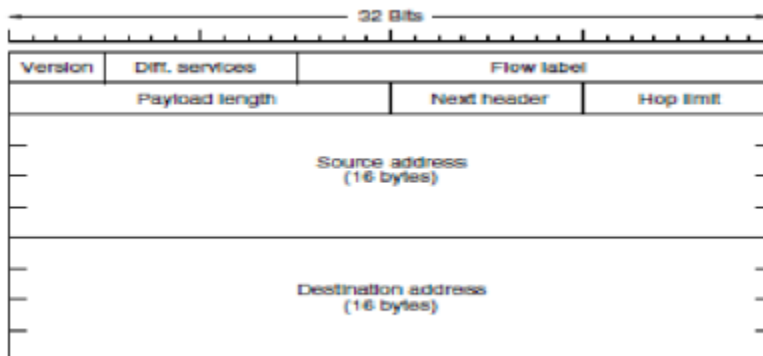
IP has been in heavy use for decades. It has worked extremely well, as demonstrated by the exponential growth of the Internet. Unfortunately, IP has become a victim of its own popularity: it is close to running out of addresses. Even with CIDR and NAT using addresses more sparingly, the last IPv4 addresses are expected to be assigned by ICANN before the end of 2012. This looming disaster was recognized almost two decades ago, and it sparked a great deal of discussion and controversy within the Internet community about what to do about it. In this section, we will describe both the problem and several proposed solutions. The only long-term solution is to move to larger addresses. **IPv6 (IP version 6)** is a replacement design that does just that. It uses 128-bit addresses; a shortage of these addresses is not likely any time in the foreseeable future. However, IPv6 has proved very difficult to deploy. It is a different network layer protocol that does not really interwork with IPv4, despite many similarities. Also, companies and users are not really sure why they should want IPv6 in any case. The result is that IPv6 is deployed and used on only a tiny fraction of the Internet (estimates are 1%) despite having been an Internet Standard since 1998. The next several years will be an interesting time, as the few remaining IPv4 addresses are allocated. Will people start to auction off their IPv4 addresses on eBay? Will a black market in them spring up? Who knows? In addition to the address problems, other issues loom in the background. In its early years, the Internet was largely used by universities, high-tech industries, and the U.S. Government (especially the Dept. of Defense). With the explosion of interest in the Internet starting in the mid-1990s, it began to be used by a different group of people, often with different requirements. For one thing, numerous people with smart phones use it to keep in contact with their home bases. For another, with the impending convergence of the computer, communication, and entertainment industries, it may not be that long before every telephone and television set in the world is an Internet node, resulting in a billion machines being used for audio and video on demand. Under these circumstances, it became apparent that IP had to evolve and become more flexible. Seeing these problems on the horizon, in 1990 IETF started work on a new version of IP, one that would never run out of addresses, would solve a variety of other problems, and be more flexible and efficient as well. Its major goals were:

1. Support billions of hosts, even with inefficient address allocation.
2. Reduce the size of the routing tables.
3. Simplify the protocol, to allow routers to process packets faster.
4. Provide better security (authentication and privacy).
5. Pay more attention to the type of service, particularly for real-time data.
6. Aid multicasting by allowing scopes to be specified.
7. Make it possible for a host to roam without changing its address.
8. Allow the protocol to evolve in the future.
9. Permit the old and new protocols to coexist for years.

### **The Main IPv6 Header**

The *Version* field is always 6 for IPv6 (and 4 for IPv4). During the transition period from IPv4, which has already taken more than a decade, routers will be able to examine this field to tell what kind of packet they have. As an aside, making this test wastes a few instructions in the critical path, given that the data link header usually indicates the network protocol for de-

multiplexing, so some routers may skip the check. For example, the Ethernet *Type* field has different values to indicate an IPv4 or an IPv6 payload. The discussions between the “Do it right” and “Make it fast” camps will no doubt be lengthy and vigorous.



### The IPv6 fixed header (required).

The *Differentiated services* field (originally called *Traffic class*) is used to distinguish the class of service for packets with different real-time delivery requirements. It is used with the Differentiated service architecture for quality of service in the same manner as the field of the same name in the IPv4 packet. Also, the low-order 2 bits are used to signal explicit congestion indications, again in the same way as with IPv4.

The *Flow label* field provides a way for a source and destination to mark groups of packets that have the same requirements and should be treated in the same way by the network, forming a pseudo connection. For example, a stream of packets from one process on a certain source host to a process on a specific destination host might have stringent delay requirements and thus need reserved bandwidth. The flow can be set up in advance and given an identifier. When a packet with a nonzero *Flow label* shows up, all the routers can look it up in internal tables to see what kind of special treatment it requires. In effect, flows are an attempt to have it both ways: the flexibility of a datagram network and the guarantees of a virtual-circuit network. Each flow for quality of service purposes is designated by the source address, destination address, and flow number. This design means that up to 220 flows may be active at the same time between a given pair of IP addresses. It also means that even if two flows coming from different hosts but with the same flow label pass through the same router, the router will be able to tell them apart using the source and destination addresses. It is expected that flow labels will be chosen randomly, rather than assigned sequentially starting at 1, so routers are expected to hash them. The *Payload length* field tells how many bytes follow the 40-byte header of Fig. The name was changed from the IPv4 *Total length* field because the meaning was changed slightly: the 40 header bytes are no longer counted as part of the length (as they used to be). This change means the payload can now be 65,535 bytes instead of a mere 65,515 bytes.

The *Next header* field lets the cat out of the bag. The reason the header could be simplified is that there can be additional (optional) extension headers. This field tells which of the (currently) six extension headers, if any, follow this one.

If this header is the last IP header, the *Next header* field tells which transport protocol handler (e.g., TCP, UDP) to pass the packet to. The *Hop limit* field is used to keep packets from living forever. It is, in practice, the same as the *Time to live* field in IPv4, namely, a field that is

decremented on each hop. In theory, in IPv4 it was a time in seconds, but no router used it that way, so the name was changed to reflect the way it is actually used. Next come the *Source address* and *Destination address* fields. Deering's original proposal, SIP, used 8-byte addresses, but during the review process many people felt that with 8-byte addresses IPv6 would run out of addresses within a few decades, whereas with 16-byte addresses it would never run out. Other people argued that 16 bytes was overkill, whereas still others favored using 20-byte addresses to be compatible with the OSI datagram protocol. Still another faction wanted variable-sized addresses. After much debate and more than a few words unprintable in an academic textbook, it was decided that fixed-length 16-byte addresses were the best compromise.

A new notation has been devised for writing 16-byte addresses. They are written as eight groups of four hexadecimal digits with colons between the groups, like this:

8000:0000:0000:0000:0123:4567:89AB:CDEF Since many addresses will have many zeros inside them, three optimizations have been authorized. First, leading zeros within a group can be omitted, so 0123 can be written as 123. Second, one or more groups of 16 zero bits can be replaced by a pair of colons. Thus, the above address now becomes 8000::123:4567:89AB:CDEF Finally, IPv4 addresses can be written as a pair of colons and an old dotted decimal number, for example: ::192.31.20.46 Perhaps it is unnecessary to be so explicit about it, but there are a lot of 16- addresses per square meter. Students of chemistry will notice that this number is larger than Avogadro's number. While it was not the intention to give every molecule on the surface of the earth its own IP address, we are not that far off. In practice, the address space will not be used efficiently, just as the telephone number address space is not (the area code for Manhattan, 212, is nearly full, but that for Wyoming, 307, is nearly empty). In RFC 3194, Durand and Huitema calculated that, using the allocation of telephone numbers as a guide, even in the most pessimistic scenario there will still be well over 1000 IP addresses per square meter of the entire earth's surface (land and water). In any likely scenario, there will be trillions of them per square meter. In short, it seems unlikely that we will run out in the foreseeable future. It is instructive to compare the IPv4 header (Fig.) with the IPv6 header (Fig) to see what has been left out in IPv6. The *IHL* field is gone because the IPv6 header has a fixed length. The *Protocol* field was taken out because the *Next header* field tells what follows the last IP header (e.g., a UDP or TCP segment). All the fields relating to fragmentation were removed because IPv6 takes a different approach to fragmentation. To start with, all IPv6-conformant hosts are expected to dynamically determine the packet size to use. They do this using the path MTU discovery procedure we described in Sec. 5.5.5. In brief, when a host sends an IPv6 packet that is too large, instead of fragmenting it, the router that is unable to forward it drops the packet and sends an error message back to the sending host. This message tells the host to break up all future packets to that destination. Having the host send packets that are the right size in the first place is ultimately much more efficient than having the routers fragment them on the fly. Also, the minimum-size packet that routers must be able to forward has been raised from 576 to 1280 bytes to allow 1024 bytes of data and many headers. Finally, the *Checksum* field is gone because calculating it greatly reduces performance. With the reliable networks now used, combined with the fact that the data link layer and transport layers normally have their own checksums, the value of yet another checksum was deemed not worth the performance price it extracted. Removing all these features has resulted in a lean and mean network layer protocol. Thus, the goal of IPv6—a fast, yet flexible, protocol with plenty of address space—is met by this design.

## Unit-V

### **Transport Layer:**

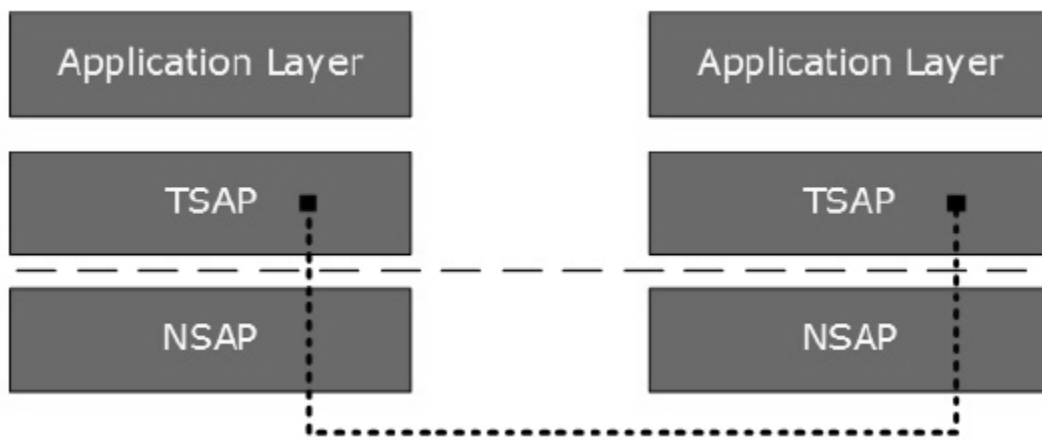
Transport layer offers peer-to-peer and end-to-end connection between two processes on remote hosts. Transport layer takes data from upper layer (i.e. Application layer) and then breaks it into smaller size segments, numbers each byte, and hands over to lower layer (Network Layer) for delivery.

### **Functions**

- This Layer is the first one which breaks the information data, supplied by Application layer in to smaller units called segments. It numbers every byte in the segment and maintains their accounting.
- This layer ensures that data must be received in the same sequence in which it was sent.
- This layer provides end-to-end delivery of data between hosts which may or may not belong to the same subnet.
- All server processes intend to communicate over the network are equipped with well-known Transport Service Access Points (TSAPs) also known as port numbers.

### **End to End Communication**

A process on one host identifies its peer host on remote network by means of TSAPs, also known as Port numbers. TSAPs are very well defined and a process which is trying to communicate with its peer knows this in advance.



For example, when a DHCP client wants to communicate with remote DHCP server, it always requests on port number 67. When a DNS client wants to communicate with remote DNS server, it always requests on port number 53 (UDP).

The two main Transport layer protocols are:

1. **Transmission Control Protocol**

It provides reliable communication between two hosts.

2. **User Datagram Protocol**

It provides unreliable communication between two hosts.

## **TRANSMISSION CONTROL PROTOCOL**

The transmission Control Protocol (TCP) is one of the most important protocols of Internet Protocols suite. It is most widely used protocol for data transmission in communication network such as internet.

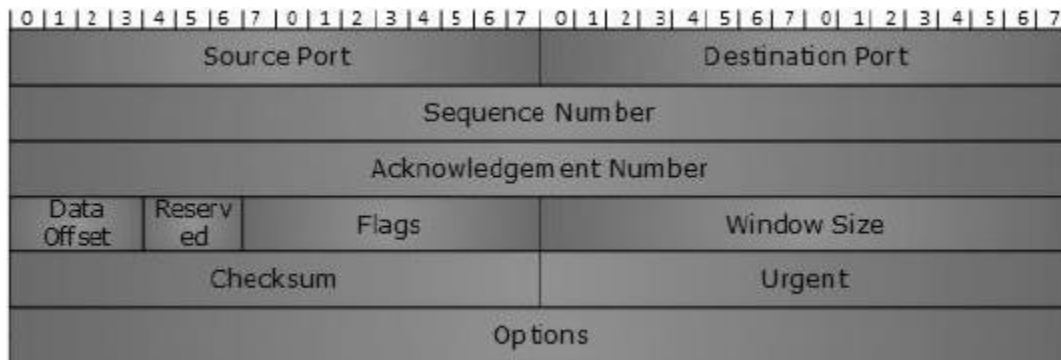
### **Features**

- TCP is reliable protocol. That is, the receiver always sends either positive or negative acknowledgement about the data packet to the sender, so that the sender always has bright clue about whether the data packet is reached the destination or it needs to resend it.
- TCP ensures that the data reaches intended destination in the same order it was sent.
- TCP is connection oriented. TCP requires that connection between two remote points be established before sending actual data.
- TCP provides error-checking and recovery mechanism.
- TCP provides end-to-end communication.
- TCP provides flow control and quality of service.
- TCP operates in Client/Server point-to-point mode.
- TCP provides full duplex server, i.e. it can perform roles of both receiver and sender.



## Header

The length of TCP header is minimum 20 bytes and maximum 60 bytes.



- **Source Port (16-bits):** It identifies source port of the application process on the sending device.
- **Destination Port (16-bits):** It identifies destination port of the application process on the receiving device.
- **Sequence Number (32-bits):** Sequence number of data bytes of a segment in a session.
- **Acknowledgement Number (32-bits):** When ACK flag is set, this number contains the next sequence number of the data byte expected and works as acknowledgement of the previous data received.
- **Data offset (4-bits):** This field implies both, the size of TCP header (32-bit words) and the offset of data in current packet in the whole TCP segment.
- **Reserved (3-bits):** Reserved for future use and all are set zero by default.
- **Flags (1-bit each):**

**NS:** Nonce Sum bit is used by Explicit Congestion Notification signaling process.

**CWR:** When a host receives packet with ECE bit set, it sets Congestion Windows Reduced to acknowledge that ECE received.

**ECE:** It has two meanings:

- If SYN bit is clear to 0, then ECE means that the IP packet has its CE (congestion experience) bit set.
- If SYN bit is set to 1, ECE means that the device is ECT capable.

**URG:** It indicates that Urgent Pointer field has significant data and should be processed.

**ACK:** It indicates that Acknowledgement field has significance. If ACK is cleared to 0, it indicates that packet does not contain any acknowledgement.

**PSH:** When set, it is a request to the receiving station to PUSH data as soon as it comes to the receiving application without buffering it.

**RST:** Reset flag has the following features:

- It is used to refuse an incoming connection.
- It is used to reject a segment.
- It is used to restart a connection.

**SYN:** This flag is used to set up a connection between hosts.

**FIN:** This flag is used to release a connection and no more data is exchanged thereafter. Because packets with SYN and FIN flags have sequence numbers, they are processed in correct order.

**Windows Size:** This field is used for flow control between two stations and indicates the amount of buffer (in bytes) the receiver has allocated for a segment, i.e. how much data is the receiver expecting.

**Checksum:** This field contains the checksum of Header, Data, and Pseudo Headers.

**Urgent Pointer:** It points to the urgent data byte if URG flag is set to 1.

**Options:** It facilitates additional options which are not covered by the regular header. Option field is always described in 32-bit words. If this field contains data less than 32-bit, padding is used to cover the remaining bits to reach 32-bit boundary.

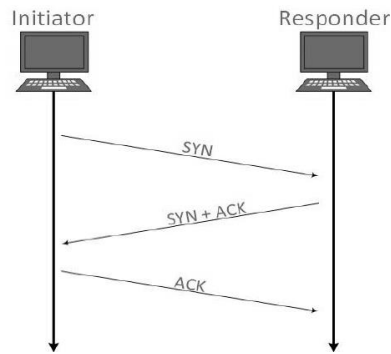
## Addressing

TCP communication between two remote hosts is done by means of port numbers (TSAPs). Ports numbers can range from 0 – 65535 which are divided as:

- System Ports (0 – 1023)
- User Ports ( 1024 – 49151)
- Private/Dynamic Ports (49152 – 65535)

## Connection Management

TCP communication works in Server/Client model. The client initiates the connection and the server either accepts or rejects it. Three-way handshaking is used for connection management



### **Establishment**

Client initiates the connection and sends the segment with a Sequence number. Server acknowledges it back with its own Sequence number and ACK of client's segment which is one more than client's Sequence number. Client after receiving ACK of its segment sends an acknowledgement of Server's response.

### **Release**

Either of server and client can send TCP segment with FIN flag set to 1. When the receiving end responds it back by Acknowledging FIN, that direction of TCP communication is closed and connection is released.

### **Bandwidth Management**

TCP uses the concept of window size to accommodate the need of Bandwidth management. Window size tells the sender at the remote end the number of data byte segments the receiver at this end can receive. TCP uses slow start phase by using window size 1 and increases the window size exponentially after each successful communication.

For example, the client uses windows size 2 and sends 2 bytes of data. When acknowledgement of this segment received the windows size is doubled to 4 and next the segment sent will be 4 data bytes long. When the acknowledgement of 4-byte data segment is received, the client sets windows size to 8 and so on.

If an acknowledgement is missed, i.e. data lost in transit network or it received NACK, then the window size is reduced to half and slow start phase starts again.

### **Error Control and Flow Control**

TCP uses port numbers to know what application process it needs to handover the data segment. Along with that, it uses sequence numbers to synchronize itself with the remote host. All data segments are sent and received with sequence numbers. The Sender knows which last data segment was received by the Receiver when it gets ACK. The Receiver knows about the last segment sent by the Sender by referring to the sequence number of recently received packet. If the sequence number of a segment recently received does not match with the sequence number the receiver was expecting, then it is discarded and NACK is sent back. If two segments arrive with the same sequence number, the TCP timestamp value is compared to make a decision.

## **Multiplexing**

The technique to combine two or more data streams in one session is called Multiplexing. When a TCP client initializes a connection with Server, it always refers to a well-defined port number which indicates the application process. The client itself uses a randomly generated port number from private port number pools.

Using TCP Multiplexing, a client can communicate with a number of different application process in a single session. For example, a client requests a web page which in turn contains different types of data (HTTP, SMTP, FTP etc.) the TCP session timeout is increased and the session is kept open for longer time so that the three-way handshake overhead can be avoided. This enables the client system to receive multiple connection over single virtual connection. These virtual connections are not good for Servers if the timeout is too long.

## **Congestion Control**

When large amount of data is fed to system which is not capable of handling it, congestion occurs. TCP controls congestion by means of Window mechanism. TCP sets a window size telling the other end how much data segment to send. TCP may use three algorithms for congestion control:

- Additive increase, Multiplicative Decrease
- Slow Start
- Timeout React

## **Timer Management**

TCP uses different types of timers to control and management various tasks:

### **Keep-alive timer:**

- This timer is used to check the integrity and validity of a connection.
- When keep-alive time expires, the host sends a probe to check if the connection still exists.

### **Retransmission timer:**

- This timer maintains stateful session of data sent.
- If the acknowledgement of sent data does not receive within the Retransmission time, the data segment is sent again.

### **Persist timer:**

- TCP session can be paused by either host by sending Window Size 0.
- To resume the session a host needs to send Window Size with some larger value.

- If this segment never reaches the other end, both ends may wait for each other for infinite time.
- When the Persist timer expires, the host resends its window size to let the other end know.
- Persist Timer helps avoid deadlocks in communication.

### **Timed-Wait:**

- After releasing a connection, either of the hosts waits for a Timed-Wait time to terminate the connection completely.
- This is in order to make sure that the other end has received the acknowledgement of its connection termination request.
- Timed-out can be a maximum of 240 seconds (4 minutes).

### **Crash Recovery**

TCP is very reliable protocol. It provides sequence number to each of byte sent in segment. It provides the feedback mechanism i.e. when a host receives a packet, it is bound to ACK that packet having the next sequence number expected (if it is not the last segment).

When a TCP Server crashes mid-way communication and re-starts its process, it sends TPDU broadcast to all its hosts. The hosts can then send the last data segment which was never unacknowledged and carry onwards.

## **USER DATAGRAM PROTOCOL**

The User Datagram Protocol (UDP) is simplest Transport Layer communication protocol available of the TCP/IP protocol suite. It involves minimum amount of communication mechanism. UDP is said to be an unreliable transport protocol but it uses IP services which provides best effort delivery mechanism.

In UDP, the receiver does not generate an acknowledgement of packet received and in turn, the sender does not wait for any acknowledgement of packet sent. This shortcoming makes this protocol unreliable as well as easier on processing.

### **Requirement of UDP**

A question may arise, why do we need an unreliable protocol to transport the data? We deploy UDP where the acknowledgement packets share significant amount of bandwidth along with the actual data. For example, in case of video streaming, thousands of packets are forwarded towards its users. Acknowledging all the packets is troublesome and may contain huge amount of bandwidth wastage. The best delivery mechanism of underlying IP protocol ensures best efforts

to deliver its packets, but even if some packets in video streaming get lost, the impact is not calamitous and can be ignored easily. Loss of few packets in video and voice traffic sometimes goes unnoticed.

## Features

- UDP is used when acknowledgement of data does not hold any significance.
- UDP is good protocol for data flowing in one direction.
- UDP is simple and suitable for query based communications.
- UDP is not connection oriented.
- UDP does not provide congestion control mechanism.
- UDP does not guarantee ordered delivery of data.
- UDP is stateless.
- UDP is suitable protocol for streaming applications such as VoIP, multimedia streaming

## UDP Header

UDP header is as simple as its function.



UDP header contains four main parameters:

- 1. Source Port:** This 16 bits information is used to identify the source port of the packet.
- 2. Destination Port:** This 16 bits information is used identify application level service on destination machine.
- 3. Length:** Length field specifies the entire length of UDP packet (including header). It is 16-bits field and minimum value is 8-byte, i.e. the size of UDP header itself.
- 4. Checksum:** This field stores the checksum value generated by the sender before sending. IPv4 has this field as optional so when checksum field does not contain any value, it is made 0 and all its bits are set to zero.

## UDP application

Here are few applications where UDP is used to transmit data:

- Domain Name Services
- Simple Network Management Protocol
- Trivial File Transfer Protocol
- Routing Information Protocol
- Kerberos

## **Congestion Control**

What is **congestion**?

A state occurring in network layer when the message traffic is so heavy that it slows down network response time.

### **Effects of Congestion**

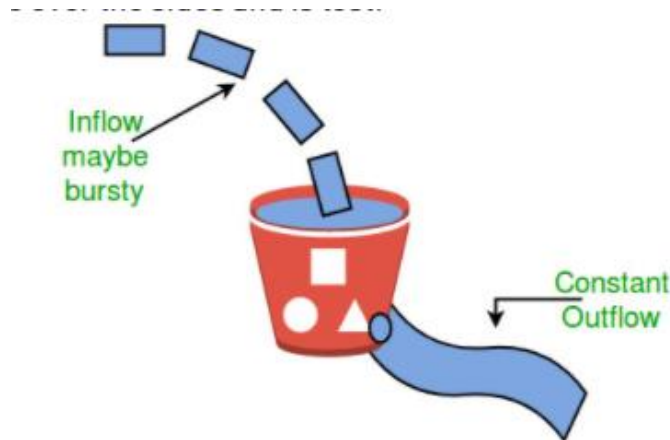
- As delay increases, performance decreases.
- If delay increases, retransmission occurs, making situation worse.

### **Congestion control algorithms**

#### **Leaky Bucket Algorithm**

Let us consider an example to understand

Imagine a bucket with a small hole in the bottom. No matter at what rate water enters the bucket, the outflow is at constant rate. When the bucket is full with water additional water entering spills over the sides and is lost.



Similarly, each network interface contains a leaky bucket and the following **steps** are involved in leaky bucket algorithm:

1. When host wants to send packet, packet is thrown into the bucket.
2. The bucket leaks at a constant rate, meaning the network interface transmits packets at a constant rate.
3. Bursty traffic is converted to a uniform traffic by the leaky bucket.
4. In practice the bucket is a finite queue that outputs at a finite rate.

### Token bucket Algorithm

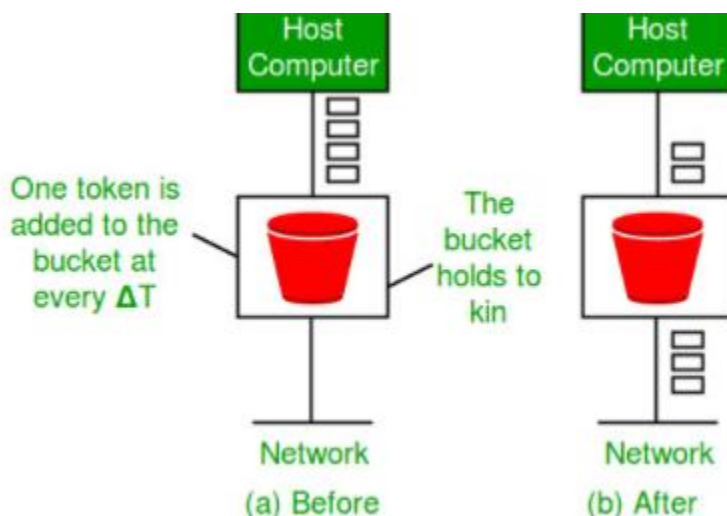
**Need** of token bucket Algorithm:-

The leaky bucket algorithm enforces output pattern at the average rate, no matter how bursty the traffic is. So in order to deal with the bursty traffic we need a flexible algorithm so that the data is not lost. One such algorithm is token bucket algorithm.

**Steps** of this algorithm can be described as follows:

1. In regular intervals tokens are thrown into the bucket.  $f$
2. The bucket has a maximum capacity.  $f$
3. If there is a ready packet, a token is removed from the bucket, and the packet is sent.
4. If there is no token in the bucket, the packet cannot be sent.

Let's understand with an example



In figure (A) we see a bucket holding three tokens, with five packets waiting to be transmitted. For a packet to be transmitted, it must capture and destroy one token. In figure (B) We see that three of the five packets have gotten through, but the other two are stuck waiting for more tokens to be generated.



### **Ways in which token bucket is superior to leaky bucket:**

The leaky bucket algorithm controls the rate at which the packets are introduced in the network, but it is very conservative in nature. Some flexibility is introduced in the token bucket algorithm. In the token bucket, algorithm tokens are generated at each tick (up to a certain limit). For an incoming packet to be transmitted, it must capture a token and the transmission takes place at the same rate. Hence some of the busty packets are transmitted at the same rate if tokens are available and thus introduces some amount of flexibility in the system.

**Formula:**  $M * s = C + \rho * s$

where S – is time taken

M – Maximum output rate

$\rho$  – Token arrival rate

C – Capacity of the token bucket in byte

## **Unit-VI**

### **Application Layer:**

Application layer is the topmost layer in OSI and TCP/IP layered model. This layer exists in both layered Models because of its significance, of interacting with user and user applications. This layer is for applications which are involved in communication system.

A user may or may not directly interacts with the applications. Application layer is where the actual communication is initiated and reflects. Because this layer is on the top of the layer stack, it does not serve any other layers. Application layer takes the help of Transport and all layers below it to communicate or transfer its data to the remote host.

When an application layer protocol wants to communicate with its peer application layer protocol on remote host, it hands over the data or information to the Transport layer. The transport layer does the rest with the help of all the layers below it.

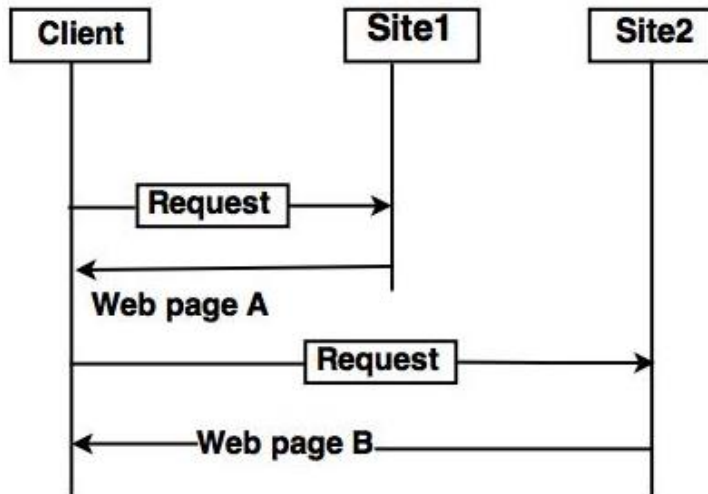
There is an ambiguity in understanding Application Layer and its protocol. Not every user application can be put into Application Layer except those applications which interact with the communication system. For example, designing software or text editor cannot be considered as application layer programs.

On the other hand, when we use a Web Browser, which is actually using Hyper Text Transfer Protocol (HTTP) to interact with the network, HTTP is Application Layer protocol.

Another example is File Transfer Protocol, which helps a user to transfer text based or binary files across the network. A user can use this protocol in either GUI based software like FileZilla or Cute FTP and the same user can use FTP in Command Line mode. Hence, irrespective of which software you use, it is the protocol which is considered at Application Layer used by that software. DNS is a protocol which helps user application protocols such as HTTP to accomplish its work.

### **World Wide Web (WWW)**

- The World Wide Web (WWW) is a collection of documents and other web resources which are identified by URLs, interlinked by hypertext links, and can be accessed and searched by browsers via the Internet.
- World Wide Web is also called the Web and it was invented by Tim Berners-Lee in 1989.
- Website is a collection of web pages belonging to a particular organization.
- The pages can be retrieved and viewed by using browser.



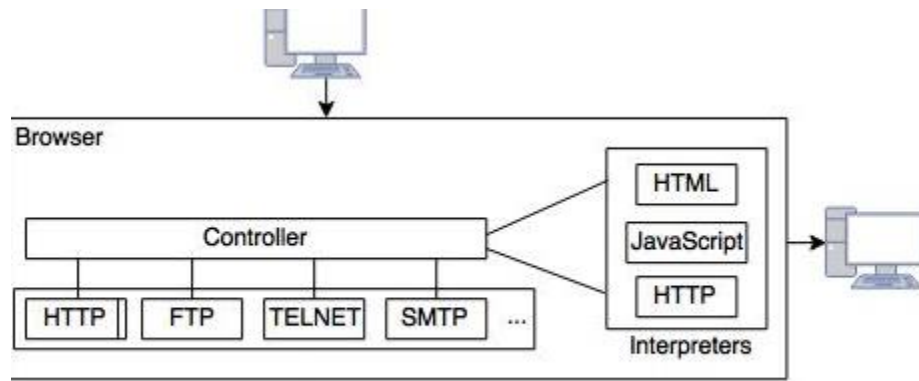
**Architecture of WWW**

**Let us go through the scenario shown in above fig.**

- The client wants to see some information that belongs to site 1.
- It sends a request through its browser to the server at site 2.
- The server at site 1 finds the document and sends it to the client.

#### **Client (Browser):**

- Web browser is a program, which is used to communicate with web server on the Internet.
- Each browser consists of three parts: a controller, client protocol and interpreter.
- The controller receives input from input device and use the programs to access the documents.
- After accessing the document, the controller uses one of the interpreters to display the document on the screen



**Fig: Client (Browser)**

### **Server:**

- A computer which is available for the network resources and provides service to the other computer on request is known as server.
- The web pages are stored at the server.
- Server accepts a TCP connection from a client browser.
- It gets the name of the file required.
- Server gets the stored file. Returns the file to the client and releases the top connection.

## **Domain Name System**

The Domain Name System (DNS) works on Client Server model. It uses UDP protocol for transport layer communication. DNS uses hierarchical domain based naming scheme.

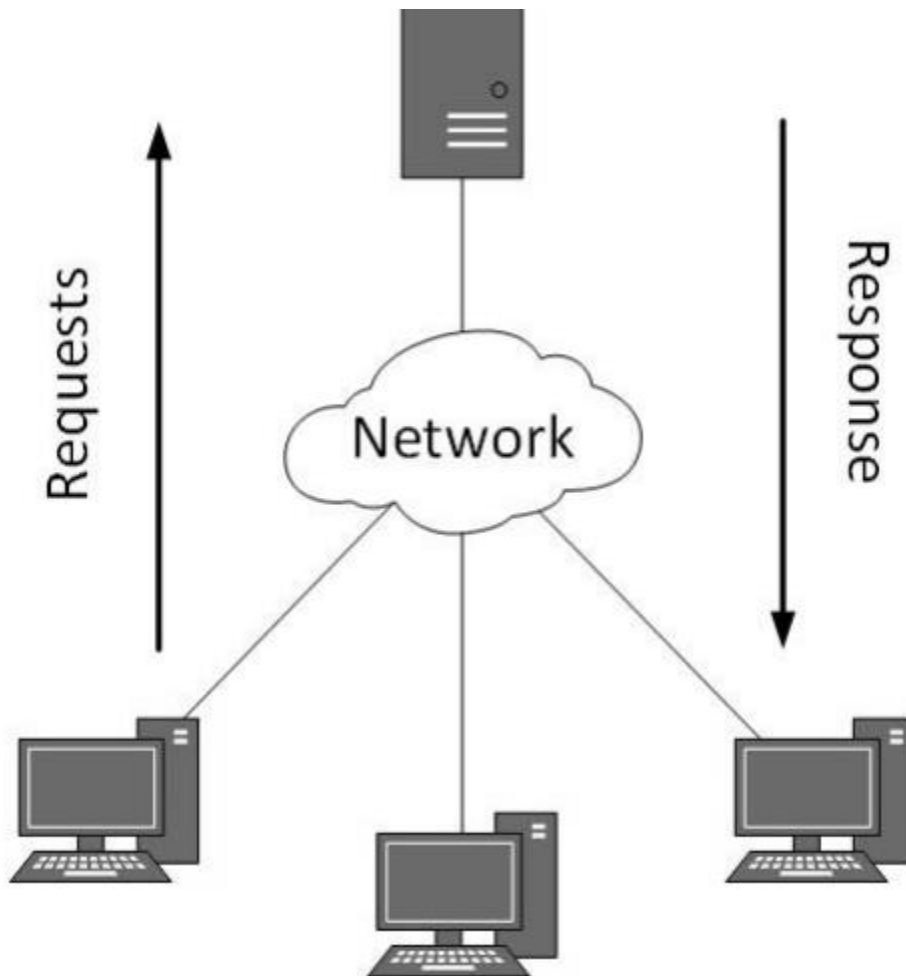
The DNS server is configured with Fully Qualified Domain Names (FQDN) and email addresses mapped with their respective Internet Protocol addresses. A DNS server is requested with FQDN and it responds back with the IP address mapped with it. DNS uses UDP port 53.

## **CLIENT-SERVER MODEL**

Two remote application processes can communicate mainly in two different fashions:

- **Peer-to-peer:** Both remote processes are executing at same level and they exchange data using some shared resource.
- **Client-Server:** One remote process acts as a Client and requests some resource from another application process acting as Server.

In client-server model, any process can act as Server or Client. It is not the type of machine, size of the machine, or its computing power which makes it server; it is the ability of serving request that makes a machine a server.



A system can act as Server and Client simultaneously. That is, one process is acting as Server and another is acting as a client. This may also happen that both client and server processes reside on the same machine.

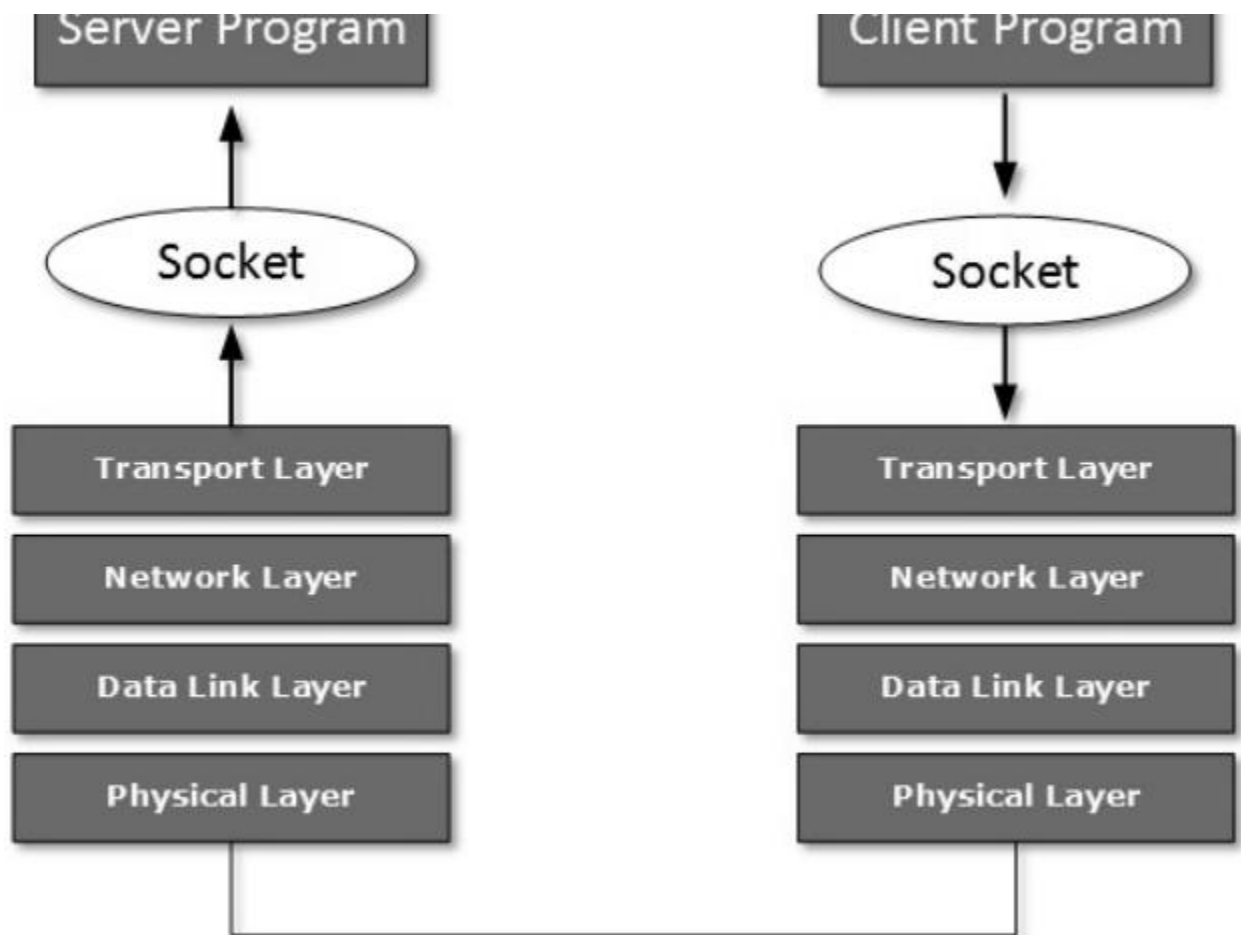
## Communication

Two processes in client-server model can interact in various ways:

- Sockets
- Remote Procedure Calls (RPC)

### Sockets

In this paradigm, the process acting as Server opens a socket using a well-known (or known by client) port and waits until some client request comes. The second process acting as a Client also opens a socket; but instead of waiting for an incoming request, the client processes 'requests first'.



When the request is reached to server, it is served. It can either be an information sharing or resource request.

### Remote Procedure Call

This is a mechanism where one process interacts with another by means of procedure calls. One process (client) calls the procedure lying on remote host. The process on remote host is said to be Server. Both processes are allocated stubs. This communication happens in the following way:

- The client process calls the client stub. It passes all the parameters pertaining to program local to it.
- All parameters are then packed (marshalled) and a system call is made to send them to other side of the network.
- Kernel sends the data over the network and the other end receives it.
- The remote host passes data to the server stub where it is unmarshalled.
- The parameters are passed to the procedure and the procedure is then executed.
- The result is sent back to the client in the same manner.

## **Simple Mail Transfer Protocol**

The Simple Mail Transfer Protocol (SMTP) is used to transfer electronic mail from one user to another. This task is done by means of email client software (User Agents) the user is using. User Agents help the user to type and format the email and store it until internet is available. When an email is submitted to send, the sending process is handled by Message Transfer Agent which is normally comes inbuilt in email client software.

Message Transfer Agent uses SMTP to forward the email to another Message Transfer Agent (Server side). While SMTP is used by end user to only send the emails, the Servers normally use SMTP to send as well as receive emails. SMTP uses TCP port number 25 and 587.

Client software uses Internet Message Access Protocol (IMAP) or POP protocols to receive emails.

## **File Transfer Protocol**

The File Transfer Protocol (FTP) is the most widely used protocol for file transfer over the network. FTP uses TCP/IP for communication and it works on TCP port 21. FTP works on Client/Server Model where a client requests file from Server and server sends requested resource back to the client.

FTP uses out-of-band controlling i.e. FTP uses TCP port 20 for exchanging controlling information and the actual data is sent over TCP port 21.

The client requests the server for a file. When the server receives a request for a file, it opens a TCP connection for the client and transfers the file. After the transfer is complete, the server

closes the connection. For a second file, client requests again and the server reopens a new TCP connection.

## **Hyper Text Transfer Protocol (HTTP)**

The Hyper Text Transfer Protocol (HTTP) is the foundation of World Wide Web. Hypertext is well organized documentation system which uses hyperlinks to link the pages in the text documents. HTTP works on client server model. When a user wants to access any HTTP page on the internet, the client machine at user end initiates a TCP connection to server on port 80. When the server accepts the client request, the client is authorized to access web pages.

To access the web pages, a client normally uses web browsers, who are responsible for initiating, maintaining, and closing TCP connections. HTTP is a stateless protocol, which means the Server maintains no information about earlier requests by clients.

HTTP versions:

- HTTP 1.0 uses non persistent HTTP. At most one object can be sent over a single TCP connection.
- HTTP 1.1 uses persistent HTTP. In this version, multiple objects can be sent over a single TCP connection.

## **Introduction to Network Security**

### **SECURITY MECHANISMS**

One of the most specific security mechanisms in use is cryptographic techniques. Encryption or encryption-like transformations of information are the most common means of providing security. Some of the mechanisms are

- Encipherment
- Digital Signature
- Access Control

### **SECURITY ATTACKS**

There are four general categories of attack which are listed below.

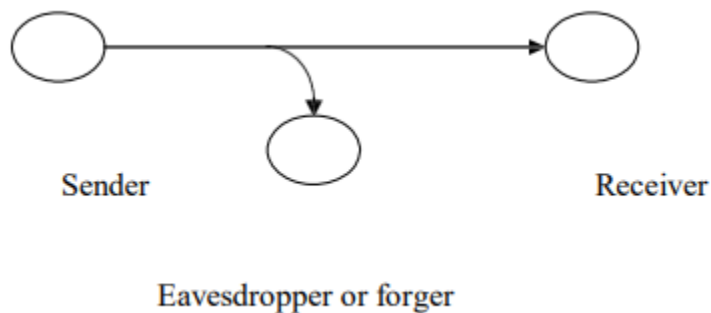
#### **Interruption**

An asset of the system is destroyed or becomes unavailable or unusable. This is an attack on availability e.g., destruction of piece of hardware, cutting of a communication line or Disabling of file management system.



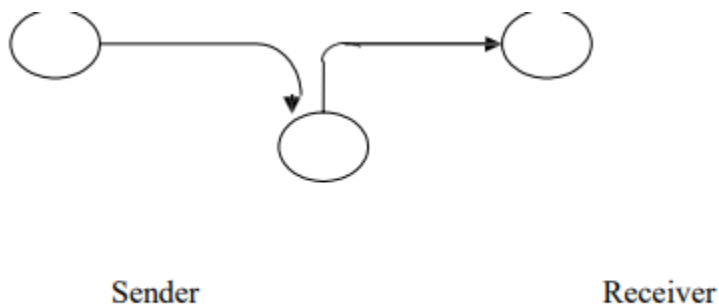
## Interception

An unauthorized party gains access to an asset. This is an attack on confidentiality. Unauthorized party could be a person, a program or a computer. e.g. Wire tapping to capture data in the network, illicit copying of files



## Modification

An unauthorized party not only gains access to but tampers with an asset. This is an attack on integrity. e.g., changing values in data file, altering a program, modifying the contents of messages being transmitted in a network.



## Fabrication

An unauthorized party inserts counterfeit objects into the system. This is an attack on authenticity. e.g., insertion of spurious message in a network or addition of records to a file.

