

# **FUNDAMENTALS AND LINK LAYER**

---

---

## **1.1. OVERVIEW OF DATA COMMUNICATIONS**

A network is a set of devices (also referred to as nodes) connected by communication links. A node can be a computer, printer or any other device capable of sending data and receiving data generated by other nodes on the network.

Data communication is the exchange of data between two devices via some form of transmission medium. The effectiveness of a data communication system depends on;

- (i) Delivery:** Data must be delivered to the correct destination. Data must be received by the intended device or user and only by that device or user.
- (ii) Accuracy:** The system must deliver the data without any change. Data that have been altered in transmission and left uncorrected are unusable.
- (iii) Timeliness:** The system must deliver the data in time. The system must deliver the audio or video data as they are produced, in the same order that they are produced, and without significant delay. This kind of delivery is called real-time transmission.
- (iv) Jitter:** Jitter refers to the variation in the packet arrival time. It is the uneven delay in the delivery of audio or video packets.

### **1.1.1 Components**

A data communication system consists of five components. They are

- (i) Message:** The message is the information or data to be communicated. Some forms of data representations are text, number, images, audio and video.
- (ii) Sender:** The sender is a device that sends the message.
- (iii) Receiver:** The receiver is a device that receives the message, sent by the sender.
- (iv) Medium:** The medium is a physical path through which the message can be passed between the sender and the receiver.
- (v) Protocol:** The protocol is a set of rules which governs the data communication. Without the protocol, two systems can be connected but not communicating. The key elements of a protocol are syntax, semantics and timing.

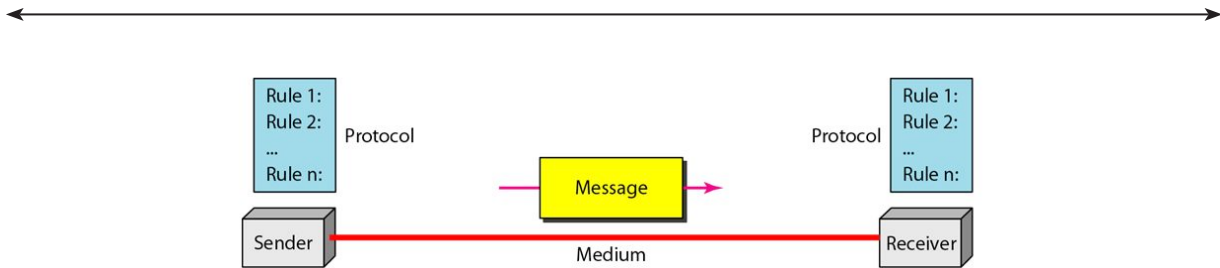


Figure 1.1 Five components of data communication

### 1.1.2 Data Representation

Five different forms are text, numbers, images, audio, and video.

#### *Text*

- Text is represented as a bit-pattern. (Bit-pattern → sequence of bits: 0s or 1s).
- Different sets of bit-patterns are used to represent symbols (or characters).
- Each set is called a code.
- The process of representing symbols is called encoding.
- Popular encoding system: ASCII, Unicode.

#### *Number*

- Number is also represented as a bit-pattern.
- ASCII is not used to represent number. Instead, number is directly converted to binary-form.

#### *Image*

- Image is also represented as a bit-pattern.
- An image is divided into a matrix of pixels (picture-elements).
- A pixel is the smallest element of an image. (Pixel → Small dot)
- The size of an image depends upon number of pixels (also called resolution). For example: An image can be divided into 1000 pixels or 10,000 pixels.
- Two types of images:
  - a) Black & White Image**
    - If an image is black & white, each pixel can be represented by a value either 0 or 1.
    - example: Chessboard
  - b) Color Image**
    - There are many methods to represent color images.
    - RGB is one of the methods to represent color images.
    - RGB is called so called ‘.’ each color is combination of 3 colors: red, green & blue.

**Audio**

- Audio is a representation of sound.
- By nature, audio is different from text, numbers, or images. Audio is continuous, not discrete.

**Video**

- Video is a representation of movie.
- Video can either
  - be produced as a continuous entity (e.g., by a TV camera), or
  - be a combination of images arranged to convey the idea of motion.

**1.1.3 Direction of Data Flow**

Communication between two devices can be of three types. They are;

- Simplex
- Half-duplex
- Full-duplex

**Simplex**

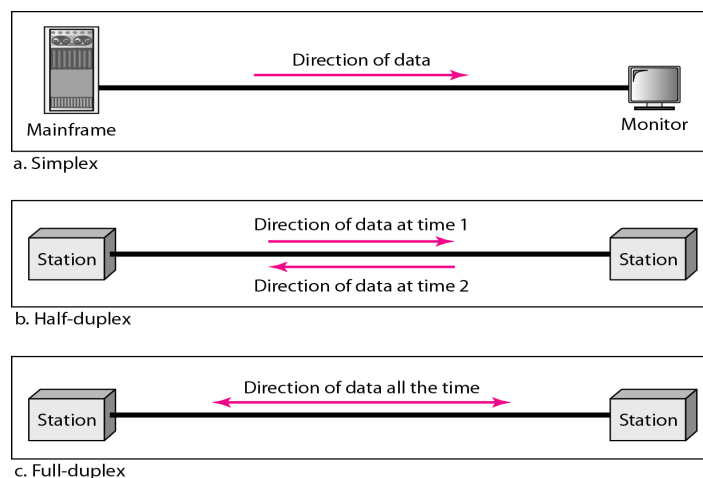
The communication is unidirectional. Only one of the two stations on a link can transmit and other can only receive.

**Half-duplex**

Each station can both transmit and receive, but not at the same time. The entire capacity of the channel is taken by the station which transmits the data.

**Full-duplex**

Both stations can transmit and receive the data at the same time. The capacity of the channel is divided between the signals traveling in opposite directions.



**Figure 1.2** Data flow (simplex, half-duplex, and full-duplex)

## 1.2. NETWORKS

- A network is defined as a set of devices interconnected by communication-links.
- This interconnection among computers facilitates information sharing among them.
- Computers may connect to each other by either wired or wireless media.
- Often, devices are referred to as nodes.
- A node can be any device capable of sending/receiving data in the network.
- *For example:* Computer & Printer
- The best-known computer network is the Internet.

### 1.2.1 Network Criteria

A network must meet following 3 criteria's:

- (i) Performance:** It depends on the factors such as transit time, response time, and the number of users, type of transmission medium, capabilities of the hardware and the efficiency of the software.
- (ii) Reliability:** Network reliability is measured by the accuracy of delivery, the frequency of failure and the time taken by a link to recover from failure.
- (iii) Security:** It is concerned with protection of data from unauthorized access.

#### *(i) Performance*

It depends on the factors such as transit time, response time, and the number of users, type of transmission medium, capabilities of the hardware and the efficiency of the software.

- Transit Time is defined as time taken to travel a message from one device to another.
- Response Time is defined as the time elapsed between enquiry and response.
- Often, performance is evaluated by 2 networking-metrics: i) throughput and ii) delay.
- Good performance can be obtained by achieving higher throughput and smaller delay times

#### *(ii) Reliability*

Network reliability is measured by the accuracy of delivery, the frequency of failure and the time taken by a link to recover from failure.

#### *(iii) Security*

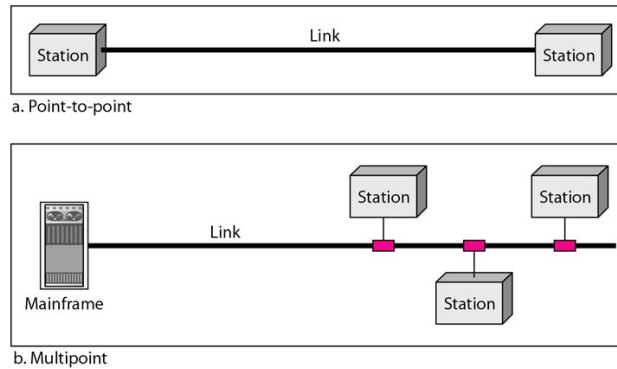
Security refers to the protection of data from the unauthorized access or damage. It also involves implementing policies for recovery from data-losses.

### 1.2.2 Physical Structures

#### 1.2.2.1 Type of Connection

Line configuration refers to the attachment of communication devices to a link. There are two types of line configurations:

- (i) **Point-to-point:** Provides a dedicated link between two devices. The entire channel capacity is reserved for the transmission between two devices only.
- (ii) **Multipoint:** More than two specific devices share a single link. The channel capacity is shared either spatially or temporally.



**Figure 1.3** Types of connection: *Point-to-point and Multipoint*

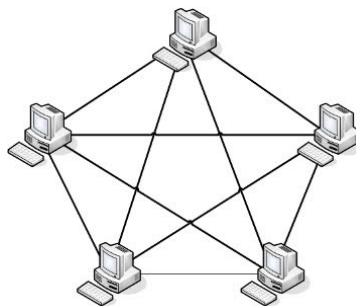
### 1.2.2.2 Physical Topology

The term physical topology refers to the way in which a network is laid out physically. The topology of a network is the geometric representation of the relationship of all the links and nodes to another. There are five types of topologies. They are,

- (i) Mesh topology
- (ii) Star topology
- (iii) Bus topology
- (iv) Ring topology
- (v) Hybrid topology

#### Mesh topology

In a mesh topology, every device has a dedicated point-to-point link to every other device. The term dedicated means that the link carries traffic only between the two devices it connects. A fully connected mesh network has  $n(n-1)$  physical channels to link  $n$  devices. To accommodate the links every device on the network must have  $(n-1)$  I/O ports.



**Figure 1.4** Mesh topology

**Advantages:**

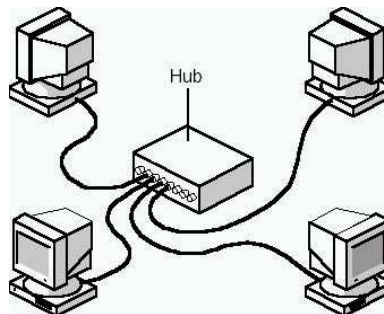
- (a) Mesh topology is robust.
- (b) Better privacy and security.
- (c) Failure of one link will not disturb other links.
- (d) Helps the network manager to find the precise location of the fault and solution.

**Disadvantages:**

- (a) Large amount of cabling and I/O ports are required.
- (b) Installation and reconnection are difficult.

**Star Topology**

In a star topology, each device has a dedicated point-to-point link to a central controller (HUB) only. If one link fails, that link is affected. All other links remain active.



*Figure 1.5 Star topology*

**Advantages:**

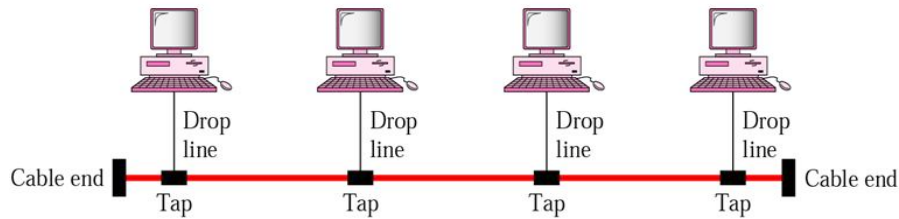
- (a) Less expensive.
- (b) Star topology is robust.
- (c) Fault identification and fault isolation are easy.
- (d) Modification of star network is easy.

**Disadvantages:**

- (a) If the central hub fails, the whole network will not work.
- (b) Communication is possible only through the hub.

**Bus topology**

One long cable acts as a backbone to link all the devices in the network. Nodes are connected to the back bone by taps and drop lines. Drop line is establishing the connection between the devices and the cable. The taps are used as connectors. To keep the energy level of the signal the taps are placed in the limited distance.



**Figure 1.6 Bus topology**

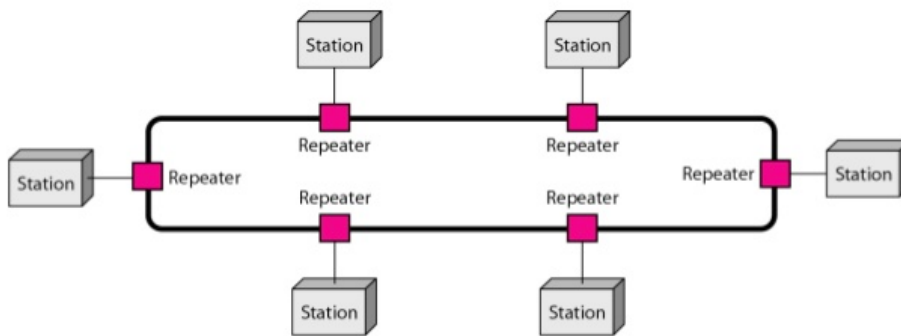
**Advantages:**

- (a) Easy installation.
- (b) Less cabling and less number of I/O port is required.
- (c) Less cost.

**Disadvantages:**

- (a) Network traffic is high.
- (b) Fault isolation and reconnection is difficult.
- (c) Adding new device is difficult.
- (d) A break in the bus cable stops all transmissions.

**Ring topology**



**Figure 1.7 Ring topology**

In a ring topology, each device has a dedicated point-to-point link with other devices. Each device is linked only to its immediate neighbors. A signal is travel along the ring in only one direction from device to device until it reaches its destination. The repeater is used to regenerate the signals during the transmission.

**Advantages:**

- (a) Easy to install and reconfigure.
- (b) Link failure can be easily found.

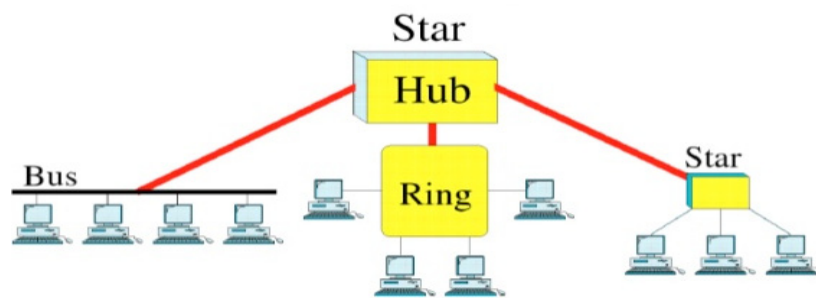
**Disadvantages:**

- (a) Maximum ring length and number of devices is limited.
- (b) Failure of one node on the ring affects the entire network.
- (c) Addition of nodes or removal of nodes disrupts the network.
- (d) Signal traffic is unidirectional.

**Hybrid Topology**

Integration of two or more different topologies to form a resultant topology which has good points of all the constituent basic topologies rather than having characteristics of one specific topology. This combination of topologies is done according to the requirements of the organization.

For example, if there exists a ring topology in one office department while a bus topology in another department, connecting these two will result in hybrid topology. Connecting two similar topologies cannot be termed as Hybrid topology. Star-Ring and Star-Bus networks are most common examples of hybrid network.



*Figure 1.8 Hybrid topology*

**1.3. BUILDING NETWORK AND ITS TYPES**

A computer network must provide a general, cost effective, fair and robust connectivity among a large number of computers. Networks do not remain fixed at any single point in time, but must evolve to accommodate changes in both the underlying technologies upon which they are based as well as changes in the demands placed on them by application programs.

To deal with this complexity, network designers have developed general blue prints called network architecture.

A network architecture is defined as which identifies the available hardware and software components and shows how they can be arranged to form a complete network system.

To build a network we must know the following things.

- Discover the requirements that different applications and different communities of people place on the network
- A network architecture on which the applications are going to be developed



- The key elements in the implementation of computer networks
- Identifying the key metrics that are used to evaluate the performance of computer networks.

### 2.3.1 Applications

Some applications of the computer networks are the World Wide Web, email, streaming audio and video, chat rooms, and music (file) sharing. Most people know the Internet through its applications only. The Web, presents an intuitively simple interface. Users can view pages full of textual and graphical objects. By clicking the selective objects on a page the people may go for the next page to be viewed. This can be done with the help of an identifier, called a uniform resource locator (URL), uniquely names every possible page that can be viewed from your Web browser. For example,

`http://www.mkp.com/pd3e`

The string `http` indicates that the Hypertext Transfer Protocol (HTTP) should be used to download the page, `www.mkp.com` is the name of the machine that serves the page, and `pd3e` uniquely identifies the page at the publisher's site.

Whenever the URL is going to be clicked 17 messages may be exchanged over the Internet. This number includes;

- (i) Six messages to translate the server name into its internet address.
- (ii) Three messages to set up a Transmission Control Protocol connection between the browser and the server.
- (iii) Four messages for the browser to send the HTTP get request and the server to respond with the request page.
- (iv) Four messages to tear down the Transmission Control Protocol connection.

Another emerging application of the Internet is the delivery of streaming audio and video. The video file is completely fetched from a remote machine and then played on the local machine. In video streaming, the sender and the receiver must have synchronization between them. There are different types of video applications. They include;

- (i) Video-on-demand: Reads a preexisting movie from disk and transmit it over the network.
- (ii) Video conferencing: Videoconferencing (or video conference) means to conduct a conference between two or more participants at different sites by using computer networks to transmit audio and video data. `vic` is the example tool for video conferencing.

### 2.3.2 Requirements

Networks are constantly changing as the technology evolves and new applications are invented. For building a computer network we must identify the set of constraints and requirements that influence network design and the expectations we have for a network depend on our perspective:

- (i) An application programmer would list the services that the application needs.

- (ii) A network designer would list the properties of a cost-effective design.
- (iii) A network provider would list the characteristics of a system that is easy to administer and manage.

### 1.3.2.1 Connectivity

In figure 1.4, each and every node is attached to one or more point-to-point links with the help of switches. The node attached to more than one link is running software that forwards the data received from one link to another link.

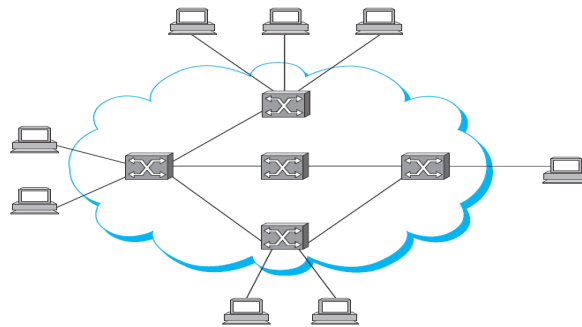


Figure 1.9 Switched network

Switches are devices capable of creating temporary connections between two or more devices linked to the switch. In a switched network, some of these nodes are connected to the end systems (computers or telephones, for example) and others are used only for routing.

Important three methods of switching are:

- (i) Circuit switching
- (ii) Packet switching
- (iii) Message switching

Packet-switched networks can further be divided into two subcategories:

- Virtual-circuit networks
- Datagram networks

In the above figure, the cloud distinguishes between the nodes on the inside of the cloud that implement the network and the nodes on the outside of the cloud that use the network. A set of independent clouds (networks) are interconnected to form an internetwork (internet). Internetwork is formed with the help of router and gateway, which forwards message from one network to another by using the IP addresses of the source and destination nodes. There are two types of routing;

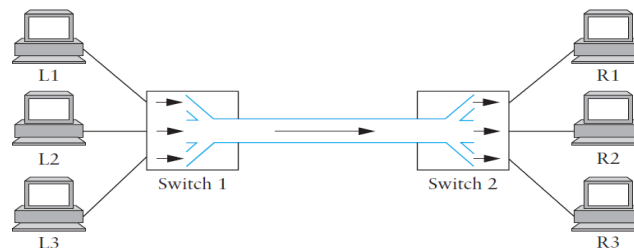
- (i) **Unicast:** Source node wants to send a message to a single destination node.
- (ii) **Multicast:** Source node to sends a message to some subset of the other nodes, but not all of them.

### 1.3.2.2 Cost-Effective Resource Sharing

During data communication, the hosts share a network by multiplexing, which means that a system resource is shared among multiple users.

#### *Multiplexing and de-multiplexing*

The fundamental idea of packet switching is to multiplex multiple flows of data over a single physical link. This can be achieved by adding identifier to the header message. It is also known as de-multiplexing key. It gives the address to which it has to communicate. The messages are de-multiplexed at the destination side.

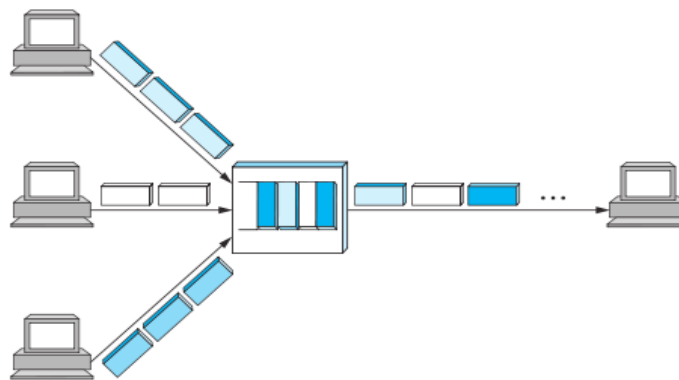


**Figure 1.10** multiplexing multiple logical flows over a single physical link

There are several different methods for multiplexing;

- (i) **Synchronous time-division multiplexing (STDM):** In STDM the time is divided into equal-sized quanta and, in a round-robin fashion, gives each flow a chance to send its data over the physical link.
- (ii) **Frequency division multiplexing (FDM):** The idea of FDM is to transmit each flow over the physical link at a different frequency.

In the packet switched network, the message is divided into packets of fixed or variable size. The size of the packet is determined by the network and the governing protocol. In packet switching, there is no resource allocation for a packet (no reserved bandwidth on the links, and no scheduled processing time for each packet). Resources are allocated on demand. The allocation is done on a first-come, first-served basis. When a switch receives a packet, no matter what is the source or destination, the packet must wait if there are other packets being processed.



**Figure 1.11** Packet switching- packets from multiple sources onto one shared link

### 1.3.2.3 Support for Common Services

The next requirement of a computer network is that the application programs running on the hosts connected to the network must be able to communicate in a meaningful way. This can be done by providing logical channels over which application-level processes can communicate with each other; each channel provides the set of services required by that application. The challenge is to recognize what functionality the channels should provide to application programs. For example,

- (i) Does the application require a guarantee that messages sent over the channel are delivered, or is it acceptable if some messages fail to arrive?
- (ii) Is it necessary that messages arrive at the recipient process in the same order in which they are sent?
- (iii) Does the network need to ensure that no third parties are able to eavesdrop on the channel, or is privacy not a concern?

#### **Identifying Common Communication Patterns**

While designing the channels, understanding the communication needs a collection of applications, then extracting their common communication requirements and finally incorporating the functionality that meets these requirements in the network. FTP (File Transfer Protocol) or NFS (Network File System) are the example for file access program over a network. The process that requests access to the file is called the client, and the process that supports access to the file is called the server.

#### **Reliability**

A network must satisfy the following important criteria;

- (iv) **Performance:** It depends on the factors such as transit time, response time, and the number of users, type of transmission medium, capabilities of the hardware and the efficiency of the software.
- (v) **Reliability:** Network reliability is measured by the accuracy of delivery, the frequency of failure and the time taken by a link to recover from failure.
- (vi) **Security:** It is concerned with protection of data from unauthorized access.

### 1.3.3 Categories of Network

Two popular types of networks:

- (i) LAN (Local Area Network) &
- (ii) WAN (Wide Area Network)

#### 1.3.3.1 LAN

- LAN is used to connect computers in a single office, building or campus (Figure 1.12).
- LAN is usually privately owned network.
- A LAN can be simple or complex.
  - a) Simple: LAN may contain 2 PCs and a printer.



b) Complex: LAN can extend throughout a company.

- Each host in a LAN has an address that uniquely defines the host in the LAN.
- A packet sent by a host to another host carries both source host's and destination host's addresses.
- LANs use a smart connecting switch.
- The switch is able to
  - recognize the destination address of the packet &
  - guide the packet to its destination.
- The switch
  - reduces the traffic in the LAN &
  - allows more than one pair to communicate with each other at the same time.

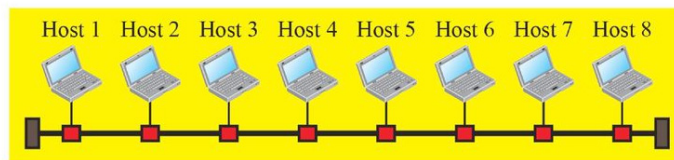
**Advantages:**

**(i) Resource Sharing**

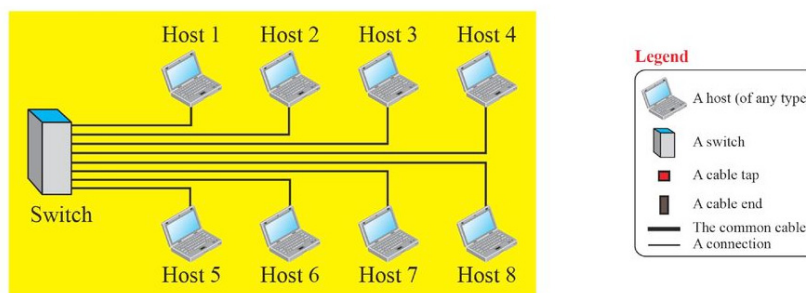
- Computer resources like printers and hard disks can be shared by all devices on the network.

**(ii) Expansion**

- Nowadays, LANs are connected to WANs to create communication at a wider level.



a. LAN with a common cable (past)



b. LAN with a switch (today)

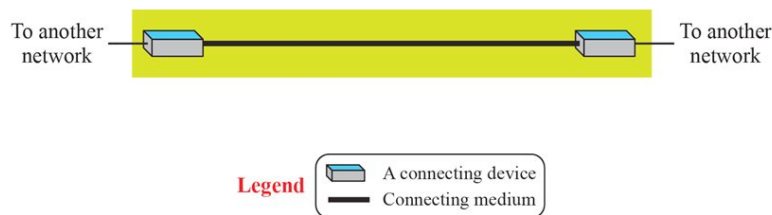
**Figure 1.12 An isolated LAN**

**1.3.3.2 WAN**

- WAN is used to connect computers anywhere in the world.
- WAN can cover larger geographical area. It can cover cities, countries and even continents.

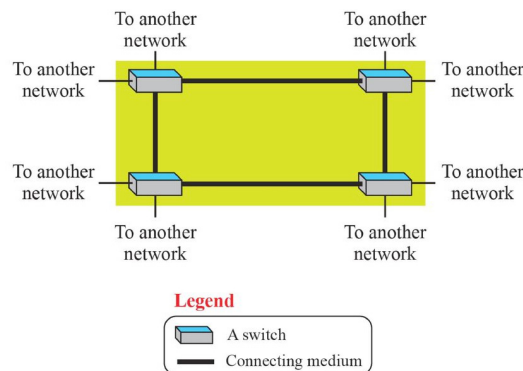
## 1.14 Communication Networks

- WAN interconnects connecting devices such as switches, routers, or modems.
- Normally, WAN is
  - created & run by communication companies (Ex: BSNL, Airtel)
  - leased by an organization that uses it.
- A WAN can be of 2 types:
  - a) Point-to-Point WAN
    - A point-to-point WAN is a network that connects 2 communicating devices through a transmission media (Figure 1.13).



**Figure 1.13 A point-to-point WAN**

- b) Switched WAN
  - A switched WAN is a network with more than two ends.
  - The switched WAN can be the backbones that connect the Internet.
  - A switched WAN is a combination of several point-to-point WANs that are connected by switches (Figure 1.14).



**Figure 1.14 A switched WAN**

### 1.3.3.2.1 Internetwork

- A network of networks is called an internet. (Internet → inter-network) (Figure 1.16).
- For example (Figure 1.15):
  - Assume that an organization has two offices,
    - i) First office is on the east coast &

ii) Second office is on the west coast.

- Each office has a LAN that allows all employees in the office to communicate with each other.
- To allow communication between employees at different offices, the management leases a point-to-point dedicated WAN from a ISP and connects the two LANs.

(ISP → Internet service provider such as a telephone company ex: BSNL).

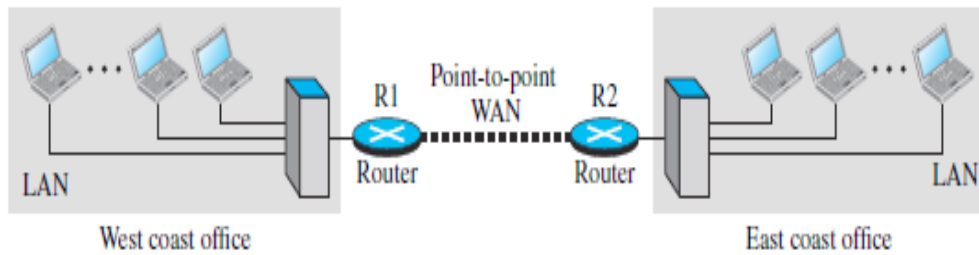


Figure 1.15 An internetwork made of two LANs and one point-to-point WAN

- When a host in the west coast office sends a message to another host in the same office, the router blocks the message, but the switch directs the message to the destination.
- On the other hand, when a host on the west coast sends a message to a host on the east coast, router R1 routes the packet to router R2, and the packet reaches the destination.

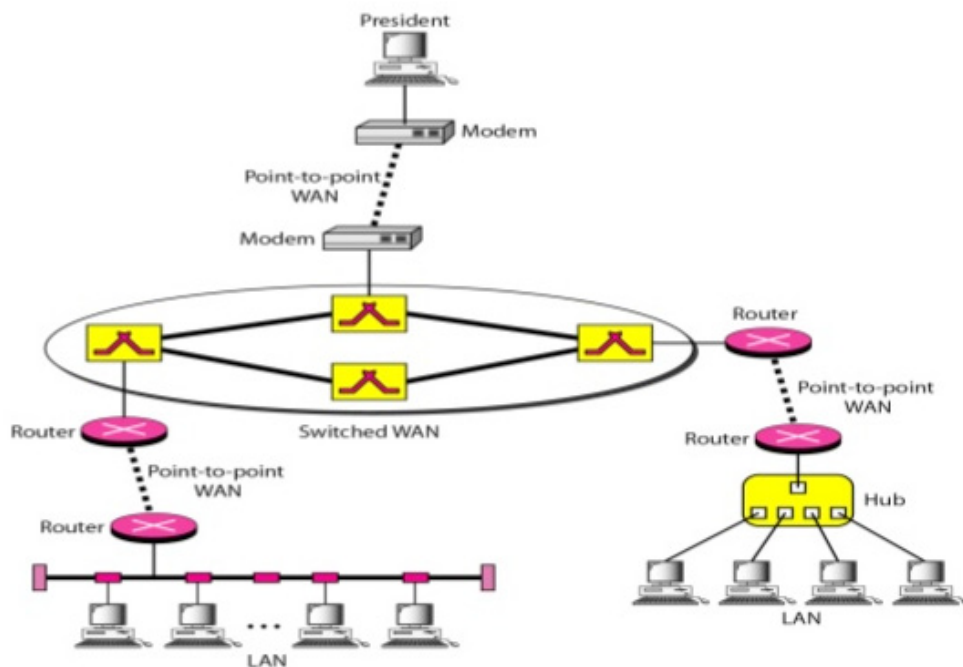


Figure 1.16 A heterogeneous network made of four WANs and three LANs

1.3.3.2 LAN vs. WAN

Parameters	LAN	WAN
Expands to	Local Area Network	Wide Area Network
Meaning	LAN is used to connect computers in a single office, building or campus	WAN is used to connect computers in a large geographical area such as countries
Ownership of network	Private	Private or public
Range	Small: up to 10 km	Large: Beyond 100 km
Speed	High: Typically 10, 100 and 1000 Mbps	Low: Typically 1.5 Mbps
Propagation Delay	Short	Long
Cost	Low	High
Congestion	Less	More
Design & maintenance	Easy	Difficult
Fault Tolerance	More Tolerant	Less Tolerant
Media used	Twisted pair	Optical fiber or radio waves
Used for	College, Hospital	Internet
Interconnects	LAN interconnects hosts	WAN interconnects connecting devices such as switches, routers, or modems

1.3.3.3 SWITCHING

- An internet is a switched network in which a switch connects at least two links together.
- A switch needs to forward data from a network to another network when required.
- Two types of switched networks are 1) circuit-switched and 2) packet-switched networks.

3.3.3.1 Circuit Switched Network

- A dedicated connection, called a circuit, is always available between the two end systems.
- The switch can only make it active or inactive.

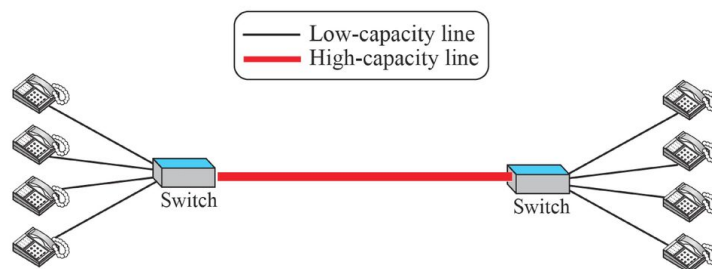


Figure 1.17 A circuit switched network



As shown in Figure 1.17, the 4 telephones at each side are connected to a switch.

- The switch connects a telephone at one side to a telephone at the other side.
- A high-capacity line can handle 4 voice communications at the same time.
- The capacity of high line can be shared between all pairs of telephones.
- The switch is used for only forwarding.

**Advantage:**

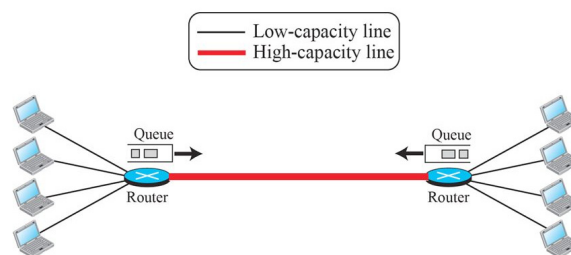
- A circuit-switched network is efficient only when it is working at its full capacity.

**Disadvantage:**

- Most of the time, the network is inefficient because it is working at partial capacity.

### 1.3.3.2 Packet Switched Network

- In a computer network, the communication between the 2 ends is done in blocks of data called packets.
- The switch is used for both storing and forwarding because a packet is an independent entity that can be stored and sent later.



**Figure 1.18** A packet switched network

As shown in Figure 1.18, the 4 computers at each side are connected to a router.

- A router has a queue that can store and forward the packet.
- The high-capacity line has twice the capacity of the low-capacity line.
- If only 2 computers (one at each site) need to communicate with each other, there is no waiting for the packets.
- However, if packets arrive at one router when high-capacity line is at its full capacity, the packets should be stored and forwarded.

**Advantages:**

- A packet-switched network is more efficient than a circuit switched network.

**Disadvantage:**

- The packets may encounter some delays.

## **1.4. OVERVIEW OF INTERNET**

Internet architecture is a meta-network, a constantly changing collection of thousands of individual networks intercommunicating with a common protocol.

The Internet's architecture is described in its name, a short form of the compound word "inter-networking". This architecture is based in the very specification of the standard TCP/IP protocol, designed to connect any two networks which may be very different in internal hardware, software, and technical design.

Once two networks are interconnected, communication with TCP/IP is enabled end-to-end, so that any node on the Internet has the near magical ability to communicate with any other, no matter where they are. This openness of design has enabled the Internet architecture to grow to a global scale. The Internet architecture, which is also sometimes called the TCP/IP architecture. Internet is made up of (Figure 1.19)

- (1) Backbones
- (2) Provider networks &
- (3) Customer networks

### **(1) Backbones**

- Backbones are large networks owned by communication companies such as BSNL and Airtel.
- The backbone networks are connected through switching systems, called peering points.

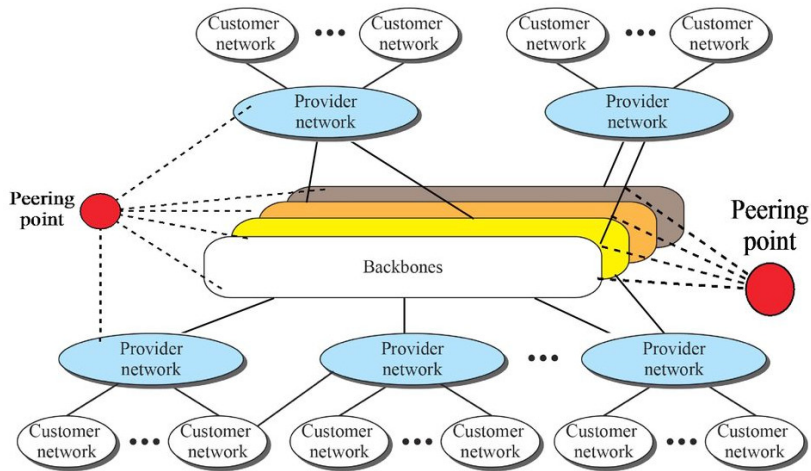
### **(2) Provider Networks**

- Provider networks use the services of the backbones for a fee.
- Provider networks are connected to backbones and sometimes to other provider networks.

### **(3) Customer Networks**

- Customer networks actually use the services provided by the Internet.
- Customer networks pay fees to provider networks for receiving services.

Backbones and provider networks are also called Internet Service Providers (ISPs). The backbones are often referred to as international ISPs. The provider networks are often referred to as national or regional ISPs.



**Figure 1.19** *The Internet today*

### 1.4.1 Accessing The Internet

- The Internet today is an internetwork that allows any user to become part of it.
- However, the user needs to be physically connected to an ISP.
- The physical connection is normally done through a point-to-point WAN.

#### *(a) Using Telephone Networks*

- Most residences have telephone service, which means they are connected to a telephone network.
- Most telephone networks have already connected themselves to the Internet.
- Thus, residences can connect to the Internet using a point-to-point WAN.
- This can be done in two ways:

##### *(i) Dial-up service*

- A modem can be added to the telephone line.
- A modem converts data to voice.
- The software installed on the computer
  - dials the ISP &
  - imitates making a telephone connection.

##### *Disadvantages:*

- The dial-up service is very slow.
- When line is used for Internet connection, it cannot be used for voice connection.
- It is only useful for small residences.

**(ii) DSL Service**

- DSL service also allows the line to be used simultaneously for voice & data communication.
- Some telephone companies have upgraded their telephone lines to provide higher speed Internet services to residences.

**(b) Using Cable Networks**

- A residence can be connected to the Internet by using cable service.
- Cable service provides a higher speed connection.
- The speed varies depending on the number of neighbors that use the same cable.

**(c) Using Wireless Networks**

- A residence can use a combination of wireless and wired connections to access the Internet.
- A residence can be connected to the Internet through a wireless WAN.

**(d) Direct Connection to the Internet**

- A large organization can itself become a local ISP and be connected to the Internet.
- The organization
  - leases a high-speed WAN from a carrier provider and
  - connects itself to a regional ISP.

## **1.4.2 Standards And Administration**

### **1.4.2.1 Internet Standards**

- An Internet standard is a thoroughly tested specification useful to those who work with the Internet.
- The Internet standard is a formalized-regulation that must be followed.
- There is a strict procedure by which a specification attains Internet standard status.
- A specification begins as an Internet draft.
- An Internet draft is a working document with no official status and a 6-month lifetime.
- Upon recommendation from the Internet authorities, a draft may be published as a RFC.
- Each RFC is edited, assigned a number, and made available to all interested parties.
- RFCs go through maturity levels and are categorized according to their requirement level. (working document → a work in progress RFC → Request for Comment)

### **1.4.2.2 Maturity Levels**

- An RFC, during its lifetime, falls into one of 6 maturity levels (Figure 1.20):

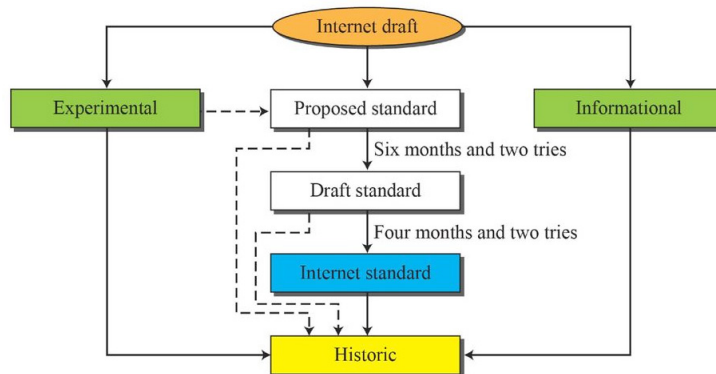


Figure 1.20 Maturity levels of an RFC

**(i) Proposed Standard**

- Proposed standard is specification that is stable, well-understood & of interest to Internet community.
- Specification is usually tested and implemented by several different groups.

**(ii) Draft Standard**

- A proposed standard is elevated to draft standard status after at least 2 successful independent and interoperable implementations.

**(iii) Internet Standard**

- A draft standard reaches Internet standard status after demonstrations of successful implementation.

**(iv) Historic**

- The historic RFCs are significant from a historical perspective.
- They either have been superseded by later specifications or have never passed the necessary maturity levels to become an Internet standard.

**(v) Experimental**

- An RFC classified as experimental describes work related to an experimental situation.
- Such an RFC should not be implemented in any functional Internet service.

**(vi) Informational**

- An RFC classified as informational contains general, historical, or tutorial information related to the Internet.
- Usually, it is written by a vendor.

(ISOC → Internet Society

IAB → Internet Architecture Board)

(IETF → Internet Engineering Task Force

IRTF → Internet Research Task Force)

(IESG → Internet Engineering Steering Group

IRSG → Internet Research Steering Group)

### **1.4.2.3 Requirement Levels**

- RFCs are classified into 5 requirement levels:

#### **(i) Required**

- An RFC labeled required must be implemented by all Internet systems to achieve minimum conformance.
- For example, IP and ICMP are required protocols.

#### **(ii) Recommended**

- An RFC labeled recommended is not required for minimum conformance.
- It is recommended because of its usefulness.
- For example, FTP and TELNET are recommended protocols.

#### **(iii) Elective**

- An RFC labeled elective is not required and not recommended.
- However, a system can use it for its own benefit.

#### **(iv) Limited Use**

- An RFC labeled limited use should be used only in limited situations.
- Most of the experimental RFCs fall under this category.

#### **(v) Not Recommended**

- An RFC labeled not recommended is inappropriate for general use.
- Normally a historic RFC may fall under this category.

### **1.4.2.4 Internet Administration**

#### **(a) ISOC**

- ISOC is a nonprofit organization formed to provide support for Internet standards process (Fig 1.21).
- ISOC maintains and supports other Internet administrative bodies such as IAB, IETF, IRTF, and IANA.

#### **(b) IAB**

- IAB is the technical advisor to the ISOC.
- Two main purposes of IAB:
  - (i) To oversee the continuing development of the TCP/IP Protocol Suite
  - (ii) To serve in a technical advisory capacity to research members of the Internet community.
- Another responsibility of the IAB is the editorial management of the RFCs.

- IAB is also the external liaison between the Internet and other standards organizations and forums.
- IAB has 2 primary components: i) IETF and ii) IRTF.
  - (i) IETF
    - IETF is a forum of working groups managed by the IESG.
    - IETF is responsible for identifying operational problems & proposing solutions to the problems
    - IETF also develops and reviews specifications intended as Internet standards.
    - The working groups are collected into areas, and each area concentrates on a specific topic.
    - Currently 9 areas have been defined. The areas include applications, protocols, routing, network management next generation (IPng), and security.
  - (ii) IRTF
    - IRTF is a forum of working groups managed by the IRSG.
    - IRTF focuses on long-term research topics related to Internet protocols, applications, architecture, and technology.

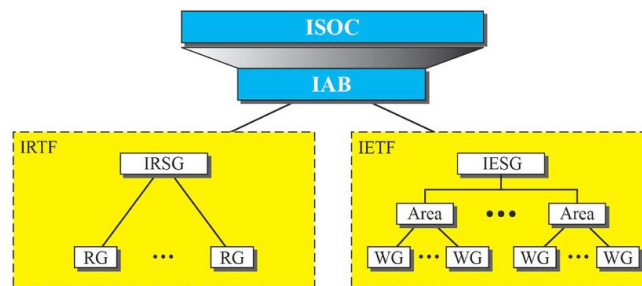


Figure 1.21 Internet administration

## 1.5 PROTOCOL LAYERING

- A protocol defines the rules that both the sender and receiver and all intermediate devices need to follow to be able to communicate effectively.
- When communication is simple, we may need only one simple protocol. When communication is complex, we need to divide the task b/w different layers. We need a protocol at each layer, or protocol layering.

### 1.5.1 Scenarios

#### First Scenario

- In the first scenario, communication is so simple that it can occur in only one layer (Figure 1.22).

- Assume Maria and Ann are neighbors with a lot of common ideas.
- Communication between Maria and Ann takes place in one layer, face to face, in the same language



Figure 1.22 A single - layer protocol

**Second Scenario**

- Maria and Ann communicate using regular mail through the post office (Figure 1.23).
- However, they do not want their ideas to be revealed by other people if the letters are intercepted.
- They agree on an encryption/decryption technique.
- The sender of the letter encrypts it to make it unreadable by an intruder; the receiver of the letter decrypts it to get the original letter.

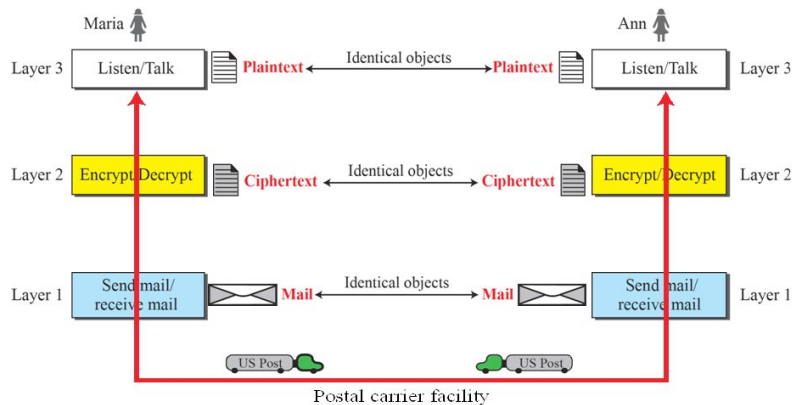


Figure 1.23 A three - layer protocol

**1.5.1.1 Protocol Layering**

- Protocol layering enables us to divide a complex task into several smaller and simpler tasks.
- Modularity means independent layers.
- A layer (module) can be defined as a black box with inputs and outputs, without concern about how inputs are changed to outputs.
- If two machines provide the same outputs when given the same inputs, they can replace each other.



**Advantages:**

- (1) It allows us to separate the services from the implementation.
- (2) There are intermediate systems that need only some layers, but not all layers.

**Disadvantage:**

- (1) Having a single layer makes the job easier. There is no need for each layer to provide a service to the upper layer and give service to the lower layer.

**1.5.2 Principles of Protocol Layering**

**(i) First Principle**

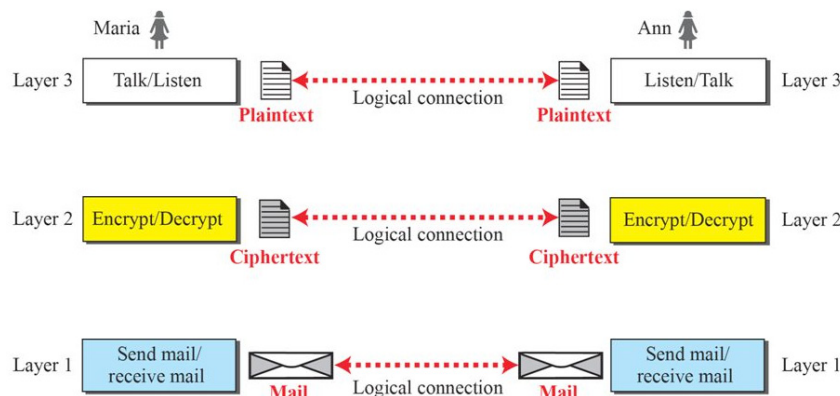
- If we want bidirectional communication, we need to make each layer able to perform 2 opposite tasks, one in each direction.
- For example, the third layer task is to listen (in one direction) and talk (in the other direction).

**(ii) Second Principle**

- The two objects under each layer at both sites should be identical.
- For example, the object under layer 3 at both sites should be a plaintext letter.

**1.5.3 Logical Connections**

- We have layer-to-layer communication (Figure 1.24).
- There is a logical connection at each layer through which 2 end systems can send the object created from that layer.



**Figure 1.24 Logical connections between peer layers**

**1.6 OSI MODEL**

An ISO standard that covers all the aspects of network communication is the Open System Interconnection Model. Open system is a set of protocols that allows any two different systems to communicate regardless of their underlying architecture. Without changing the logic of the hardware and software, two systems can communicate with the help of open system. OSI model consists of

seven layers. The layers define the process of moving the information across the network. The seven layers of the OSI model are;

- (1) physical layer
- (2) data link layer
- (3) network layer
- (4) transport layer
- (5) session layer
- (6) presentation layer and
- (7) application layer

### Layered architecture

When a message travels from the sender to receiver, it may pass through many intermediate nodes. Only the first three layers of the intermediate nodes are involved in all communication. Each layer calls upon the services of the layers just below it. This is done with the help of protocols.

The processes on each machine that communicate at a given layer are called Peer-to-peer. The passing of data and network information between the layers are carried out with the help of interfaces. Interface is used to define the information and services to be provided by each layer.

#### 1.6.1 OSI vs. TCP/IP

- i) The four bottommost layers in the OSI model & the TCP/IP model are same (Figure 1.25). However, the Application-layer of TCP/IP model corresponds to the Session, Presentation & Application Layer of OSI model.

Two reasons for this are:

- a) TCP/IP has more than one transport-layer protocol.
  - b) Many applications can be developed at Application layer
- ii) The OSI model specifies which functions belong to each of its layers. In TCP/IP model, the layers contain relatively independent protocols that can be mixed and matched depending on the needs of the system.

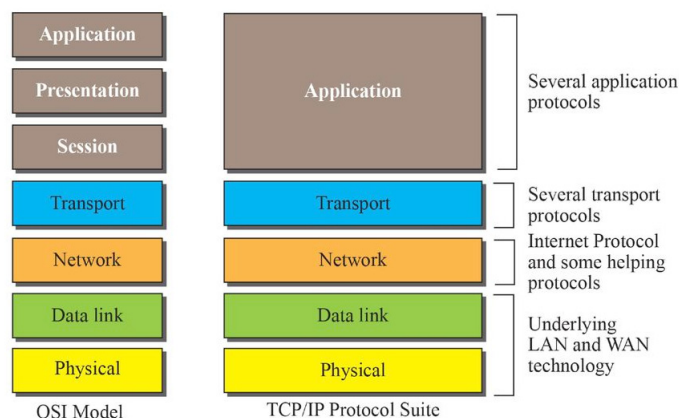


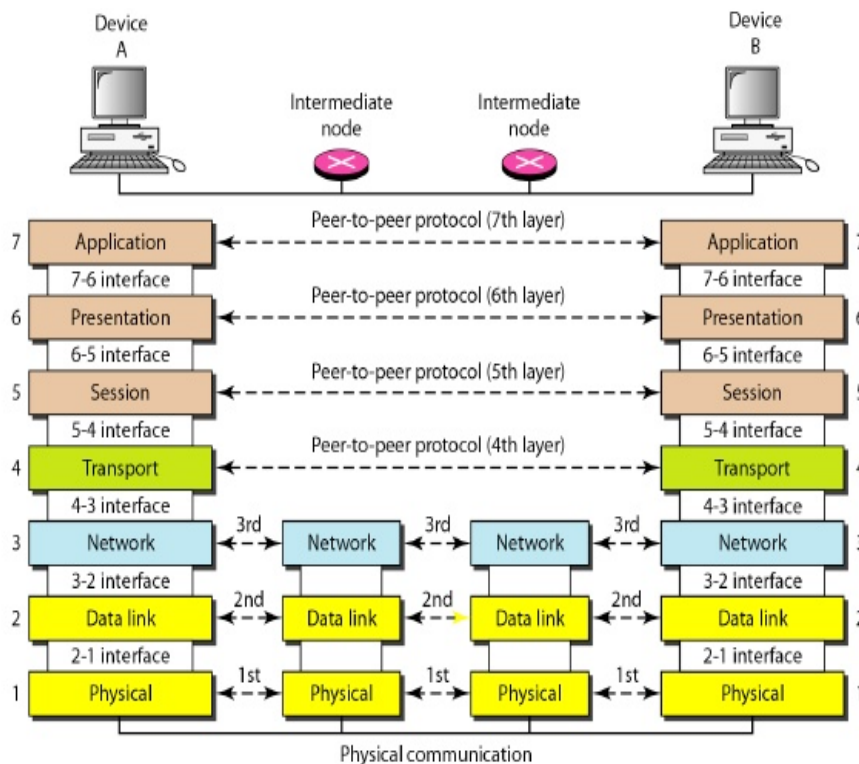
Figure 1.26 TCP/IP and OSI model

**Lack of OSI Model's Success**

- OSI was completed when TCP/IP was fully in place and a lot of time and money had been spent on the suite; changing it would cost a lot.
- Some layers in the OSI model were never fully defined.
- When OSI was implemented by an organization in a different application, it did not show a high enough level of performance

**1.6.2 Organization of the Layers**

The below figure 1.26 gives an overall view of the OSI layers. The seven layers are categorized into three subgroups. Layers 1, 2, and 3-physical, data link, and network-are the network support layers. Layers 5, 6, and 7-session, presentation, and application – are the user support layers. Layer 4, the transport layer, links the two subgroups and ensures that, what the lower layers have transmitted is in a form that the upper layers can use.



**Figure 1.26 The interaction between layers in the OSI model**

Network support layers deal with the physical aspects of moving data from one device to another such as electrical specifications, physical connections, physical addressing, and transport timing and reliability. User support layers allow interoperability among unrelated software systems. The upper OSI layers are always implemented in software; lower layers are a combination of hardware and software, except for the physical layer, which is mostly hardware.

The process starts at the application layer then moves from layer to layer in descending, sequential order. At each layer, a **header**, or possibly a **trailer**, can be added to the data unit. The trailer is added only at layer 2. When the formatted data unit passes through the physical layer, it

is changed into an electromagnetic signal and transported along a physical link. Upon reaching its destination, the signal passes into physical layer and is transformed back into digital form. The data units are then moved back up through the OSI layers.

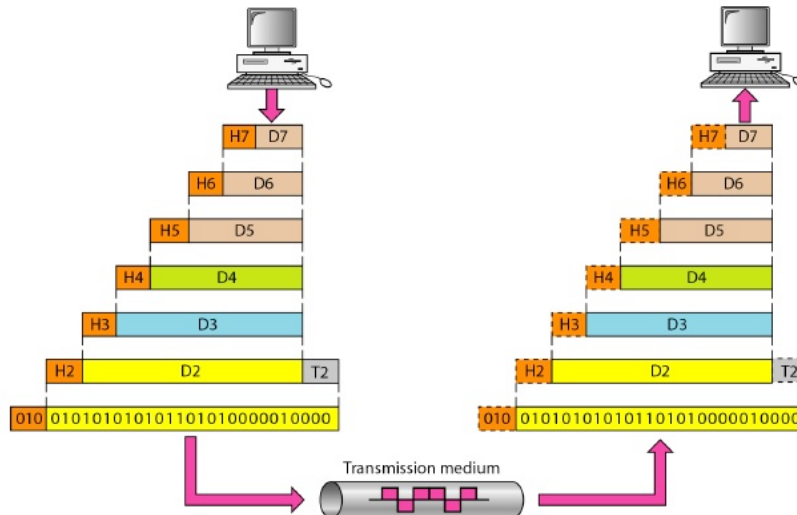


Figure 1.27 An exchange using the OSI model

When the block of data reaches the next higher layer, the headers and trailers attached by the sending layer are removed. When the data unit reaches the application layer, the message is again in a form appropriate to the application and is made available to the recipient.

### 1.6.3 Layers in the OSI Model

#### *Physical Layer*

The physical layer is responsible for movements of individual bits from one hop (node) to the next. 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. Physical layer defines the procedures and functions that physical devices and interfaces have to perform for transmission of data. The physical layer is also concerned with the following:

- (a) **Physical characteristics of interfaces and medium:** Defines the characteristics of the interface between the devices and the transmission medium. It also defines the type of transmission medium.
- (b) **Representation of bits:** A stream of bits is encoded into signals (electrical or optical). It defines the type of encoding.
- (c) **Data rate:** The number of bits sent/Sec is also defined by the physical layer.
- (d) **Synchronization of bits:** The sender and the receiver clocks must be synchronized.
- (e) **Line configuration:** The connection of devices to the media (point-to-point configuration or multipoint configuration).
- (f) **Physical topology:** The physical topology defines how devices are connected to make a network.

- ←—————→
- (g) **Transmission mode:** The physical layer also defines the direction of transmission between two devices: simplex, half-duplex, or full-duplex.

### **Data Link Layer**

The data link layer is responsible for moving frames from one hop (node) to the next. Other responsibilities of the data link layer include the following:

- (a) **Framing:** The data link layer divides the stream of bits received from the network layer into manageable data units called frames.
- (b) **Physical addressing:** It adds a header to the frame to define the sender and/or receiver of the frame
- (c) **Flow control:** The data link layer imposes a flow control mechanism to avoid overwhelming the receiver.
- (d) **Error control:** It adds reliability by adding mechanisms to detect and retransmit damaged or lost frames. It also uses a mechanism to recognize duplicate frames by adding the trailer to the end of the frame.
- (e) **Access control:** It determine which device has control over the link at any given time, when two or more devices are connected to the same link.

### **Network Layer**

The network layer is responsible for the delivery of individual packets from the source host to the destination host. Other responsibilities of the network layer include the following;

- (a) **Logical addressing:** When a packet passes the network boundary, the network layer adds the logical addresses of the sender and receiver.
- (b) **Routing:** When independent networks or links are connected to create internetworks, the connecting devices (called routers or switches) route or switch the packets to their final destination.

### **Transport Layer**

The transport layer is responsible for the delivery of a message from one process to another. Other responsibilities of the transport layer include the following;

- (a) **Service-point addressing:** The transport layer gets the entire message to the correct process on the destination system by adding a type of address called a service-point address (or port address).
- (b) **Segmentation and reassembly:** A message is divided into transmittable segments, with each segment containing a sequence number. These numbers are used to reassemble the message at the destination and to identify and replace packets that were lost in transmission.
- (c) **Connection control:** In a connectionless service each segment is treated as independent packet and in connection oriented service each segment is treated as dependent packet. After all the data are transferred, the connection is terminated.

- (d) **Flow control:** Flow control is performed from end to end rather than across a single link.
- (e) **Error control:** At this layer the error control is performed in a process-to-process rather than across a single link.

### ***Session Layer***

The session layer is responsible for dialog control and synchronization. Specific responsibilities of the session layer include the following;

- (a) **Dialog control:** The session layer allows two systems to enter into a dialog. It allows the communication between two processes to take place in either half-duplex or full-duplex mode.
- (b) **Synchronization:** The session layer allows a process to add checkpoints, or synchronization points, to a stream of data. For example, if a system is sending a file of 100 pages, it is advisable to insert checkpoints after every 10 pages to ensure that each 10-page unit is received and acknowledged independently. In this case, if a crash happens during the transmission of page 23, the only pages that need to be resent after system recovery are pages 21 to 30.

### ***Presentation Layer***

The presentation layer is responsible for translation, compression, and encryption. Specific responsibilities of the presentation layer include the following:

- (a) **Translation:** The presentation layer is responsible for the interoperability between different encoding methods.
- (b) **Encryption:** To carry sensitive information, a system must be able to ensure privacy. Encryption means that the sender transforms the original information to another form and sends the resulting message out over the network. Decryption reverses the original process to transform the message back to its original form.
- (c) **Compression:** Data compression reduces the number of bits contained in the information. Data compression is important in the transmission of multimedia such as text, audio, and video.

### ***Application Layer***

The application layer is responsible for providing services to the user. Specific services provided by the application layer include the following:

- (a) **Network virtual terminal:** A network virtual terminal is a software version of a physical terminal and it allows a user to log on to a remote host.
- (b) **File transfer, access, and management:** This application allows a user to access files in a remote host, to retrieve files from a remote computer for use in the local computer, and to manage or control files in a remote computer locally.
- (c) **Mail services:** This application provides the basis for e-mail forwarding and storage.
- (d) **Directory services:** This application provides distributed database sources and access for global information about various objects and services.

## 1.7. PHYSICAL LAYER

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.

#### 1.7.1 Data and Signals

##### 1.7.1.1 Analog & Digital Data

- To be transmitted, data must be transformed to electromagnetic-signals.
- Data can be either analog or digital.

(a) *Analog Data refers to information that is continuous.*

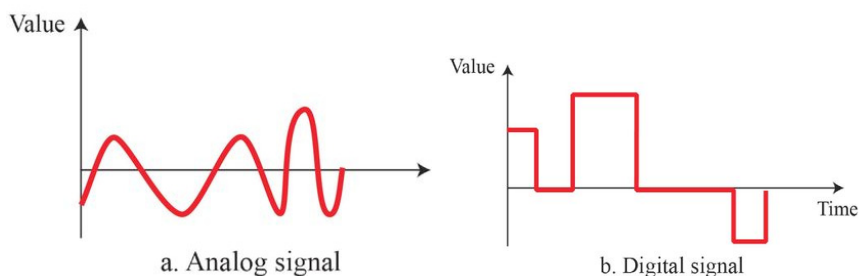
- For example:
  - The sounds made by a human voice.

(b) *Digital Data refers to information that has discrete states.*

- For example:
  - Data are stored in computer-memory in the form of 0s and 1s.

##### 1.7.1.2 Analog & Digital Signals

- Signals can be either analog or digital (Figure 1.28).
- *Analog Signal* has infinitely many levels of intensity over a period of time.
- *Digital Signal* can have only a limited number of defined values.



**Figure 1.28 Comparison of analog and digital signals**

### 1.7.1.3 Periodic & Non-Periodic Signals

- The signals can take one of 2 forms: periodic or non-periodic.

#### (a) Periodic Signal

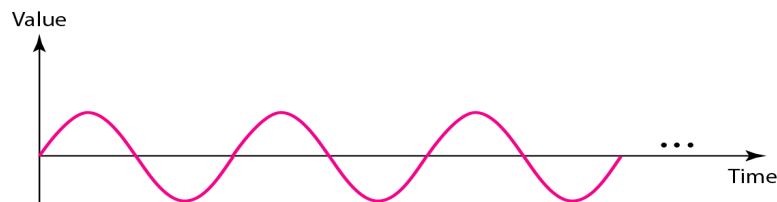
- Signals which repeat itself after a fixed time period are called Periodic Signals.
- The completion of one full pattern is called a cycle.

#### (b) Non-Periodic Signal

- Signals which do not repeat itself after a fixed time period are called Non-Periodic Signals.

#### 1.7.1.3.1 Periodic Analog Signal

- Analog Signal is continuously varying electromagnetic waves that may be propagated over a variety of media
- Generally, analog signals are used to represent analog data such as audio & video
- The simplest form of a periodic analog signal is a sine wave.



**Figure 1.29 A sign wave**

Sine wave can be represented by three parameters

- (i) Amplitude (A) - A max value of the signal over time (Volts))
- (ii) Frequency (f) - The rate (in cycles per seconds or Hertz) at which the signal repeats
- (iii) Period (T) - Is the amount of time it takes for one repetition  $T=1/f$
- (iv) Phase - A measure of the relative position in time within a single period of a signal

**Frequency and period are the inverse of each other.**

$$f = \frac{1}{T} \quad \text{and} \quad T = \frac{1}{f}$$

#### Example 7.1

The power we use at home has a frequency of 60 Hz. The period of this sine wave can be determined as follows:

$$T = \frac{1}{f} = \frac{1}{60} = 0.0166 \text{ s} = 0.0166 \times 10^3 \text{ ms} = 16.6 \text{ ms}$$



**Example 7.2**

The period of a signal is 100 ms. What is its frequency in kilohertz?

**Solution:**

First we change 100 ms to seconds, and then we calculate the frequency from the period (1 Hz = 10<sup>-3</sup> kHz).

$$100 \text{ ms} = 100 \times 10^{-3} \text{ s} = 10^{-1} \text{ s}$$

$$f = \frac{1}{T} = \frac{1}{10^{-1}} \text{ Hz} = 10 \text{ Hz} = 10 \times 10^{-3} \text{ kHz} = 10^{-2} \text{ kHz}$$

**Frequency**

- Frequency is the rate of change with respect to time.
- Change in a short span of time means high frequency.
- Change over a long span of time means low frequency.
- If a signal does not change at all, its frequency is zero.
- If a signal changes instantaneously, its frequency is infinite.
- Phase describes the position of the waveform relative to time 0.

**Example 7.3**

A sine wave is offset 1/6 cycle with respect to time 0. What is its phase in degrees and radians?

**Solution:**

We know that 1 complete cycle is 360°. Therefore, 1/6 cycle is

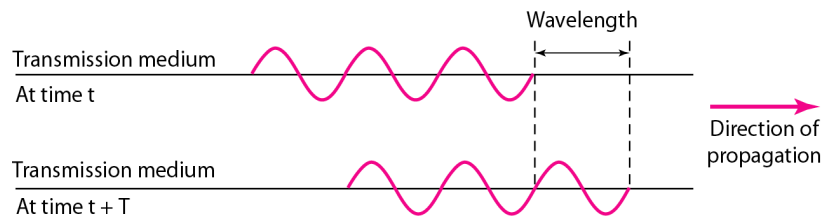
$$\frac{1}{6} \times 360 = 60^\circ = 60 \times \frac{2\pi}{360} \text{ rad} = \frac{\pi}{3} \text{ rad} = 1.046 \text{ rad}$$

**Wavelength and period**

Wave length binds the period or the frequency of the simple sine wave to the propagation speed of the medium.

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

$$= \text{Propagation speed} / \text{Frequency}$$



**Signals and Communication**

- A single-frequency sine wave is not useful in data communications
- We need to send a composite signal, a signal made of many simple sine waves.
- According to Fourier analysis, any composite signal is a combination of simple sine waves with different frequencies, amplitudes, and phases.

**Composite Signals and Periodicity**

- If the composite signal is periodic, the decomposition gives a series of signals with discrete frequencies.
- If the composite signal is non-periodic, the decomposition gives a combination of sine waves with continuous frequencies.

**Bandwidth and Signal Frequency**

- The bandwidth of a composite signal is the difference between the highest and the lowest frequencies contained in that signal.

**Example 7.4**

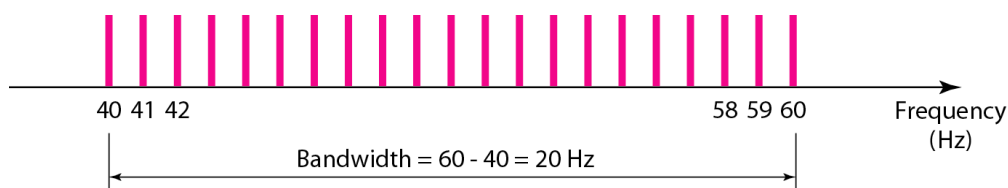
If a periodic signal is decomposed into five sine waves with frequencies of 100, 300, 500, 700, and 900 Hz, what is its bandwidth? Draw the spectrum, assuming all components have maximum amplitude of 10 V.

**Solution:**

Let  $f_h$  be the highest frequency,  $f_l$  the lowest frequency, and  $B$  the bandwidth. Then

$$B = f_h - f_l \Rightarrow 20 = 60 - f_l \Rightarrow f_l = 60 - 20 = 40 \text{ Hz}$$

The spectrum has only five spikes, at 100, 300, 500, 700, and 900 Hz (see Figure 1.30).



**Figure 1.30** The bandwidth for Example 7.4

**Example 7.5**

A periodic signal has a bandwidth of 20 Hz. The highest frequency is 60 Hz. What is the lowest frequency? Draw the spectrum if the signal contains all frequencies of the same amplitude.

**Solution:**

Let  $f_h$  be the highest frequency,  $f_l$  the lowest frequency, and  $B$  the bandwidth. Then

$$B = f_h - f_l \Rightarrow 20 = 60 - f_l \Rightarrow f_l = 60 - 20 = 40 \text{ Hz}$$

The spectrum contains all integer frequencies. We show this by a series of spikes (see Figure 1.31).

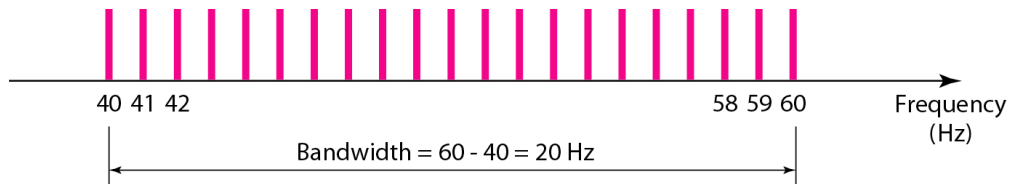


Figure 1.31 The bandwidth for Example 7.5

**Example 7.6**

A non-periodic composite signal has a bandwidth of 200 kHz, with a middle frequency of 140 kHz and peak amplitude of 20 V. The two extreme frequencies have amplitude of 0. Draw the frequency domain of the signal.

**Solution:**

The lowest frequency must be at 40 kHz and the highest at 240 kHz. Figure 3.15 shows the frequency domain and the bandwidth.

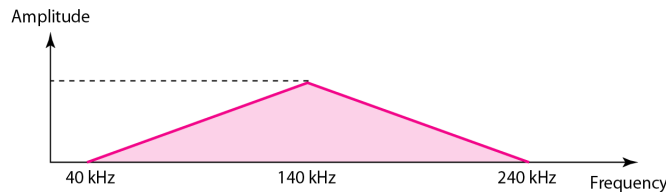


Figure 1.32 The bandwidth for Example 7.6

**1.7.2 Digital Signals**

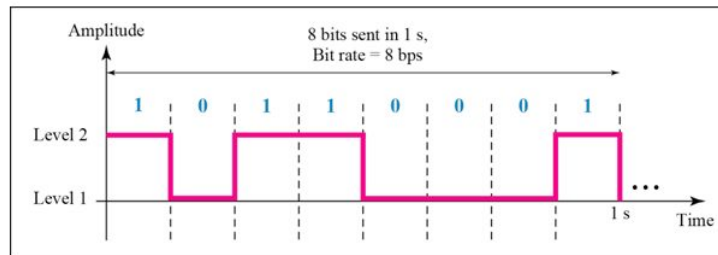
Digital Signal is a sequence of voltage pulses that may be transmitted over a wire medium. For example, constant positive voltage level represents binary 1 and a constant negative voltage level may represent binary 0. Generally, digital signals are used to represent digital data such as text.

Digital technology generates stores and processes data in terms of two states (0's and 1's). Each of these state digits is referred to as a bit. (8bits = 1 byte).

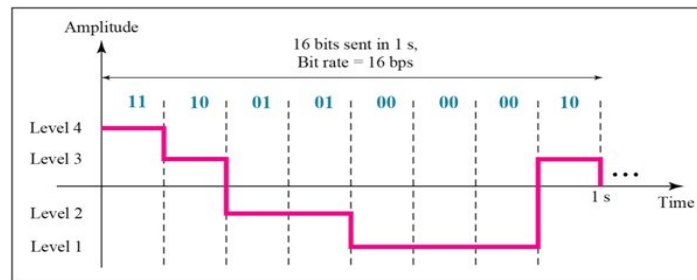
**Bit rate**

Most digital signals are non-periodic, and thus period and frequency are not appropriate characteristics.

- Bit rate is used to describe digital signals.
- The bit rate is the number of bits sent in 1 second, expressed in bits per second (bps).
- Bit interval is used (instead of period) to refer to the time required to send one single bit.



**Figure 1.33** A digital signal with two signal levels



**Figure 1.34** A digital signal with four signal levels

### Example 7.7

A digital signal has eight levels. How many bits are needed per level?

#### Solution:

We calculate the number of bits from the formula

$$\text{Number of bits per level} = \log_2 8 = 3$$

Each signal level is represented by 3 bits.

### Example 7.8

Assume we need to download text documents at the rate of 100 pages per minute. What is the required bit rate of the channel?

#### Solution:

A page is an average of 24 lines with 80 characters in each line. If we assume that one character requires 8 bits, the bit rate is

$$100 \times 24 \times 80 \times 8 = 1,636,000 \text{ bps} = 1.636 \text{ Mbps}$$

### Example 7.9

A digitized voice channel is made by digitizing a 4-kHz bandwidth analog voice signal. We need to sample the signal at twice the highest frequency (two samples per hertz). We assume that each sample requires 8 bits. What is the required bit rate?

#### Solution:

The bit rate can be calculated as

$$2 \times 4000 \times 8 = 64,000 \text{ bps} = 64 \text{ kbps}$$

**Example 7.10**

**What is the bit rate for high-definition TV (HDTV)?**

**Solution:**

HDTV uses digital signals to broadcast high quality video signals. The HDTV Screen is normally a ratio of 16: 9 (in contrast to 4: 3 for regular TV), which means the screen is wider. There are 1920 by 1080 pixels per screen, and the screen is renewed 30 times per second. Twenty-four bits represents one color pixel. We can calculate the bit rate as

$$1920 \times 1080 \times 30 \times 24 = 1,492,992,000 \text{ or } 1.5 \text{ Gbps}$$

The TV stations reduce this rate to 20 to 40 Mbps through compression.

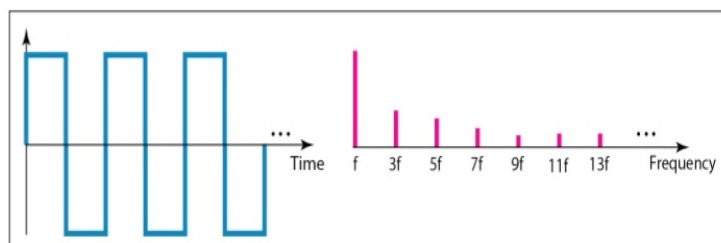
**Bit Length**

The bit length is the distance one bit occupies on the transmission medium.

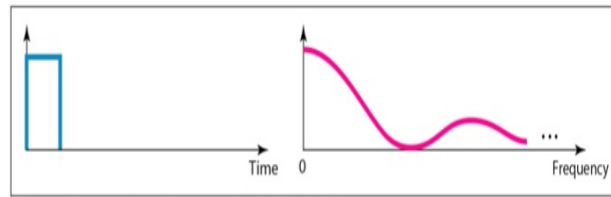
$$\text{Bit length} = \text{propagation speed} \times \text{bit duration}$$

**1.7.2.1 Digital Signal as a Composite Analog Signal**

- A digital signal is a composite analog signal.
- A digital signal, in the time domain, comprises connected vertical and horizontal line segments.
  - a) A vertical line in the time domain means a frequency of infinity (sudden change in time);
  - b) A horizontal line in the time domain means a frequency of zero (no change in time).
- Fourier analysis can be used to decompose a digital signal.
  - (i) If the digital signal is periodic, the decomposed signal has a frequency domain representation with an infinite bandwidth and discrete frequencies (Figure 1.35).
  - (ii) If the digital signal is non-periodic, the decomposed signal has a frequency domain representation with an infinite bandwidth and continuous frequencies (Figure 1.36).



**Figure 1.35** The time and frequency domains of periodic digital signals



*Figure 1.36 The time and frequency domains of non-periodic digital signals*

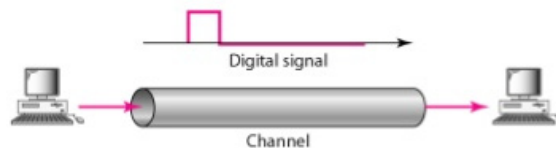
### 1.7.3 Transmission of Digital Signals

Two methods for transmitting digital signals are

- 1) Baseband transmission
- 2) Broadband transmission (using modulation).

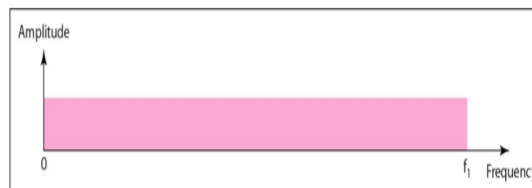
#### 1.7.3.1 Baseband Transmission

Baseband transmission means sending a digital signal over a channel without changing the digital signal to an analog signal (Figure 1.37).

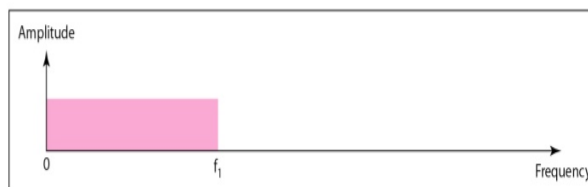


*Figure 1.37 Baseband transmission*

- Baseband transmission requires that we have a low-pass channel.
- Low-pass channel means a channel with a bandwidth that starts from zero.
- For example, we can have a dedicated medium with a bandwidth constituting only one channel.



*Figure 1.38 Low-pass channel with wide bandwidth*

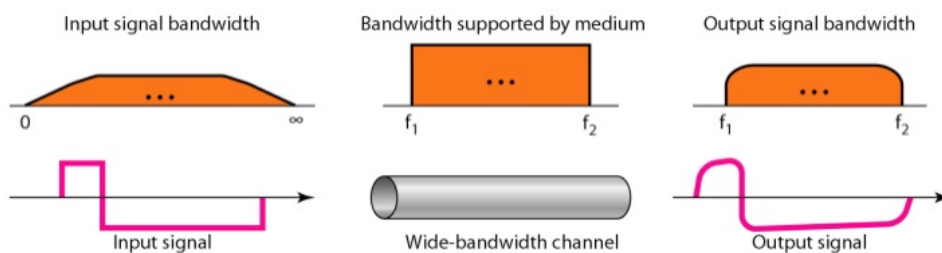


*Figure 1.39 Low-pass channel with narrow bandwidth*

- Two cases of a baseband communication are
  - Case 1: Low-pass channel with a wide bandwidth (Figure 1.38)
  - Case 2: Low-pass channel with a limited bandwidth (Figure 1.39)

**Case 1: Low-Pass Channel with Wide Bandwidth**

- To preserve the shape of a digital signal, we need to send the entire spectrum i.e. the continuous range of frequencies between zero and infinity.
- This is possible if we have a dedicated medium with an infinite bandwidth between the sender and receiver.
- If we have a medium with a very wide bandwidth, 2 stations can communicate by using digital signals with very good accuracy.
- Although the output signal is not an exact replica of the original signal, the data can still be deduced from the received signal.



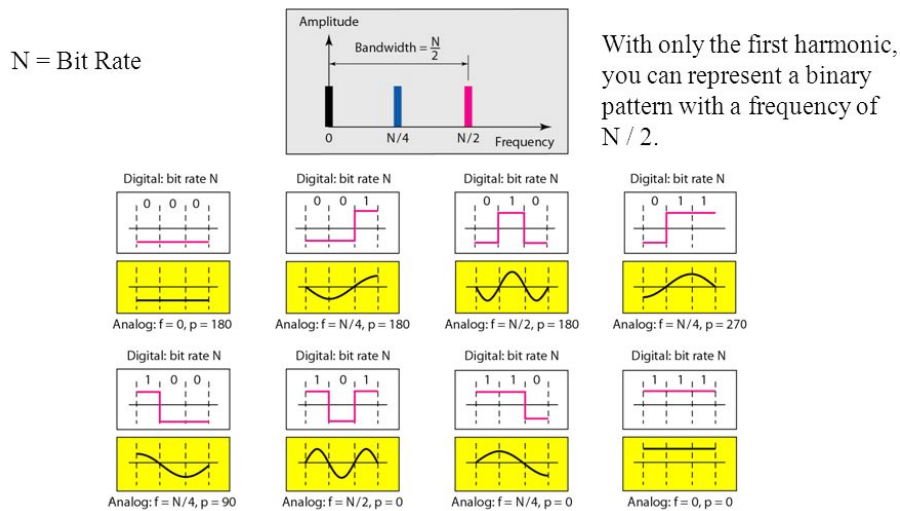
**Figure 1.40 Baseband transmission using a dedicated medium**

**Case 2: Low-Pass Channel with Limited Bandwidth**

- In a low-pass channel with limited bandwidth, we approximate the digital signal with an analog signal.
- The level of approximation depends on the bandwidth available.

**A) Rough Approximation**

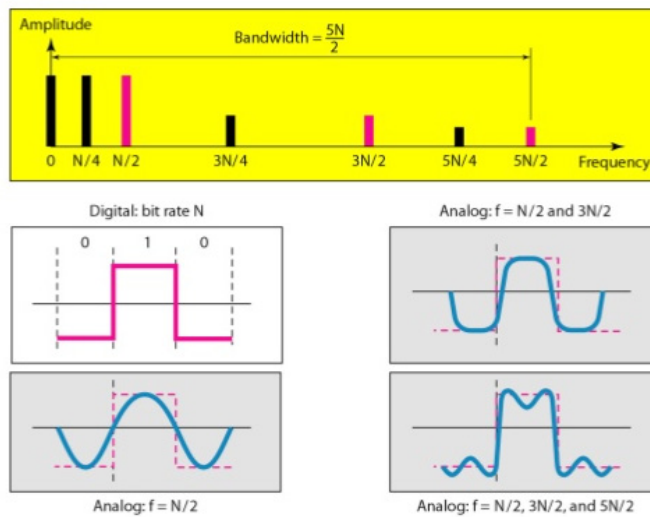
- Assume that we have a digital signal of bit rate  $N$  (Figure 1.41).
- If we want to send analog signals to roughly simulate this signal, we need to consider the worst case, a maximum number of changes in the digital signal.
- This happens when the signal carries the sequence 01010101... or 10101010....
- To simulate these two cases, we need an analog signal of frequency  $f = N/2$ .
- Let 1 be the positive peak value and 0 be the negative peak value.
- We send 2 bits in each cycle; the frequency of the analog signal is one-half of the bit rate, or  $N/2$ .
- This rough approximation is referred to as using the first harmonic ( $N/2$ ) frequency. The required bandwidth is



**Figure 1.41** Rough approximation of a digital signal using the first harmonic for worst case

**B) Better Approximation**

- To make the shape of the analog signal look more like that of a digital signal, we need to add more harmonics of the frequencies (Figure 1.42).
- We can increase the bandwidth to  $3N/2$ ,  $5N/2$ ,  $7N/2$ , and so on.
- In baseband transmission, the required bandwidth is proportional to the bit rate; If we need to send bits faster, we need more bandwidth



**Figure 1.42** Simulating a digital signal with three first harmonics

Bit Rate	Harmonic 1	Harmonics 1,3	Harmonics 1, 3, 5
$n = 1 \text{ kbps}$	$B=500\text{Hz}$	$B= 1.5 \text{ kHz}$	$B= 2.5 \text{ kHz}$
$n = 10 \text{ kbps}$	$B=5 \text{ kHz}$	$B= 15\text{kHz}$	$B= 25\text{kHz}$
$n = 100 \text{ kbps}$	$B= 50\text{kHz}$	$B = 150 \text{ kHz}$	$B = 250 \text{ kHz}$

**Table 1.1** Bandwidth requirements



**Example 7.11**

What is the required bandwidth of a low-pass channel if we need to send 1 Mbps by using baseband transmission?

**Solution:**

The answer depends on the accuracy desired.

- The minimum bandwidth, a rough approximation, is  $B = \text{bit rate} / 2$ , or 500 kHz. We need a low-pass channel with frequencies between 0 and 500 kHz.
- A better result can be achieved by using the first and the third harmonics with the required bandwidth  $B = 3 \times 500 \text{ kHz} = 1.5 \text{ MHz}$
- Still a better result can be achieved by using the first, third and fifth harmonics with  $B = 5 \times 500 \text{ kHz} = 2.5 \text{ MHz}$

**Example 7.12**

We have a low-pass channel with bandwidth 100 kHz. What is the maximum bit rate of this channel?

**Solution:**

The maximum bit rate can be achieved if we use the first harmonic. The bit rate is 2 times the available bandwidth, or 200 kbps.

**1.7.3.2 Broadband Transmission (Using Modulation)**

- Broadband transmission or modulation means changing the digital signal to an analog signal for transmission.
- Modulation allows us to use a band pass channel (Figure 1.43).
- Bandpass channel means a channel with a bandwidth that does not start from zero.
- This type of channel is more available than a low-pass channel.



**Figure 1.43 Bandwidth of a band pass channel**

- If the available channel is a band pass channel, we cannot send the digital signal directly to the channel; we need to convert the digital signal to an analog signal before transmission (Figure 1.44).

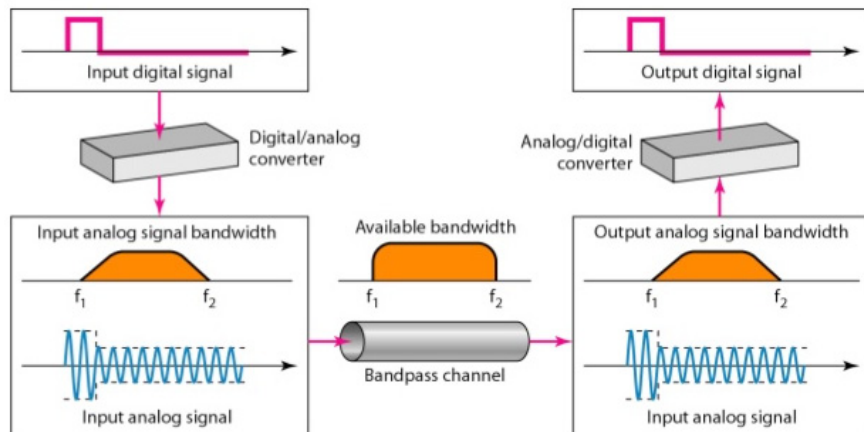


Figure 1.44 Modulation of a digital signal for transmission on a band pass channel

### 1.7.4 Transmission Impairment

- Signals travel through transmission media, which are not perfect.
- The imperfection causes signal-impairment.
- This means that signal at beginning of the medium is not the same as the signal at end of medium.
- What is sent is not what is received.
- Three causes of impairment are ;
  - (i) Attenuation
  - (ii) Distortion &
  - (iii) Noise.

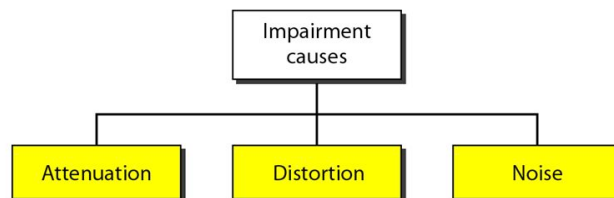


Figure 1.45 Causes of impairment

#### 1.7.4.1 Attenuation

- As signal travels through the medium, its strength decreases as distance increases. This is called attenuation.
- As the distance increases, attenuation also increases.
- For example:
  - Voice-data becomes weak over the distance & loses its contents beyond a certain distance.
- To compensate for this loss, amplifiers are used to amplify the signal.

**Decibel**

- The decibel (dB) measures the relative strengths of 2 signals or one signal at 2 different points.
- The decibel is negative if a signal is attenuated. The decibel is positive if a signal is amplified.
- $\text{dB} = 10 \log_{10} P_2/P_1$
- Variables  $P_1$  and  $P_2$  are the powers of a signal at points 1 and 2, respectively.
- To show that a signal has lost or gained strength, engineers use the unit of decibel.

**Example 7.13**

Suppose a signal travels through a transmission medium and its power is reduced to one-half. This means that  $P_2 = \sim P_1$ . In this case, the attenuation (loss of power) can be calculated as

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

A loss of 3 dB (-3 dB) is equivalent to losing one-half the power.

**Example 7.14**

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

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

**Example 7.15**

Sometimes the decibel is used to measure signal power in milli watts. In this case, it is referred to as dBm and is calculated as  $\text{dBm} = 10 \log_{10} P_m$ , where  $P_m$  is the power in milli watts. Calculate the power of a signal with  $\text{dBm} = -30$ .

**Solution:**

We can calculate the power in the signal as

$$\begin{aligned} \text{dB}_m &= 10 \log_{10} P_m = -30 \\ \log_{10} P_m &= -3 & P_m &= 10^{-3} \text{ mW} \end{aligned}$$

**Example 7.16**

The loss in a cable is usually defined in decibels per kilometer (dB/km). If the signal at beginning of a cable with  $-0.3 \text{ dB/km}$  has a power of 2 mW, what is the power of the signal at 5 km?

**Solution:**

The loss in the cable in decibels is  $5 \times (-0.3) = -1.5$  dB. We can calculate the power as

$$\text{dB} = 10 \log_{10} \frac{P_2}{P_1} = -1.5$$

$$\frac{P_2}{P_1} = 10^{-0.15} = 0.71$$

$$P_2 = 0.71P_1 = 0.7 \times 2 = 1.4 \text{ mW}$$

**1.7.4.2 Distortion**

- Distortion means that the signal changes its form or shape.
- Distortion can occur in a composite signal made of different frequencies.
- Different signal-components
  - Have different propagation speed through a medium.
  - Have different delays in arriving at the final destination.
- Differences in delay create a difference in phase if delay is not same as the period-duration.
- Signal-components at the receiver have phases different from what they had at the sender.
- The shape of the composite signal is therefore not the same.

**1.7.4.3 Noise**

- Noise is defined as an unwanted data (Figure 3.30).
- In other words, noise is the external energy that corrupts a signal.
- Due to noise, it is difficult to retrieve the original data/information.
- Four types of noise:
  - (i) Thermal Noise: It is random motion of electrons in wire which creates extra signal not originally sent by transmitter.
  - (ii) Induced Noise: Induced noise comes from sources such as motors & appliances. These devices act as a sending-antenna. The transmission-medium acts as the receiving-antenna.
  - (iii) Crosstalk: Crosstalk is the effect of one wire on the other. One wire acts as a sending-antenna and the other as the receiving-antenna.
  - (iv) Impulse Noise: Impulse Noise is a spike that comes from power-lines, lightning, and so on. (spike is a signal with high energy in a very short time).

**Signal-to-Noise Ratio (SNR)**

- SNR is used to find the theoretical bit-rate limit.
- SNR is defined as

SNR = Average signal power / Average noise power

- SNR is actually the ratio of what is wanted (signal) to what is not wanted (noise).
- A high-SNR means the signal is less corrupted by noise.
- A low-SNR means the signal is more corrupted by noise.
- Because SNR is the ratio of 2 powers. It is often described in decibel units as SNRdB.
- SNRdB can be defined as

$$\text{SNRdB} = 10 \log_{10} \text{SNR}$$

### 1.7.5 Data Rate Limits

- Data-rate depends on 3 factors:
  - (i) Bandwidth available
  - (ii) Level of the signals
  - (iii) Quality of channel (the level of noise)
- Two theoretical formulas can be used to calculate the data-rate:
  - (a) Nyquist for a noiseless channel and
  - (b) Shannon for a noisy channel.

#### 1.7.5.1 Noiseless Channel: Nyquist Bit Rate

- For a noiseless channel, the Nyquist bit-rate formula defines the theoretical maximum bit-rate as

$$\text{Bit Rate} = 2 \times \text{Bandwidth} \times \log_2 \text{Levels}$$

- where
  - (a) bandwidth = bandwidth of the channel
  - (b) Levels = number of signal-levels used to represent data
  - (c) Bit Rate = bit rate of channel in bps
- According to the formula,
  - By increasing number of signal-levels, we can increase the bit-rate.
  - Although the idea is theoretically correct, practically there is a limit.
  - When we increase the number of signal-levels, we impose a burden on the receiver.
  - If no. of levels in a signal is 2, the receiver can easily distinguish b/w 0 and 1.
  - If no. of levels is 64, the receiver must be very sophisticated to distinguish b/w 64 different levels.
  - In other words, increasing the levels of a signal reduces the reliability of the system.

**Example 7.17**

Consider a noiseless channel with a bandwidth of 3000 Hz transmitting a signal with two signal levels. What is the maximum bit rate?

**Solution:**

$$\text{Bit rate} = 2 * 3000 * \log_2 2 = 6000 \text{ bps}$$

**Example 7.18**

Consider the same noiseless channel transmitting a signal with four signal levels (for each level, we send 2 bits). What is the maximum bit rate?

**Solution:**

$$\text{Bit rate} = 2 * 3000 * \log_2 4 = 12000 \text{ bps}$$

**Example 7.19**

We need to send 265 kbps over a noiseless channel with a bandwidth of 20 kHz. How many signal levels do we need?

**Solution:**

We can use the Nyquist formula as

$$265,000 = 2 * 20,000 * \log_2 \text{Levels}$$

$$\log_2 \text{Levels} = 6.625$$

$$\text{Levels} = 2^{6.625}$$

$$= 98.7 \text{ Levels}$$

**1.7.5.2 Noisy Channel: Shannon Capacity**

- In reality, we cannot have a noiseless channel; the channel is always noisy.
- For a noisy channel, the Shannon capacity formula defines the theoretical maximum bit-rate.

$$\text{Capacity} = \text{Band width} * \log_2 (1 + \text{SNR})$$

- where
  - (a) bandwidth = bandwidth of channel in bps.
  - (b) SNR = signal-to-noise ratio
  - (c) Capacity = capacity of channel in bps.
- This formula does not consider the no. of levels of signals being transmitted (as done in the Nyquist bit rate).
- This means that no matter how many levels we have, we cannot achieve a data-rate higher than the capacity of the channel.

- In other words, the formula defines a characteristic of the channel, not the method of transmission.

**Example 7.20**

Let's calculate the theoretical highest bit rate of a regular telephone line. A telephone line normally has a bandwidth of 3000. The signal-to-noise ratio is usually 3162. What is the channel capacity?

**Solution:**

$$\begin{aligned} C &= B \log_2 (1 + \text{SNR}) = 3000 \log_2 (1 + 3162) = 3000 \log_2 3163 \\ &= 3000 * 11.62 = 34,860 \text{ bps} \end{aligned}$$

**Example 7.21**

The signal-to-noise ratio is often given in decibels. Assume that SNR<sub>dB</sub> = 36 and the channel bandwidth is 2 MHz. What is the theoretical channel capacity?

**Solution:**

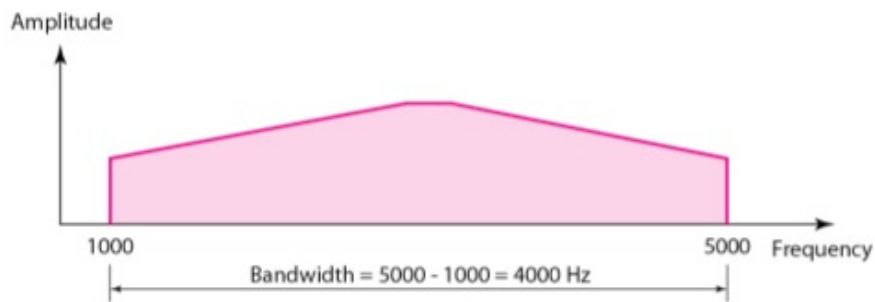
$$\begin{aligned} \text{SNR}_{\text{db}} &= 10 \log_{10} \text{SNR} \\ \rightarrow \text{SNR} &= 10^{\text{SNR}_{\text{db}}/10} \\ \rightarrow \text{SNR} &= 10^{3.6} \\ &= 3981 \\ C &= B \log_2 (1 + \text{SNR}) = 2 * 10^6 * \log_2 3982 \\ &= 24 \text{ Mbps} \end{aligned}$$

**1.7.6 Performance****1.7.6.1 Bandwidth**

- One characteristic that measures network-performance is bandwidth.
- Bandwidth of analog and digital signals is calculated in separate ways.

**Bandwidth of an Analog Signal (in Hz)**

- Bandwidth of an analog signal is expressed in terms of its frequencies.
- Bandwidth is defined as the range of frequencies that the channel can carry.
- It is calculated by the difference b/w the maximum frequency and the minimum frequency.



**Figure 1.46** *The band width of signals*

- In figure 1.46, the signal has a minimum frequency of  $F_1 = 1000\text{Hz}$  and maximum frequency of  $F_2 = 5000\text{Hz}$
- Hence, the bandwidth is given by  $F_2 - F_1 = 5000 - 1000 = 4000\text{ Hz}$

#### ***Bandwidth of a Digital Signal (in bps)***

- Bandwidth refers to the number of bits transmitted in one second in a channel (or link).
- For example: The bandwidth of a Fast Ethernet is a maximum of 100 Mbps. This means that this network can send 100 Mbps.

#### ***Relationship between Analog and Digital Signal***

- There is an explicit relationship between the bandwidth in hertz and bandwidth in bits per seconds.
- Basically, an increase in bandwidth in hertz means an increase in bandwidth in bits per second.
- The relationship depends on
  - a) baseband transmission or
  - b) transmission with modulation

#### ***1.7.6.2 Throughput***

- The throughput is a measure of how fast we can actually send data through a network.
- Although, bandwidth in bits per second and throughput seem the same, they are actually different.
- A link may have a bandwidth of  $B$  bps, but we can only send  $T$  bps through this link with  $T$  always less than  $B$ .
- In other words,
  - a) The bandwidth is a potential measurement of a link.
  - b) The throughput is an actual measurement of how fast we can send data. For example:
- We may have a link with a bandwidth of 1 Mbps, but the devices connected to the end of the link may handle only 200 kbps.
- This means that we cannot send more than 200 kbps through this link.



**Example 7.22**

A network with bandwidth of 10 Mbps can pass only an average of 12,000 frames per minute with each frame carrying an average of 10,000 bits. What is the throughput of this network?

**Solution:**

We can calculate the throughput as

$$\text{Throughput} = \frac{12,000 \times 10,000}{60} = 2 \text{ Mbps}$$

**1.7.6.3 Latency (Delay)**

- The latency defines how long it takes for an entire message to completely arrive at the destination from the time the first bit is sent out from the source.

$$\text{Latency} = \text{Propagation time} + \text{Transmission time} + \text{Queuing time} + \text{Processing Delay}$$

**Propagation Time**

- Propagation time is defined as the time required for a bit to travel from source to destination.
- Propagation time is given by
 
$$\text{Propagation time} = \text{Distance} / \text{Propagation Speed}$$
- Propagation speed of electromagnetic signals depends on medium and frequency of the signal.

**Transmission Time**

- The time required for transmission of a message depends on size of the message and bandwidth of the channel.
- The transmission time is given by

$$\text{Transmission Time} = \text{Message Size} / \text{Band Width}$$

**Queuing Time**

- Queuing-time is the time needed for each intermediate-device to hold the message before it can be processed. Intermediate device may be a router or a switch.
- The queuing-time is not a fixed factor. This is because
  - (a) Queuing-time changes with the load imposed on the network.
  - (b) When there is heavy traffic on the network, the queuing-time increases.
- An intermediate-device
  - (a) queues the arrived messages and

(b) processes the messages one by one.

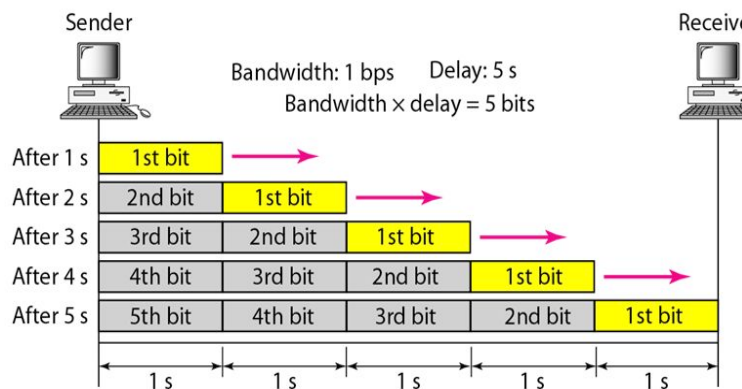
- If there are many messages, each message will have to wait.

**Processing Delay**

- Processing delay is the time taken by the routers to process the packet header.

**1.7.6.4 Bandwidth - Delay Product**

- Two performance-metrics of a link are
  - a) Bandwidth
  - b) Delay
- The bandwidth-delay product is very important in data-communications.
- Let us elaborate on this issue, using 2 hypothetical cases as examples.

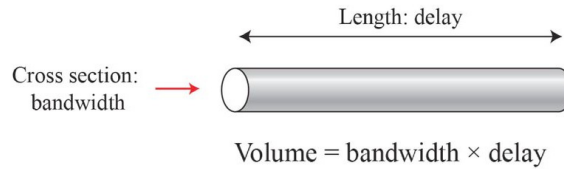


**Figure 1.47 Filling the link with bits for case 1**

**Case 1: Figure 1.47 shows case 1.**

- Let us assume,
  - Bandwidth of the link = 4 bps
  - Delay of the link = 5s.
- From the figure 1.47, bandwidth-delay product is  $5 \times 5 = 25$ .
  - (i) Thus, there can be maximum 25 bits on the line.
  - (ii) At each second, there are 5 bits on the line, thus the duration of each bit is 0.20s.
- The above 2 cases (i & ii) show that the **bandwidth × delay** is the number of bits that can fill the link.
- This measurement is important if we need to **send data in bursts and wait for the acknowledgment of each burst**.
- To use the maximum capability of the link
  - a) We need to make the burst-size as  $(2 \times \text{bandwidth} \times \text{delay})$ .

- b) We need to fill up the full-duplex channel (two directions)
- Amount ( $2 \times \text{bandwidth} \times \text{delay}$ ) is the number of bits that can be in transition at any time (Figure 1.48).



**Figure 1.48** Concept of bandwidth delay product

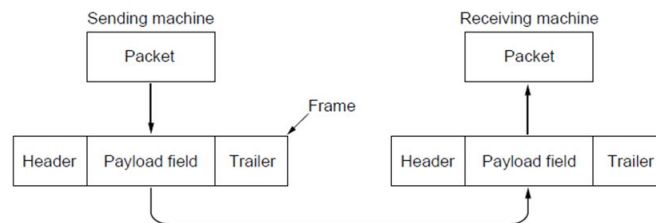
### 1.7.6.5 Jitter

- Another performance issue that is related to delay is jitter.
- We can say that jitter is a problem
  - a) if different packets of data encounter different delays and
  - b) if the application using the data at the receiver site is time-sensitive (for ex: audio/video).
- For example: If the delay for the first packet is 20 ms the delay for the second is 45 ms and the delay for the third is 40 ms then the real-time application that uses the packets suffers from jitter.

## 1.8. INTRODUCTION TO DATA LINK LAYER

The data link control (DLC) deals with procedures for communication between two adjacent nodes i.e. node-to-node communication. The data link layer has a number of specific functions it can carry out. Some of the functions are;

- (i) Providing a well defined service interface to the network
- (ii) Dealing with transmission errors
- (iii) Regulating the flow of data ( to synchronize the sender and the receiver)



**Figure 1.49** Relationship between packets and frames

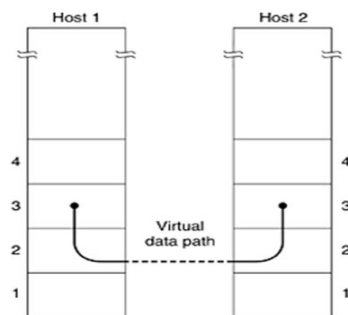
To obtain these functions, the data link layer takes the packets it gets from the network layer and encapsulates them into frames for transmission. Each frame contains a frame header, a payload field for holding the packet and a frame trailer. Figure 1.49 explains the same. The data link layer is having some issues. They are,

- (i) Services provided to the network layer
- (ii) Framing
- (iii) Error control
- (iv) Flow control

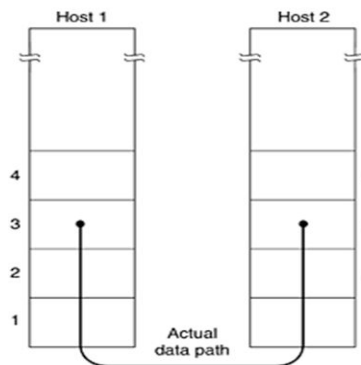
### 1.8.1 Services Provided to the Network Layer

The principal service is transferring data from the network layer on the source machine to the network layer on the destination machine. On the source machine is an entity, call it a process, in the network layer that hands some bits to the data link layer for transmission to the destination.

The job of the data link layer is to transmit the bits to the destination machine so that they can be handed over to the network layer, which is shown in the below figure.



**Figure 1.50** *Virtual communication*



**Figure 1.51** *Actual communication*

The data link layer can be designed to offer various services. The actual services offered can vary from system to system. Three reasonable possibilities that are commonly provided are

- (a) Unacknowledged connectionless service
- (b) Acknowledged connectionless service
- (c) Acknowledged connection-oriented service.

#### **(a) Unacknowledged connectionless service**

It contains the source machine send independent frames to the destination machine without having the acknowledgement from the destination machine. No logical connection is established

beforehand or released afterward. If a frame is lost due to noise on the line, no attempt is made to detect the loss or recover from it in the data link layer. Most LANs use unacknowledged connectionless service in the data link layer.

**(b) Acknowledged connectionless service**

When this service is offered, there are still no logical connections used, but each frame sent is individually acknowledged. Here the sender knows whether a frame has arrived correctly. If it has not arrived within a specified time interval, it can be sent again. This service is useful over unreliable channels, such as wireless systems.

**(c) Acknowledged connection-oriented service**

Each frame sent over the connection is numbered, and the data link layer guarantees that each frame sent is indeed received. Furthermore, it guarantees that each frame is received exactly once and that all frames are received in the right order.

## 1.8.2 Framing

The data link layer to break the bit stream up into discrete frames and compute the checksum for each frame. 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.

A frame is a group of bits. Framing means organizing the bits into a frame that are carried by the physical layer. The data-link-layer needs to form frames, so that each frame is distinguishable from another. Framing separates a message from other messages by adding sender-address & destination-address. The destination-address defines where the packet is to go. The sender-address helps the recipient acknowledge the receipt.

Here we may have a question that, why the whole message is not packed in one frame? The reason behind that is the large frame makes flow and error-control very inefficient. Even a single-bit error requires the re-transmission of the whole message. When a message is divided into smaller frames, a single-bit error affects only that small frame.

### 1.8.2.1 Frame Size

Two types of frames are;

i) Fixed Size Framing

- There is no need for defining boundaries of frames; the size itself can be used as a delimiter.
- For example: ATM WAN uses frames of fixed size called cells.

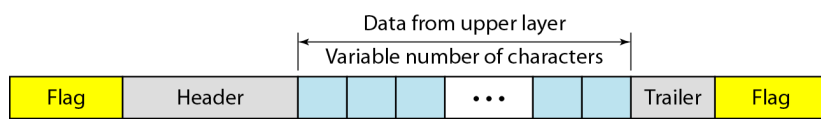
ii) Variable Size Framing

- We need to define the end of the frame and the beginning of the next frame.
- Two approaches are used;

- (a) Character-oriented approach
- (b) Bit-oriented approach.

### 1.8.2.2. Character Oriented Framing

Data to be carried are 8-bit characters from a coding system such as ASCII (Figure 1.52). The header and the trailer are also multiples of 8 bits. Header carries the source and destination addresses and other control information. Trailer carries error-detection or error-correction redundant bits. To separate one frame from the next frame, an 8-bit (1-byte) flag is added at the beginning and the end of a frame. The flag is composed of protocol-dependent special characters. The flag signals the start or end of a frame.

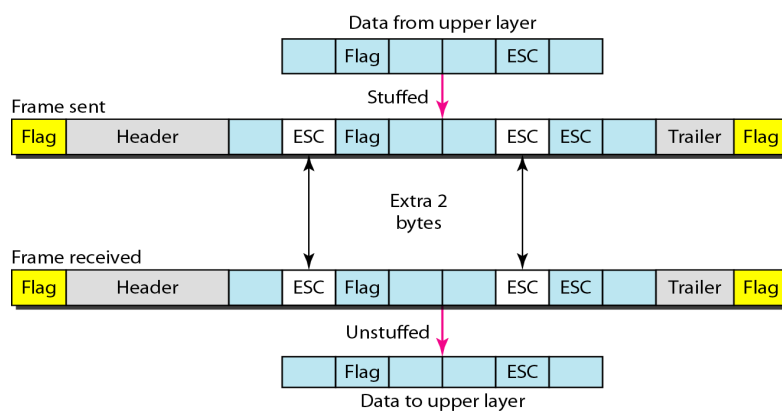


**Figure 1.52** A frame in a character-oriented protocol

Character-oriented framing is suitable when only text is exchanged by the data-link-layers. When other type of information like audio or video is going to be sent, the flag pattern may also look like a part of the information. If the flag-pattern appears in the data-section, the receiver might think that it has reached the end of the frame.

In order to solve this issue a byte-stuffing (character stuffing) is used. In byte stuffing, a special byte is added to the data-section of the frame when there is a character with the same pattern as the flag. The data-section is stuffed with an extra byte. This byte is called the escape character (ESC), which has a predefined bit pattern. When a receiver encounters the ESC character, the receiver **removes ESC character from the data-section and treats the next character as data, not a delimiting flag.**

What happens if the text contains one or more escape characters followed by a flag? In this case, the receiver removes the escape character but keeps the flag, which is incorrectly interpreted as the end of the frame. If escape characters are present as a part of the text, it must also be marked by another escape character as shown in Figure 1.53.



**Figure 1.53** Byte stuffing and unstuffing

In short, byte stuffing is the process of adding one extra byte whenever there is a flag or escape character in the text.

### 1.8.2.3 Bit Oriented Framing

The data-section of a frame is a sequence of bits to be interpreted by the upper layer as text, audio, video, and so on. However, in addition to headers and trailers, we need a delimiter to separate one frame from the other. Most protocols use a special 8-bit pattern flag 01111110 as the delimiter to define the beginning and the end of the frame (Figure 1.54).

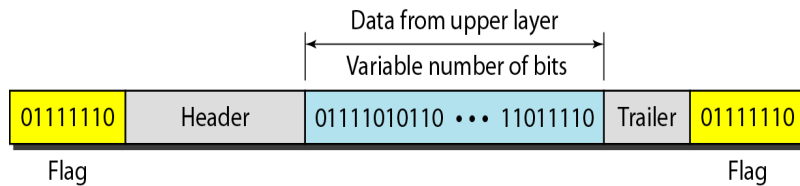


Figure 1.54 A frame in a bit-oriented protocol

If the flag-pattern appears in the data-section, the receiver might think that it has reached the end of the frame. To solve the confusion at the receiver side a bit-stuffing is used. In bit stuffing, if a 0 and five consecutive 1 bits are encountered, an extra 0 is added. This extra stuffed bit is eventually removed from the data by the receiver as shown in figure 1.55. This guarantees that the flag field sequence does not inadvertently appear in the frame.

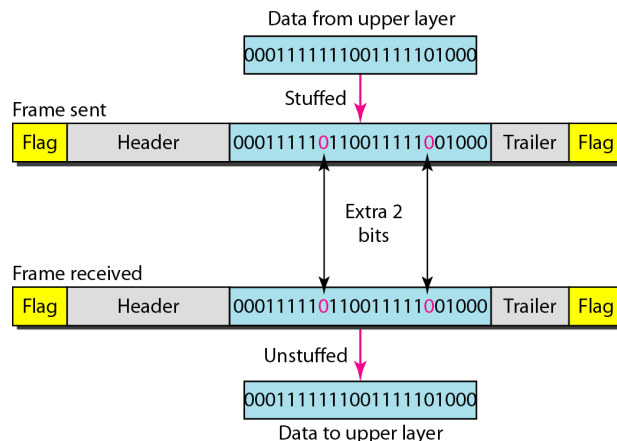


Figure 1.55 Bit stuffing and unstuffing

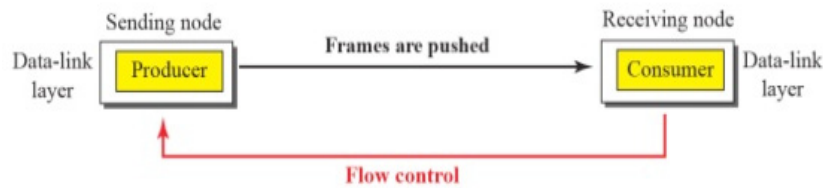
In short, bit stuffing is the process of adding one extra 0 whenever five consecutive 1s follow a 0 in the data, so that the receiver does not mistake the pattern 0111110 for a flag.

## 1.8.3 Flow Control

One of the responsibilities of the DLC sublayer is flow and error control at the data-link layer.

### Flow Control

Whenever an entity produces items and another entity consumes them, there should be a balance between production and consumption rates. If the items are produced faster than they can be consumed, the consumer can be overwhelmed and may need to discard some items. We need to prevent losing the data items at the consumer site.



**Figure 1.56** Flow control at the data link layer

At the sending node, the data-link layer tries to push frames toward the data-link layer at the receiving node (Figure 1.56). If the receiving node cannot process and deliver the packet to its network at the same rate that the frames arrive, it becomes overwhelmed with frames. Here, flow control can be feedback from the receiving node to the sending node to stop or slow down pushing frames.

### Buffers

Flow control can be implemented by using buffer. A buffer is a set of memory locations that can hold packets at the sender and receiver. Normally, two buffers can be used.

- (i) First buffer at the sender.
- (ii) Second buffer at the receiver.

The flow control communication can occur by sending signals from the consumer to the producer. When the buffer of the receiver is full, it informs the sender to stop pushing frames.

## 1.8.4 Error Control

Error-control includes both error-detection and error-correction. Error-control allows the receiver to inform the sender of any frames lost/damaged in transmission. **A CRC is added to the frame header by the sender and checked by the receiver.** At the data-link layer, error control is normally implemented using one of the following two methods.

- (i) **First method:** If the frame is corrupted, it is discarded; If the frame is not corrupted, the packet is delivered to the network layer. This method is used mostly in wired LANs such as Ethernet.
- (ii) **Second method:** If the frame is corrupted, it is discarded; If the frame is not corrupted, an acknowledgment is sent to the sender. Acknowledgment is used for the purpose of both flow and error control.

### 1.8.4.1 Combination of Flow and Error Control

Flow and error control can be combined. The acknowledgment that is sent for flow control can also be used for error control to tell the sender the packet has arrived uncorrupted. The lack of acknowledgment means that there is a problem in the sent frame. A frame that carries an acknowledgment is normally called an ACK to distinguish it from the data frame.

### Connectionless and Connection-Oriented

A Data Link Control protocol can be either connectionless or connection-oriented.



(i) **Connectionless Protocol**

- Frames are sent from one node to the next without any relationship between the frames; each frame is independent.
- The term connectionless does not mean that there is no physical connection (transmission medium) between the nodes; it means that there is no connection between frames.
- The frames are not numbered and there is no sense of ordering.
- Most of the data-link protocols for LANs are connectionless protocols.

(ii) **Connection Oriented Protocol**

- A logical connection should first be established between the two nodes (setup phase).
- After all frames that are somehow related to each other are transmitted (transfer phase), the logical connection is terminated (teardown phase).
- The frames are numbered and sent in order.
- If the frames are not received in order, the receiver needs to wait until all frames belonging to the same set are received and then deliver them in order to the network layer.
- Connection oriented protocols are rare in wired LANs, but we can see them in some point-to-point protocols, some wireless LANs, and some WANs.

## 1.9 LINK LAYER ADDRESSING

The Internet is a combination of networks glued together by connecting devices (routers or switches). If a packet is to travel from a host to another host, it needs to pass through these networks. Figure 1.57 shows the communication at the data-link layer.

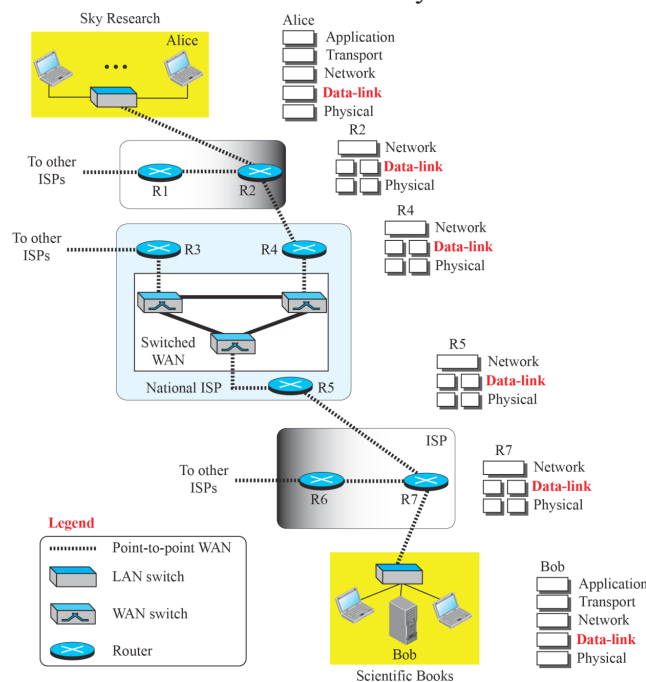
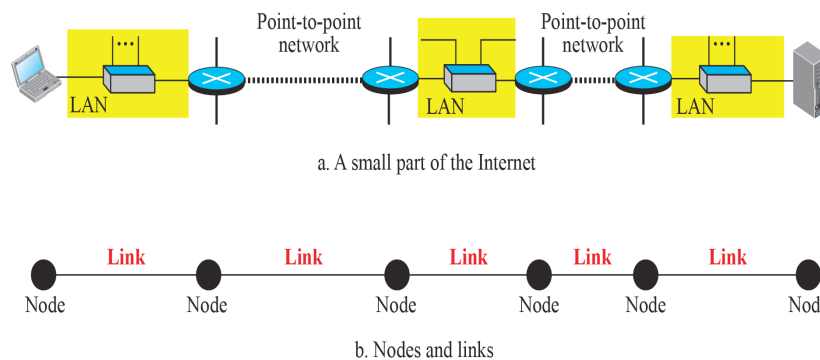


Figure 1.57 Communication at the data-link layer

Communication at the data-link layer is made up of five separate logical connections between the data-link layers in the path. The data-link layer at Alice's computer communicates with the data-link layer at router R2. The data-link layer at router R2 communicates with the data-link layer at router R4 and so on. Finally, the data-link layer at router R7 communicates with the data-link layer at Bob's computer. Only one data-link layer is involved at the source or the destination, but two data-link layers are involved at each router. The reason is that Alice's and Bob's computers are each connected to a single network, but each router takes input from one network and sends output to another network.

### 1.9.1 Nodes and Links

Communication at the data-link layer is node-to-node. A data unit from one point in the Internet needs to pass through many networks (LANs and WANs) to reach another point. These LANs and WANs are connected by routers. It is customary to refer to the two end hosts and the routers as nodes and the networks in between as links. Figure 1.58 is a simple representation of links and nodes when the path of the data unit is only six nodes.



**Figure 1.58 Nodes and Links**

#### Categories of Link

Although two nodes are physically connected by a transmission medium such as cable or air, we need to remember that the data-link layer controls how the medium is used. We can have a data-link layer that uses the whole capacity of the medium; we can also have a data-link layer that uses only part of the capacity of the link. In other words, we can have a point-to-point link or a broadcast link.

In a point-to-point link, the link is dedicated to the two devices; in a broadcast link, the link is shared between several pairs of devices. For example, when two friends use the traditional home phones to chat, they are using a point-to-point link; when the same two friends use their cellular phones, they are using a broadcast link (the air is shared among many cell phone users).

### 1.9.2 Services

The data-link layer is located between the physical and the network layers. The data-link layer provides services to the network layer; it receives services from the physical layer. Let us discuss services provided by the data-link layer.

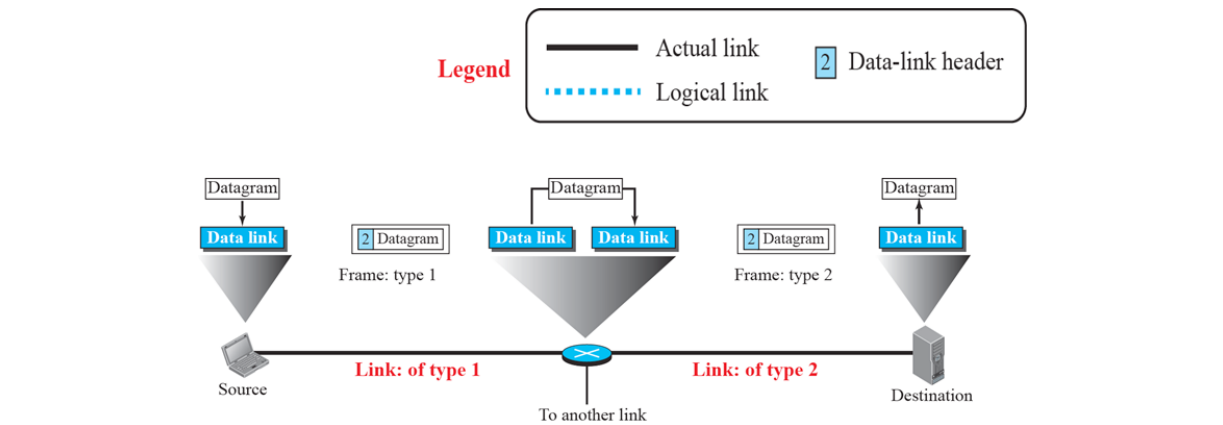


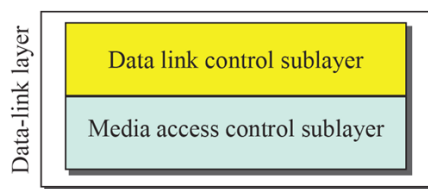
Figure 1.59 A communication with only three nodes

The datagram received by the data-link layer of the source host is encapsulated in a frame. The frame is logically transported from the source host to the router. The frame is decapsulated at the data-link layer of the router and encapsulated at another frame. The new frame is logically transported from the router to the destination host.

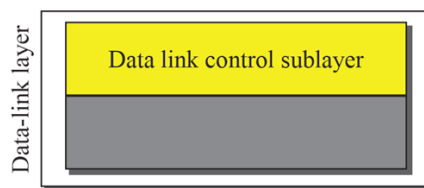
**Sublayer**

To better understand the functionality of and the services provided by the link layer, we can divide the data-link layer into two sub layers;

- (i) Data link control (DLC)
- (ii) Media access control (MAC).



a. Data-link layer of a broadcast link



b. Data-link layer of a point-to-point link

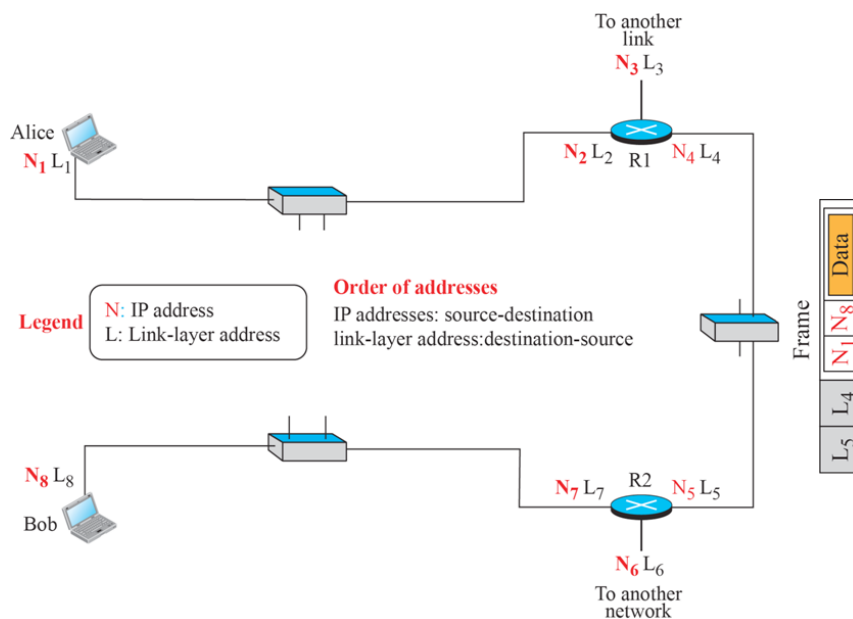
Figure 1.60 Dividing the data-link layer into two sub layers

The DLC sublayer deals with all issues common to both point-to-point and broadcast links. The MAC sublayer deals only with issues specific to broadcast links.

In an internetwork such as the Internet we cannot make a datagram reach its destination using only IP addresses. The source and destination IP addresses define the two ends but cannot define which links the packet should pass through. The reason is that each datagram in the Internet, from the same source host to the same destination host, may take a different path. Also, the IP

addresses in a datagram should not be changed. If the destination IP address in a datagram changes, the packet never reaches its destination.

For that, we need another addressing mechanism in a connectionless internetwork. The link-layer addresses of the two nodes. A link-layer address is sometimes called a **link address**, **sometimes a physical address and sometimes a MAC address**. When a datagram passes from the network layer to the data-link layer, the datagram will be encapsulated in a frame and two data-link addresses are added to the frame header. These two addresses are changed every time the frame moves from one link to another.



**Figure 1.61** IP addresses and link-layer addresses in a small internet

Each frame carries the same datagram with the same source and destination IP addresses (N<sub>1</sub> and N<sub>8</sub>), but the link-layer addresses of the frame change from link to link. In link 1, the link-layer addresses are L<sub>1</sub> and L<sub>2</sub>. In link 2, they are L<sub>4</sub> and L<sub>5</sub>. In link 3, they are L<sub>7</sub> and L<sub>8</sub>. Note that For IP addresses, the source address comes before the destination address and for link-layer addresses, and the destination address comes before the source.

### 1.9.3 Addressing

Data link-layer protocols define three types of addresses. They are,

- (i) **Unicast:** Each host or each interface of a router is assigned a unicast address. Unicasting means one-to-one communication. A frame with a unicast address destination is destined only for one entity in the link.
- (ii) **Multicast:** one-to-many communication. However, the jurisdiction is local (inside the link).
- (iii) **Broadcast:** one-to-all communication. A frame with a destination broadcast address is sent to all entities in the link.

**Example 9.1**

The unicast link-layer addresses in the most common LAN, Ethernet, are 48 bits (six bytes) that are presented as 12 hexadecimal digits separated by colons; for example, the following is a link-layer address of a computer. The second digit needs to be an odd number. The following shows a unicast address.

A3:34:45:11:92:F1

**Example 9.2**

The multicast link-layer addresses in the most common LAN, Ethernet, are 48 bits (six bytes) that are presented as 12 hexadecimal digits separated by colons. The second digit, however, needs to be an even number in hexadecimal. The following shows a multicast address.

A2:34:45:11:92:F1

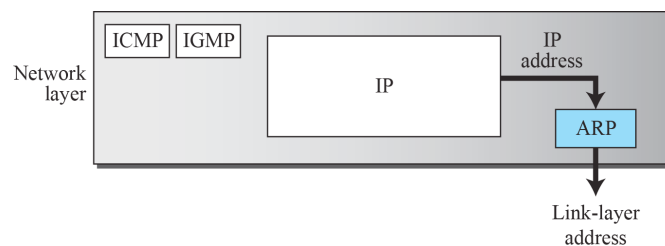
**Example 9.3**

The broadcast link-layer addresses in the most common LAN, Ethernet, are 48 bits, all 1s, that are presented as 12 hexadecimal digits separated by colons. The following shows a broadcast address.

FF: FF: FF: FF: FF: FF:

**1.9.4 ARP**

Anytime a node has an IP datagram to send to another node in a link, it has the IP address of the receiving node. However, the IP address of the next node is not helpful in moving a frame through a link; we need the link-layer address of the next node. This is the time when the Address Resolution Protocol (ARP) becomes helpful.



**Figure 1.62** Position of ARP in TCP/IP protocol suite

The ARP protocol is one of the network layer protocols. ARP accepts an IP address from the IP protocol, maps the address to the corresponding link-layer address, and passes it to the data-link layer.

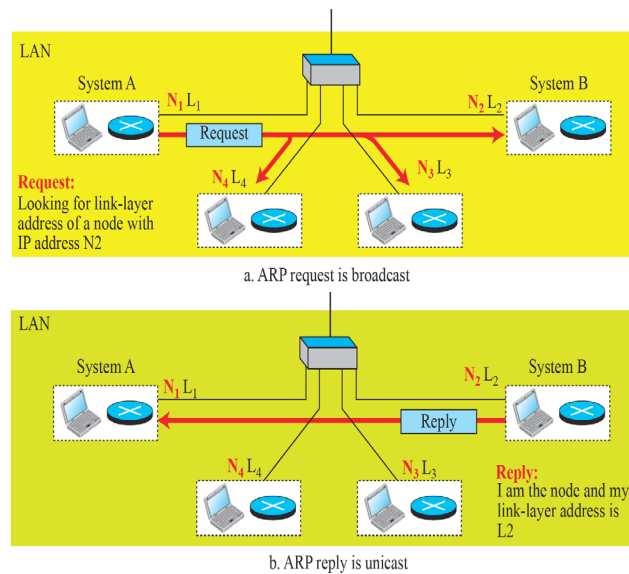


Figure 1.63 ARP operation

When a host or a router needs to find the link-layer address of another host or router in its network, a host or a router sends an ARP request packet that includes the link-layer and IP addresses of the sender and the IP address of the receiver. Because the sender does not know the link-layer address of the receiver, the query is broadcast over the link using the link-layer broadcast address.

Every host or router on the network receives and processes the ARP request packet, but only the intended recipient recognizes its IP address and sends back an ARP response packet. The response packet contains the recipient’s IP and link-layer addresses. The packet is unicast directly to the node that sent the request packet.

0		8	16	31
Hardware Type		Protocol Type		
Hardware length	Protocol length	Operation <b>Request:1, Reply:2</b>		
Source hardware address				
Source protocol address				
Destination hardware address (Empty in request)				
Destination protocol address				

Hardware: LAN or WAN protocol  
Protocol: Network-layer protocol

Figure 1.64 ARP packet

The hardware type field defines the type of the link-layer protocol; Ethernet is given the type 1. The protocol type field defines the network-layer protocol: IPv4 protocol is (0800)16. The source hardware and source protocol addresses are variable-length fields defining the link-layer and network-layer addresses of the sender. The destination hardware address and destination protocol address fields define the receiver link-layer and network-layer addresses. An ARP packet is encapsulated directly into a data-link frame.

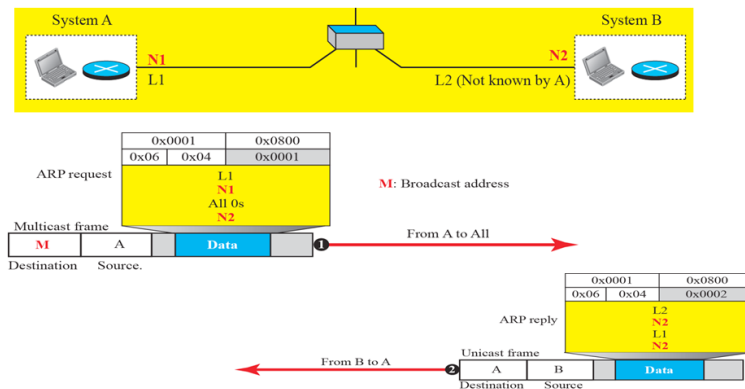


Figure 1.65 Example for the ARP request and response messages

In figure 1.65 a host with IP address N1 and MAC address L1 has a packet to send to another host with IP address N2 and physical address L2 (which is unknown to the first host). The two hosts are on the same network.



Figure 1.66 The internet for our example

In figure 1.66, assume Alice needs to send a datagram to Bob, who is three nodes away in the Internet. Assume that Alice knows the following

- (a) Data to be sent.
- (b) The IP address of Alice's host (each host needs to know its IP address).
- (c) Network-layer (IP) address of Bob.

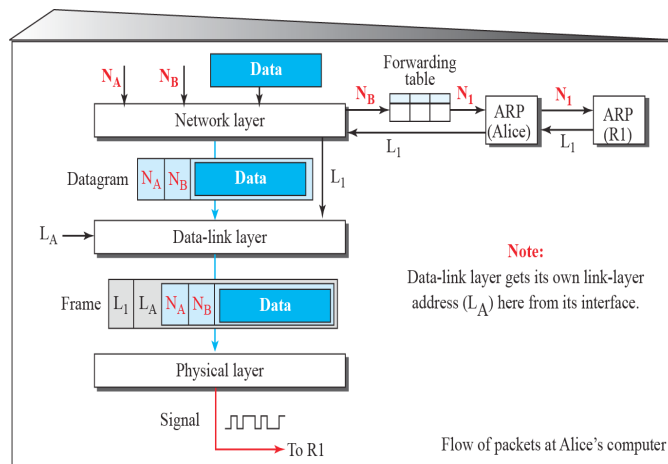
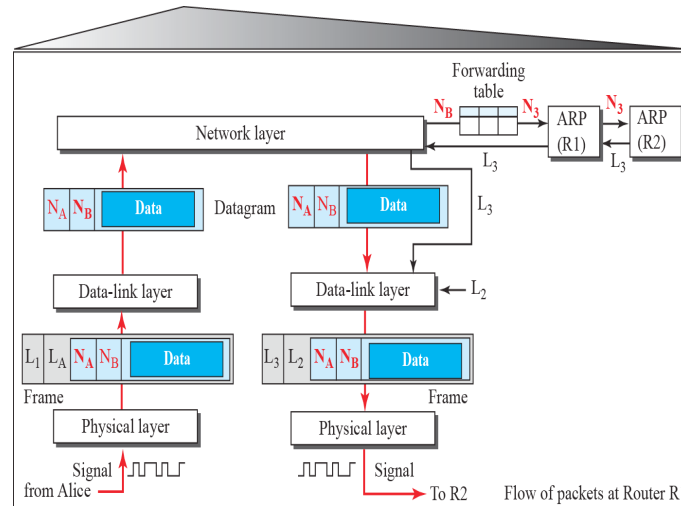


Figure 1.67 Flow of packets at Alice site

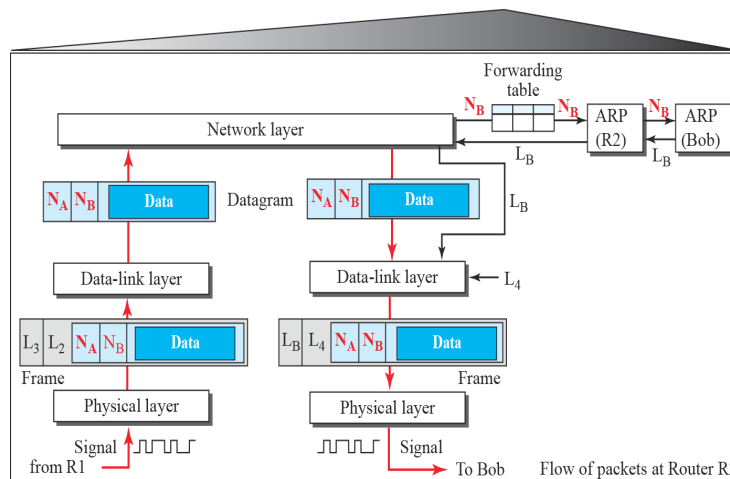
Figure 1.67 shows that, the network layer knows it's given NA, NB, and the packet, but it needs to find the link-layer address of the next node. The network layer consults its routing table and tries to find which router is next for the destination NB. The routing table gives N1, but the network

layer needs to find the link-layer address of router R1. It uses its ARP to find the link-layer address L1. The network layer can now pass the datagram with the link-layer address to the data-link layer. The data-link layer knows its own link-layer address, LA. It creates the frame and passes it to the physical layer, where the address is converted to signals and sent through the media.



**Figure 1.68** *Flow of activities at router R1*

In figure 1.68 Router R1 has only three lower layers. The packet received needs to go up through these three layers and come down. At arrival, the physical layer of the left link Change the signal received from the link to a frame and passes it to the data-link layer. The data-link layer decapsulates the datagram and passes it to the network layer. The network layer examines the network-layer address of the datagram and finds that the datagram needs to be delivered to the device with IP address NB.



**Figure 1.69** *Flow of activities at router R2*

The network layer consults its routing table to find out which is the next node (router) in the path to NB. The forwarding table returns N3. The IP address of router R2 is in the same link with R1. The network layer now uses the ARP to find the link-layer address of this router, which comes up as L3. The network layer passes the datagram and L3 to the data-link layer belonging to the link at the right side. The link layer encapsulates the datagram, adds L3 and L2 (its own link-layer



address), and passes the frame to the physical layer. The physical layer encodes the bits to signals and sends them through the medium to R2.

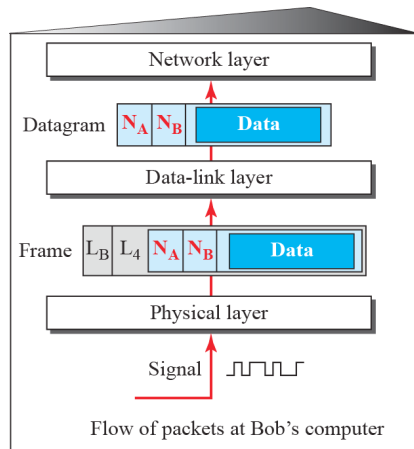


Figure 1.70 Activities at Bob's site

Figure 1.70 shows that, the signals at Bob's site are changed to a message. At Bob's site there are no more addresses or mapping needed. The signal received from the link is changed to a frame. The frame is passed to the data-link layer, which decapsulates the datagram and passes it to the network layer. The network layer decapsulates the message and passes it to the transport layer.

### 1.10 ERROR DETECTION AND CORRECTION

Network must be able to transfer data from one device to another with complete accuracy. Data can be corrupted during transmission. For reliable communication, errors are detected and corrected.

#### 10.1 Types of Errors

Whenever electromagnetic signal flows from one point to another, it is subjected to unpredictable interference from heat, magnetism and other forms of electricity. This interference can change the shape on timing of the signal. There are two types of errors,

- (i) Single-Bit Error
- (ii) Burst Error

##### Single-Bit Error

In a single-bit error, only 1 bit in the data unit has changed. Single bit error can happen if we are sending data using parallel transmission. Single bit error is produced very rarely in serial data transmission.

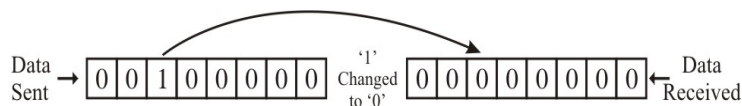


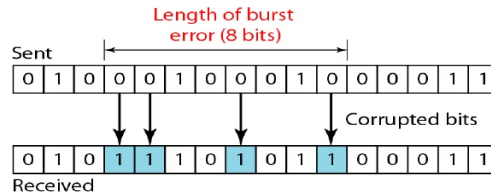
Figure 1.71 Single-bit error

##### Example

The data rate is 1Mbps. For a single bit error, possibility is 1/1,000,000 (or) 1 Microseconds.

**Burst Error**

A burst error means that 2 or more bits in the data unit have changed. Burst error will be produced by serial transmission. Number of bits affected in burst error will depend on the data transmission rate and duration of noise.



**Figure 1.72 Burst error of length 8**

**Example**

- (i) Data rate = 1 Kbps; noise = 1/100 sec, then  
No. of bits affected = 10
- (ii) Data rate = 1 Mbps; noise = 1/100 sec, then  
No. of bits affected = 10,000

**1.10.2 Redundancy**

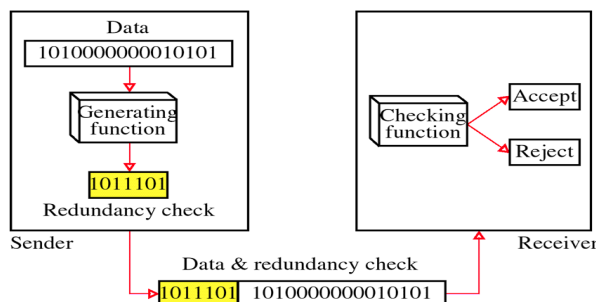
One error detection mechanism would send every data unit twice. The receiving device would then be able to do a bit-by-bit comparison between the two versions of data. Any discrepancy would indicate an error. Any error will found the necessary correction mechanism should take place.

**Disadvantages**

- (i) Transmission time is double.
- (ii) Time taken for bit-by-bit comparison is high

To overcome this drawback, instead of repairing the entire data stream, a shorter group of bits (redundant) may be appended to the end of each data unit. The technique involving the addition of extra bits to the data unit is called redundancy.

These redundant bits are added by the sender and removed by the receiver. Their presence allows the receiver to detect or correct corrupted bits. By suitably appending the required parity (either odd or even parity) may be obtained. In the appended unit, if the total number of 1s is even then it is called even parity and if the total number of 1s is odd then it is called odd parity.



**Figure 1.73 Redundancy**

### 1.10.3 Types of Redundancy Check

Four types of redundancy checks are common in data communications. They are;

- (i) Vertical redundancy check (VRC)
- (ii) Longitudinal redundancy check(LRC)
- (iii)Cyclic redundancy check (CRC)
- (iv)Checksum

#### Vertical redundancy check (VRC)

- Most common and least expensive mechanism.
- VRC is also called as parity check method.
- Redundant bits or parity bit is appended to every data unit so that the total number of 1's in the data unit becomes even.
- Some systems may use odd parity checking.
- It can detect only the single bit error.

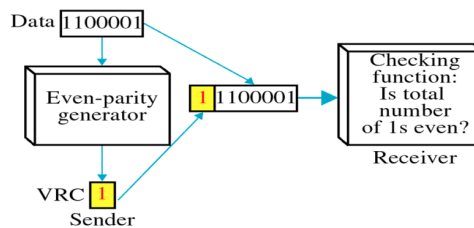


Figure 1.74 Vertical redundancy check

#### Longitudinal redundancy check (LRC)

- In LRC block of bit is organized in to a table.
- For example 32 bit data unit is arranged as four rows and eight columns.
- Check the parity bit for each column and create a new row of eight bits which are the parity bits for the whole block.
- Original data with eight parity bits are transferred to the receiver.

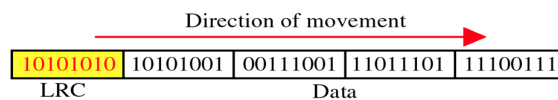


Figure 1.75 Longitudinal redundancy check

#### Cyclic redundancy check (CRC)

- Unlike VRC and LCR, CRC method is working based on division.

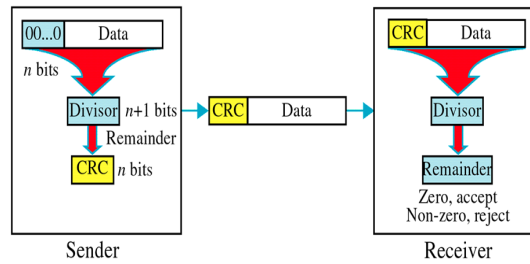


Figure 1.76 Cyclic redundancy check

**CRC generator**

- CRC generator uses modulo-2 division.

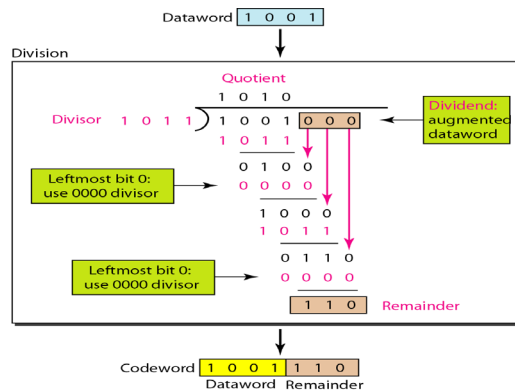


Figure 1.77 CRC generator

**CRC Checker**

- The CRC checker functions exactly like the CRC generator
- After receiving the data appended with the CRC, the checker does the same modulo-2 division.
- If the remainder is all 0's the CRC is dropped and the data accepted. Otherwise the data will be discarded (It should be resent by the sender).

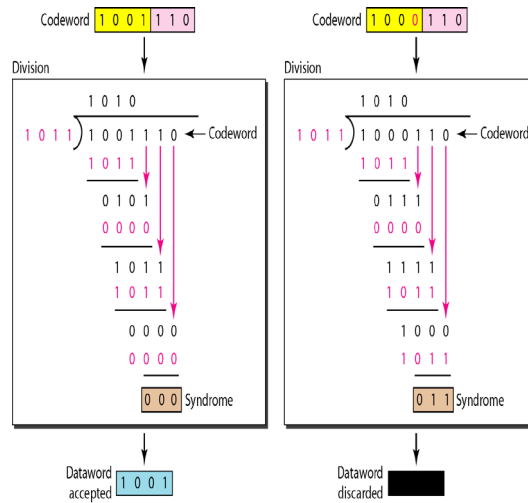


Figure 1.78 CRC checker

### 1.10.4 Polynomials

- More common representation than binary form
- Easy to analyze
- Divisor is commonly called generator polynomial

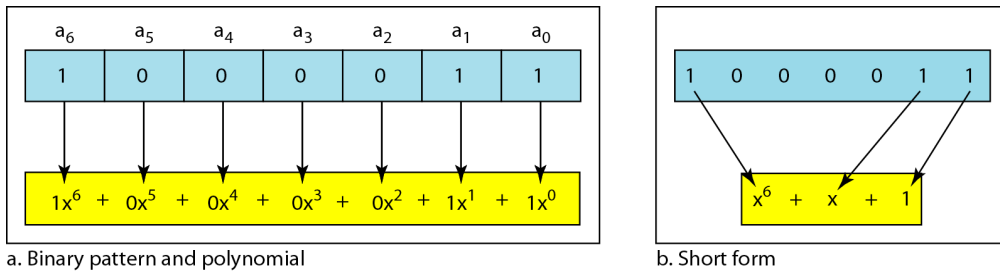


Figure 1.79 Polynomial representation

#### Properties of polynomials

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

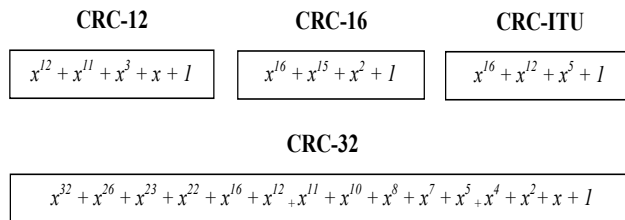


Figure 1.80 Standard polynomials

#### Checksum

- It is the error detection mechanism used by the higher layer protocols.
- Checksum is working on the redundancy concept.

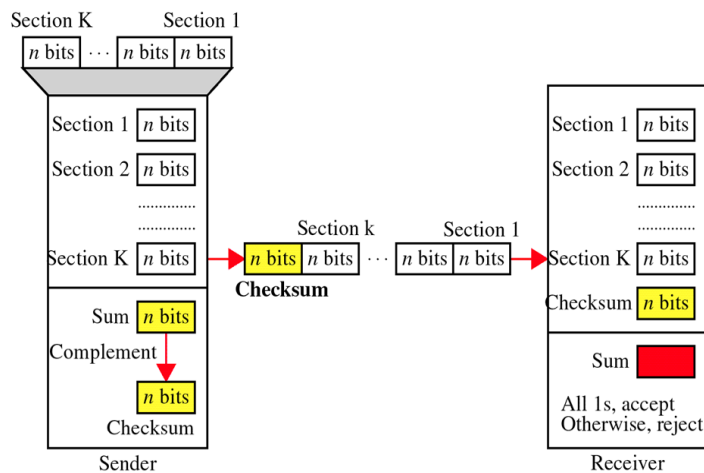


Figure 1.81 Check sum generator and checker

←—————→  
**Checksum Generator**

- The data unit is divided into K sections, each of n Bits
- All sections are added together using one's complement to get the sum.
- The sum is then complemented and becomes the checksum.
- The check sum is sent with the data
- If the sum of the data segment is T, the checksum will be  $-T$ .

**Checksum checker**

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

**Performance**

- The checksum detects all errors involving an odd number of bits.
- It detects most errors involving an even number of bits.
- If one or more bits of a segment are damaged and the corresponding bit or bits of opposite value in a second segment are also damaged, the sums of those columns will not change and the receiver will not detect a problem.

**1.10.5 Error Correction**

It can be handled in two ways:

- (i) Receiver can have the sender retransmit the entire data unit.
- (ii) The receiver can use an error-correcting code, which automatically corrects certain errors.

**Single-bit error correction**

- To correct an error, the receiver reverses the value of the altered bit. To do so, it must know which bit is in error.
- Number of redundancy bits can be calculated as follows.

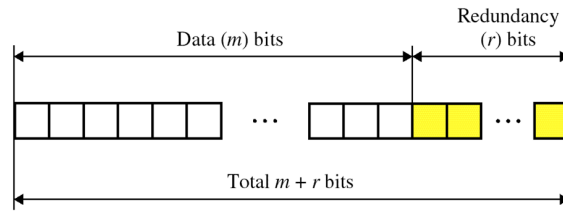
Let data bits = m

Redundancy bits = r

∴ Total message sent = m+r

- The value of r must satisfy the following relation:

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



Number of data bits k	Number of redundancy bit r	Total bits k + r
1	2	3
2	2	5
3	3	6
4	3	7
5	4	9

**Hamming Code**

- Hamming code is used to positioning the redundancy bits.
- For example
  - If  $m = 7$  then  $r = 4$ ;
  - So total number of bits =  $7 + 4$
  - = 11
- The redundancy bits are  $r_1, r_2, r_3$  and  $r_4$ .
- The position of the redundancy bits will be

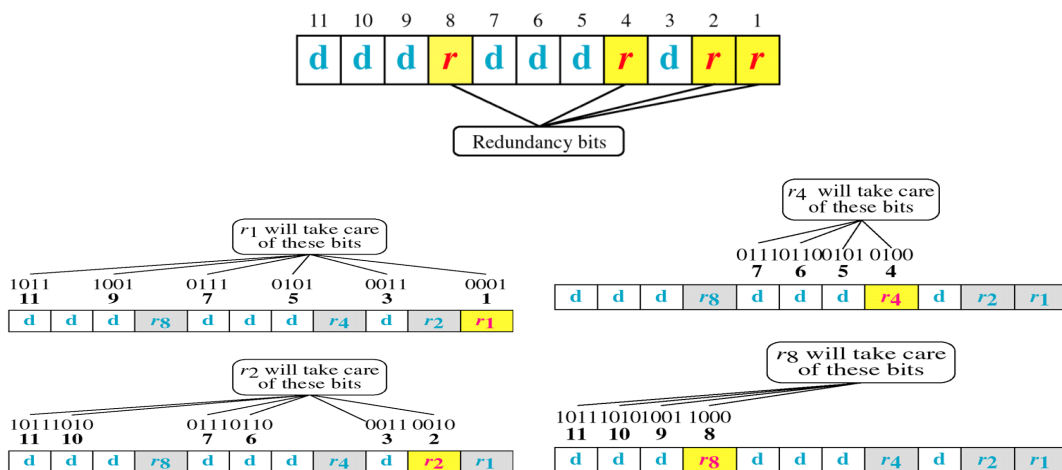


Figure 1.82 Redundancy bit calculation

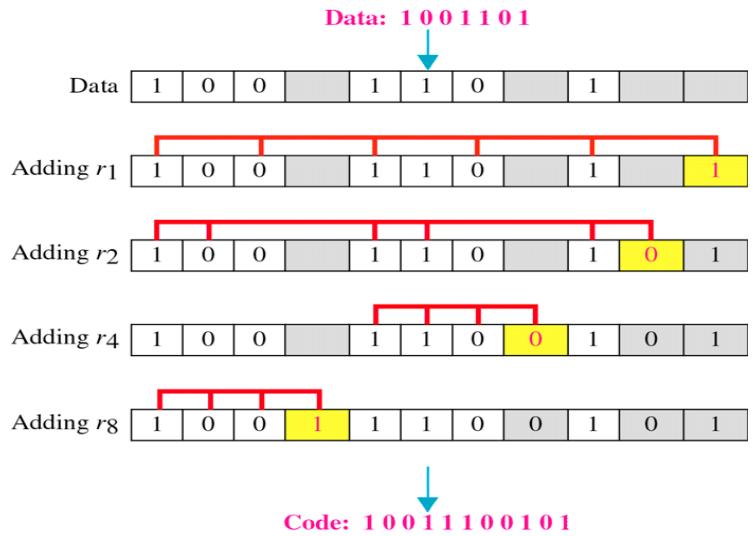


Figure 1.83 Example of hamming code

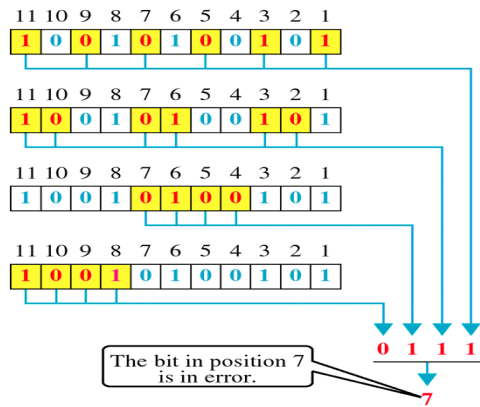


Figure 1.84 Example of error detection



# MEDIA ACCESS & INTERNETWORKING

---

## 2.1 OVERVIEW OF DATA LINK CONTROL

The two main functions of the data link layer are data link control and media access control. The data link control deals with the design and procedures for communication between two adjacent nodes (node-to-node communication). The second function of the data link layer is media access control, or how to share the link.

Data link control functions include framing, flow and error control, and software implemented protocols that provide smooth and reliable transmission of frames between nodes. To implement data link control, we need protocols. Protocol is a set of rules that need to be implemented in software and run by the two nodes involved in data exchange at the data link layer.

Data transmission in the physical layer means moving bits in the form of a signal from the source to the destination. The physical layer provides bit synchronization to ensure that the sender and receiver use the same bit durations and timing.

### 2.1.1 Framing

The data link layer needs to pack bits into frames, so that each frame is distinguishable from another. Framing in the data link layer separates a message from one source to a destination, or from other messages to other destinations, by adding a sender address and a destination address. The destination address defines where the packet is to go. The sender address helps the recipient acknowledge the receipt. Frames can be of fixed or variable size.

#### *(i) Fixed-Size Framing*

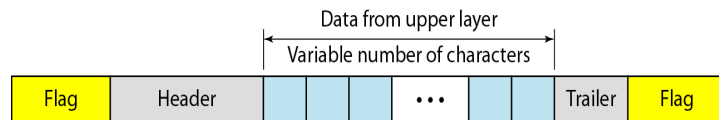
In fixed-size framing, there is no need for defining the boundaries of the frames; the size itself can be used as a delimiter. An example of this type of framing is the ATM wide-area network, which uses frames of fixed size called cells. ATM (Asynchronous Transfer Mode) is a connection oriented, high-speed network technology that is used in both LAN and WAN over optical fiber and operates up to gigabit speed.

#### *(ii) Variable-Size Framing*

In variable-size framing, we need a way to define the end of the frame and the beginning of the next. Two approaches were used for this purpose: ***a character-oriented approach and a bit oriented approach.***

### Character-Oriented Protocols

In a character-oriented protocol, data to be carried are 8-bit characters from a coding system such as ASCII. The header, which normally carries the source and destination addresses and other control information, and the trailer, which carries error detection or error correction redundant bits, are also multiples of 8 bits.

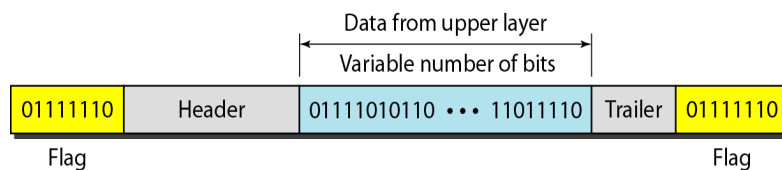


**Figure 2.1** A frame in a character-oriented protocol

To separate one frame from the next, an 8-bit (1-byte) flag is added at the beginning and the end of a frame. The flag, composed of protocol-dependent special characters, signals the start or end of a frame. Figure 2.1 shows the format of a frame in a character-oriented protocol

### Bit-Oriented Protocols

In a bit-oriented protocol, the data section of a frame is a sequence of bits to be interpreted by the upper layer as text, graphic, audio, video, and so on. However, in addition to headers (and possible trailers), we still need a delimiter to separate one frame from the other. Most protocols use a special 8-bit pattern flag 01111110 as the delimiter to define the beginning and the end of the frame, as shown in Figure 3.2.



**Figure 2.2A** frame in a bit-oriented protocol

#### 2.1.2 FLOW AND ERROR CONTROL

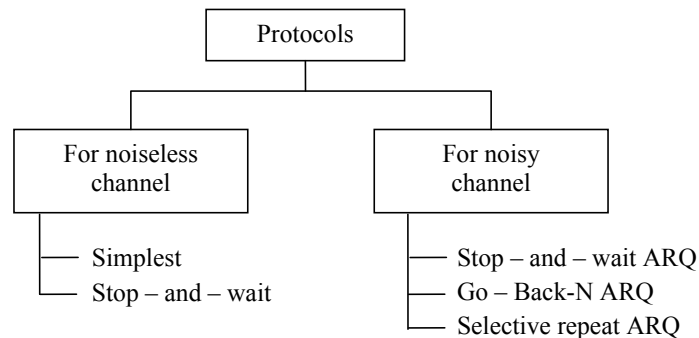
The most important responsibilities of the data link layer are flow control and error control. Collectively, these functions are known as data link control. Flow control refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgment. Each receiving device has a block of memory, called a buffer, reserved for storing incoming data until they are processed. If the buffer begins to fill up, the receiver must be able to tell the sender to halt transmission until it is once again able to receive.

Error control is both error detection and error correction. It allows the receiver to tell the sender of any frames lost or damaged in transmission and coordinates the retransmission of those frames by the sender. Error control in the data link layer is based on automatic repeat request (ARQ), which is the retransmission of data.

### 2.2 PROTOCOLS USED FOR FLOW CONTROL

All the protocols are unidirectional. The data frames travel from one node, called the sender, to another node, called the receiver. Special frames, called acknowledgment (ACK) and negative acknowledgment (NAK) can flow in the opposite direction.

In bidirectional data flow – the protocol includes the control information such as ACKs and NAKs with the data frames. This technique is called piggybacking.



*Figure 2.3 Taxonomy of protocols used for flow control*

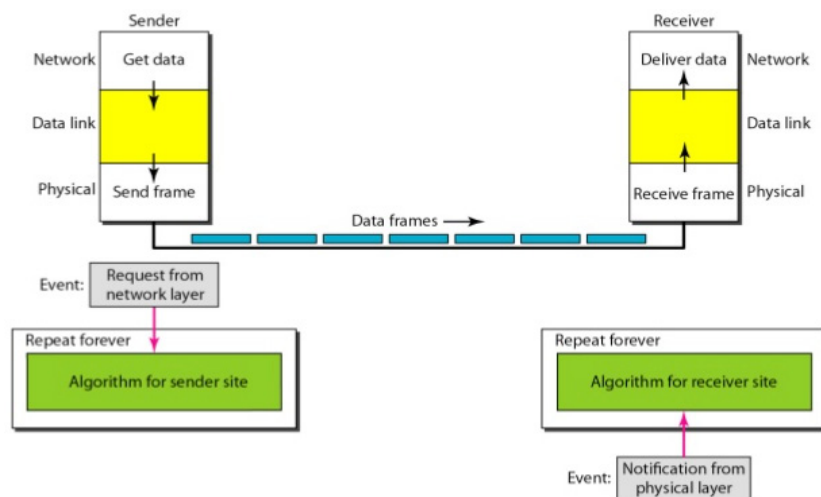
## Noiseless Channels

An ideal channel in which no frames are lost, duplicated, or corrupted. We introduce two protocols for this type of channel. The first is a protocol that does not use flow control; the second is the one that does.

### 2.2.1 Simplest Protocol

It is a unidirectional protocol in which data frames are traveling in only one direction—from the sender to receiver. The receiver can immediately handle any frame it receives within a processing time. The data link layer of the receiver immediately removes the header from the frame and hands the data packet to its network layer, which can also accept the packet immediately. Here the receiver can never be overwhelmed with incoming frames.

#### Design



*Figure 2.4 Simplest Protocol Design*

There is no need for flow control in this scheme. The data link layer at the sender site gets data from its network layer, makes a frame out of the data, and sends it. The data link layer at the

receiver site receives a frame from its physical layer, extracts data from the frame, and delivers the data to its network layer. The data link layers of the sender and receiver provide transmission services for their network layers. The data link layers use the services provided by their physical layers (such as signaling, multiplexing, and so on) for the physical transmission of bits.

### ***The procedure used by both data link layers***

The sender site cannot send a frame until its network layer has a data packet to send. The receiver site cannot deliver a data packet to its network layer until a frame arrives. The procedure / event of a protocol are as follows;

- (i) The procedure at the sender site is constantly running; there is no action until there is a request from the network layer.
- (ii) The procedure at the receiver site is also constantly running, but there is no action until notification from the physical layer arrives.

Both procedures are constantly running because they do not know when the corresponding events will occur.

### ***Sender-site algorithm for the simplest protocol***

1	while (true)	//Repeat forever
2	{	
3	WaitForEven( )I	// Sleep until an event occurs
4	If(Event(RequestToSend>>	// There is a packet to send
5	{	
6	GetData( )i	
7	MakeFrame( )i	
8	sendFrame( )i	// Send the frame
9	}	
10	}	

Where,

- (i) ***GetData( )*** - takes a data packet from the network layer.
- (ii) ***MakeFrame( )*** - adds a header and delimiter flags to the data packet to make a frame.
- (iii) ***SendFrame( )*** - delivers the frame to the physical layer for transmission.

### ***Receiver-site algorithm for the simplest protocol***

1	While(true)	// Repeat forever
2	{	
3	waitForEvent( ) I	II Sleep until an event occur
4	if (Event(ArrivalNotification>>	II Data frame arrived
5	{	

```

6   ReceiverFrame() i
7   ExtractData() i
8   DeliverData() I      // Deliver data to network layer
9   }
10  }

```

### Flow diagram

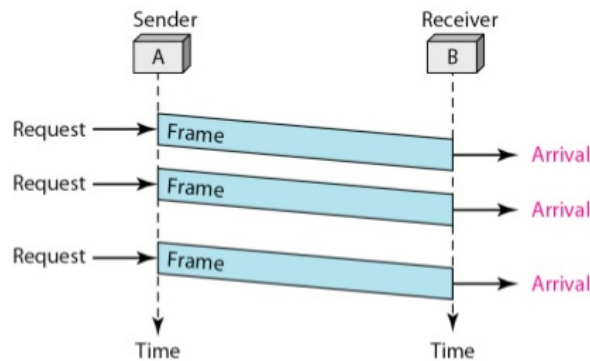


Figure 2.5 Simplest Protocol – Flow diagram

### 2.2.2 Stop-and-Wait Protocol

If data frames arrive at the receiver site faster than they can be processed, the frames must be stored until their use. The receiver does not have enough storage space, especially if it is receiving data from many sources. This may result in either the discarding of frames or denial of service. To prevent the receiver from becoming overwhelmed with frames, we need to tell the sender to slow down. There must be a feedback from the receiver to the sender. In the Stop-and-Wait Protocol sender sends one frame, stops until it receives confirmation from the receiver and then sends the next frame.

In Stop-and-Wait Protocol

- (i) Data frames will follow the unidirectional communication.
- (ii) ACK frames (simple tokens of acknowledgment) can travel from the other direction.

### Design

At any time, there is either one data frame on the forward channel or one ACK frame on the reverse channel. We therefore need a half-duplex link.

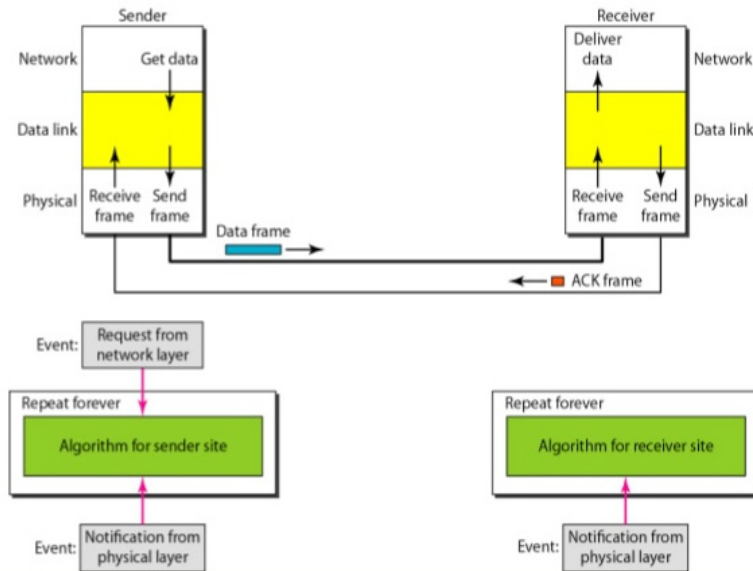


Figure 2.6 Design of Stop-and- Wait Protocol

**Sender-site algorithm for Stop-and- Wait Protocol**

```

while (true)
canSend = true
{
    waitForEvent ( )i
    if (Event(RequestToSend) AND canSend)
    {
        GetData( )i
        MakeFrame ( );
        SendFrame ( )i
        canSend = false;
    }
    waitForEvent ( )i
    if(Event(Arrival Notification)
    {
        ReceiverFrame( )i
        cansend = true;
    }
}

```

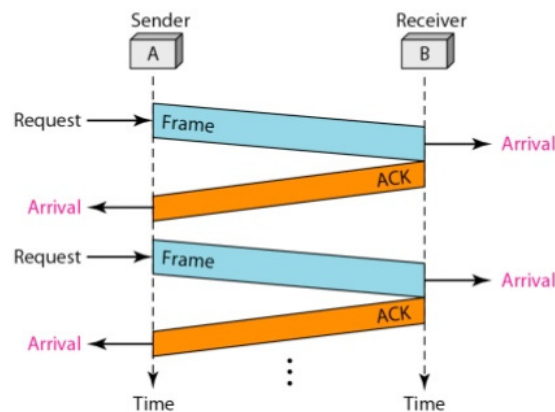
*// Repeat forever*  
*// Allow the first frame to go*  
*// Sleep until an event occurs*  
*// Send the data frame*  
*// cannot send until ACK arrives*  
*// Sleep until an event occurs*  
*// An ACK has arrived*  
*// Receive the ACK frame*

**Receiver-site algorithm for Stop-and-Wait Protocol**

```

while (true)                                // Repeat forever
{
  waitForEvent ( )i                          // Sleep until an event occurs
  if(Event(Arrival Notification)             // Data frame arrives
  {
    ReceiverFrame( }i                        // Receive the ACK frame
    ExtractData( }i
    Deliver(data);                            // I Deliver data to network layer
    SendFrame( );                             // Send an ACK frame
  }
}

```

**Data flow diagram for Stop and wait Protocol****Figure 2.7 Flow diagram for Stop-and- Wait 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 discuss three protocols in this section that use error control.

**2.2.3 Stop-and-Wait Automatic Repeat Request**

The Stop-and-Wait Automatic Repeat Request protocol adds a simple error control mechanism. To detect and correct the corrupted frames - 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 is manifested by the silence of the receiver. To number the frames - handle the corrupted frames, duplicate, or a frame out of order.

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 before receiving the ACK. 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.

### Flow diagram for Stop-and-Wait ARQ

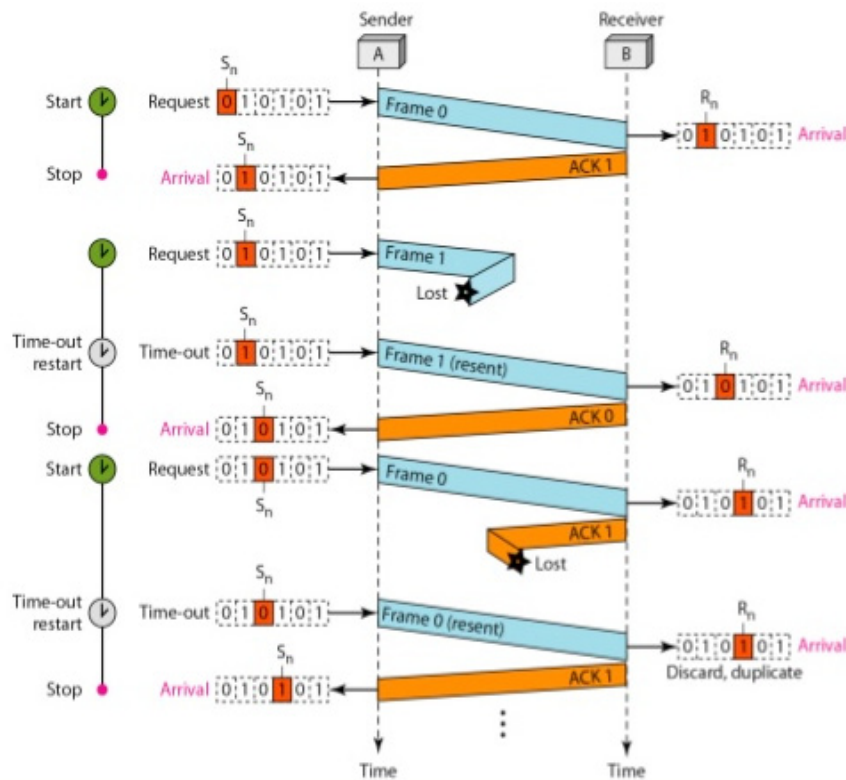


Figure 2.8 Flow diagram for Stop-and- Wait ARQ

### Efficiency

The Stop-and-Wait ARQ discussed in the previous section is very inefficient if our channel is thick and long. By thick, we mean that our channel has a large and width; by long, we mean the round-trip delay is long.

### 2.2.4 Go-Back-N Automatic Repeat Request

To improve the efficiency of transmission (filling the pipe), multiple frames must be in transition by the sender while waiting for acknowledgment (to keep the channel busy). The first is called Go-Back-N ARQ – 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. So the sequence numbers are 0, 1,2,3,4,5,6, 7,8,9, 10, 11, 12, 13, 14, 15,0, 1,2,3,4,5,6,7,8,9,10, 11, ...



### Control Variables

- Sender has 3 variables: S, SF, and SL
- S holds the sequence number of recently sent frame
- SF holds the sequence number of the first frame
- SL holds the sequence number of the last frame
- Receiver only has the one variable, R that holds the sequence number of the frame it expects to receive.
- If the seq. no. is the same as the value of R, the frame is accepted, otherwise rejected.

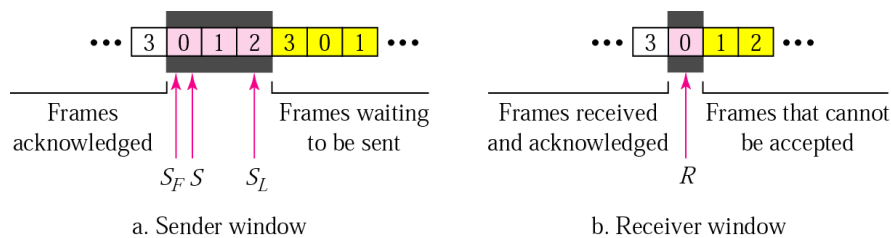


Figure 2.9 Go-Back-N ARQ

### Normal operation of Go-Back-N ARQ

The sender keeps track of the outstanding frames and updates the variables and windows as the ACKs arrive.

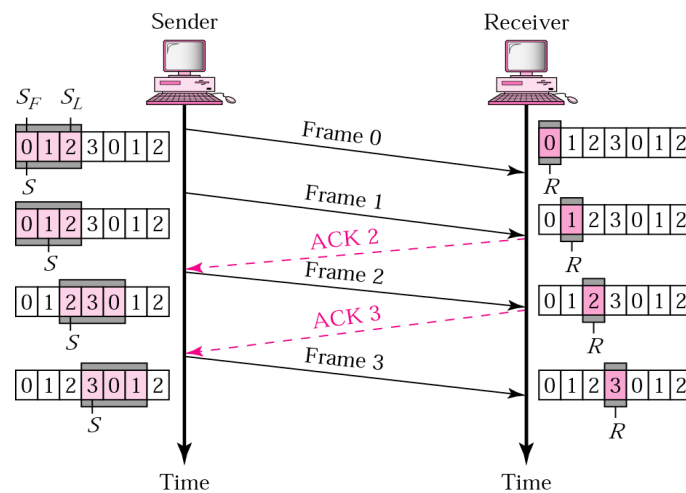
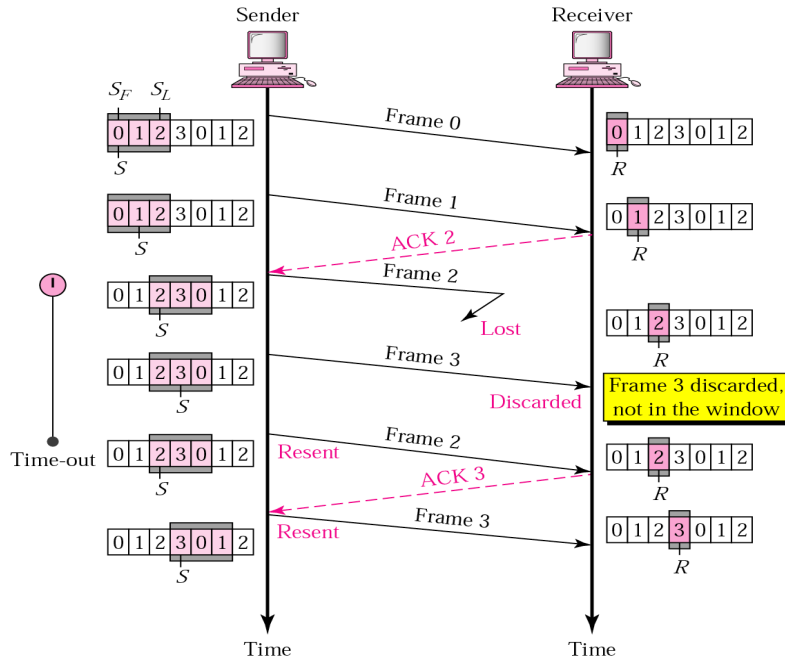


Figure 2.10 Normal operation of Go-Back-N ARQ

**Go-Back-N ARQ - Lost frame**



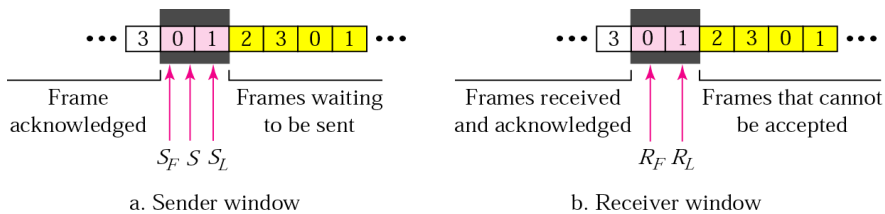
**Figure 2.11 Go-Back-N ARQ- Lost frame**

Consider a situation that if frame 2 is lost and the receiver receives frame 3, it discards frame 3 as it is expecting frame 2 (according to window). After the timer for frame 2 expires at the sender site, the sender sends frame 2 and 3. (Go back to 2)

**2.2.5 Selective Repeat ARQ**

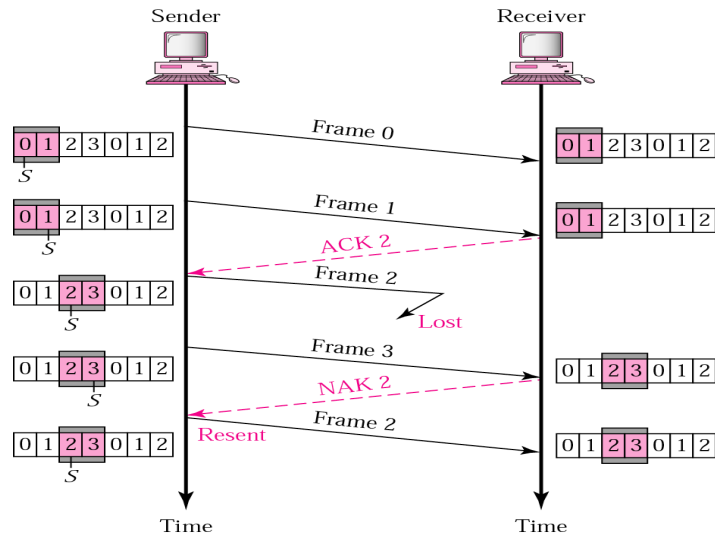
**Sender and receiver windows**

Go-Back-N ARQ simplifies the process at the receiver site. Receiver only keeps track of only one variable, and there is no need to buffer out-of-order frames, they are simply discarded. However, Go-Back-N ARQ protocol is inefficient for noisy link. It bandwidth inefficient and slows down the transmission. In Selective Repeat ARQ, only the damaged frame is resent. It may give more bandwidth efficiency but more complex processing at receiver site. It defines a negative ACK (NAK) to report the sequence number of a damaged frame before the timer expires.



**Figure 2.12 Selective Repeat ARQ**

**Selective Repeat ARQ- Lost frame**

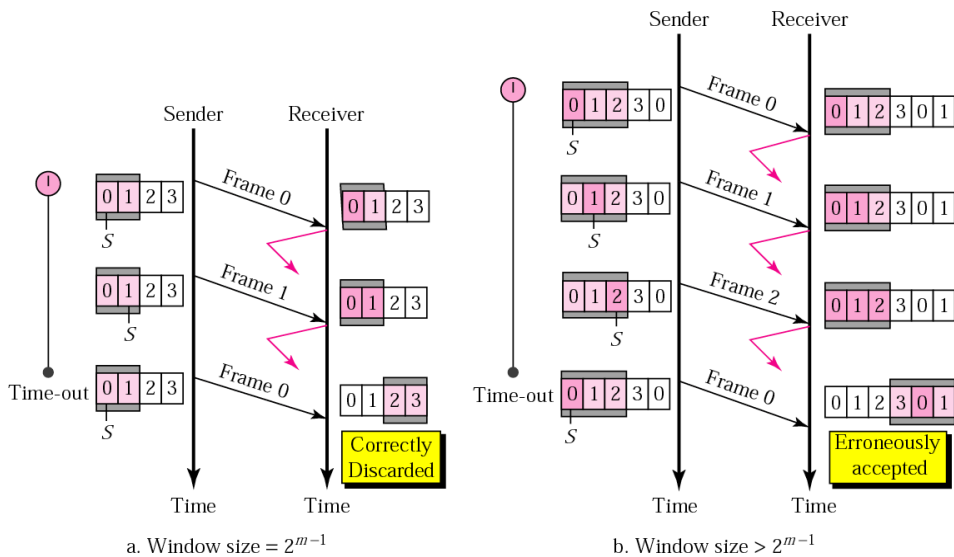


**Figure 2.13 Selective Repeat ARQ- Lost frames**

Frames 0 and 1 are accepted when received because they are in the range specified by the receiver window. Same thing will be followed for frame 3. Receiver sends a NAK2 to show that frame 2 has not been received and then sender resends only frame 2 and it is accepted as it is in the range of the window.

**Selective Repeat ARQ - Sender window size**

Size of the sender and receiver windows must be at most one-half of  $2^m$ . If  $m = 2$ , window size should be  $2^m / 2 = 2$ . Fig compares a window size of 2 with a window size of 3. Window size is 3 and all ACKs are lost, sender sends duplicate of frame 0, window of the receiver expect to receive frame 0 (part of the window), so accepts frame 0, as the 1st frame of the next cycle – an error.



**Figure 2.14 Selective Repeat ARQ – Sender window**

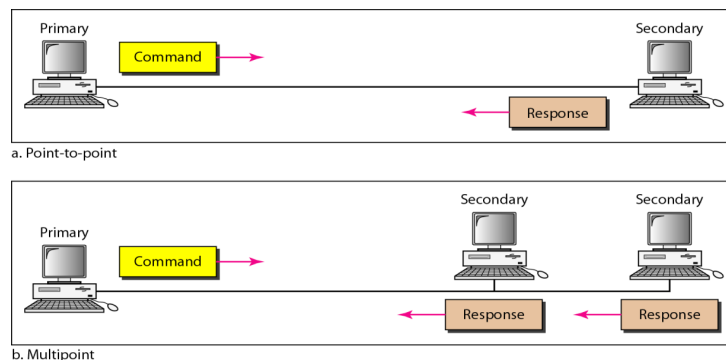
### 2.3. HIGH-LEVEL DATA LINK CONTROL (HDLC)

High-level Data Link Control (HDLC) is a bit-oriented protocol for communication over point-to-point and multipoint links. It implements the ARQ mechanisms. The HDLC protocol embeds information in a data frame that allows devices to control data flow and correct errors. In 1979, the ISO made HDLC the standard as a Bit-oriented control protocol. The HDLC provides a transparent transmission service at the data link layer of the OSI. The users of the HDLC service provide PDUs which are encapsulated to form data link layer frames. These frames are *separated by HDLC "flags" and are modified by "zero bit insertion" to guarantee transparency*.

Each piece of data is encapsulated in an HDLC frame by adding a trailer and a header. The header contains an HDLC address and an HDLC control field. The trailer is found at the end of the frame, and contains a (CRC) which detects any errors which may occur during transmission. The frames are separated by HDLC flag sequences which are transmitted between each frame and whenever there is no data to be transmitted. HDLC provides two common transfer modes that can be used in different configurations: *normal response mode (NRM) and asynchronous balanced mode (ABM)*.

#### (i) Normal response mode (NRM)

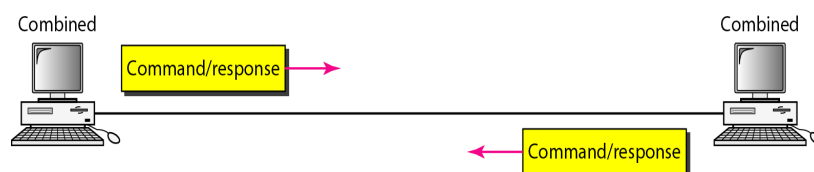
In normal response mode (NRM), the station configuration is unbalanced. We have one primary station and multiple secondary stations. A primary station can send commands and a secondary station can only respond. The NRM is used for both point-to-point and multiple-point links, as shown in Figure 2.15.



*Figure 2.15 Normal response mode*

#### (ii) Asynchronous Balanced Mode

In asynchronous balanced mode (ABM), the configuration is balanced. The link is point-to-point, and each station can function as a primary and a secondary (acting as peers), as shown in Figure 2.16. This is the common mode today.



*Figure 2.16 Asynchronous balanced mode*

### 2.3.1 Frames

To provide the flexibility, HDLC defines three types of frames namely *information frames (I-frames)*, *supervisory frames (S-frames)*, and *unnumbered frames (V-frames)*. Each type of frame serves as an envelope for the transmission of a different type of message.

- I-frames are used to transport user data and control information relating to user data (piggybacking).
- S-frames are used only to transport control information.
- V-frames are reserved for system management. Information carried by V-frames is intended for managing the link itself.

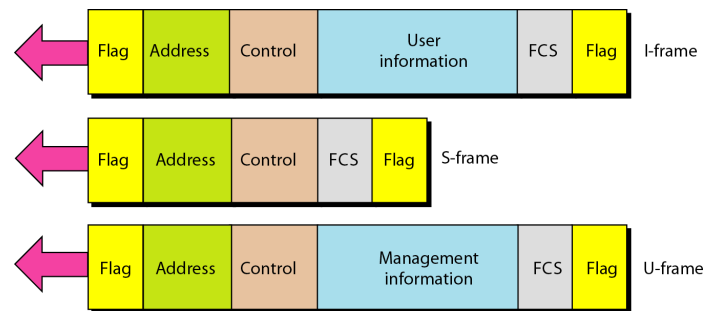


Figure 2.17 HDLC frames

#### Frame Format

Each frame in HDLC contain up to six fields, as shown in Figure 2.17.

- Beginning flag field
- An address field
- A control field
- An information field
- A frame check sequence (FCS) field
- An ending flag field.

In multiple-frame transmissions, the ending flag of one frame can serve as the beginning flag of the next frame.

#### Fields and their use in different frame types

- Flag field:** The flag field of an HDLC frame is an 8-bit sequence with the bit pattern 01111110 that identifies both the beginning and the end of a frame and serves as a synchronization pattern for the receiver. The ending flag of one frame can be used as the beginning flag of the next frame.
- Address field:** The second field of an HDLC frame contains the address of the secondary station. If a primary station created the frame, it contains a *to address*. If a secondary creates the frame, it contains a *from address*. An address field can be 1 byte or several bytes long, depending on the needs of the network. One byte can identify up to 128 stations.

- If the address field is only 1 byte, the last bit is always a 1.
- If the address is more than 1 byte, all bytes but the last one will end with 0; only the last will end with 1.
- Ending each intermediate byte with 0 indicates to the receiver that there are more address bytes to come.

**(iii) Control field:** The control field is a 1- or 2-byte segment of the frame used for flow and error control. The interpretation of bits in this field depends on the frame type.

**(iv) Information field:** The information field contains the user's data from the network layer or management information. Its length can vary from one network to another.

**(v) FCS field:** The frame check sequence (FCS) is the HDLC error detection field. It can contain either a 2- or 4-byte ITU-T CRC.

### 2.3.1.1 Bit Stuffing

HDLC uses a process called **Bit Stuffing**. Bit stuffing is the process of adding one extra zero whenever there are 5 consecutive 1's in the data, so that the receiver doesn't mistake the data for a flag. Every time a sender wants to transmit a bit sequence having more than 6 consecutive 1's, it inserts 1 redundant 0 after the 5<sup>th</sup> 1.

#### Exceptions

- When the **bit sequence** is really a **flag**.
- When **transmission** is being **aborted**.
- When the **channel** is being put into **idle**.

#### Example

A frame before bit stuffing

01111110 01111100 101101111 110010

After

011111010 011111000 101101111 1010010

#### How does the receiver identify a stuffed bit?

- Receiver reads incoming bits and counts 1's.
- When number of consecutive 1s after a zero is 5, it checks the next bit (7<sup>th</sup> bit).
- If 7<sup>th</sup> bit = zero → receiver recognizes it as a stuffed bit, discard it and resets the counter.
- If the 7<sup>th</sup> bit = 1 → then the receiver checks the 8<sup>th</sup> bit; If the 8<sup>th</sup> bit = 0, the sequence is recognized as a flag.

01111010 011111000 101101111 1010010

### 2.3.2 Control Field

The control field determines the type of frame and defines its functionality. Figure 2.18 shows the control field format for the different frame types.

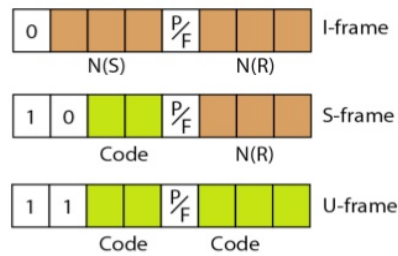


Figure 2.18 Control field format for the different frame types

### Control Field for I-Frames

- I-frames are designed to carry user data from the network-layer.
- In addition, they can include flow and error-control information (piggybacking).
- The subfields in the control field are:
  - (i) The first bit defines the type. If the first bit of the control field is 0, this means the frame is an I-frame.
  - (ii) The next 3 bits N(S) define the sequence-number of the frame. With 3 bits, we can define a sequence-number between 0 and 7.
  - (iii) The last 3 bits N(R) correspond to the acknowledgment-number when piggybacking is used.
  - (iv) The single bit between N(S) and N(R) is called the P/F bit. The P/F field is a single bit with a dual purpose. It can mean poll or final.
    - (a) It means poll when the frame is sent by a primary station to a secondary (when the address field contains the address of the receiver).
    - (b) It means final when the frame is sent by a secondary to a primary (when the address field contains the address of the sender).

### Control Field for S-Frames

- Supervisory frames are used for flow and error-control whenever piggybacking is either impossible or inappropriate (e.g., when the station either has no data of its own to send or needs to send a command or response other than an acknowledgment).
- S-frames do not have information fields.
- The subfields in the control field are:
  - (i) If the first 2 bits of the control field is 10, this means the frame is an S-frame.
  - (ii) The last 3 bits N(R) corresponds to the acknowledgment-number (ACK) or negative acknowledgment-number (NAK).
  - (iii) The 2 bits called code is used to define the type of S-frame itself. With 2 bits, we can have four types of S-frames:
    - (a) **Receive Ready (RR) = 00**
      - This acknowledges the receipt of frame or group of frames.

- The value of  $N(R)$  is the acknowledgment-number.

**(b) Receive Not Ready (RNR) = 10**

- This is an RR frame with 1 additional function.
- It announces that the receiver is busy and cannot receive more frames.
- It acts as congestion control mechanism by asking the sender to slow down.
- The value of  $N(R)$  is the acknowledgment-number.

**(c) ReJect (REJ) = 01**

- It is a NAK frame used in Go-Back-N ARQ to improve the efficiency of the process.
- It informs the sender, before the sender time expires, that the last frame is lost or damaged.
- The value of  $N(R)$  is the negative acknowledgment-number.

**(d) Selective REJect (SREJ) = 11**

- This is a NAK frame used in Selective Repeat ARQ.
- The value of  $N(R)$  is the negative acknowledgment-number.

**Control Field for U-Frames**

- Unnumbered frames are used to exchange session management and control information between connected devices.
- U-frames contain an information field used for system management information, but not user data.
- Much of the information carried by U-frames is contained in codes included in the control field.
- U-frame codes are divided into 2 sections:
  - (i) A 2-bit prefix before the P/F bit
  - (ii) A 3-bit suffix after the P/F bit.
- Together, these two segments (5 bits) can be used to create up to 32 different types of U-frames.

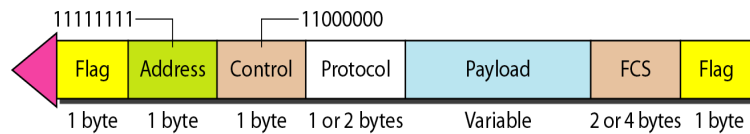
**2.3.3 Point-To-Point Protocol (PPP)**

- PPP is one of the most common protocols for point-to-point access.
- Today, millions of Internet users who connect their home computers to the server of an ISP use PPP.

**Framing**

- PPP uses a character-oriented (or byte-oriented) frame as shown in figure 2.19.





**Figure 2.19 PPP frame format**

### Various fields of PPP frame

#### i) Flag

- This field has a synchronization pattern 01111110.
- This field identifies both the beginning and the end of a frame.

#### ii) Address

- This field is set to the constant value 11111111 (broadcast address).

#### iii) Control

- This field is set to the constant value 00000011 (imitating unnumbered frames in HDLC).
- PPP does not provide any flow control.
- Error control is also limited to error detection.

#### iv) Protocol

- This field defines what is being carried in the payload field.
- Payload field carries either i) user data or ii) other control information.
- By default, size of this field = 2 bytes.

#### v) Payload field

- This field carries either i) user data or ii) other control information.
- By default, maximum size of this field = 1500 bytes.
- This field is byte-stuffed if the flag-byte pattern appears in this field.
- Padding is needed if the payload-size is less than the maximum size.

#### vi) FCS

- This field is the PPP error-detection field.
- This field can contain either a 2- or 4-byte standard CRC.

### Byte Stuffing

Since PPP is a byte-oriented protocol, the flag in PPP is a byte that needs to be escaped whenever it appears in the data section of the frame. The escape byte is 01111101, which means that every time the flag like pattern appears in the data, this extra byte is stuffed to tell the receiver that the next byte is not a flag. Obviously, the escape byte itself should be stuffed with another escape byte.

### Transition Phases

- The transition diagram starts with the dead state as shown in figure 2.20.

#### (a) Dead State

- In dead state, there is no active carrier and the line is quiet.

#### (b) Establish State

- When 1 of the 2 nodes starts communication, the connection goes into the establish state.
- In establish state, options are negotiated between the two parties.

#### (c) Authenticate State

- If the 2 parties agree that they need authentication, then the system needs to do authentication; otherwise, the parties can simply start communication.

#### (d) Open State

- Data transfer takes place in the open state.

#### (e) Terminate State

- When 1 of the endpoints wants to terminate connection, the system goes to terminate state.

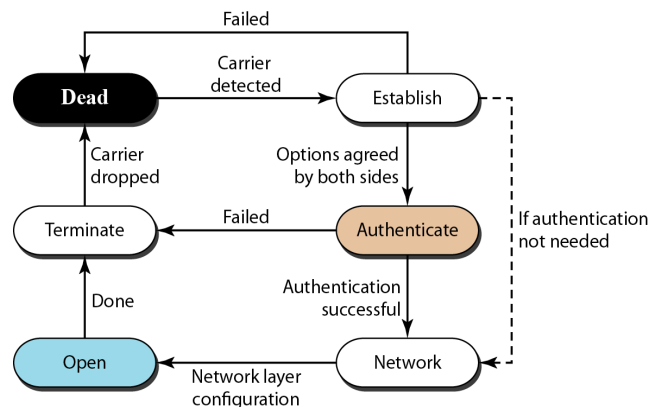


Figure 2.20 Transition phases

## 2.4. MEDIA ACCESS CONTROL

The two main functions of the data link layer are data link control and media access control. The data link control deals with the design and procedures for communication between two adjacent nodes: node-to-node communication. The second function of the data link layer is media access control, or how to share the link.

When nodes or stations are connected and use a common link, called a multipoint or broadcast link, we need a multiple-access protocol to coordinate access to the link. The upper sub-layer of the DLL that is responsible for flow and error control is called the logical link control (LLC) layer. The lower sub-layer that is mostly responsible for multiple access resolution is called the media access control (MAC) layer. Many formal protocols have been devised to handle access to a shared links; we categorize them into three groups.

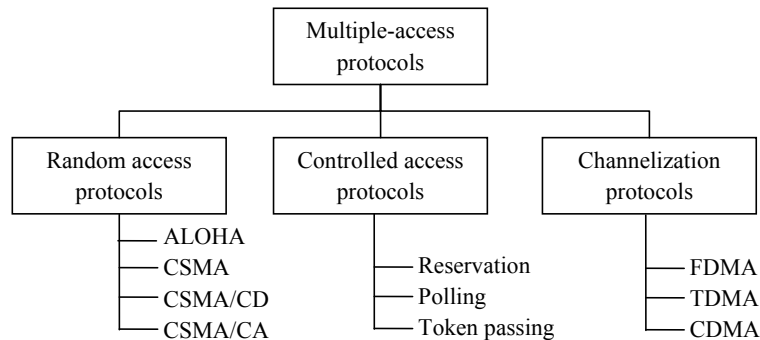


Figure 2.21 Taxonomy of multiple-access protocols

### 2.4.1 Random Access or Contention Method

In random access no station is superior to another station and none is assigned the control over another. A station that has data to send uses a procedure defined by the protocol to make a decision on whether or not to send. This decision depends on the state of the medium (idle or busy). Two features of random access are;

- (i) There is no scheduled time for a station to transmit. Transmission is random among the stations. That is why these methods are called random access.
- (ii) No rules specify which station should send next. Stations compete with one another to access the medium. That is why these methods are also called contention methods.

In a random access method, each station has the right to the medium without being controlled by any other station. If more than one station tries to send, there is an access conflict-collision-and the frames will be either destroyed or modified. To avoid access conflict or to resolve it when it happens, each station follows a procedure that answers the following questions:

- (1) When can the station access the medium?
- (2) What can the station do if the medium is busy?
- (3) How can the station determine the success or failure of the transmission?
- (4) What can the station do if there is an access conflict?

The random access method using ALOHA protocol which used a very simple procedure called multiple access (MA). The method was improved with the addition of a procedure that forces the station to sense the medium before transmitting. This was called carrier sense multiple access. This method later evolved into two parallel methods:

- (i) **Carriers sense multiple access with collision detection (CSMA/CD)** : CSMA/CD tells the station what to do when a collision is detected
- (ii) **Carrier sense multiple access with collision avoidance (CSMA/CA)**: CSMA/CA tries to avoid the collision.

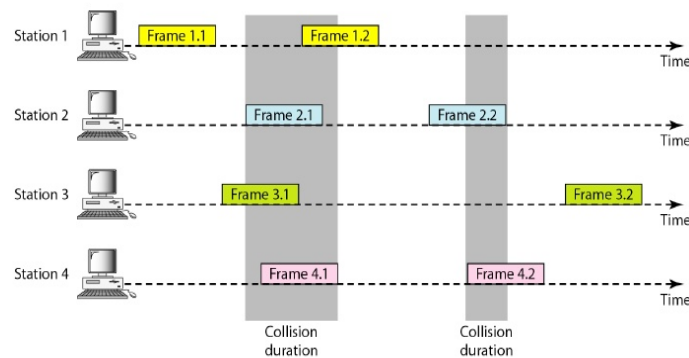
#### **ALOHA**

ALOHA, the earliest random access method was designed for a radio (wireless) LAN, but it can be used on any shared medium. When the medium is shared between the stations, the data from the two stations collide and become garbled.

**Pure ALOHA**

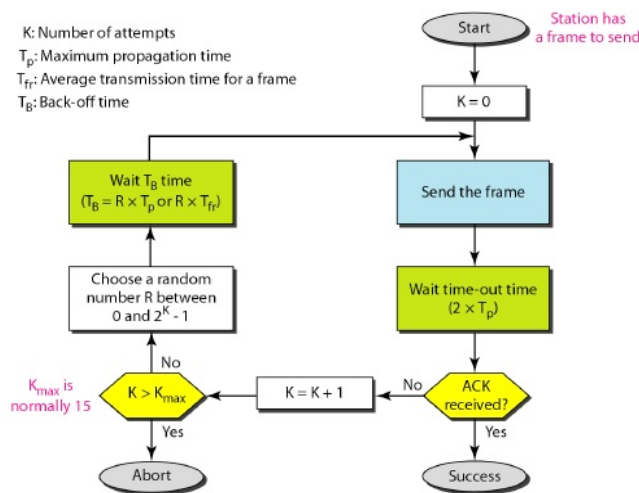
The original ALOHA protocol is called pure ALOHA. The idea is that each station sends a frame whenever it has a frame to send. When the channel is shared, there is the possibility of collision between frames from different stations. Figure 2.22 shows an example of frame collisions in pure ALOHA.

There are four stations that contend with one another for access to the shared channel. The figure 2.22 shows that each station sends two frames; there are a total of eight frames on the shared medium. Some of these frames collide because multiple frames are in contention for the shared channel. Only two frames survive: frame 1.1 from station 1 and frame 3.2 from station 3. The pure ALOHA protocol relies on acknowledgments from the receiver. If the acknowledgment does not arrive after a time-out period, the station assumes that the frame (or the acknowledgment) has been destroyed and resends the frame.



**Figure 2.22 Frames in a pure ALOHA network**

A collision involves two or more stations. If all these stations try to resend their frames after the time-out, the frames will collide again. Pure ALOHA dictates that when the time-out period passes, each station waits a random amount of time before resending its frame. The randomness will help avoid more collisions. We call this time the **back-off time TB**. Pure ALOHA has a 2nd method to prevent congesting the channel with retransmitted frames. After a maximum number of retransmission attempts  $K_{max}$  a station must give up and try later.



**Figure 2.23 Procedure for pure ALOHA protocol**

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  $T_{fr}$  to send. From figure 2.24, we see that the vulnerable time, during which a collision may occur in pure ALOHA, is 2 times the frame transmission time.

$$\text{Pure ALOHA vulnerable time} = 2 \times T_{fr}$$

The throughput for pure ALOHA is

$$S = G \times e^{-2G}.$$

The maximum throughput

$$S_{max} = 0.184 \text{ when } G = (1/2)$$

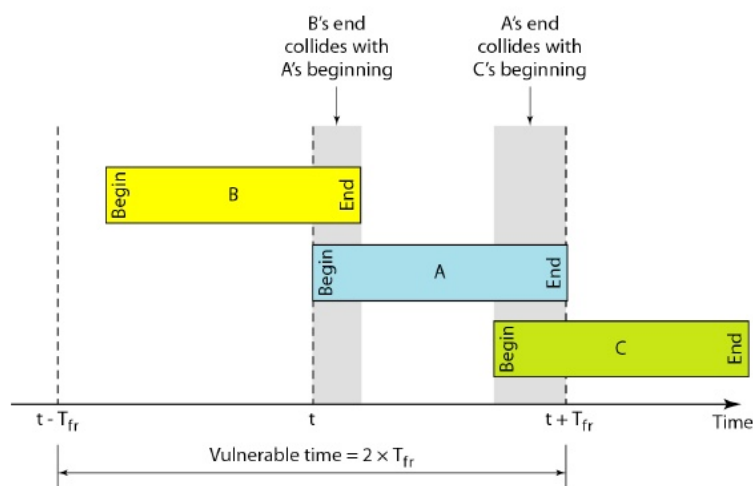


Figure 2.24 Vulnerable time for pure ALOHA protocol

### Slotted ALOHA

Pure ALOHA has a vulnerable time of  $2 \times T_{fr}$ . This is so because there is no rule that defines when the station can send. A station may send soon after another station has started or soon before another station has finished. Slotted ALOHA was invented to improve the efficiency of pure ALOHA. In slotted ALOHA we divide the time into slots of  $T_{fr}$  and force the station to send only at the beginning of the time slot. Figure 2.25 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. The vulnerable time is now reduced to one-half, equal to  $T_{fr}$ .

$$\text{Slotted ALOHA vulnerable time} = T_{fr}$$

### Throughput

It can be proved that the average number of successful transmissions for slotted ALOHA is,

$$S = G \times e^{-G}$$

The maximum throughput  $S_{max}$  is 0.368, when  $G = 1$ .

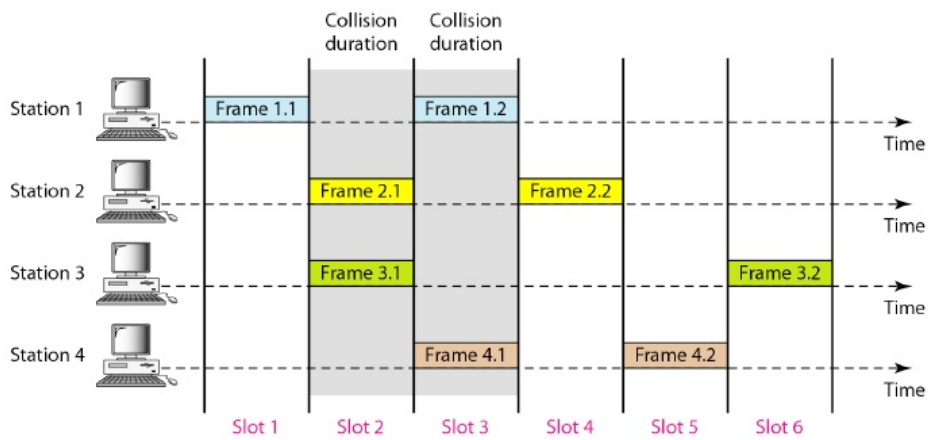


Figure 2.25 Frames in a slotted ALOHA network

### Carrier Sense Multiple Access (CSMA)

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 propagation delay (Stations are connected to a shared channel usually a dedicated medium). The possibility of collision still exists because of at time  $t_1$  station B senses the medium and finds it idle, so it sends a frame.

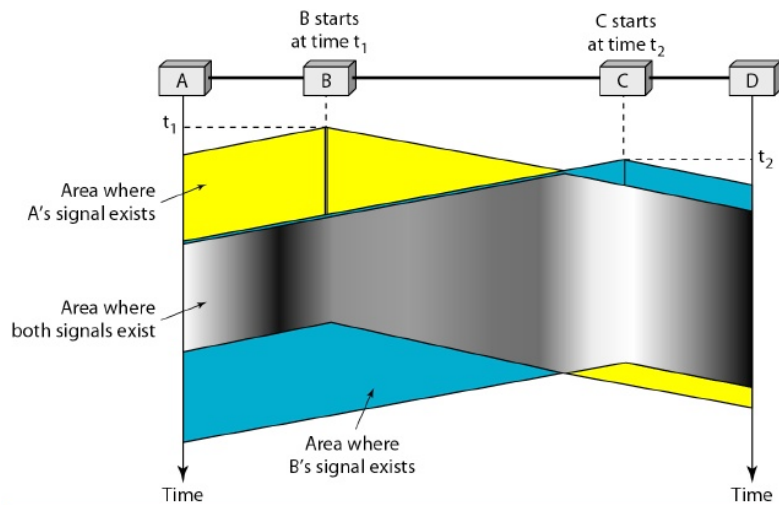


Figure 2.26 Space/time model of the collision in CSMA

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. So station C also sends a frame. The two signals collide and both frames are destroyed.

### 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.

### Persistence Methods

What should a station do if the channel is busy? What should a station do if the channel is idle? Three persistence methods have been devised to answer these questions:

- (i) 1-persistent method
- (ii) non-persistent method
- (iii) P-persistent method.

#### 1-Persistent

In this method, after the station finds the line idle, it sends its frame immediately (with probability 1). This method has the highest chance of collision because two or more stations may find the line idle and send their frames immediately.

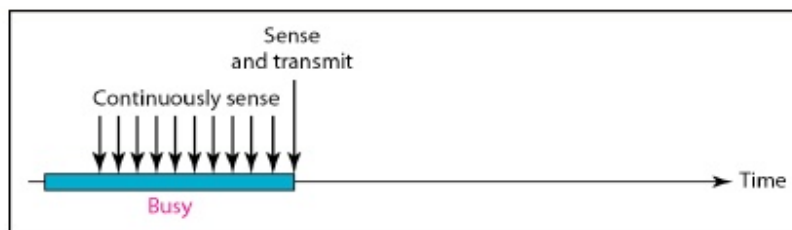


Figure 2.27 Behavior of 1-Persistence methods

#### Non-persistent

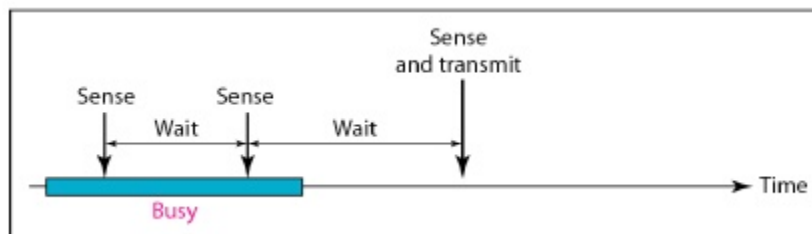


Figure 2.28 Behavior of Non-Persistence methods

In this method, a station that has a frame to send senses the line. If the line is idle, it sends immediately. If the line is not idle, it waits a random amount of time and then senses the line again. The non-persistent approach reduces the chance of collision. This method reduces the efficiency of the network because the medium remains idle when there may be stations with frames to send.

#### The P-persistent method

It is used if the channel has time slots with slot duration equal to or greater than the maximum propagation time. This approach reduces the chance of collision and improves efficiency. In this method, after the station finds the line idle it follows these steps:

- (1) With probability  $p$ , the station sends its frame.
- (2) With probability  $q = 1 - p$ , the station waits for the beginning of the next time slot and checks the line again.

- (a) If the line is idle, it goes to step 1.
- (b) If the line is busy, it acts as though a collision has occurred and uses the back-off procedure.

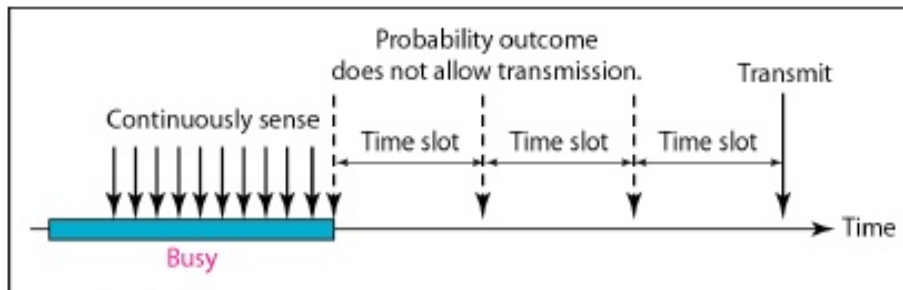


Figure 2.29 Behavior of P-Persistence methods

**Flow diagram for three persistence methods**

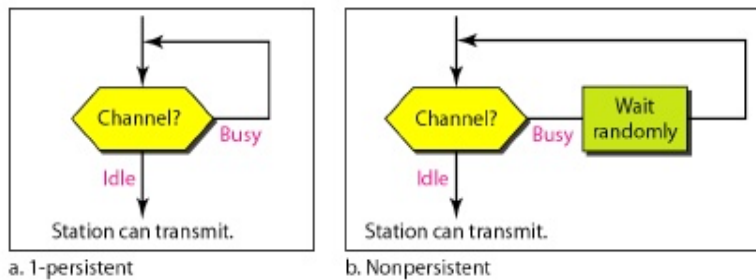


Figure 2.30 Flow diagram for three persistence methods

**Carrier sense multiple access with collision detection (CSMA/CD)**

CSMA/CD augments the algorithm to handle the collision. 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.

**Procedure**

We need to sense the channel before we start sending the frame by using one of the persistence processes. Transmission and collision detection is a continuous process. We do not send the entire frame (bit by bit). By sending a short jamming signal, we can enforce the collision in case other stations have not yet sensed the collision.

**Carrier sense multiple access with collision avoidance (CSMA/CA)**

CSMA/CA was invented to avoid collisions on wireless networks. Collisions are avoided through the use of CSMA/CA's three strategies:

- (i) The inter frame space (used to define the priority of a station)
- (ii) The contention window
- (iii) Acknowledgments



### Interframe Space (IFS)

When an idle channel is found, the station does not send immediately. It waits for a period of time called the interframe space or IFS. Even though the channel may appear idle when it is sensed, a distant station may have already started transmitting. The distant station's signal has not yet reached this station.

### Contention Window

The contention window is an amount of time divided into slots. A station that is ready to send chooses a random number of slots as its wait time. The station needs to sense the channel after each time slot. However, if the station finds the channel busy, it does not restart the process; it just stops the timer and restarts it when the channel is sensed as idle. This gives priority to the station with the longest waiting time.

### Acknowledgment

With all these precautions, there still may be a collision resulting in destroyed data, and the data may be corrupted during the transmission. The positive acknowledgment and the time-out timer can help guarantee that the receiver has received the frame.

## 2.4.2 Controlled Access

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

- (i) Reservation
- (ii) Polling
- (iii) Token passing

### Reservation

In the reservation method, a station needs to make a reservation before sending data. Time is divided into intervals. In each interval, a reservation frame precedes the data frames sent in that interval. Figure 2.31 shows a situation with five stations and a five-mini slot reservation frame. In the first interval, only stations 1, 3, and 4 have made reservations. In the second interval, only station 1 has made a reservation.

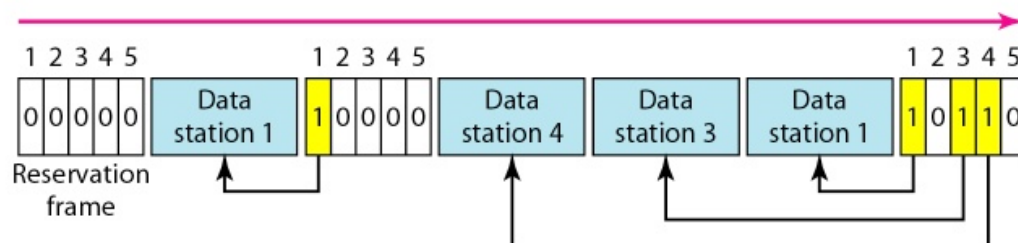


Figure 2.31 Reservation process in controlled access

### Polling

Here one device is designated as a primary station and the other devices are secondary stations. All data exchanges must be made through the primary device. The primary device controls the link; the secondary devices follow its instructions. The primary device is always the initiator of a session. If the primary wants to receive data it asks the secondary if they have anything to send; this is called poll function. If the primary wants to send data, it tells the secondary to get ready to receive; this is called select function.

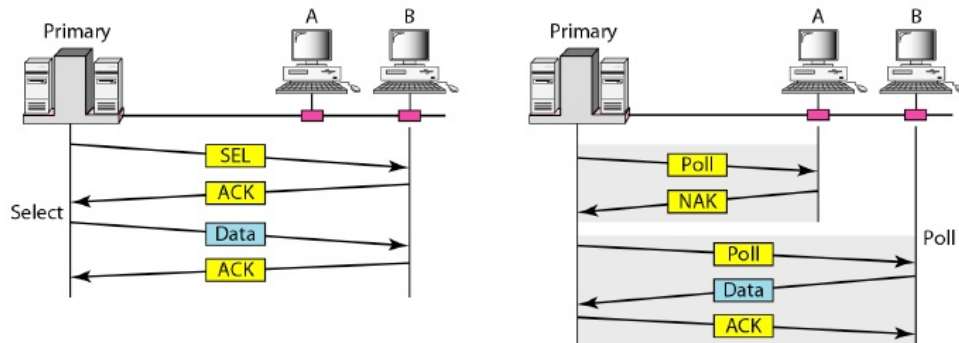


Figure 2.32 Polling in controlled access

### Token Passing

In the token-passing method, the stations in a network are organized in a logical ring. For each station, there is a predecessor and a successor.

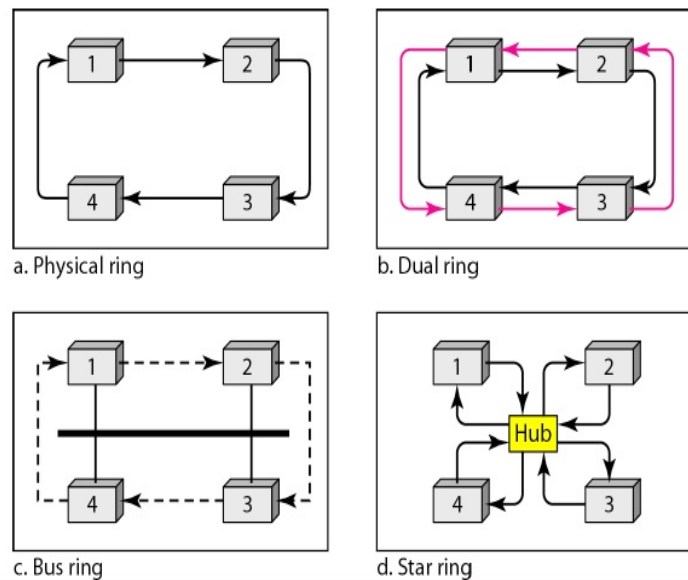


Figure 2.33 Token passing methods in controlled access

### 2.4.3. Channelization

Channelization is a multiple-access method in which the available bandwidth of a link is shared in time, frequency, or through code, between different stations. Three Channelization protocols are used. They are,

- (i) FDMA
- (ii) TDMA
- (iii) CDMA

### ***Frequency-division multiple access (FDMA)***

In frequency-division multiple access (FDMA), the available bandwidth is divided into frequency bands. Each station is allocated a band to send its data. Each band is reserved for a specific station, and it belongs to the station all the time. Each station also uses a band-pass filter to confine the transmitter frequencies.

To prevent station interferences, the allocated bands are separated from one another by small guard bands. FDMA specifies a predetermined frequency band for the entire period of communication (a continuous flow of data that may not be packetized).

### ***Time-Division Multiple Access (TDMA)***

In time-division multiple access (TDMA), the stations share the bandwidth of the channel in time. Each station is allocated a time slot during which it can send data. Each station transmits its data in assigned time slot.

The main problem with TDMA lies in achieving synchronization between the different stations. Each station needs to know the beginning of its slot and the location of its slot. This is difficult because of propagation delays introduced in the system if the stations are spread over a large area.

To compensate for the delays, we can insert guard times. Synchronization is normally accomplished by having some synchronization bits (normally referred to as preamble bits) at the beginning of each slot.

### ***Code-Division Multiple Access (CDMA)***

CDMA differs from FDMA because only one channel occupies the entire bandwidth of the link. It differs from TDMA because all stations can send data simultaneously; there is no timesharing. In CDMA, one channel carries all transmissions simultaneously.

## **2.5 ETHERNET (IEEE 802.3)**

A LAN can be used as an isolated network to connect computers in an organization for sharing resources. Most of the LANs today are linked to a wide area network (WAN) or the Internet. The LAN market has seen several technologies such as,

- (i) Ethernet
- (ii) Token Ring
- (iii) Token Bus
- (iv) FDDI
- (v) ATM LAN.

The IEEE Standard Project 802 is designed to regulate the manufacturing and interconnectivity between different LANs.

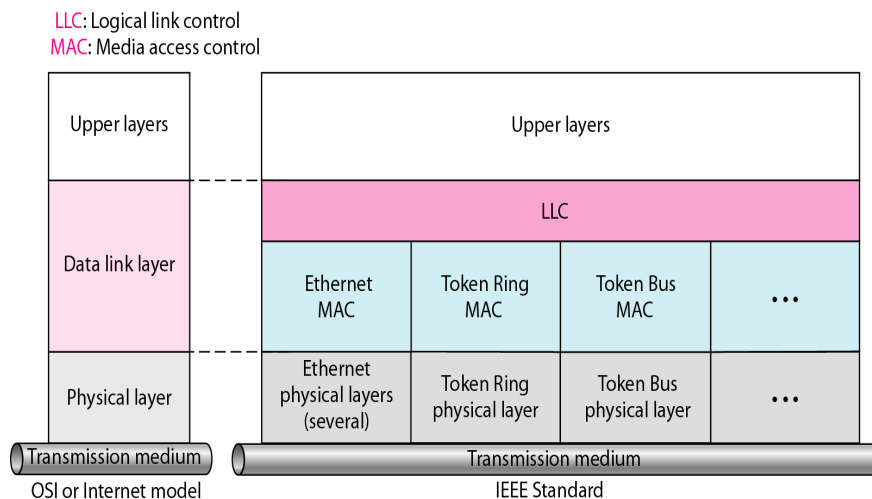
### 2.5.1 IEEE Standards

The IEEE 802 standard was adopted by the American National Standards Institute (ANSI). In 1987, the International Organization for Standardization (ISO) also approved it as an international standard. The relationship of the 802 Standard to the traditional OSI model is shown in figure 2.34. The IEEE has subdivided the data link layer into two sub layers:

- (i) Logical link control (LLC)
- (ii) Media access control (MAC).

The data link layer in the IEEE standard is divided into two sublayer. They are,

- (i) Logical Link Control (LLC)
- (ii) Media Access Control (MAC)



**Figure 2.34 IEEE standard for LANs**

#### **Logical Link Control (LLC)**

In IEEE Project 802, flow control, error control, and part of the framing duties are collected into a sublayer called the logical link control. Framing is handled in both the LLC sublayer and the MAC sublayer. The LLC provides a single data link control protocol for all IEEE LANs, but the MAC sublayer provides different protocols for different LANs. A single LLC protocol can provide interconnectivity between different LANs because it makes the MAC sublayer transparent.

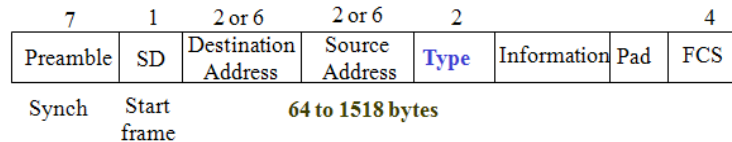
#### **Media Access Control (MAC)**

IEEE Project 802 has created a sublayer called media access control that defines the specific access method for each LAN. For example, it defines CSMA/CD as the media access method for Ethernet LANs and the token passing method for Token Ring and Token Bus LANs. A part of the framing function is also handled by the MAC layer. The MAC sublayer contains a number of distinct modules for defining the access method and the framing format specific to the corresponding

## MAC Sublayer

In standard Ethernet, the MAC sublayer governs the operation of the access method. It also frames the data received from the upper layer and passes them to the physical layer.

### Frame Format



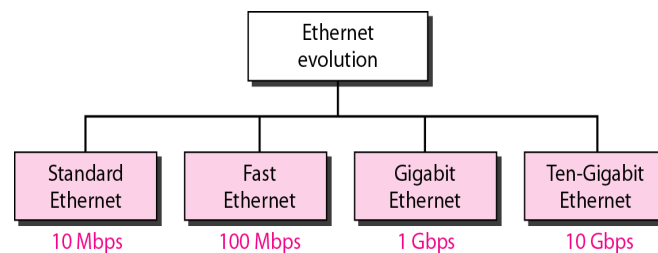
**Figure 2.35 Frame Format**

The Ethernet frame contains the following seven fields.

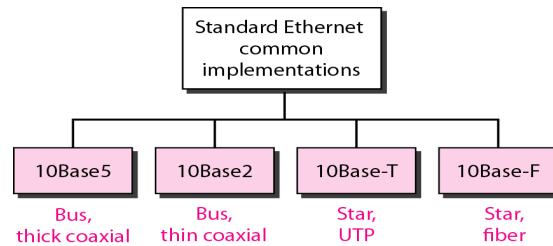
- (i) **Preamble:** 8 bytes with pattern 10101010 used to synchronize receiver, sender clock rates.
- (ii) **SD:** Eighth byte is used to indicate the start of frame (10101011)
- (iii) **Addresses:** The DA field is 6 bytes and contains the physical address of the destination station or stations to receive the packet. The Source address (SA) field is also 6 bytes and contains the physical address of the sender of the packet.
- (iv) **Type (DIX):** Indicates the type of the Network layer protocol being carried in the payload field (IP, IP (0800), Novell IPX (8137) and AppleTalk (809B), ARP (0806) )
- (v) **Length:** Number of bytes in the data field (Maximum 1500 bytes).
- (vi) **CRC:** Checked at receiver, if error is detected, the frame is discarded CRC-32.
- (vii) **Data:** Carries data encapsulated from the upper-layer protocols
- (viii) **Pad:** Zeros are added to the data field to make the minimum data length = 46 bytes

## 2.5.2 Standard Ethernet

Ethernet data link layer protocol provides connectionless service to the network layer. (No handshaking between sending and receiving machine). It also provides an unreliable service to the network layer. Here the receiver doesn't send ACK or NAK to sender. This means that the stream of datagram's passed to network layer can have gaps (missing data).



**Figure 2.36 Ethernet evolution**



**Figure 2.37** Categories of traditional Ethernet

### **10BASE5**

- Data transfer rate is 10 Mbps.
- 500 meter segment length.
- Signal regeneration can be done with help of repeaters.
- Thick Coax is used as a transmission medium.

#### **Advantages:**

- (i) Low attenuation,
- (ii) Excellent noise immunity
- (iii) Superior mechanical strength

#### **Disadvantages:**

- (i) Bulky
- (ii) Difficult to pull
- (iii) Transceiver boxes are too expensive
- (iv) Wiring represented a significant part of total installed cost.

### **10BASE2 (Cheaper net)**

- Data transfer rate is 10 Mbps
- 185 meter segment length.
- Signal regeneration can be done with help of repeaters.
- Transceiver was integrated onto the adapter.
- Thin Coax is used as a transmission medium.

#### **Advantages:**

- (i) Easier to install
- (ii) Reduced hardware cost
- (iii) BNC connectors widely deployed (lower installation costs).

**Disadvantages:**

- (i) Attenuation is not good
- (ii) Could not support as many stations due to signal reflection caused by BNC Tee Connector.

**10BaseT**

- Uses twisted pair Cat3 cable.
- Star-wire topology.
- A hub functions as a repeater with additional functions.

**Advantages:**

- (i) Fewer cable problems
- (ii) Easier to troubleshoot than coax.

**Disadvantages:**

- (i) Cable length at most 100 meters.

**1 BASE 5 (Star LAN)**

- Data transfer rate is 1 Mbps
- 250 meter segment length.
- Signal regeneration can be done with help of repeaters.
- Transceiver integrated onto the adapter.
- Implemented with the help of star topology
- Two pairs of unshielded twisted pair cable are used as a transmission media.

**Advantages:**

- (i) It is easier to use installed wiring in the walls.

**10BASE - T**

- Most popularly used.
- Data transfer rate is 10 Mbps.
- 100 meter segment length.
- Signal regeneration can be done with help of repeaters.
- Transceiver is integrated onto adapter.
- Two pairs of UTP cable are used as a transmission media.
- Implemented with the help of star topology (Hub in the closet).

***Advantages:***

- (i) Could be done without pulling new wires.
- (ii) Each hub amplifies and restores incoming signal.

***Hub Concept***

It is used to separate transmit and receive pair of wires. The repeater in the hub retransmits the signal received on any input pair onto all output pairs. The hub emulates a broadcast channel with collisions detected by receiving nodes.

**2.5.3 Changes in the Standard**

***2.5.3.1 Bridged Ethernet***

The first step in the Ethernet evolution was the division of a LAN by bridges. Bridges have two effects on an Ethernet LAN. They are,

- (i) Raise the bandwidth
- (ii) Separate collision domains.

***Raising the Bandwidth***

In an un-bridged Ethernet network, the total capacity (10 Mbps) is shared among all stations with a frame to send. The stations share the bandwidth of the network. For example, if two stations have a lot of frames to send, they probably alternate in usage. When one station is sending, the other one refrains from sending.

A bridge divides the network into two or more networks. Bandwidth-wise, each network is independent. For example, a network with 12 stations is divided into two networks, each with 6 stations. Now each network has a capacity of 10 Mbps. The 10-Mbps capacity in each segment is now shared between 6 stations not 12 stations.

In a network with a heavy load, each station is offered  $10/6$  Mbps instead of  $10/12$  Mbps. If we use a four-port bridge, each station is now offered  $10/3$  Mbps, which is 4 times more than an un-bridged network.

***Separating Collision Domains***

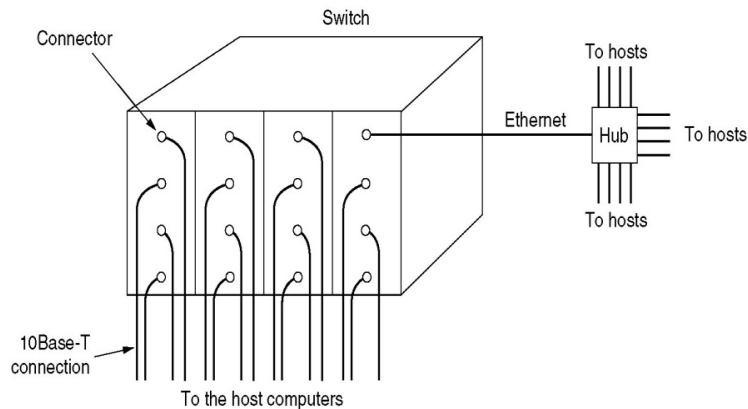
In the bridged network, the collision domain becomes much smaller and the probability of collision is reduced tremendously.

***2.5.3.2 Switched Ethernet***

The basic idea behind the switched Ethernet is to overcome the drawbacks of Hub concept. The switch learns destination locations by remembering the ports of the associated source address in a table. The switch may not have to broadcast to all output ports. It may be able to send the frame only to the destination port. A big performance advantage of a switch over a hub is that, more than one frame transfer can go through it concurrently.



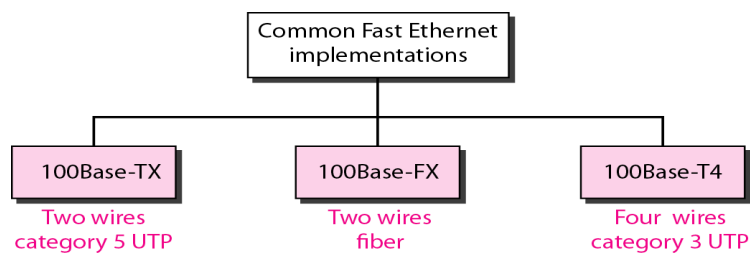
The advantage comes when the switched Ethernet backplane is able to repeat more than one frame in parallel (a separate backplane bus line for each node). The frame is relayed onto the required output port via the port's own backplane bus line. Under this scheme collisions are still possible when two concurrently arriving frames are destined for the same station. Each parallel transmission can take place at 10Mbps.



**Figure 2.38 Example of switched Ethernet**

### 2.5.3.3 Fast Ethernet

- Data transmission rate is 100 Mbps.
- Using the same frame format, media access, and collision detection rules as 10 Mbps Ethernet.
- It is possible to combine 10 Mbps Ethernet and Fast Ethernet on same network using a switch.
- Twisted pair (CAT 5) or fiber optic cable (no coax) can be used as a transmission media.
- Implemented with star-wire topology.



**Figure 2.39 Fast Ethernet implementations**

### 2.5.3.4 Gigabit Ethernet

- Data transmission rate is 1,000Mbps.
- Compatible with lower speeds.
- Uses standard framing and CSMA/CD algorithm.
- Distances are severely limited.

- Typically used for backbones and inter-router connectivity.
- Becoming cost competitive.
- Minimum frame length is 512 bytes
- Operates in full/half duplex modes mostly full duplex.
- In the full-duplex mode of Gigabit Ethernet, there is no collision.
- The maximum length of the cable is determined by the signal attenuation in the cable.

Name	Cable	Max. segment	Advantages
1000Base-SX	Fiber optics	550m	Multimode fiber (50, 62.5 microns)
1000Base-LX	Fiber optics	5000m	Single (10 $\mu$ ) or multimode (50, 62.5 $\mu$ )
1000Base-CX	2 pairs of STP	25m	Shielded twisted pair
1000Base-T	4 pairs of UTP	100m	Standard category 5 UTP

**Table 2.1 Gigabit Ethernet implementations**

### **10Gbps Ethernet**

- Maximum link distances cover 300 m to 40 km.
- Operates only on full-duplex mode.
- No CSMA/CD.
- Uses optical fiber only.

### **2.5.4 Experiences With Ethernet**

- Ethernets work best under light loads (Utilization over 30% is considered heavy).
- Network capacity is wasted by collisions
- Most networks are limited to about 200 hosts (Specification allows for up to 1024).
- Most networks are much shorter (5 to 10 microseconds RTT).
- Transport level flow control helps reduce load (number of back to back packets)
- Ethernet is inexpensive, fast and easy to administer.

### **Ethernet Problems**

- Ethernet's peak utilization is pretty low (like Aloha)
- Peak throughput worst with
  - More hosts:** More collisions needed to identify single sender.
  - Smaller packet sizes:** More frequent arbitration.
  - Longer links:** Collisions take longer to observe, more wasted bandwidth.
  - Efficiency is improved by avoiding these conditions.

### Why does Ethernet Win?

- (i) There are lots of LAN protocols
- (ii) Price
- (iii) Performance
- (iv) Availability
- (v) Ease of use
- (vi) Scalability

## 2.6. WIRELESS LAN

Wireless communication is one of the fastest-growing technologies because the demand for connecting devices without the use of cables is increasing everywhere. Wireless LANs can be found on college campuses, in office buildings, and in many public areas. IEEE 802.11 wireless LANs sometimes called wireless Ethernet. IEEE 802.11 operates on the physical and data link layers.

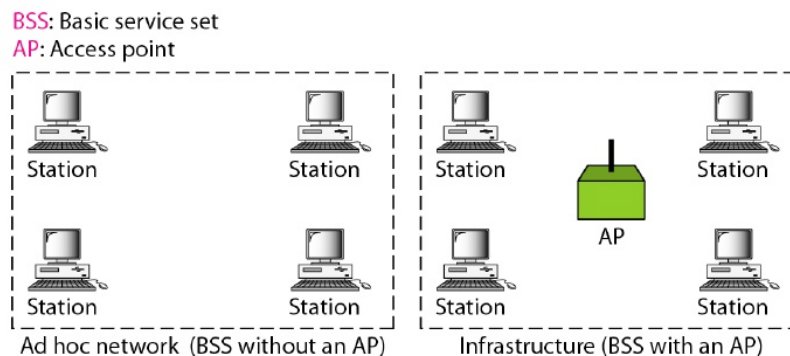
### 2.6.1 Architecture

IEEE 802.11 defines two kinds of services. They are,

- (i) Basic service set (BSS)
- (ii) Extended service set (ESS).

#### *Basic Service Set (BSS)*

BSS - the building block of a wireless LAN. A basic service set is made of stationary or mobile wireless stations and an optional central base station, known as the access point (AP). The BSS without an AP is a stand-alone network and cannot send data to other BSSs. It is called an ad hoc architecture. In this architecture, stations can form a network without the need of an AP; they can locate one another and agree to be part of a BSS. A BSS with an AP is sometimes referred to as an infrastructure network.



**Figure 2.40 Architecture of IEEE 802.11 (BSS)**

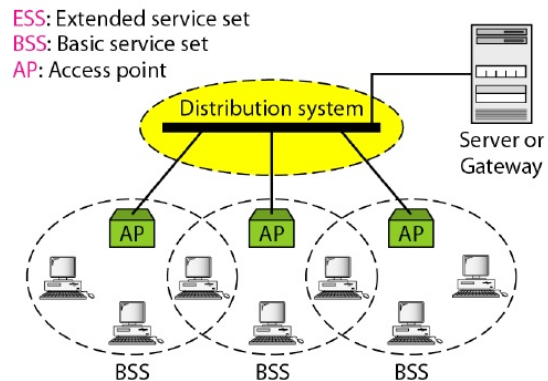
**Extended Service Set (ESS)**

An extended service set (ESS) is made up of two or more BSSs with APs. In this case, the BSSs are connected through a distribution system, which is usually a wired LAN such as an Ethernet. The distribution system connects the APs in the BSSs. The extended service set uses two types of stations. They are,

- (i) Mobile stations
- (ii) Stationary stations.

The mobile stations are normal stations inside a BSS. The stationary stations are AP stations that are part of a wired LAN.

When BSSs are connected, the stations within reach of one another can communicate without the use of an AP. However, communication between two stations in two different BSSs usually occurs via two APs.



**Figure 2.41 Architecture of IEEE 802.11 (ESS)**

**Station Types**

IEEE 802.11 defines three types of stations based on their mobility in a wireless LAN:

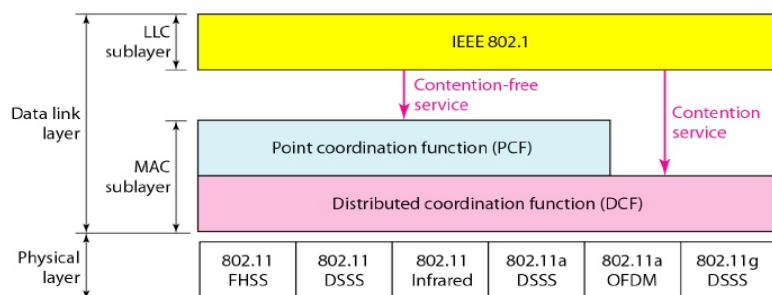
- (i) No-transition mobility
- (ii) BSS-transition mobility
- (iii) ESS-transition mobility.

A station with no-transition mobility is either stationary (not moving) or moving only inside a BSS. A station with BSS-transition mobility can move from one BSS to another, but the movement is confined inside one ESS. A station with ESS-transition mobility can move from one ESS to another. However, IEEE 802.11 does not guarantee that communication is continuous during the move.

**2.6.2 MAC Sublayer**

IEEE 802.11 defines two types of MAC sub-layers. They are;

- (i) The distributed coordination function (DCF)
- (ii) The point coordination function (PCF).



**Figure 2.42 MAC layers in IEEE 802.11 standard**

### Distributed Coordination Function

One of the two protocols defined by IEEE at the MAC sublayer is called the distributed coordination function (DCF). DCF uses CSMA/CA as the access method. Wireless LANs cannot implement CSMA/CD for the following three reasons:

- (i) For collision detection, a station must be able to send data and receive collision signals at the same time. This can mean costly stations and increased bandwidth requirements.
- (ii) Collision may not be detected because of the hidden station problem.
- (iii) The distance between stations can be great. Signal fading could prevent a station at one end from hearing a collision at the other end.

### Process Flowchart

The following figure 2.43 shows the process flowchart for CSMA/CA as used in wireless LANs. This includes the following steps;

- (i) Before sending a frame, the source station senses the medium by checking the energy level at the carrier frequency.
  - (a) The channel uses a persistence strategy with back-off until the channel is idle.
  - (b) After the station is found to be idle, the station waits for a period of time called the distributed interframe space (DIFS); then the station sends a control frame called the request to send (RTS).

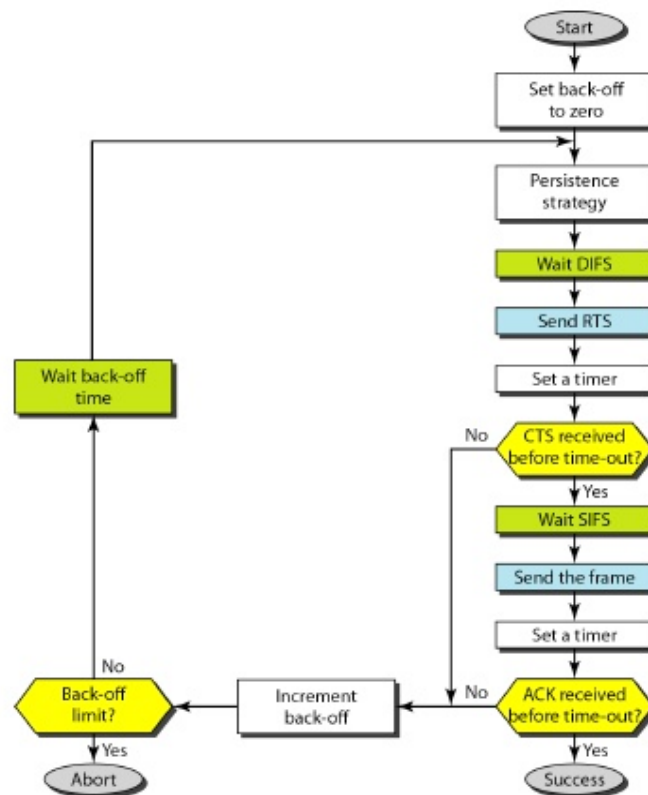


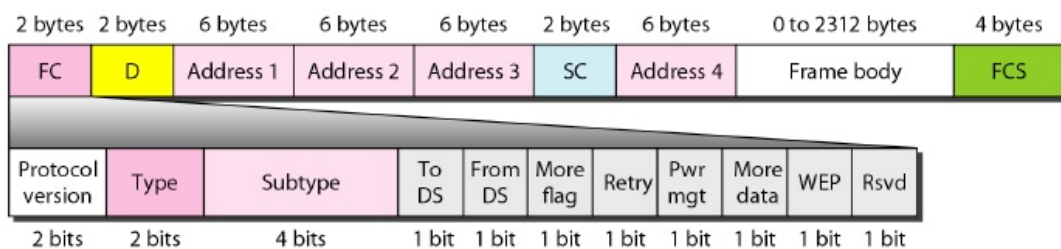
Figure 2.43 CSMA/CA flowchart

- (ii) After receiving the RTS and waiting a period of time called the short interframe space (SIFS), the destination station sends a control frame, called the clear to send (CTS), to the source station. This control frame indicates that the destination station is ready to receive data.
- (iii) The source station sends data after waiting an amount of time equal to SIFS.
- (iv) The destination station, after waiting an amount of time equal to SIFS, sends an acknowledgment to show that the frame has been received. Acknowledgment is needed in this protocol because the station does not have any means to check for the successful arrival of its data at the destination.

### ***Point Coordination Function (PCF)***

The point coordination function (PCF) is an optional access method that can be implemented in an infrastructure network (not in an ad hoc network). It is used mostly for time-sensitive transmission. PCF has a centralized, contention-free polling access method. The AP performs polling for stations that are capable of being polled. The stations are polled one after another, sending any data they have to the AP.

### ***Frame Format***



***Figure 2.44 Frame format***

The MAC layer frame consists of nine fields.

- (1) Frame control (FC) - The FC field is 2 bytes long and defines the type of frame and some control information.
- (2) D - In all frame types except one, this field defines the duration of the transmission. In the control frame - this field defines the ID of the frame.
- (3) Addresses - There are four address fields, each 6 bytes long. The meaning of each address field depends on the value of the To DS and From DS subfields.
- (4) Sequence control - This field defines the sequence number of the frame to be used in flow control.
- (5) Frame body - This field, which can be between 0 and 2312 bytes, contains information based on the type and the subtype defined in the FC field.
- (6) FCS - The FCS field is 4 bytes long and contains a CRC-32 error detection sequence.

Below table describes the subfields of the Frame control (FC) field and the Values of subfields in control frames.

Subtype	Meaning
1011	Request to send (RTS)
1100	Clear to send (CTS)
1100	Acknowledgement (ACK)

**Table 2.2 Values of subfields in control frames**

Field	Explanation
Version	Current version is 0
Type	Type of information: management (00), control (01), or data (10)
Subtype	Subtype of each type
To DS	Defined later
From DS	Defined later
More flag	When set to 1, means more fragments
Retry	When set to 1, means retransmitted frame
Pwr mgt	When set to 1, means station is in power management mode
More data	When set to 1, means station has more data to send
WEP	Wired equivalent privacy (encryption implemented)
Rsvd	Reserved

**Table 2.3 Subfields of the Frame control (FC) field**

### Frame Types

IEEE 802.11 has the following three categories of frames.

- (i) Management frames
- (ii) Control frames
- (iii) Data frames

Management frames are used for the initial communication between stations and access points. Control frames are used for accessing the channel and acknowledging. Data frames are used for carrying data and control information.

### 2.6.3 Addressing Mechanism

The IEEE 802.11 addressing mechanism specifies four cases, defined by the value of the two flags in the FC field, To DS and From DS. Each flag can be either 0 or 1, resulting in four different situations. The interpretation of the four addresses (address 1 to address 4) in the MAC frame depends on the value of these flags, as shown in below Table.

- Address 1 is always the address of the next device.
- Address 2 is always the address of the previous device.
- Address 3 is the address of the final destination station, if the address is not defined by address 1.
- Address 4 is the address of the original source station if it is not the same as address 2.

To DS	From DS	Address 1	Address 2	Address 3	Address 4
0	0	Destination	Source	BSS ID	N/A
0	1	Destination	Sending AP	Source	N/A
1	0	Receiving AP	Source	Destination	N/A
1	1	Receiving AP	Sending AP	Destination	Source

Table 2.4 Addresses

#### Four possible cases of addressing

**Case 1: 00** In this case,  $To DS = 0$  and  $From DS = 0$ .

This means that the frame is not going to a distribution system ( $To DS = 0$ ) and is not coming from a distribution system ( $From DS = 0$ ). The frame is going from one station in a BSS to another without passing through the distribution system. The ACK frame should be sent to the original sender.

**Case 2: 01** In this case,  $To DS = 0$  and  $From DS = 1$ .

This means that the frame is coming from a distribution system ( $From DS = 1$ ). The frame is coming from an AP and going to a station. The ACK should be sent to the AP. Note that address 3 contains the original sender of the frame (in another BSS).

**Case 3: 10** In this case,  $To DS = 1$  and  $From DS = 0$ .

This means that the frame is going to a distribution system ( $To DS = 1$ ). The frame is going from a station to an AP. The ACK is sent to the original station. Note that address 3 contains the final destination of the frame (in another BSS).

**Case 4: 11** In this case,  $To DS = 1$  and  $From DS = 1$ .

In this case the frame is going from one AP to another AP in a wireless distribution system. Here, we need four addresses to define the original sender, the final destination, and two intermediate APs.

#### 2.6.4 Physical Layer

All implementations, except the infrared, operate in the industrial, scientific, and medical (ISM) band, which defines three unlicensed bands in the three ranges: 902-928 MHz, 2.400-4.835 GHz, and 5.725-5.850 GHz. We discuss six specifications, as shown in Below Table.

IEEE	Technique	Band	Modulation	Rate (Mbps)
802.11	FHSS	2.4 GHz	FSK	1 and 2
	DSSS	2.4 GHz	FSK	1 and 2
		Infrared	PPM	1 and 2
802.11 a	OFDM	5.725 GHz	PSK or QAM	6 to 54
802.11 b	DSSS	2.4 GHz	PSK	5.5 and 11
802.11 g	OFDM	2.4 GHz	Different	22 to 54

Table 2.5 Physical layers



**IEEE 802.11 FHSS**

- It uses the frequency-hopping spread spectrum (FHSS) method.
- FHSS uses the 2.4 GHz ISM band.
- The band is divided into 79 sub-bands of 1 MHz (and some guard bands).
- A pseudorandom number generator selects the hopping sequence.
- The modulation technique in this specification is either two-level FSK or four-level FSK with 1 or 2 bits/ baud, which results in a data rate of 1 or 2 Mbps,

**IEEE 802.11 DSSS**

- DSSS uses the direct sequence spread spectrum (DSSS) method.
- DSSS uses the 2.4-GHz ISM band.
- The modulation technique in this specification is PSK at 1 Mbaud/s.
- The system allows 1 or 2 bits/ baud which results in a data rate of 1 or 2 Mbps,

**IEEE 802.11 Infrared**

- IEEE 802.11 infrared uses infrared light in the range of 800 to 950 nm.
- The modulation technique is called pulse position modulation (PPM).
- For a 1-Mbps data rate, a 4-bit sequence is first mapped into a 16-bit sequence in which only one bit is set to 1 and the rest are set to 0.
- For a 2-Mbps data rate, a 2-bit sequence is first mapped into a 4-bit sequence in which only one bit is set to 1 and the rest are set to 0.
- The mapped sequences are then converted to optical signals; the presence of light specifies 1, the absence of light specifies 0

**IEEE 802.11a – OFDM**

- IEEE 802.11a OFDM describes the orthogonal frequency-division multiplexing (OFDM) method for signal generation in a 5-GHz ISM band.
- OFDM is similar to FDM with one major difference: All the subbands are used by one source at a given time.
- The band is divided into 52 subbands, with 48 subbands for sending 48 groups of bits at a time and 4 subbands for control information.
- OFDM uses PSK and QAM for modulation.
- The common data rates are 18 Mbps (PSK) and 54 Mbps (QAM).

**IEEE 802.11b DSSS**

- IEEE 802.11 b DSSS describes the high-rate direct sequence spread spectrum (HRDSSS) method for signal generation in the 2.4-GHz ISM band.
- HR-DSSS is similar to DSSS except for the encoding method, which is called complementary code keying (CCK).
- CCK encodes 4 or 8 bits to one CCK symbol.
- HR-DSSS defines four data rates: 1, 2, 5.5, and 11 Mbps.

- The first two use the same modulation techniques as DSSS.
- The 5.5-Mbps version uses BPSK and transmits at 1.375 Mbaud/s with 4-bit CCK encoding.
- The 11-Mbps version uses QPSK and transmits at 1.375 Mbps with 8-bit CCK encoding.

### **IEEE 802.11g**

- This new specification using the OFDM with 2.4-GHz ISM band and forward error correction method.
- The modulation technique achieves a 22- or 54-Mbps data rate.

## **2.7 BLUETOOTH**

Bluetooth is a wireless LAN technology designed to connect devices of different functions such as telephones, notebooks, computers, cameras, printers, coffee makers, and so on. A Bluetooth LAN is an ad hoc network, which means that the network is formed spontaneously.

The device sometimes called gadgets, find each other and make a network called a piconet. A Bluetooth LAN can even be connected to the Internet if one of the gadgets has this capability. A Bluetooth LAN, by nature, cannot be large. If there are many gadgets that try to connect, there is confusion.

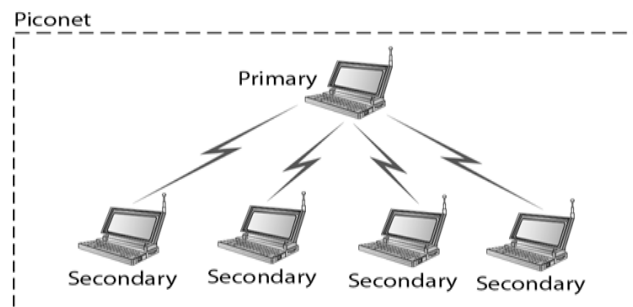
### **2.7.1 Architecture**

Bluetooth defines two types of networks.

- (i) Piconet
- (ii) Scatternet.

#### ***Piconet***

- A Bluetooth network is called a piconet, or a small net.
- The communication between the primary and the secondary can be one-to-one or one-to-many.
- It can have up to eight stations, one of which is called the master; the rest are called slaves.
- Maximum of seven slaves and only one master.



**Figure 2.45 Piconet**

- Slaves synchronize their clocks and hopping sequence with the master.
- But an additional eight slaves can stay in parked state, which means they can be synchronized with the master but cannot take part in communication until it is moved from the parked state.

### Scatternet

- Piconets can be combined to form what is called a scatternet.
- This station can receive messages from the primary in the first piconet (as a secondary) and, acting as a primary, deliver them to secondaries in the second piconet.
- A slave station in one piconet can become the master in another piconet.
- A Bluetooth device has a built-in short-range radio transmitter.

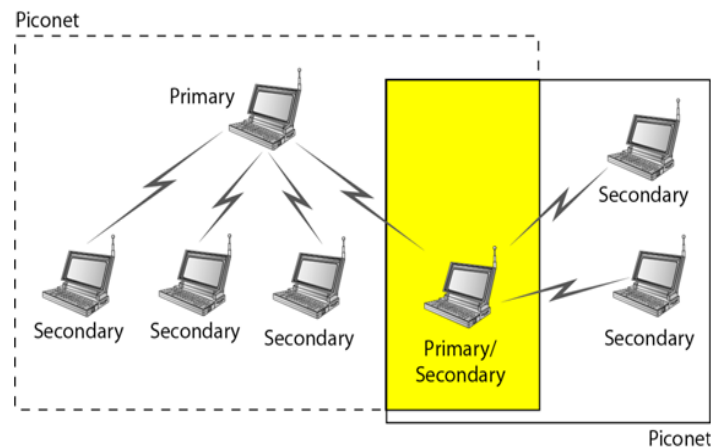


Figure 2.46 Scatternet

### 2.7.2 Bluetooth Layers

Bluetooth uses several layers that do not exactly match those of the Internet model. Bluetooth devices are low-power and have a range 10 centimeters to 10 meters. Bluetooth uses a 2.4-GHz ISM band divided into 79 channels of 1 MHz each.

#### (i) Radio Layer

- Roughly equivalent to physical layer of the Internet model. Physical links can be synchronous or asynchronous.
- Uses Frequency-hopping spread spectrum [Changing frequency of usage].
- Changes its modulation frequency 1600 times per second.
- Uses frequency shift keying (FSK) with Gaussian bandwidth filtering to transform bits to a signal.

#### (ii) Baseband layer

- Roughly equivalent to MAC sublayer in LANs. Access is using Time Division (Time slots).

- Length of time slot = dwell time = 625 microsecond. So, during one frequency, a sender sends a frame to a slave, or a slave sends a frame to the master.
- Time division duplexing TDMA (TDD-TDMA) is a kind of half-duplex communication in which the slave and receiver send and receive data, but not at the same time (half-duplex).
- However, the communication for each direction uses different hops, like walkie-talkies.

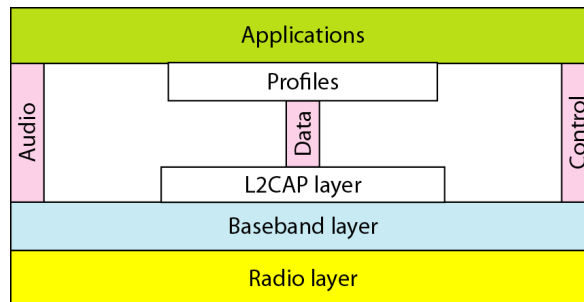


Figure 2.47 Blue tooth layers

### 2.7.2.1 Single-secondary communication

- Also called Single-slave communication
- If the piconet has only one secondary, the TDMA operation is very simple.
- The time is divided into slots of 625 micro seconds.

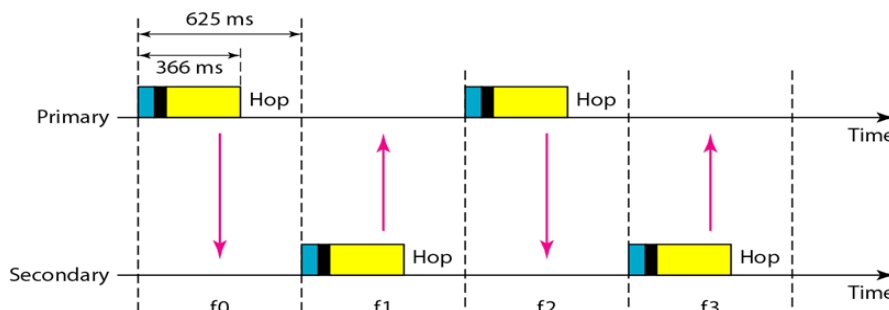


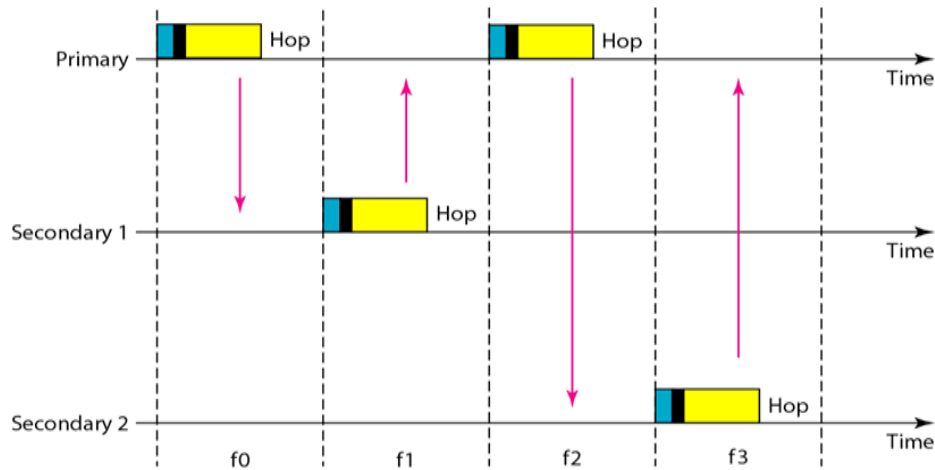
Figure 2.48 Single-secondary communications

- The primary uses even numbered slots (0, 2, 4 ...) and the secondary uses odd-numbered slots (1, 3, 5 ...).
- TDD-TDMA allows the primary and the secondary to communicate in half-duplex mode.
- In slot 0, the primary sends, and the secondary receives; in slot 1, the secondary sends, and the primary receives. The cycle is repeated.

### 2.7.3 Multiple-Secondary Communication

- Also called Multiple-slave communication (If there is more than one secondary in the piconet).
- Master uses even-numbered slots.

- Slave sends in the next odd-numbered slot if the packet in the previous slot was addressed to it.
- The below figure 2.49 shows a multiple-secondary communication scenario.



**Figure 2.49 Multiple-secondary communications**

Let us elaborate on the figure 2.49.

- (1) In slot 0, the primary sends a frame to secondary 1.
- (2) In slot 1, only secondary 1 sends a frame to the primary because the previous frame was addressed to secondary 1; other secondaries are silent.
- (3) In slot 2, the primary sends a frame to secondary 2.
- (4) In slot 3, only secondary 2 send a frame to the primary because the previous frame was addressed to secondary 2; other secondaries are silent.
- (5) The cycle continues.
  - We can say that this access method is similar to a poll/select operation with reservations.
  - When the primary selects a secondary, it also polls it. The next time slot is reserved for the polled station to send its frame.
  - If the polled secondary has no frame to send, the channel is silent.

## 2.7.4 Physical Links

Two types of links can be created between a primary and a secondary: SCQ links and ACL links. SCQ is used for real-time audio where avoiding delay is all-important. A secondary can create up to three SCQ links with the primary, sending digitized audio (PCM) at 64 kbps in each link.

### (i) Synchronous connection-oriented (SCQ)

- Latency is important than integrity.
- Transmission using slots.
- No retransmission.

**(ii) Asynchronous connectionless link (ACL)**

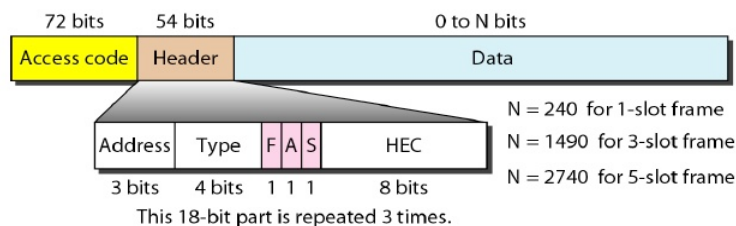
- Integrity is important than latency.
- Does like multiple-slave communication.
- Retransmission is done.

**2.7.5 Frame Format**

A frame in the baseband layer can be one of three types: one-slot, three-slot, or five-slot.

A slot, as we said before, is 625 micro seconds.

- (1) In a one-slot frame exchange, 259 micro seconds is needed for hopping and control mechanisms. The size of a one-slot frame is 366 (625 – 259) bits.
- (2) A three-slot frame occupies three slots. Since 259 micro seconds is used for hopping, the length of the frame is  $3 \times 625 - 259 = 1616$  micro seconds or 1616 bits.
- (3) A five-slot frame also uses 259 bits for hopping, which means that the length of the frame is  $5 \times 625 - 259 = 2866$  bits.



**Figure 2.50 Format of the three frame types**

The frame includes the following fields.

- (1) **Access code:** This 72-bit field normally contains synchronization bits and the identifier of the primary to distinguish the frame of one piconet from another.
- (2) **Header:** This 54-bit field is a repeated 18-bit pattern. Each pattern has the following subfields:
  - (i) **Address:** The 3-bit address subfield can define up to seven secondaries (1 to 7). If the address is zero, it is used for broadcast communication from the primary to all secondaries.
  - (ii) **Type:** The 4-bit type subfield defines the type of data coming from the upper layers.
  - (iii) **F:** This 1-bit subfield is for flow control. If it is set to 1, it indicates that the device is unable to receive more frames (buffer is full).
  - (iv) **A:** This 1-bit subfield is for acknowledgment. Bluetooth uses Stop-and-Wait ARQ; 1 bit is sufficient for acknowledgment.
  - (v) **S:** This 1-bit subfield holds a sequence number. Bluetooth uses Stop-and-Wait ARQ; 1 bit is sufficient for sequence numbering.
  - (vi) **HEC:** The 8-bit header error correction subfield is a checksum to detect errors in each 18-bit header section.

- (3) **Payload:** This subfield can be 0 to 2740 bits long. It contains data or control information coming from the upper layers.

### 2.7.6 L2CAP (Logical Link Control and Adaptation Protocol)

- Equivalent to LLC sublayer in LANs.
- Used for data exchange on ACL Link. SCQ channels do not use L2CAP.
- Frame format contains following three fields: Length, Channel ID, Data and Control.
- L2CAP can do Multiplexing, segmentation and Reassembly, QoS and group management.



Figure 2.51 L2CAP data packet format

## 2.8. NETWORKING PROTOCOLS FOR INTERNET OF THINGS

Decades ago, only computers have the abilities to communicate over the Internet. After continuous development of modern technologies, the future trend is that any object in the real world can interact with one another to exchange messages through the Internet, so that management and communication can be easily carried on. The idea for all objects tied together is called Internet of Things (IoT). Any object including computers, mobile phones, and sensors will all have a unique IP address connecting to the Internet. In large scale IoT deployment there will be rich combination of sensors and intelligent management schemes.

As a result, the research of wireless sensor networks (WSNs) and related technology played a very important role in IoT. WSN has wide applications in many fields such as smart energy, smart logistics, health care, home automation and so forth. To support these applications, a **protocol IEEE 802.15.4** was proposed for wireless personal area network communication. It specifies the physical layer and data link layer, *with short distance transmission, low power consumption and low cost characteristics*.

Based on IEEE 802.15.4, **ZigBee is a protocol widely used in smart grids**. It deals with the upper network layer and application layer. Because ZigBee was designed for local networks in home environments, it does not directly communicate with servers on the Internet. If administrators want to remotely control ZigBee devices through the Internet, or ZigBee devices need to send collected data back to a managing server on the Internet, an additional mechanism is required.

For example, a **gateway can be deployed to connect a ZigBee network to the Internet**. In a ZigBee network, end devices collect data and send data to the gateway, which then translates the data from ZigBee protocol format to Internet Protocol format, and vice versa. This allows ZigBee devices to communicate with servers on the Internet. With rapid development of wireless network applications, how to efficiently manage WSN devices and monitor the status of WSNs is a very important topic.

The abovementioned gateway mechanism only facilitates sending data from ZigBee networks to the Internet. On the other hand, it does not provide easy mechanisms to manage WSN devices from the Internet. **The Session Initiation Protocol (SIP) is an application layer protocol used to manage WSNs**.

## 2.8.1 6LoWPAN and Zigbee Protocols

### What is ZigBee?

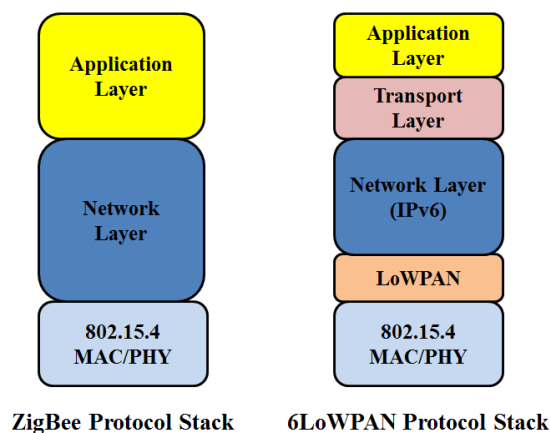
ZigBee, like 6LoWPAN, is designed for low data-rate and battery-powered applications. ZigBee is the most popular, low-cost, low-power wireless mesh networking standard on the market right now—and the more mature technology of the two (ZigBee, 6LoWPAN). It is typically implemented for personal or home-area networks, or in a wireless mesh for networks that operate over longer ranges. The ZigBee IP is built on the IEEE 802.15.4 standard, but unlike 6LoWPAN, it cannot easily communicate with other protocols. A benefit of ZigBee, however, is that nodes can stay in sleep mode most of the time, drastically extending battery life.

### What is 6LoWPAN?

6LoWPAN combines the latest version of the *Internet Protocol (IPv6) and Low-power Wireless Personal Area Networks (LoWPAN)*. 6LoWPAN, therefore, allows for the smallest devices with limited processing ability to transmit information wirelessly using an internet protocol. It's the newest competitor to ZigBee. The concept was created because engineers felt like the smallest devices were being left out from the Internet of Things. 6LoWPAN can communicate with 802.15.4 devices as well as other types of devices on an IP network link like WiFi. A bridge device can connect the two.

The physical layer and data link layer of ZigBee are based on the existing IEEE 802.15.4 protocol, in order to achieve the goal of low-power and low-energy consumption. However, for the upper layer protocols (network layer and application layer), many development experiences show that ZigBee has some technical shortcomings, such as address allocation, scalability, management tools, routing mechanisms, and interoperability with the Internet.

One competitive alternative to ZigBee is 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks). As shown in Figure 2.52, at the physical layer and the data link layer, it uses the same IEEE 802.15.4 protocol as ZigBee. For the network layer, it uses Internet Protocol version 6 (IPv6). It supports 2<sup>128</sup> IP addresses, so the numbers of addresses are more than sufficient. Even if there are a large number of devices deployed in a WSN, each device can also be assigned with a unique IP address. This feature makes it *easy to support end-to-end communication*.



**Figure 2.52 ZigBee / 6LoWPAN protocol stack**



## 2.8.2 Comparison of ZigBee with IPv6

### (i) Compatibility with Internet

If no extra conversion mechanism is deployed, ZigBee devices can not directly communicate with devices on the Internet. Currently there is no perfect solution for ZigBee/IP conversion. Proposed conversion mechanisms such as SOAP / REST, GRIP and tunnel mechanism, will all impose extra cost, which will increase the total cost of the network. On the contrary, if WSN devices can support IP, it can directly communicate with servers on the Internet. This does not require any application-layer translation, so the cost and efficiency will be greatly improved.

### (ii) Address allocation

When a ZigBee node joins a network, its parent node will arbitrarily assign an unused random address to the newly added ZigBee node. From the perspective of network management, the randomly assigned ZigBee address is difficult to control. IPv6 has two address allocation approaches.

- Stateful auto-configuration mode, which utilizes DHCPv6 (Dynamic Host Configuration Protocol for IPv6) to assign addresses for specific devices. This makes it easy to manage sensors in a large deployment.
- Stateless Auto-configuration mode, in which the device can use EUI-64 method to obtain its own IPv6 address from the MAC address of its network interface card.

### (iii) Network management

When the number of devices in a WSN gradually increases, and the deployment range gets larger, it is important to have a good tool to monitor and analyze the network. Otherwise, it is quite easy to waste lots of time and effort in trouble-shooting trivial problems. Currently, a few software tools were developed for specific ZigBee platforms, such as *ZigBee Sensor Monitor* developed by Texas Instruments.

This tool supports ZigBee CC2530ZDK module delivered by Texas Instrument. It can also display ZigBee network topology and the temperature collected by sensors. ZigBee Operator is developed by the company Serial Port Tool; it can be used to manage WSNs built by Digi's XBee ZigBee modules. It can read the module information, set module parameters, and show the network topology.

ZigBee may use the *SNMP (Simple Network Management Protocol)* as an IP-based network management standard which can *collect, modify and exchange network management information between network devices*. With SNMP, it is easy to monitor and manage network devices. SIP (Session Initiation Protocol) is a common protocol used in voice and video communications. In addition to making phone calls and starting video conferences, SIP can also display the on-line status of friends, and deliver instant text messages. It can also be applied to network management.

### (iv) Routing

There are two ZigBee network routing protocols: *Tree routing and AODV (Ad hoc On-Demand Distance Vector Routing)*.

- *Tree routing mechanism* is suitable for stationary devices or less mobile devices. The disadvantage is that the chosen path may not be optimal. Moreover, after a single node fails in the path, the data will not be sent to the destination. Therefore, it is less used.

- **Ad hoc On-Demand Distance Vector Routing** is suitable for mobile devices because it has the route repair mechanism. When the routing path between two nodes fails, its route repair mechanism uses broadcasting to discover a new path.

### 2.8.3 System Architecture

Smart grid is a new generation of electric power network which uses **advanced metering infrastructure (AMI)** to monitor and control power plants, substations, and power transmission lines. It can clearly oversee the status of the entire electrical power network, and adjust electric power scheduling for devices to increase energy utilization efficiency. How to efficiently transmit electricity related data (such as the power consumption of users) back to the back-end server, so that the server can make decisions to adjust the power network accordingly. For example, if a portion of the power network is suffering severe power loss, this implies there must be something wrong with those electrical devices or transmission lines. Some actions must be taken quickly to remedy the problem.

#### ZigBee-based smart grid system architecture

ZigBee is a popularly adopted communication technology in smart grid systems. There are three types of devices in a ZigBee network: **a coordinator, routers, and end devices**. Figure 2.53 shows typical ZigBee-based smart grid system architecture and the protocol stacks for each node.

- A **coordinator** is responsible for establishing, maintaining, and controlling a ZigBee network. It allocates network addresses to other nodes which join the network successively.
- **Routers**, which are sometimes called relay nodes, take care of data transmission and have capability to extend the scope of a ZigBee network.
- **End devices** collect data and transmit then tor routers or coordinators.

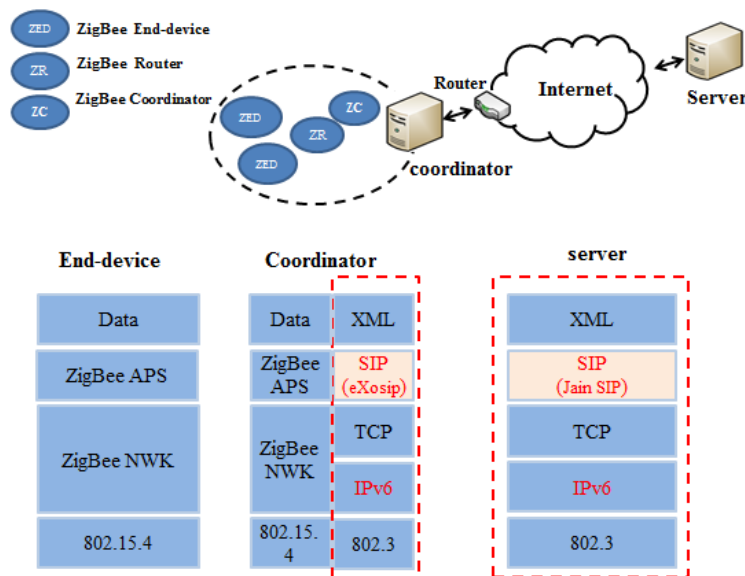


Figure 2.53 ZigBee-based smart grid system architecture

### 6LoWPAN-based smart grid system architecture

According to the comparison of ZigBee with IPv6, the IP-based WSN allows us to manage devices over the Internet easily. Although many current smart grid systems are built using the ZigBee system architecture shown in Figure 2.53. In Figure 2.54, all sensor nodes are required to support 6LoWPAN. Upon IPv6, the SIP application layer protocol is used to control end devices. The gateway (edge router in Figure 2.54) between wired and wireless networks only needs to handle IP packets forwarding, while upper application layers do no need to do any protocol conversion. This significantly reduces the loading of this intermediate gateway.

With this improvement, both types of smart grid systems, no matter they are based on 6LoWPAN or ZigBee, can be managed with the same SIP protocol. Moreover, 6LoWPAN devices have public IPv6 addresses, so servers can directly communicate with the end devices by their addresses, and easily discover the whole WSN topology. When any WSN device breaks down, the server can quickly notice that. Servers can collect data directly from end devices, without waiting the coordinator to handle the requests. In contrast, a ZigBee network is managed by a coordinator which must perform application-layer protocol translations and send data to servers. Therefore, this imposes heavy burdens on coordinators, and will easily cause data loss and transmission latency. Moreover, suppose a coordinator failed, the ZigBee network would be completely unable to communication with the Internet.

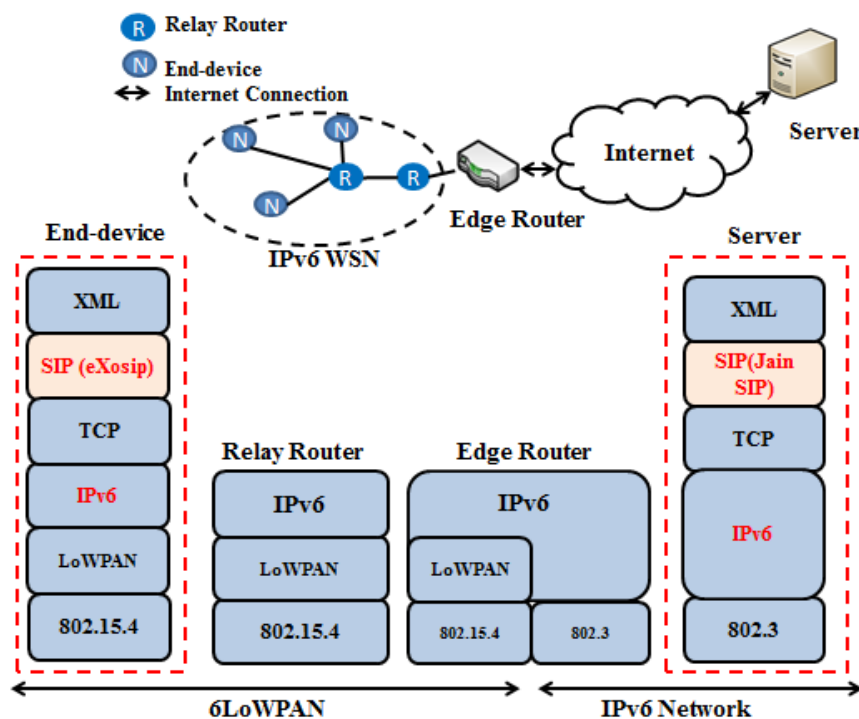


Figure 2.54 6LoWPAN-based smart grid system architecture

The single point of failure (SPOF) problem is a fatal issue to ZigBee networks. The advantage of the 6LoWPAN-based architecture is that, if there are legacy ZigBee-based smart grid systems, they can be easily managed by SIP. Newly deployed smart grid systems can choose 6LoWPAN protocol to enhance the communication performance, but the same SIP protocol can be used to manage devices on these two different types of networks. This greatly simplifies the management

framework. Figure 2.55 shows a hybrid smart grid network. These two different types of networks can both be managed by SIP.

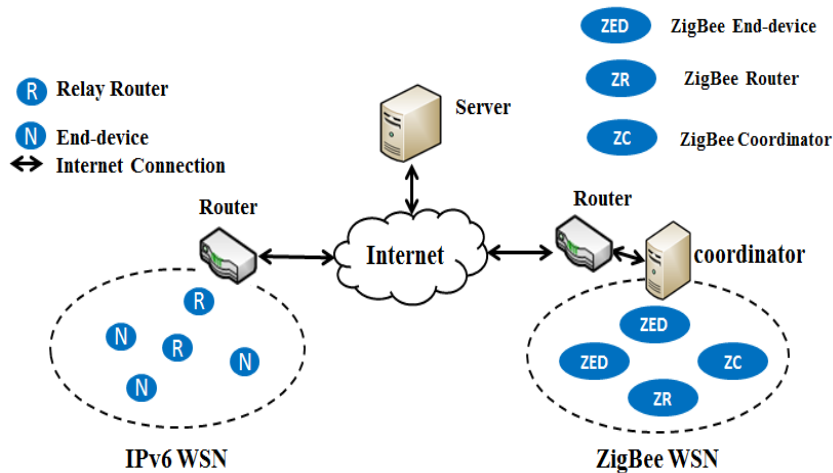


Figure 2.55 Hybrid smart grid system architecture

### 2.8.4 SIP Network Management Module and System Architecture

One of the main issue in smart grid is how to quickly transmit data and commands between servers and WSN devices. So far, SNMP and SIP are commonly used management protocols in IP networks. In addition to that, in order to integrate heterogeneous systems, International Engineering Consortium proposed IEC 61968 Common Information Model (CIM), and W3C also proposed Efficient XML Interchange (EXI), which suggests using *text based XML as the standard format* to transport smart grids data. Since SIP is also a text-based protocol, we chose SIP as the upper application-layer protocol to provide three functions in managing the smart grid system:

- (i) Subscription
- (ii) Notification
- (iii) Instant message delivery

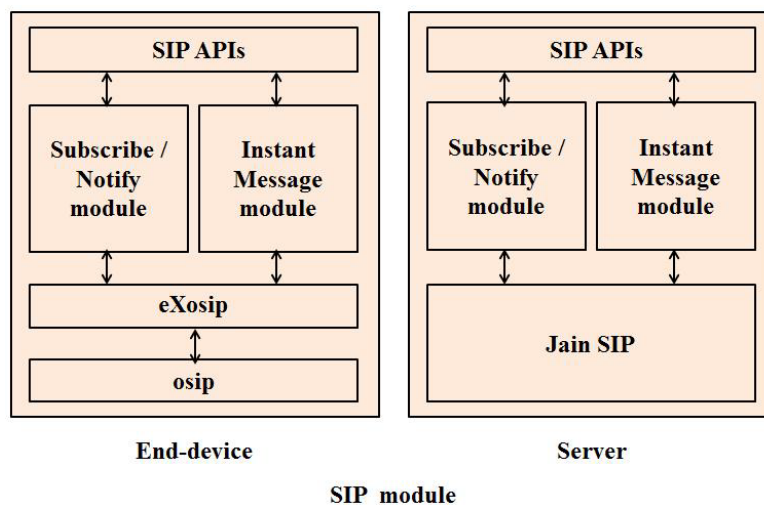


Figure 2.56 SIP Module Software Architecture

The software architecture of servers and end devices are shown in Figure 2.56, which includes the following;

**(i) osip, eXosip and Jain-SIP library**

**a) End device**

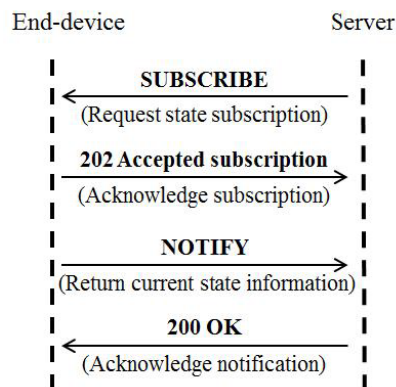
- Using GNU osip library to implement SIP related application programs.
- eXosip is an extended osip library which encapsulates osip library and make it easier to develop SIP application programs.

**b) Server**

- Java Server Pages (JSP) is used to develop a web-based management system.
- To integrate web programs and communication programs, Jain-SIP library is chosen to develop Java programs which communicate between servers and end devices.

**(ii) Subscrible / Notify module**

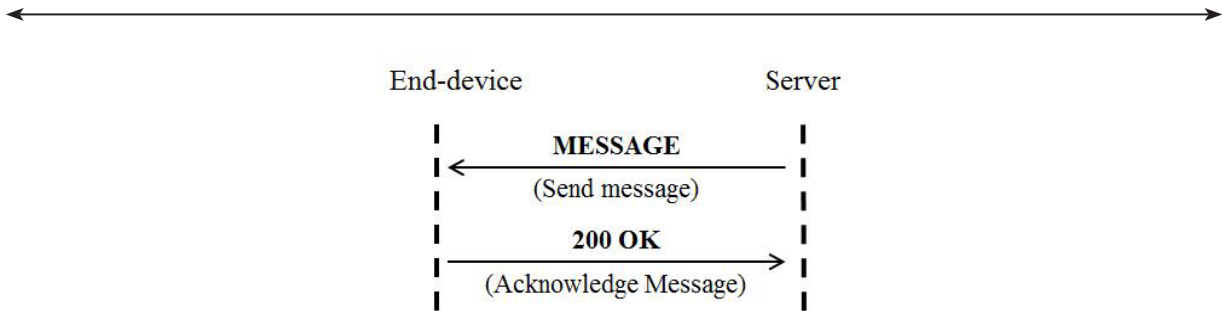
- Providing subscription and notification capabilities.
- As shown in Figure 2.57, servers can subscribe data which it wants to monitor from end devices.
- These data may include electricity consumption, temperature of meter, and so on.
- The servers may also specify a condition (for example, when electricity consumption exceeds a certain threshold), and ask end devices to automatically notify servers when that happens.



**Figure 2.57 Signal Flow of SUBSCRIBE/NOTIFY Messages**

**(iii) Instant Message module**

- Sending messages to control end devices.
- In Figure 2.58, the server can send commands to devices, such as shutting down or starting devices, changing devices settings, and so on.



**Figure 2.58 Signal Flow of Instant Messages**

**(iv) SIP Application programming interface (SIP API)**

- Allowing the application program to call Subscribe/Notify module or Instant Message module.

## 2.9 NETWORK LAYER SERVICES

The network layer is responsible for the delivery of individual packets from the source host to the destination host. Other responsibilities of the network layer include the following;

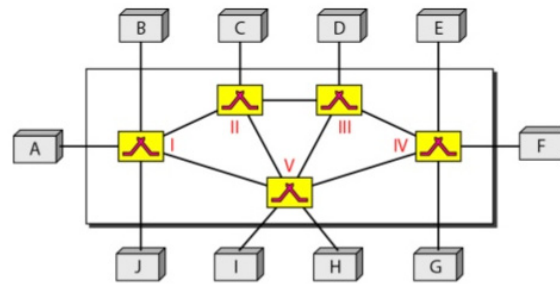
- (a) Logical addressing:** When a packet passes the network boundary, the network layer adds the logical addresses of the sender and receiver.
- (b) Routing:** When independent networks or links are connected to create internetworks, the connecting devices (called routers or switches) route or switch the packets to their final destination.

### 2.9.1 Switching

A network is a set of connected devices. When multiple devices are connected, we must find the solution how to connect them to make one-to-one communication possible. One solution is to make a point-to-point connection between each pair of devices (a mesh topology) or between a central device and every other device (a star topology). In this method, the number and length of the links require too much infrastructure to be cost-efficient, and the majority of those links would be idle most of the time.

Other topologies employing multipoint connections, such as a bus, are ruled out because the distances between devices and the total number of devices increase beyond the capacities of the media and equipment. A better solution is switching. A switched network consists of a series of interlinked nodes, called switches. Switches are devices capable of creating temporary connections between two or more devices linked to the switch.

In a switched network, some of these nodes are connected to the end systems (computers or telephones, for example). Others are used only for routing.

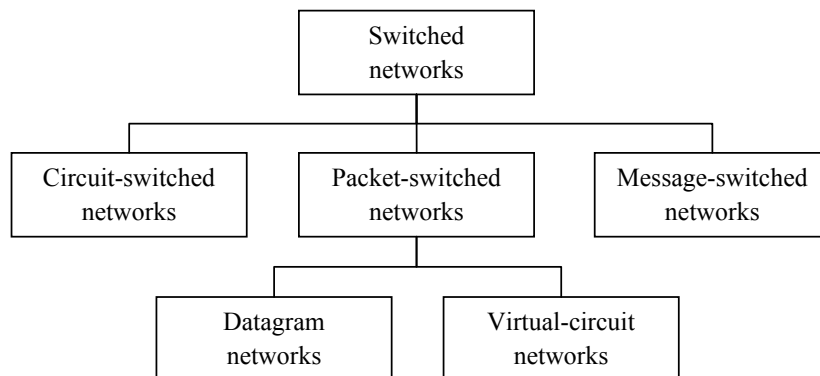


**Figure 2.59** An example switched network

The end systems (communicating devices) are labeled A, B, C, D, and so on, and the switches are labeled I, II, III, IV, and V. Each switch is connected to multiple links. Three methods of switching have been important;

- (i) Circuit switching
- (ii) Packet switching
- (iii) Message switching

We can then divide today's networks into three broad categories: circuit-switched networks, packet-switched networks, and message-switched. Packet-switched networks can further be divided into two subcategories—virtual-circuit networks and datagram networks.



**Figure 2.60** Categories of switched network

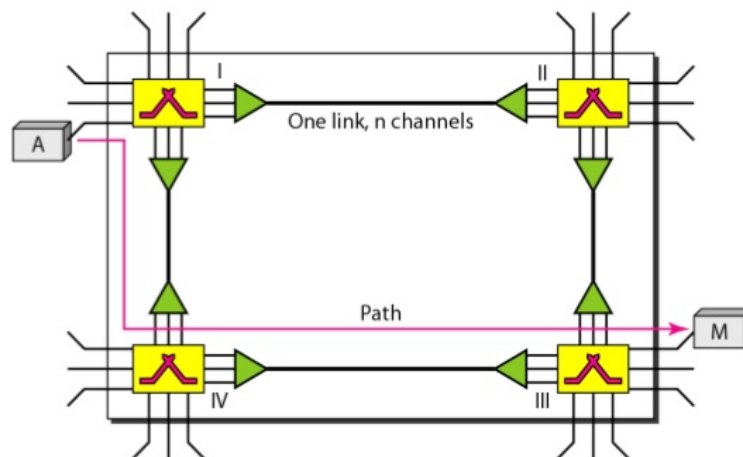
## 2.9.2 Circuit Switched Networks

A circuit-switched network is made of a set of switches connected by physical links, in which each link is divided into  $n$  channels. A circuit-switched network with four switches and four links is shown below. Each link is divided into  $n$  ( $n$  is 3 in the figure 2.61) channels by using FDM or TDM.

When end system A needs to communicate with end system M, system A needs to request a connection to M that must be accepted by all switches as well as by M itself. This is called the setup phase. A circuit (channel) is reserved on each link, and the combination of circuits or channels defines the dedicated path.

After the dedicated path made of connected circuits (channels) is established, data transfer can take place. After all data have been transferred, the circuits are torn down. In circuit switching, the resources need to be reserved during the setup phase; the resources remain dedicated for the entire duration of data transfer until the teardown phase. We need to emphasize several points here:

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



**Figure 2.61** *Circuit switched network*

- (iii) Data transferred between the two stations are not packetized. There is a continuous flow of data from the source station to receiver station.
- (iv) There is no addressing involved during data transfer. The switches route the data based on their occupied band. End-to-end addressing is used during the setup phase.

### **Three phases**

The actual communication in a circuit-switched network requires three phases: connection setup, data transfer, and connection teardown.

#### **(i) Setup phase**

- Before the communication, a dedicated circuit needs to be established.
- The end systems are normally connected through dedicated lines to the switches, so connection setup means creating dedicated channels between the switches.
- In Figure 2.61, when system A needs to connect to system M, it sends a setup request that includes the address of system M, to switch I.
- Switch I finds a channel between itself and switch IV that can be dedicated for this purpose.
- Switch I then sends the request to switch IV, which finds a dedicated channel between itself and switch III.



- Switch III informs system M of system A's intention at this time.
- In the next step to making a connection, an acknowledgment from system M needs to be sent in the opposite direction to system A.
- Only after system A receives this acknowledgment is the connection established.
- Note that end-to-end addressing is required for creating a connection between the two end systems.

**(ii) Data Transfer Phase**

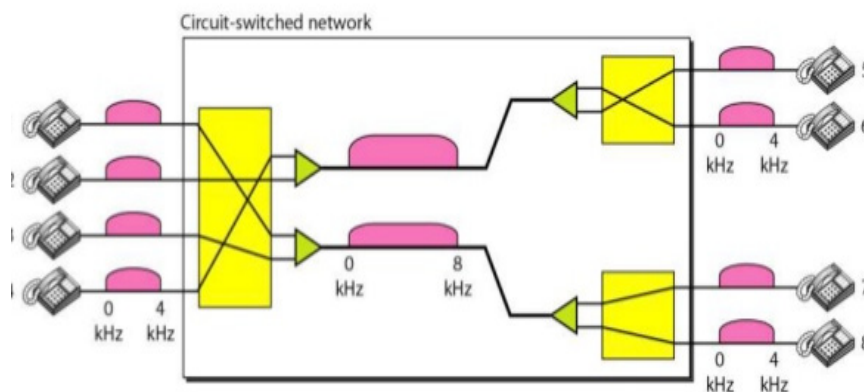
- After the establishment of the dedicated circuit, the two parties can transfer data.

**(iii) Teardown Phase**

- When one of the parties needs to disconnect, a signal is sent to each switch to release the resources.

**Efficiency**

It can be argued that circuit-switched networks are not as efficient as the other two types of networks because resources are allocated during the entire duration of the connection. These resources are unavailable to other connections. Switching at the physical layer in the traditional telephone network uses the circuit-switching approach.



**Figure 2.62 Example for circuit switched network**

**Disadvantages of circuit switched network**

- Designed for voice communication.
- Data transmission line is often idle and its facilities wasted.
- Supports less data transmission rates only.
- Circuit switching is inflexible.
- Circuit switching sees all transmission as equal

**2.9.3 Packet Switched Network**

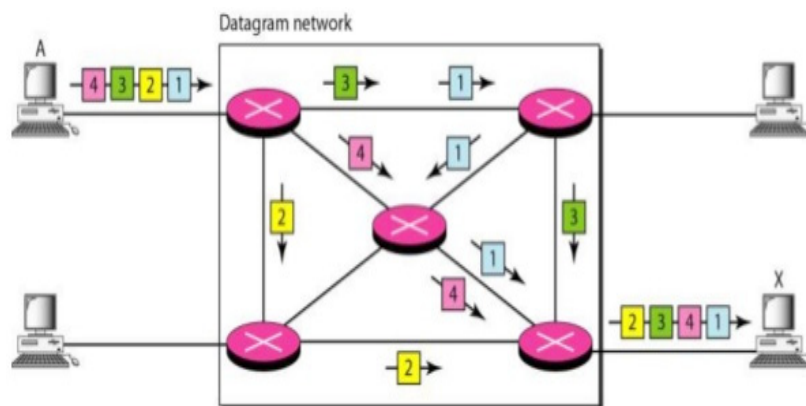
In data communications, we need to send messages from one end system to another. The message is divided into packets of fixed or variable size. The size of the packet is determined by the

network and the governing protocol. In packet switching, there is no resource allocation for a packet (no reserved bandwidth on the links, and no scheduled processing time for each packet). Resources are allocated on demand. The allocation is done on a first-come, first-served basis. When a switch receives a packet, no matter what is the source or destination, the packet must wait if there are other packets being processed.

### 2.9.3.1 *Datagram approach*

In a datagram network, each packet is treated independently of all others. Packets in this approach are referred to as datagram. Datagram switching is normally done at the network layer.

Figure 2.63 shows how the datagram approach is used to deliver four packets from station A to station X. The switches in a datagram network are traditionally referred to as routers.



**Figure 2.63** *A datagram network with five switches (routers)*

- All four packets (or datagram) belong to the same message but may travel different paths to reach their destination.
- This is so because the links may be involved in carrying packets from other sources and do not have the necessary bandwidth available to carry all the packets from A to X.
- Due to this the datagram of a transmission to arrive at their destination out of order with different delays between the packets.
- Packets may also be lost or dropped because of a lack of resources.
- It is the responsibility of an upper-layer protocol to reorder the datagrams or ask for lost datagrams before passing them on to the application.
- The datagram networks are sometimes referred to as connectionless networks.
- There are no setup or teardown phases.

### **Routing table**

If there are no setup or teardown phases, how the packets are routed to their destinations. In a datagram network each switch has a routing table which is based on the destination address, and the corresponding forwarding output ports are recorded in the tables. The routing tables are dynamic and are updated periodically. This is different from the table of a circuit switched network in which each entry is created when the setup phase is completed and deleted when the teardown phase is over.

Destination address	Output port
1232	1
4150	3
.	.
9130	3

*Table 3.2 Sample routing table*

### ***Destination Address***

Every packet in a datagram network carries a header that contains, among other information, the destination address of the packet. When the switch receives the packet, this destination address is examined; the routing table is consulted to find the corresponding port through which the packet should be forwarded. This address remains the same during the entire journey of the packet.

### ***Efficiency***

The efficiency of a datagram network is better than that of a circuit-switched network, because resources are allocated only when there are packets to be transferred. If a source sends a packet and there is a delay of a few minutes before another packet can be sent.

### ***Delay***

There may be greater delay in a datagram network than in a virtual-circuit network. Not all packets in a message necessarily travel through the same switches, so the delay is not uniform for the packets of a message.

### ***Applications***

- The Internet has chosen the datagram approach to switching at the network layer.
- It uses the universal addresses defined in the network layer to route packets from the source to the destination.

#### ***2.9.3.2 Virtual circuit networks***

A virtual-circuit network is a cross between a circuit-switched network and a datagram network. It has some characteristics of both. They are,

- (i) As in a circuit-switched network, there are setup and teardown phases in addition to the data transfer phase.
- (ii) Resources can be allocated during the setup phase, as in a circuit-switched network, or on demand, as in a datagram network.
- (iii) As in a datagram network, data are packetized and each packet carries an address in the header (it defines what should be the next switch and the channel on which the packet is being carried), not end-to-end jurisdiction.
- (iv) As in a circuit-switched network, all packets follow the same path established during the connection.

- (v) A virtual-circuit network is implemented in the DLL; a circuit-switched network is implemented in the physical layer and a datagram network in the network layer.

**Classification**

Virtual Circuit Networks are again classified into two types. They are,

- (i) Switched VC – Different VC is provided between two users
- (ii) Permanent VC – The same VC is provided between two users on a continuous basis

**Addressing**

Two types of addressing are involved in virtual circuit networks.

- (i) Global – used to create a virtual-circuit identifier (VCI)
- (ii) Local – Data transfer

**Virtual-Circuit Identifier**

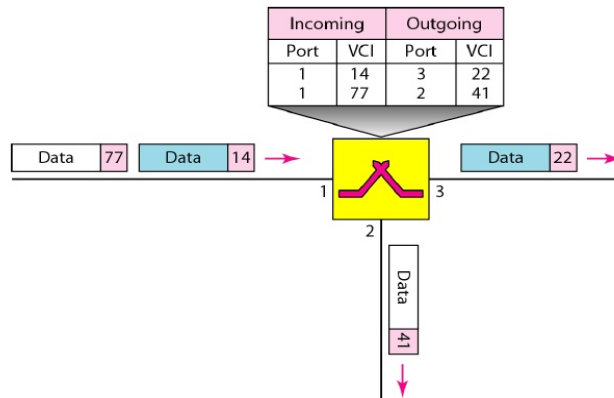


Figure 2.64 Switch and tables in a virtual-circuit network

- The identifier is a small number used by a frame between two switches.
- When a frame arrives at a switch, it has a VCI; when it leaves, it has a different VCI.

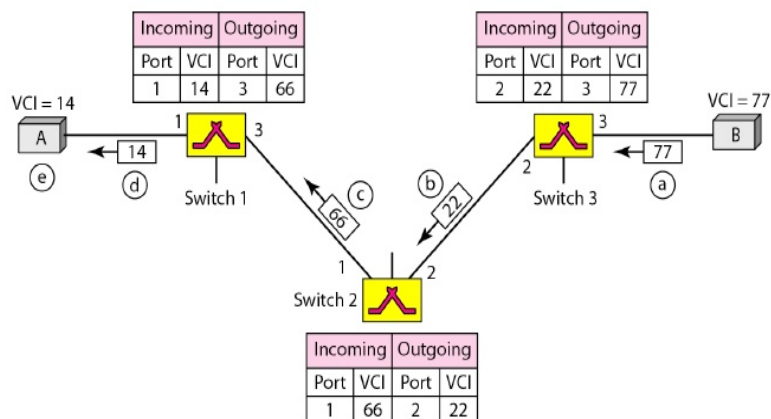


Figure 2.65 Source-to-destination data transfer in a virtual-circuit network

### Three Phases

Virtual circuit networks consists of the following three phases.

- (i) **Setup phase:** The source and destination use their global addresses to help switches make table entries for the connection.
- (ii) **Data transfer phase:** Data transfer occurs between these two phases.
- (iii) **Teardown phase:** The source and destination inform the switches to delete the corresponding entry.

### Efficiency

In virtual-circuit switching, all packets belonging to the same source and destination travel the same path, but the packets may arrive at the destination with different delays if resource allocation is on demand.

## 2.9.4 Structure of a Switch

### Crossbar Switch

A crossbar switch connects  $n$  inputs to  $m$  outputs in a grid, using electronic micro switches (transistors) at each cross point. The major limitation of this design is the number of cross points required. To connect  $n$  inputs to  $m$  outputs using a crossbar switch requires  $n \times m$  cross points.

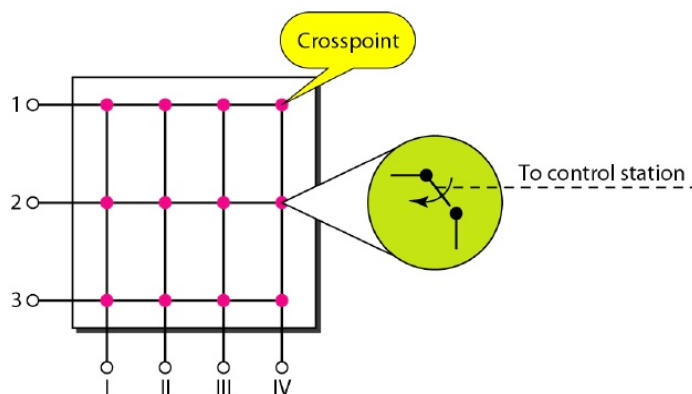


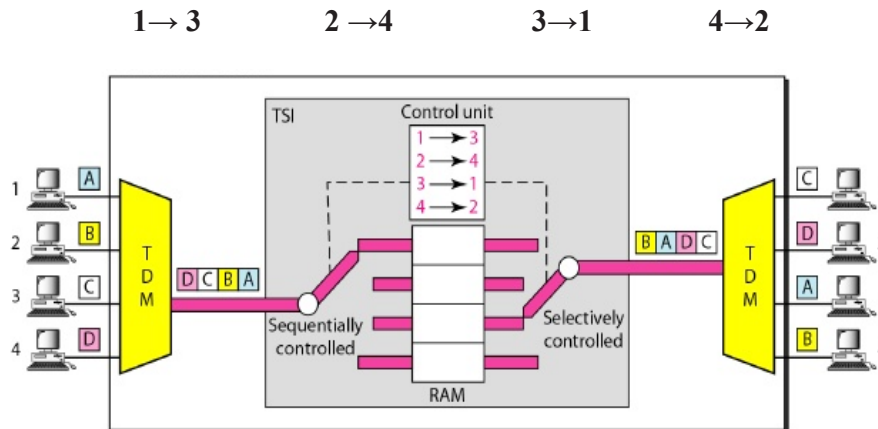
Figure 2.66 Crossbar switch with three inputs and four outputs

### Time-Division Switch

Time-division switching uses time-division multiplexing (TDM) inside a switch. The most popular technology is called the time-slot interchange (TSI).

Figure 2.67 combines a TDM multiplexer, a TDM demultiplexer, and a TSI consisting of random access memory (RAM) with several memory locations. The size of each location is the same as the size of a single time slot. The number of locations is the same as the number of inputs. The RAM fills up with incoming data from time slots in the order received. Slots are then sent out in an order based on the decisions of a control unit.

Imagine that each input line wants to send data to an output line according to the following pattern:



**Figure 2.67 Time-division switch**

### ***Time-space-time Switch***

The advantage of space-division switching is that it is instantaneous. Its disadvantage is the number of cross points required to make space-division switching. The advantage of time-division switching is that it needs no cross points. Its disadvantage is the TSI (Processing each delay at each connection). To overcome these problems, we combine space-division and time-division technologies to take advantage of the best of both.

### **2.9.5 Message Switching**

- Store and forward technology.
- When a node receives a message stores it until the appropriate route is free. If the node finds that the route is free, then it sends the message.
- No direct link between the source and the destination, Routing technology is used here.

## **2.10. INTERNET PROTOCOL (IPV4)**

The physical and data link layers of a network operate locally. These two layers are jointly responsible for data delivery on the network from one node to the next.

### ***Need for Network Layer***

To solve the problem of delivery through several links, the network layer (or the internetwork layer, as it is sometimes called) was designed. The network layer is responsible for host-to-host delivery and for routing the packets through the routers or switches.

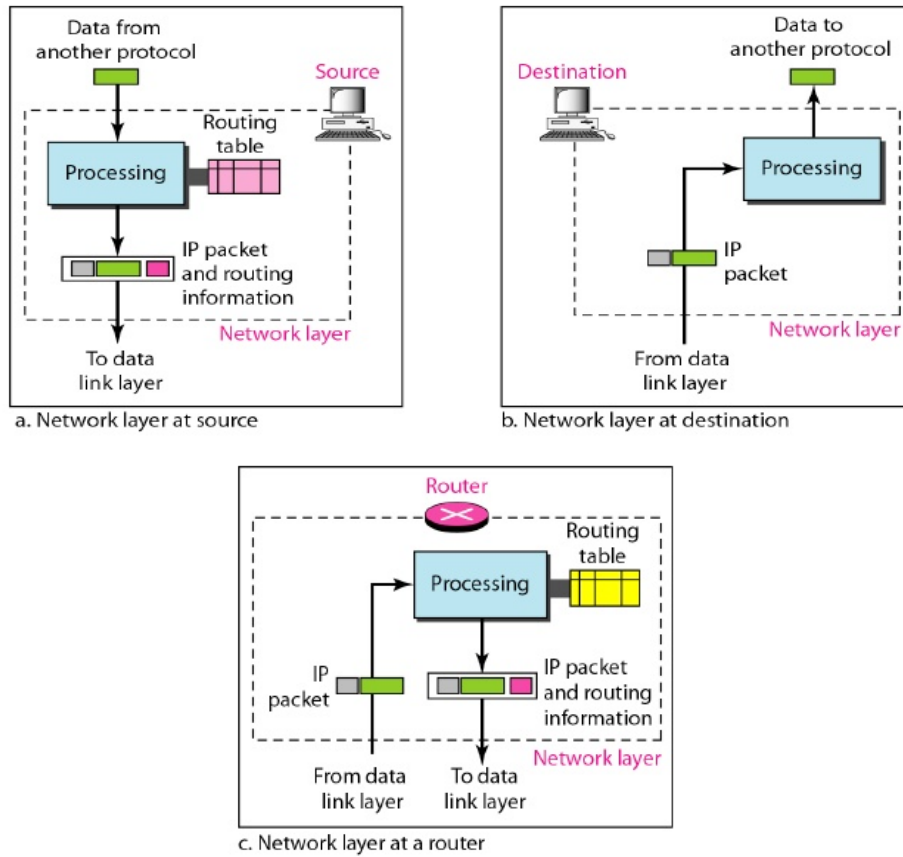


Figure 2.68 Network layer at the source, router, and destination

### 2.10.1 Internet Protocol Version 4 (IPv4)

The Internet Protocol version 4 (IPv4) is the delivery mechanism used by the TCP/IP protocols. Figure 2.69 shows the position of IPv4 in the suite.

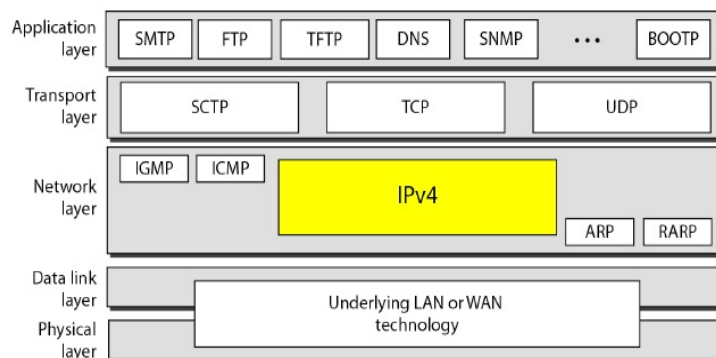


Figure 2.69 Layers and its protocols

- IPv4 is an unreliable and connectionless datagram protocol.
- IPv4 provides a best-effort delivery service (means that IPv4 provides no error control or flow control except for error detection on the header).

- IPv4 assumes no guarantees for data delivery.
- IPv4 uses the datagram approach (Each datagram is handled independently, and each datagram can follow a different route to the destination).
- This implies that datagram's sent by the same source to the same destination could arrive out of order; also, some could be lost or corrupted during transmission.
- Again, IPv4 relies on a higher-level protocol to take care of all these problems.

### 2.10.2 Datagram

Packets in the IPv4 layer are called datagram's. A datagram is a variable-length packet consisting of two parts: Header and Data. The header is 20 to 60 bytes in length and contains information essential to routing and delivery. Figure 2.70 shows the IPv4 datagram format.

A brief description of each field

#### (i) Version (VER)

- This 4-bit field defines the version of the IPv4 protocol.
- Currently the version is 4.
- Version 6 (or IPv6) may totally replace version 4 in the future.
- If the machine is using some other version of IPv4, the datagram is discarded rather than interpreted incorrectly.

#### (ii) Header length (HLEN)

- This 4-bit field defines the total length of the datagram header in 4-byte words.
- This field is needed because the length of the header is variable (between 20 and 60 bytes).
- When there are no options, the header length is 20 bytes, and the value of this field is 5 ( $5 \times 4 = 20$ ). When the option field is at its maximum size, the value of this field is 15 ( $15 \times 4 = 60$ ).

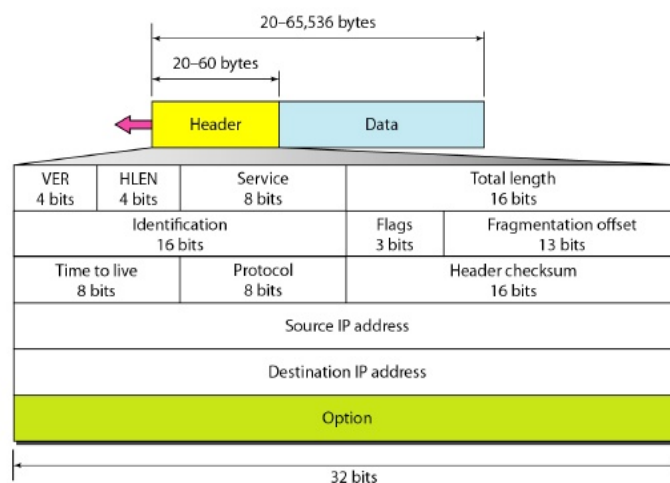


Figure 2.70 IPv4 datagram format



**(iii) Services**

- Define how the datagram should be handled.
- It includes bits that define the priority of datagram.
- It also contains some bits that specify the type of service the sender desires such as the level of throughput, reliability and delay.

**(iv) Total length**

- It defines the total length (header plus data) of the IPv4 datagram in bytes.
- Length of data = total length - header length.
- Since the field length is 16 bits, the total length of the IPv4 datagram is limited to 65,535 bytes, of which 20 to 60 bytes are the header and the rest is data from the upper layer.

**(v) Identification**

- This field is used in fragmentation.
- Provide the sequence number to each fragment.

**(vi) Flags**

- This field is used to find whether the datagram can or can't be fragmented.
- To indicate the first, last or middle fragment etc.,

**(vii) Fragmentation offset**

- It is the pointer that shows the offset of the data in the original datagram (if it is fragmented).

**(viii) Time to live**

- A datagram has a limited lifetime in its travel through an internet.
- This field was originally designed to hold a timestamp, which was decremented by each visited router.
- The datagram was discarded when the value became zero.
- However, for this scheme, all the machines must have synchronized clocks and must know how long it takes for a datagram to go from one machine to another.

**(ix) Protocol**

- This 8-bit field defines the higher-level protocol that uses the services of the IPv4 layer.
- An IPv4 datagram can encapsulate data from several higher-level protocols such as TCP, UDP, ICMP, and IGMP.
- This field specifies the final destination protocol to which the IPv4 datagram is delivered.

**(x) Checksum**

- 16 bit field used to check the integrity of the header.

**(xi) Source address**

- This 32-bit field defines the IPv4 address of the source.
- This field must remain unchanged during the time the IPv4 datagram travels from the source host to the destination host.

**(xii) Destination address**

- This 32-bit field defines the IPv4 address of the destination.
- This field must remain unchanged during the time the IPv4 datagram travels from the source host to the destination host.

**(xiii) Options**

- Gives more functionality to the IP datagram.
- It can carry fields that control routing, timing, management, and alignment.

**2.10.3 Classless Inter-Domain Routing (CIDR)**

Before CIDR was adopted, the network portion of an IP address were constrained to be 8,16, or 24-bits in length, an addressing scheme known as classful addressing since subnets with 8-,16- and 24- bit subnet address were known as class A, class B, class C network respectively.

- (i) A class C (/24) Subnet could accommodate only up to  $2^8 - 2 = 256 - 2$   
= 254 hosts (two addresses are reserved for special use).
- (ii) A class B (/16) Subnet could accommodate up to  $2^{16} - 2 = 65,536 - 2$   
= 65,534 hosts (two addresses are reserved for special use).
- (iii) A class A (/8) Subnet could accommodate up to  $2^{24} - 2 = 16,777,216 - 2$   
= 16,777,214 hosts (two addresses are reserved for special use).

A class B subnet supports up to 65,534 hosts whereas a class A subnet supports up to 16,777,214 hosts. Thus Class B and Class A subnet supports a large number of hosts. Under classful addressing, an organization with, say, 2000 hosts was typically allocated a class B (/16) Subnet address. This led to a rapid depletion of the class B address space and poor utilization of the address assigned for that address space.

For example, if the organization is using a class B address for its 2000 hosts, it is leaving more than 63,000 addresses that could not be used by other organization. A technique known as Classless Inter-Domain Routing (CIDR) solves this problem.

CIDR makes the IP addressing space classless. CIDR associates network masks with IP network numbers independent of their traditional class. CIDR requires the size of each block of addresses to be a power of two. It uses a bit mask to identify the size of the block. An IPv4 address is 32 bit long. The address space of IPv4 is  $2^{32}$  or 4,294,967,296.

**Example 1**

Change the given IPv4 address from binary notation to dotted-decimal notation.

$\leftarrow$   $\rightarrow$   
 10000001    00001011    00001011    11101111  
 Given (Binary notation)  
 10000001    00001011    00001011    11101111

**Conversation to Decimal:**

$$10000001 \rightarrow (1 \times 27) + (0 \times 26) + (0 \times 25) + (0 \times 24) + (0 \times 23) + (0 \times 22) + (0 \times 21) + (1 \times 20)$$

$$= 128 + 0 + 0 + 0 + 0 + 0 + 0 + 1 = 129.$$

$$00001011 \rightarrow (0 \times 27) + (0 \times 26) + (0 \times 25) + (0 \times 24) + (1 \times 23) + (0 \times 22) + (1 \times 21) + (1 \times 20)$$

$$= 0 + 0 + 0 + 0 + 8 + 0 + 2 + 1 = 11.$$

$$00001011 \rightarrow (0 \times 27) + (0 \times 26) + (0 \times 25) + (0 \times 24) + (1 \times 23) + (0 \times 22) + (1 \times 21) + (1 \times 20)$$

$$= 0 + 0 + 0 + 0 + 8 + 0 + 2 + 1 = 11.$$

$$11101111 \rightarrow (1 \times 27) + (1 \times 26) + (1 \times 25) + (0 \times 24) + (1 \times 23) + (1 \times 22) + (1 \times 21) + (1 \times 20)$$

$$= 128 + 64 + 32 + 0 + 8 + 4 + 2 + 1 = 239.$$

Therefore, Binary notation  $\rightarrow$  dotted-decimal notation is,

$$10000001 \ 00001011 \ 00001011 \ 11101111 \rightarrow 129.11.11.239$$

**Example 2**

Change the given IPv4 address from dotted decimal notation to binary notation.

111.56.45.78

Given (Dotted Decimal Notation)

111.56.45.78

**Conversation to binary**

111	$\rightarrow$	1101111	0110111
		(Seven bits)	(Eight bits)
56	$\rightarrow$	111000	00111000
		(Six bits)	(Eight bits)
45	$\rightarrow$	1011011	00101101
		(Six bits)	(Eight bits)
78	$\rightarrow$	1001110	01001110
		(Seven bits)	(Eight bits)

Dotted decimal notation → Binary notation is

111.56.45.78	→	01101111 00111000 00101101
--------------	---	----------------------------

**Example 3**

Find the number of addresses in a range if the first address in a range is 146.102.29.0 and the last address is 146.102.32.255.

**Solution:**

Subtract the first address from the last address.

$$\begin{array}{r}
 \text{Last address} \quad \rightarrow \quad 146. 102. 32. 255 \\
 \text{First address} \quad \rightarrow \quad \underline{146. 102. 29. 0} \\
 \hline
 \quad \quad \quad \quad \quad \quad 0. \quad 0. \quad 3. 255
 \end{array}$$

Number of addresses

$$\begin{aligned}
 &= (0 \times 2563) + (0 \times 2562) + (0 \times 2561) + (0 \times 2560) + 1 \\
 &= 0 + (3 \times 256) + 0 + (255 \times 1) + 1 \\
 &= 0 + 768 + 0 + 255 + 1 \\
 &= 1024
 \end{aligned}$$

**Example 4**

The first address in a range of addresses is 14.11.45.96. If the number of addresses in the range is 32, what is the last address?

**Solution:**

We convert the number of addresses minus 1 to base 256, which is 0.0.0.31. We then add it to the first address to set the last address. Addition is in base 256.

$$\begin{aligned}
 \text{Last address} &= \text{First address} + \text{Number of addresses} \\
 &= 14. 11. 45. 96 \quad + \\
 &= \underline{0. 0. 0. 31} \\
 &= 14. 11. 45. 127
 \end{aligned}$$

Therefore, Last address = 14.11.45.127.

**2.10.4 Classful Addressing**

In classful addressing, an IP address in class A, B and C is divided into **netid** and **hostid**. Figure 2.71 shows the netid and hostid in bytes.

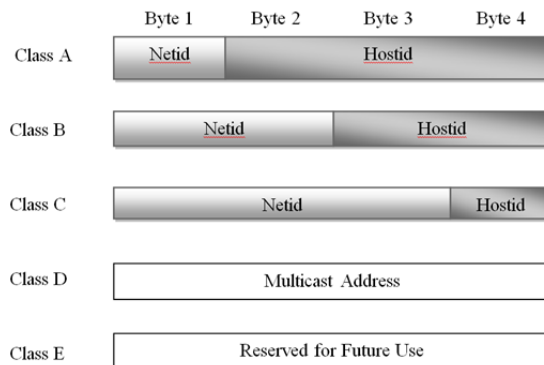


Figure 2.71 Netid and Hostid

Figure 2.72 shows an IPv4 address in classful addressing. If  $n$  bits in the class defines the net, then  $32-n$  define the host.

Net bits (prefix) =  $n$

Host bits (suffix) =  $32-n$

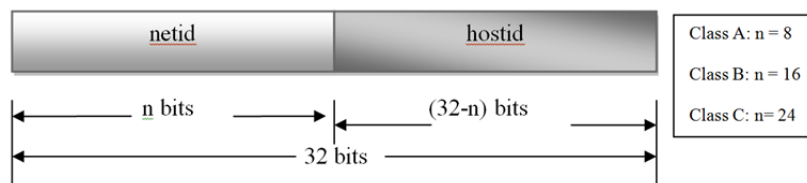


Figure 2.72 Two – level addressing in classful addressing

In classful addressing, the length of the netid,  $n$ , depends on the class of the address; it can be only 8, 16 or 24.

In classless addressing, the value of  $n$  is referred to as **prefix length**; the value of  $32-n$  is referred to as **suffix length**. Figure 2.73 shows the prefix and suffix in a classless block.

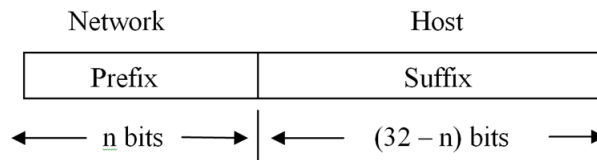


Figure 2.73 Classless block

The prefix length in classless addressing can be 1 to 32.

In class addressing, the prefix defines the network and the suffix defines the host

The number of addresses in a block is inversely related to the value of the prefix length,  $n$ . A small  $n$  means a larger block; a large  $n$  means a smaller block.

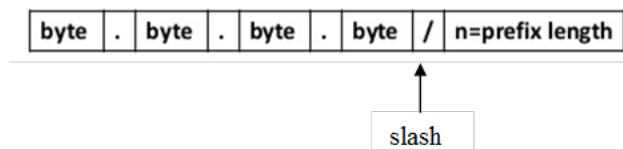
### 2.10.5 Slash Notation

The netid length in classful addressing or the prefix length in classless addressing plays a very important role when we need to extract the information about the block from a given address in the block. However, there is a difference in classful and classless addressing.

- In classful addressing, the netid length is inherent in the address. With a given address, we may know the class of the address, through that we can find the netid length (8, 10 or 24).
- In classless addressing, the prefix length cannot be found. Because we have given only an address in the block. The given address can belong to a block with any prefix length.

In classless addressing, we need to include the prefix length to each address if we need to find the block of the address. In this case the prefix length to each address if we need to find the block of the address. In this case the prefix length,  $n$ , is added to the address separated by a slash (like /  $n$ ). The notation is referred to as **Slash Notation or Classless Inter Domain Routing (CIDR) notation**.

Slash Notation: An address in classless addressing can then be represented as shown in figure 2.74.



**Figure 2.74 Slash Notation**

The slash notation is formally referred to as Classless Inter – Domain Routing (CIDR) notation.

#### Network Mask

The idea of network mask in classless addressing is the same as the one in classful addressing. A network mask is a 32-bit number with  $n$  left most bits all set to 1s and the rest of the bits [(32 –  $n$ ) bits] all set to 0s.

#### Example 5

The following addresses are defined using slash notations.

- In the address 12.33.24.78 / 8 the network mask is 255.0.0.0. The mask has eight 1s and twenty four 0s. The prefix length is 8 and the suffix length is 24.
- In the address 130.11.232.156 /16. The network mask is 255.255.0.0. The mask has sixteen 1s and sixteen 0s. The prefix length is 16 and the (suffix length is 16).
- In the address 167.199.170.82/27, the network mask is 255.255.255.224. The mask has twenty seven 1s and five 0s. The prefix length is 27 and the suffix length is 5.

#### Extracting Block Information

An address in slash notation (CIDR) contains all information we need about the block; the first address (network address), the number of addresses, and the last address.

Those three pieces of information can be found as follows:

- 1) The number of addresses in the block can be found as

$$N = 2^{32-n}$$

In which  $n$  is the prefix length and  $N$  is the number of addresses in the block.

- 2) The first address (network address) in the block can be found by AND ing the address with the network mask:

<b>First Address</b> <b>(Binary Notation)</b>	=	<b>(Any Address)</b> <b>(Binary Notation)</b>	AND	<b>(Network Mask)</b> <b>(Binary Notation)</b>
--------------------------------------------------	---	--------------------------------------------------	-----	---------------------------------------------------

Alternatively, we can keep the  $n$  leftmost bits of any address in the block and set the  $32-n$  bits to 0s to find the first address.

- 3) The last address in the block can be found by adding the first address with the number of addresses or, directly, by OR ing the addresses with the complement (NOT ing) of the network mask:

<b>Last Address</b> <b>(Binary Notation)</b>	=	<b>(Any Address)</b> <b>(Binary Notation)</b>	OR	<b>[NOT (Network Mask)]</b> <b>(Binary Notation)</b>
-------------------------------------------------	---	--------------------------------------------------	----	---------------------------------------------------------

Alternatively, we can keep the  $n$  leftmost bits of any address in the block and set the  $32 - n$  bits to 1s to find the last address.

### Example 6

One of the addresses in a block is 167.199.170.82/27. Find,

- The number of addresses in the network
- The first address
- The last address.

### Solution:

The value of  $n$  is 27. i.e., the network mask has twenty seven 1s and five 0s.

Network mask in binary notation is

11111111    11111111    11111111    11100000

Complement of the mask is

00000000    00000000    00000000    00011111

- a) The Number of address in the Network  $N = 2^{32-n}$   
 $= 2^{32-27}$   
 $= 2^5$   
 $= 32.$

$$\begin{matrix} \text{b)} & \text{First Address} & = & (\text{Any Address}) & \text{AND} & (\text{Network Mask}) \\ & (\text{Binary Notation}) & & (\text{Binary Notation}) & & (\text{Binary Notation}) \end{matrix}$$

The Number of Addresses in the Network = 32.

**Address (Any address) given (dotted-decimal notation) is 167.199.170.82**

Decimal		Binary
167	→	10100111 → 10100111
		(Eight bits) (Eight bits)
199	→	11000111 → 11000111
		(Eight bits) (Eight bits)
170	→	10101010 → 10101010
		(Eight bits) (Eight bits)
82	→	1010010 → 01010010
		(Eight bits) (Eight bits)

Given Address	= 10100111	11000111	10101010	01010010
		(AND)		
Network Mask	= 11111111	11111111	11111111	11100000
First Address	= 10100111	11000111	10101010	01000000

(Binary Notation)

$$\begin{aligned} 10100111 &= (1 \times 2^7) + (0 \times 2^6) + (1 \times 2^5) + (0 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) \\ &= 128 + 0 + 32 + 0 + 0 + 4 + 2 + 1 \\ &= 167. \end{aligned}$$

$$\begin{aligned} 11000111 &= (1 \times 2^7) + (1 \times 2^6) + (0 \times 2^5) + (0 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) \\ &= 128 + 64 + 0 + 0 + 0 + 4 + 2 + 1 \\ &= 199. \end{aligned}$$

$$\begin{aligned} 10101010 &= (1 \times 2^7) + (0 \times 2^6) + (1 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (0 \times 2^0) \\ &= 128 + 0 + 32 + 0 + 8 + 0 + 2 + 0 \\ &= 170. \end{aligned}$$

$$\begin{aligned} 01000000 &= (0 \times 2^7) + (1 \times 2^6) + (0 \times 2^5) + (0 \times 2^4) + (0 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (0 \times 2^0) \\ &= 0 + 64 + 0 + 0 + 0 + 0 + 0 + 0 \\ &= 64. \end{aligned}$$



$$\begin{array}{rcl}
 \text{c) Last Address} & = & (\text{Any Address}) \text{ OR } [\text{NOT}(\text{Network Mask})] \\
 \text{(Binary Notation)} & & \text{(Binary Notation)} \quad \text{(Binary Notation)} \\
 \\ 
 \text{Given Address} & = & 10100111 \quad 11000111 \quad 10101010 \quad 01010010 \\
 & & \text{(OR)} \\
 \text{NOT (Network mask)} & = & 00000000 \quad 00000000 \quad 00000000 \quad 00011111 \\
 \text{Last Address} & = & \underline{10100111 \quad 11000111 \quad 10101010 \quad 01011111}
 \end{array}$$

(Binary Notation)

$$\begin{array}{l}
 10100111 \quad \rightarrow \quad (1 \times 2^7) + (0 \times 2^6) + (1 \times 2^5) + (0 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) \\
 \quad \quad \quad = 128 + 0 + 32 + 0 + 0 + 4 + 2 + 1 \\
 \quad \quad \quad = 167.
 \end{array}$$

$$\begin{array}{l}
 11000111 \quad \rightarrow \quad (1 \times 2^7) + (1 \times 2^6) + (0 \times 2^5) + (0 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) \\
 \quad \quad \quad = 128 + 64 + 0 + 0 + 0 + 4 + 2 + 1 \\
 \quad \quad \quad = 199.
 \end{array}$$

$$\begin{array}{l}
 10101010 \quad \rightarrow \quad (1 \times 2^7) + (0 \times 2^6) + (1 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (0 \times 2^0) \\
 \quad \quad \quad = 128 + 0 + 32 + 0 + 8 + 0 + 2 + 0 \\
 \quad \quad \quad = 170.
 \end{array}$$

$$\begin{array}{l}
 01011111 \quad \rightarrow \quad (0 \times 2^7) + (1 \times 2^6) + (0 \times 2^5) + (1 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) \\
 \quad \quad \quad = 0 + 64 + 0 + 16 + 8 + 4 + 2 + 1 \\
 \quad \quad \quad = 95.
 \end{array}$$

So the Last Address is

10100111	11000111	10101010	01011111 (Binary Notation)
167	199	170	95 / 27 (Dotted Decimal Notation)

### 2.10.6 Method for Finding the Mask/Prefix and Cidr Block

- (1) Consider both first address and last address of the block in binary notation.
- (2) Starting from the right extreme of these two addresses check the bit values and compare them at each and every position of these addresses.
- (3) If the bit values are different. (i.e., 1 and 0 or 0 and 1) continue this process (step 2). At one particular position the bit value of the first address and the bit value of the last address will be same. (i.e., 1 and 1 or 0 and 0). Stop comparing.
- (4) The number of bits counted, from that particular position to the left extreme position of first address or last address will be equal to that value of the prefix (n).

(5) Suffix (number of host bits) = 32 – prefix (number of bits) = 32-n.

(6) Mask contains the prefix on the left side from the above fixed position (including fixed position) and suffix on the right side after the fixed position.

In the mask all bits on the prefix are 1s and bits on the suffix are 0s.

CIDR Block = First Address / Number of bits in prefix (in dotted decimal notation)

**Example 7**

**Explanation**

First address	10100111	11000111	10101010	01	0 0000
Last address	10100111    11000111    10101010			01	1 1111
	No. of bits 27				↑
	(Prefix)				5 bits
	Network				(Suffix) Host

**CIDR Block: 167 .199 .170 .64 /27**

**Example 8**

The first address and last address are given in dotted-decimal notation. Find the prefix, mask, suffix and the number of addresses (hosts) in the block. Finally write the CIDR block also.

First (Lowest) Address: 192.168.12.0

Last (Highest) Address: 192.168.13.255

First Address

(Binary Notation) → 11000000    10101000    000011    0    00000000

Last Address

(Binary Notation) → 11000000    10101000    000011    1    11111111

No. of bits 23

Network  
(Prefix)

Host (9 bits)  
(Suffix)

Prefix n:    23

Mask :    11111111    11111111    11111110    00000000

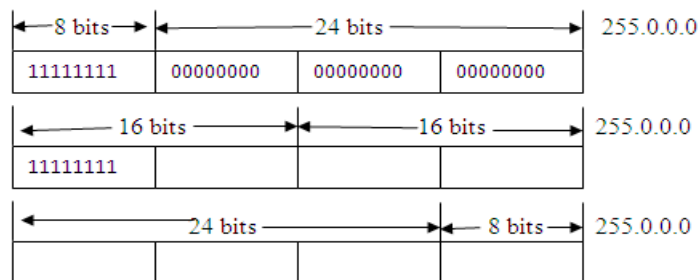
Suffix (Host Bits) h: 32-Prefix) = (32-23) = 9

Number of Hosts (Number of Host Address) = 2<sup>h</sup> = 2<sup>9</sup>  
= 512

CIDR Block: 192.168.12.0/23.

In classful addressing, the IPv4 address space is divided into five classes A, B, C, D, and E. An organization is granted a block in one of the three classes A, B or C. Classes D and E are reserved for special purposes. An IP address in classes A, B and C is divided into netid and hosted.

Since n (prefix) is different for each class in classful addressing, we have three default masks in classful addressing as shown in figure 2.75.



**Figure 2.75** Masks in classful addressing

A network mask or a default mask in classful addressing is a 32-bit number with n leftmost bits set to 1s and (32-n) right most bits all set to 0s. The n values are not changeable for the classful addressing. The mask in a classful addressing is different for different block.

CIDR Notation	Dotted Decimal	CIDR Notation	Dotted Decimal
/1	128.0.0.0	/17	255.255.128.0
/2	192.0.0.0	/18	255.255.192.0
/3	224.0.0.0	/19	255.255.224.0
/4	240.0.0.0	/20	255.255.240.0
/5	248.0.0.0	/21	255.255.248.0
/6	252.0.0.0	/22	255.255.252.0
/7	254.0.0.0	/23	255.255.254.0
/8	255.0.0.0	/24	255.255.255.0
/9	255.128.0.0	/25	255.255.255.128
/10	255.192.0.0	/26	255.255.255.192
/11	255.244.0.0	/27	255.255.255.224
/12	255.240.0.0	/28	255.255.255.240
/13	255.248.0.0	/29	255.255.255.248
/14	255.252.0.0	/30	255.255.255.252
/15	255.254.0.0	/31	255.255.255.254
/16	255.255.0.0	/32	255.255.255.255

**Table 2.7** Dotted decimal mask values for all possible CIDR prefixes

In CIDR notation,

- The masks (prefixed) are different for different blocks of addresses.

- The table below lists dotted decimal values for all possible CIDR masks.
- The /18, /16, and /24 prefixes correspond to traditional class A, B, and C divisions.

### **2.10.7 DHCP**

Three protocols are used to perform the Mapping of Physical to Logical Address. They are

- (i) RARP
- (ii) BOOTP
- (iii) DHCP

There are occasions in which a host knows its physical address, but needs to know its logical address. This may happen in two cases:

- (i) A diskless station is just booted. The station can find its physical address by checking its interface, but it does not know its IP address.
- (ii) An organization does not have enough IP addresses to assign to each station; it needs to assign IP addresses on demand. The station can send its physical address and ask for a short time lease.

#### **2.10.7.1 BOOTP**

BOOTP is not a dynamic configuration protocol. When a client requests its IP address, the BOOTP server consults a table that matches the physical address of the client with its IP address. This implies that the binding between the physical address and the IP address of the client already exists. The binding is predetermined.

##### ***Drawbacks of BOOTP***

- (i) However, what if a host moves from one physical network to another?
- (ii) What if a host wants a temporary IP address?

BOOTP cannot handle these situations because the binding between the physical and IP addresses is static and fixed in a table until changed by the administrator. BOOTP is a static configuration protocol.

#### **2.10.7.2 Dynamic Host Configuration Protocol**

The Dynamic Host Configuration Protocol has been devised to provide static and dynamic address allocation that can be manual or automatic.

##### ***Static Address Allocation***

- In this capacity DHCP acts as BOOTP does.
- It is backward compatible with BOOTP, which means a host running the DHCP client can request a static address from a DHCP server.
- A DHCP server has a database that statically binds physical addresses to IP addresses.

### *Dynamic Address Allocation*

- DHCP has a second database with a pool of available IP addresses.
- This second database makes DHCP dynamic.
- When a DHCP client requests a temporary IP address, the DHCP server goes to the pool of available (unused) IP addresses and assigns an IP address for a negotiable period of time.
- When a DHCP client sends a request to a DHCP server, the server first checks its static database.
- If an entry with the requested physical address exists in the static database, the permanent IP address of the client is returned.
- On the other hand, if the entry does not exist in the static database, the server selects an IP address from the available pool, assigns the address to the client, and adds the entry to the dynamic database.
- The dynamic aspect of DHCP is needed when a host moves from network to network or is connected and disconnected from a network (as is a subscriber to a service provider).
- DHCP provides temporary IP addresses for a limited time.
- The addresses assigned from the pool are temporary addresses.
- The DHCP server issues a lease for a specific time.
- When the lease expires, the client must either stop using the IP address or renew the lease.
- The server has the option to agree or disagree with the renewal. If the server disagrees, the client stops using the address.

### *Difference between BOOTP and DHCP*

- One major problem with the BOOTP protocol is that the table mapping the IP addresses to physical addresses needs to be manually configured.
- This means that every time there is a change in a physical or IP address, the administrator needs to manually enter the changes.
- DHCP, on the other hand, allows both manual and automatic configurations.
- Static addresses are created manually and dynamic addresses are created automatically.

## **2.10.8 Network Address Translation (NAT)**

In the beginning, small businesses and home user was connected to the Internet with a dial-up line, which means that they were connected for a specific period of time. An ISP with a block of addresses could dynamically assign an address to this user. An address was given to a user when it was needed. Now a day's home users and small businesses can be connected by an ADSL line or cable modem. In addition, many have created small networks with several hosts and need an IP address for each host. With the shortage of addresses, this is a serious problem. A quick solution to this problem is called **network address translation (NAT)**.

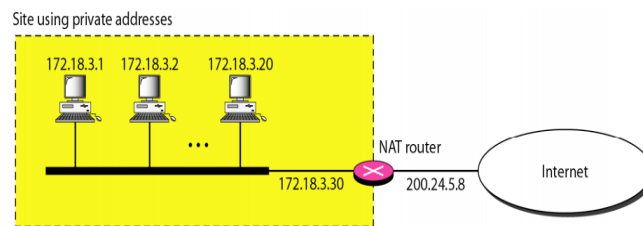
NAT enables a user to have a large set of addresses internally and one address or a small set of addresses, externally. The traffic inside can use the large set and the traffic outside may use the small set. To separate the addresses used inside the home or business and the ones used for the Internet, the Internet authorities have reserved three sets of addresses as private addresses, shown in Table 10.1.

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

**Table 10.1** Addresses for private networks

Any organization can use an address out of this set without permission from the Internet authorities. Everyone knows that these reserved addresses are for private networks. **Reserved addresses are unique inside the organization, but they are not unique globally.** No router will forward a packet that has one of these addresses as the destination address. The site must have only one single connection to the global Internet through a router that runs the NAT software. Figure 2.76 shows a simple implementation of NAT.

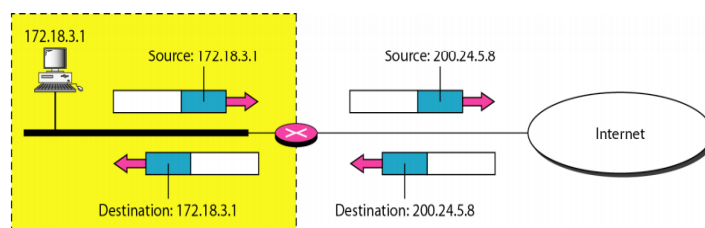
As Figure 2.76 shows, the private network uses private addresses. The router that connects the network to the global address uses one private address and one global address. The private network is transparent to the rest of the Internet; the rest of the Internet sees only the NAT router with the address 200.24.5.8.



**Figure 2.76A** NAT implementation

**2.10.8.1** Address Translation

All the outgoing packets go through the NAT router, which replaces the source address in the packet with the global NAT address. All incoming packets also pass through the NAT router, which replaces the destination address in the packet (the NAT router global address) with the appropriate private address. Figure 2.77 shows an example of address translation.



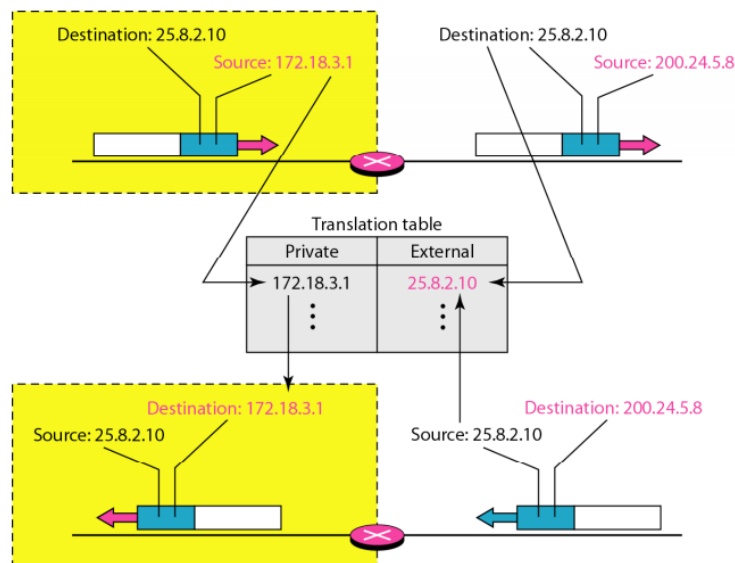
**Figure 2.77** Addresses in a NAT

### Translation Table

The NAT router has a translation table, which is used to find the destination address for a packet coming from the Internet. The translation table has only two columns;

- (i) The private address
- (ii) The external address (destination address of the packet).

When the router translates the source address of the outgoing packet, it also makes note of the destination address-where the packet is going. When the response comes back from the destination, the router uses the source address of the packet (as the external address) to find the private address of the packet which is shown in Figure 2.78.



**Figure 2.78 NAT address translation**

- In NAT, communication must always be initiated by the private network.
- NAT is used mostly by ISPs which assign one single address to a customer. The customer may be a member of a private network that has many private addresses. Here, communication with the Internet is always initiated from the customer site by using a client program such as HTTP, TELNET or FTP to access the corresponding server program.
- A private network cannot run a server program for clients outside of its network if it is using NAT technology.

#### 2.10.8.2 Using a Pool of IP Addresses

Since the NAT router has only one global address, only one private network host can access the same external host. To remove this restriction, the NAT router uses a pool of global addresses. For example, instead of using only one global address (200.24.5.8), the NAT router can use four addresses (200.24.5.8, 200.24.5.9, 200.24.5.10, and 200.24.5.11). In this case, four private network hosts can communicate with the same external host at the same time because each pair of addresses defines a connection.

**Drawbacks**

- (i) No more than four connections can be made to the same destination.
- (ii) No private-network host can access two external server programs at the same time.

In order to solve the above drawback we need more information in the translation table. The translation table may include the following five columns;

- (a) Private address
- (b) Private port
- (c) External address
- (d) External port
- (e) Transport layer protocol

For example, suppose two hosts with addresses 172.18.3.1 and 172.18.3.2 inside a private network need to access the HTTP server on external host 25.8.3.2. In this example, the ambiguity can be eliminated by using the translation table with five columns, as shown in Table 10.2.

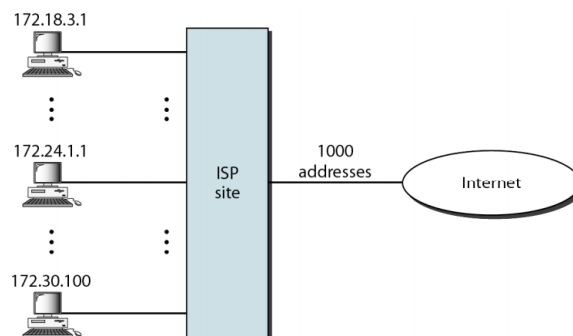
Private Address	Private Port	External Address	External Port	Transport Protocol
172.18.3.1	1400	25.8.3.2	80	TCP
172.18.3.2	1401	25.8.3.2	80	TCP
...	...	...	...	...

**Table 10.2 Five-column translation table**

From table 10.2, when the response from HTTP comes back, the combination of source address (25.8.3.2) and destination port number (1400) defines the private network host to which the response should be directed.

**2.10.8.3 NAT and ISP**

An ISP that serves dial-up customers can use NAT technology to conserve addresses. For example, suppose an ISP is granted 1000 addresses, but has 100,000 customers. Each of the customers is assigned a private network address. The ISP translates each of the 100,000 source addresses in outgoing packets to one of the 1000 global addresses. It translates the global destination address in incoming packets to the corresponding private address as shown in Figure 2.79.



**Figure 2.79 An ISP and NAT**



## 2.11 NETWORK LAYER PROTOCOLS (IP, ICMP AND MOBILE IP)

The network layer contains following 4 protocols as shown in Figure 2.80.

- (i) **Internet Protocol (IP)** : IP is the main protocol responsible for packetizing, forwarding, and delivery of a packet at the network layer.
- (ii) **Internet Control Message Protocol (ICMP)**: ICMP helps IP to handle some errors that may occur in the network-layer delivery.
- (iii) **Internet Group Management Protocol (IGMP)**: IGMP is used to help IPv4 in multicasting.
- (iv) **Address Resolution Protocol (ARP)**: ARP is used to glue the network and data-link layers in mapping network-layer addresses to link-layer addresses.

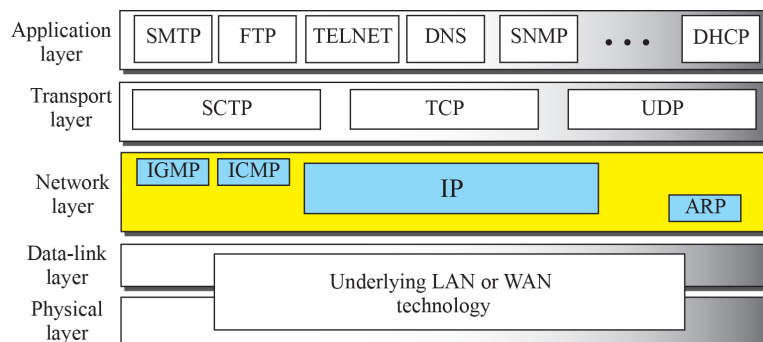


Figure 2.80 Position of IP and other network-layer protocols in TCP/IP protocol suite

### 2.11.1 Internet Protocol (IP)

IP is main protocol responsible for packetizing, forwarding & delivery of a packet at network layer. IP is an unreliable datagram protocol. IP provides a best-effort delivery service. The term best-effort means that the packets can

- be corrupted
- be lost or
- arrive out-of-order.

If reliability is important, IP must be paired with a TCP which is reliable transport-layer protocol. IP is a connectionless protocol. IP uses the datagram approach, which means

- (a) Each datagram is handled independently.
- (b) Each datagram can follow a different route to the destination.
- (c) Datagrams may arrive in out-of-order at the destination.

#### 2.11.1.1 Datagram Format

IP uses the packets called datagrams. A datagram consist of 2 parts namely **payload and header** as shown in Figure 2.81.

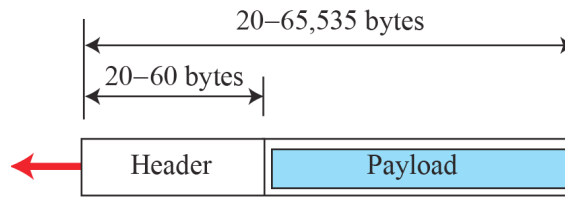


Figure 2.81 IP Datagram

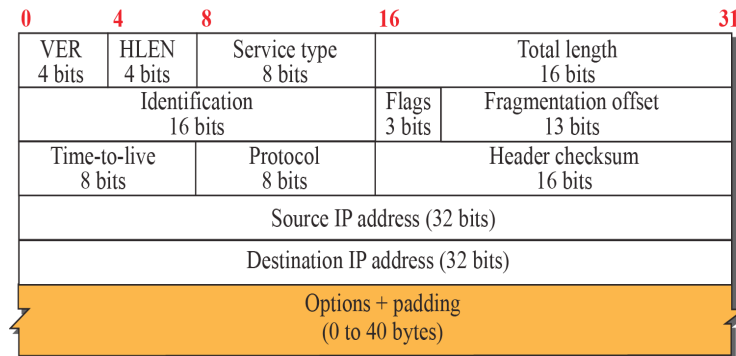


Figure 2.82 IP Datagram Header

### Fields of IP datagram

#### Payload

- Payload (or Data) is the main reason for creating a datagram.
- Payload is the packet coming from other protocols that use the service of IP.

#### Header

- Header contains information essential to routing and delivery.
- IP header contains following fields:

##### (i) Version Number (VER)

- field indicates version number used by the packet. Current version=4

##### (ii) Header Length (HLEN)

- This field specifies length of header.
- When a device receives a datagram, the device needs to know when the header stops and when the data starts.

##### (iii) Service Type

- This field specifies priority of packet based on delay, throughput, reliability & cost requirements.

##### (iv) Total Length

- This field specifies the total length of the datagram (header plus data).
- Maximum length=65535 bytes.

**(v) Identification, Flags, and Fragmentation Offset**

- These 3 fields are used for fragmentation and reassembly of the datagram.
- Fragmentation occurs when the size of the datagram is larger than the MTU of the network.

**(vi) Time-to-Live (TTL)**

- This field indicates amount of time, the packet is allowed to remain in the network.
- If TTL becomes 0 before packet reaches destination, the router discards packet and sends an error-message back to the source.

**(vii) Protocol**

- This field specifies upper-layer protocol that is to receive the packet at the destination-host.

**(viii) Header Checksum**

- This field is used to verify integrity of header only.
- If the verification process fails, packet is discarded.

**(ix) Source and Destination Addresses**

- These 2 fields contain the IP addresses of source and destination hosts.

**(x) Options**

- This field allows the packet to request special features such as
  - security level
  - route to be taken by packet and
  - timestamp at each router.
- This field can also be used for network testing and debugging.

**(xi) Padding**

- This field is used to make the header a multiple of 32-bit words.

**Example 11.1**

An IPv4 packet has arrived with the first 8 bits as  $(01000010)_2$ . The receiver discards the packet. Why?

**Solution:**

There is an error in this packet. The 4 left most bits  $(0100)_2$  show the version, which is correct. Then next 4 bits  $(0010)_2$  show an invalid header length ( $2 \times 4 = 8$ ). The minimum number of bytes in the header must be 20. The packet has been corrupted in transmission.

**Example 19.2**

In an IPv4 packet, the value of HLEN is  $(1000)_2$ . How many bytes of options are being carried by this packet?

**Solution:**

The HLEN value is 8, which means the total number of bytes in the header is  $8 \times 4$ , or 32 bytes. The first 20 bytes are the base header and the next 12 bytes are the options.

**2.11.1.2 Fragmentation**

**Maximum Transfer Unit (MTU)**

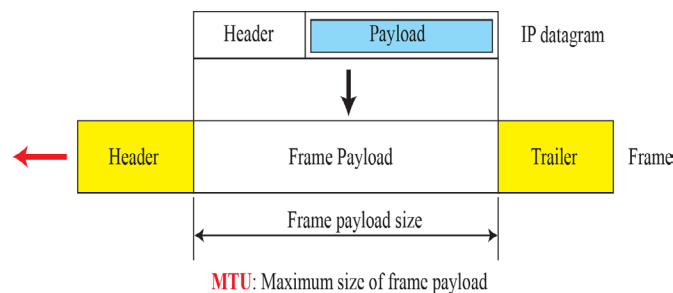
Each network imposes a restriction on maximum size of packet that can be carried. This is called the MTU (maximum transmission unit). For example;

For Ethernet, MTU = 1500 bytes

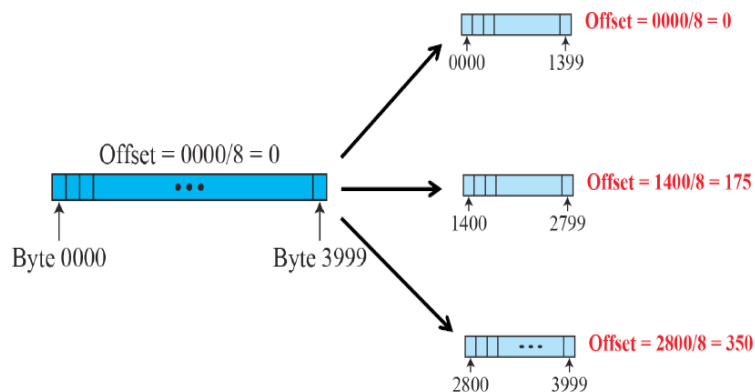
For FDDI, MTU = 4464 bytes

When IP wants to send a packet that is larger than MTU of physical-network, IP breaks packet into smaller fragments. This is called fragmentation. The maximum length of IP datagram is 65,535 bytes. This ensures that the IP protocol is independent of the physical network. When a datagram is fragmented, each fragment has its own header. A fragmented-datagram may itself be fragmented if it encounters a network with an even smaller MTU.

Source host or router is responsible for fragmentation of original datagram into the fragments. Destination host is responsible for reassembling the fragments into the original datagram.



**Figure 2.83 Maximum Transfer Unit (MTU)**



**Figure 2.84 Fragmentation example**

### ***Fields Related to Fragmentation & Reassembly***

Three fields in the IP header are used to manage fragmentation and reassembly. They are;

- Identification
- Flags
- Fragmentation offset.

#### ***Identification***

This field is used to identify to which datagram a particular fragment belongs to (so that fragments for different packets do not get mixed up).

To guarantee uniqueness, the IP protocol uses an up-counter to label the datagrams.

When the IP protocol sends a datagram, IP protocol copies the current value of the counter to the identification field and increments the up-counter by 1.

When a datagram is fragmented, the value in the identification field is copied into all fragments.

The identification number helps the destination in reassembling the datagram.

#### ***Flags***

This field has 3 bits.

- (a) The leftmost bit is not used.
- (b) DF bit (Don't Fragment):
  - If DF=1, the router should not fragment the datagram. Then, the router
    - discards the datagram and
    - sends an error-message to the source host.
  - If DF=0, the router can fragment the datagram if necessary.
- (c) MF bit (More Fragment):
  - If MF=1, there are some more fragments to come.
  - If MF=0, this is last fragment.

#### ***Fragmentation Offset***

- This field identifies location of a fragment in a packet.
- This field is the offset of the data in the original datagram.

#### ***2.11.1.3 Options***

- This field allows the packet to request special features such as
  - Security level

- Route to be taken by packet and
- Timestamp at each router
- This field can also be used for network testing and debugging.
- As the name implies, options are not required for a datagram.
- The header is made of two parts: a) Fixed part and b) Variable part.
  - Maximum size of Fixed part = 20 bytes.
  - Maximum size of Variable part = 40 bytes
- Options are divided into two broad categories: a) Single-byte options and b) Multiple-byte options.

### ***Single Byte Options***

- (i) No Operation:** This option is used as filler between options.
- (ii) End of Option:** This option is used for padding at the end of the option field.

### ***Multiple Byte Options***

#### **(i) Record Route**

- This option is used to record the routers that handle the datagram.
- This option can list up to 9 router-addresses.

#### **(ii) Strict Source Route**

- This option is used by the source to pre-determine a route for the datagram.
- Useful purposes: The sender can choose a route with a specific type of service, such as
  - minimum delay
  - maximum throughput or
  - more secure/reliable.
- All the defined-routers must be visited by the datagram.
- If the datagram visits a router that is not on the list, the datagram is discarded.

#### **(iii) Loose Source Route**

- This option is similar to the strict source route, but it is less rigid.
- Each router in the list must be visited, but the datagram can visit other routers as well.

#### **(iv) Timestamp**

- This option is used to record the time of datagram processing by a router.
- The time is expressed in milliseconds from midnight GMT (Greenwich Mean Time).
- The recorded-time can help the managers to track the behavior of the routers in the Internet.

#### 2.11.1.4 Security of IPv4 Datagrams

Nowadays, the Internet is not secure anymore. Three security issues applicable to the IP protocol, they are

- (i) Packet sniffing
- (ii) Packet modification and
- (iii) IP spoofing

##### **(i) Packet Sniffing**

- Attackers may
  - capture certain packets
  - intercept the packets and
  - make a copy of the packets.
- Packet sniffing is a passive attack.
- Passive attack means the attacker does not modify the contents of the packet.
- The attack is difficult to detect „sender & receiver may never know that the packet has been copied.
- Solution:
  - Although the attack cannot be stopped, encryption of packet may make the attacker's job difficult.
  - The attacker may still sniff the packet, but the content is not detectable (or understandable).

##### **(ii) Packet Modification**

- Attackers may succeed in accessing the content of a packet.
- Then, the attacker can
  - change the address of the packet or
  - change the data of the packet
- Solution:
  - The attack can be prevented by data integrity mechanism.
  - Data integrity guarantees that the packet is not modified during the transmission.

##### **(iii) IP Spoofing**

- The attacker pretends as a trusted entity and obtains all the secret information.
- For example: An attacker sends an IP packet to a bank pretending as legitimate customers.
- Solution: The attack can be prevented using an origin-authentication mechanism.

### ***IPSec (IP Security)***

IP packets can be protected from the various network-attacks using a protocol called IPSec. IPSec protocol & IP protocol can be used to create a connection-oriented service between 2 entities. Four services of IPSec;

#### ***(i) Defining Algorithms & Keys***

- To create a secure channel b/w two entities, the two entities can agree on some available algorithms and keys.

#### ***(ii) Packet Encryption***

- To provide privacy, the packets exchanged b/w two parties can be encrypted using the encryption-algorithms and a shared key.
- This prevents the packet sniffing attack.

#### ***(iii) Data Integrity***

- Data integrity guarantees that the packet is not modified during the transmission.
- If the received packet does not pass the data integrity test, the packet is discarded.
- This prevents the packet modification attack.

#### ***(iv) Origin Authentication***

- Origin Authentication guarantees that the packet is not created by a pretender.
- This prevents the IP Spoofing attack.

### **2.11.2 ICMP (INTERNET CONTROL MESSAGE PROTOCOL)**

The Internet Control Message Protocol (ICMP) has been designed to compensate for the above below two deficiencies. It is a companion to the IP protocol.

#### ***Draw backs of IP***

- (i) The IP protocol has no error-reporting or error-correcting mechanism.
- (ii) The IP protocol also lacks a mechanism for host and management queries.

#### ***Types of Messages***

ICMP messages are divided into two broad categories:

- (i) Error-reporting messages
- (ii) Query messages.

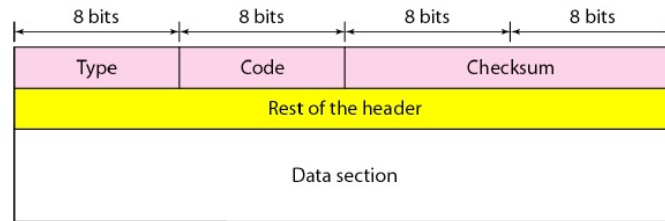
The error-reporting messages report problems that a router or a host (destination) may encounter when it processes an IP packet.

The query messages, which occur in pairs, help a host or a network manager get specific information from a router or another host. For example, nodes can discover their neighbors.



### 2.11.2.1 Message Format

An ICMP message has an 8-byte header and a variable-size data section. The general format of the header is different for each message type; the first 4 bytes are common to all.



**Figure 2.85 ICMP Message Format**

- (i) **Type:** Defines the type of the message.
- (ii) **Code:** specifies the reason for the particular message type.
- (iii) **Checksum :** Error detection and correction
- (iv) **Rest of the header:** Specific for each message type.
- (v) Data section:
  - (a) In error messages carries information for finding the original packet that had the error.
  - (b) In query messages, the data section carries extra information based on the type of the query.

### 2.11.2.2 Error Reporting

- ICMP does not correct errors-it simply reports them.
- Error correction is left to the higher-level protocols.
- Error messages are always sent to the original source because the only information available in the datagram about the route is the source and destination IP addresses.
- Five types of errors are handled:
  - (i) destination unreachable
  - (ii) source quench
  - (iii) Time exceeded
  - (iv) parameter problems
  - (v) redirection

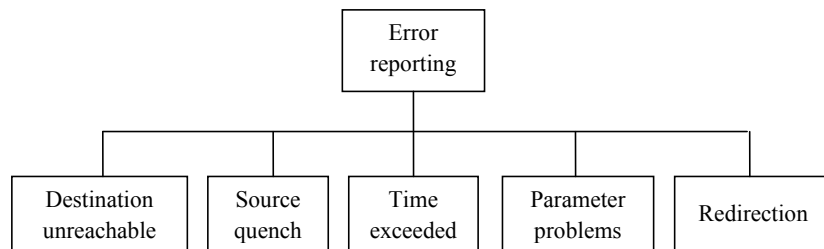
The following are important points about ICMP error messages:

- (i) No ICMP error message will be generated in response to a datagram carrying an ICMP error message.
- (ii) No ICMP error message will be generated for a fragmented datagram that is not the first fragment.

- (iii) No ICMP error message will be generated for a datagram having a multicast address.
- (iv) No ICMP error message will be generated for a datagram having a special address such as 127.0.0.0 or 0.0.0.0.

### **Types of error-reporting messages**

- ICMP error-reporting messages can be classified into following five categories;



**Figure 2.86 Types of ICMP error messages**

#### **(i) Source Quench**

- The IP protocol is a connectionless protocol.
- There is no communication between the source host, which produces the datagram, the routers, which forward it, and the destination host, which processes it (**lack off flow control**).
- The lack of flow control can create congestion in routers or the destination host.
- A router or a host has a limited-size queue (buffer) for incoming datagram's waiting to be forwarded (in the case of a router) or to be processed (in the case of a host).
- If the diagrams are received much faster than they can be forwarded or processed, the queue may overflow. In this case, the router or the host has no choice but to discard some of the diagrams.
- When a router or host discards a datagram due to congestion, it sends a source-quench message to the sender of the datagram.

#### **(ii) Destination Unreachable**

- When a router cannot route a datagram or a host cannot deliver a datagram, the datagram is discarded and the router or the host sends a destination-unreachable message back to the source host that initiated the datagram.
- The destination-unreachable messages can be created by either a router or the destination host.

#### **(iii) Time Exceeded**

- The time-exceeded message is generated in two cases:
  - (a) Routers use routing tables to find the next hop that must receive the packet. If there are errors in one or more routing tables, a packet can travel in a loop or a cycle,

going from one router to the next or visiting a series of routers endlessly. The TTL (time to live) controls this situation.

- (b) A time-exceeded message is also generated when not all fragments that make up a message arrive at the destination host within a certain time limit.

**(iv) Parameter Problem**

- Any ambiguity (missing values) in the header part of a datagram can create serious problems as the datagram travels through the Internet.
- If a router or the destination host discovers an ambiguous or missing value it will send the Parameter Problem to the source

**(v) Redirection**

- When a router needs to send a packet destined for another network, it must know the IP address of the next appropriate router (router - dynamic routing).
- The hosts usually use static routing. However, to update the routing table of the host, it sends a redirection message to the host.

### 2.11.2.3 ICMP Query Messages

Four different pairs of query messages are used for network management.

- (i) Echo Request and Reply:** Network managers and users utilize this pair of messages to identify network problems.
- (ii) Timestamp Request and Reply:** Two machines can use the timestamp request and timestamp reply messages to determine the round-trip time needed for transmission between them.
- (iii) Address-Mask Request and Reply:** Providing the necessary mask for the host (when using the subnets).
- (iv) Router Solicitation and Advertisement:** It announces not only its own presence but also the presence of all routers on the network.

### Checksum

In ICMP the checksum is calculated over the entire message (header and data).

### 2.11.2.4 Debugging Tools

Debugging tools are used to determine the viability of a host or router. Debugging tools for ICMP are ping and traceroute.

**(i) Ping :**

- (a) To find if a host is alive and responding.
- (b) To see how it uses ICMP packets.

**ii) Traceroute:** Used to trace the route of a packet from the source to the destination.

### *Ping*

- The ping program can be used to find if a host is alive and responding
- Here, ping is used to see how it uses ICMP packets
- The source host sends ICMP echo-request messages;
- The destination, if alive, responds with ICMP echo-reply messages.
- The ping program
  - sets the identifier field in the echo-request and echo-reply message and
  - starts the sequence number from 0; this number is incremented by 1 each time a new message is sent.
- Ping can calculate the round-trip time.
- It inserts the sending time in the data section of the message.
- When the packet arrives, it subtracts the arrival time from the departure time to get the round-trip time (RTT).

### *Traceroute*

- The traceroute program can be used to trace the path of a packet from a source to the destination.
- It can find the IP addresses of all the routers that are visited along the path.
- The program is usually set to check for the maximum of 30 hops (routers) to be visited.
- The traceroute program is different from the ping program.
- The ping program gets help from 2 query messages.
- The traceroute program gets help from two error-reporting messages: time-exceeded and destination-unreachable.
- The traceroute is an application layer program, but only the client program is needed. In other words, there is no traceroute server program.
- The traceroute application program is encapsulated in a UDP user datagram, but traceroute intentionally uses a port number that is not available at the destination.

## **2.11.3 MOBILE IP**

Mobile IP is the extension of IP protocol. Mobile IP allows mobile computers to be connected to the Internet.

### *2.11.3.1 Addressing*

- In Mobile IP, the main problem that must be solved is addressing.

#### *(i) Stationary Hosts*

- The original IP addressing assumed that a host is stationary.
- A router uses an IP address to route an IP datagram.

- An IP address has two parts: a prefix and a suffix.
- The prefix associates a host with a network.  
For example, the IP address 10.3.4.24/8 defines a host attached to the network 10.0.0.0/8.
- The address is valid only when the host is attached to the network.
- If the network changes, the address is no longer valid.

### (ii) Mobile Hosts

- When a host moves from one network to another, the IP addressing structure needs to be modified.
- The host has two addresses (Figure 2.87).
  - a) Home address &
  - b) Care-of address

#### Home Address

- Original address of host called the home address.
- The home address is permanent.
- The home address associates the host with its home network.
- Home network is a network that is the permanent home of the host.

#### Care-of-Address

- The care-of address is temporary.
- The care-of address changes as the mobile-host moves from one network to another.
- Care-of address is associated with the foreign network.
- Foreign network is a network to which the host moves.
- When a mobile-host visits a foreign network, it receives its care-of address during the agent discovery and registration phase.

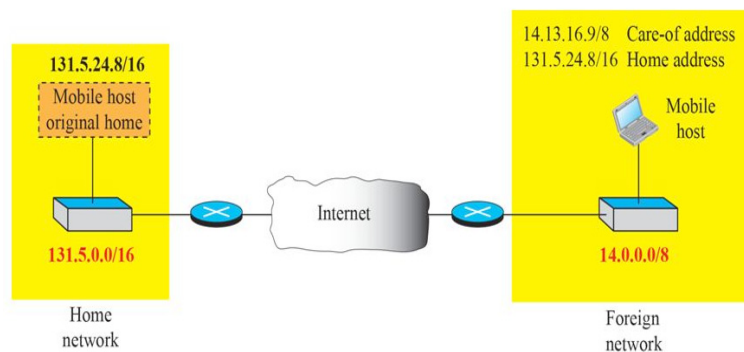
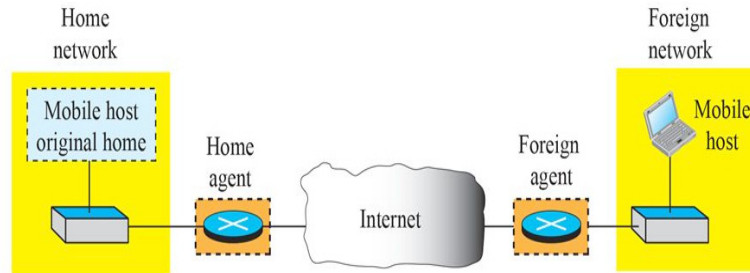


Figure 2.87 Home address and Care-of address

### 2.11.3.2 Agents

Two agents are required to make change of address transparent to rest of the Internet as shown in figure 2.88. two types of agents are;

- (a) Home-agent and
- (b) Foreign-agent.



**Figure 2.88 Home-agent and Foreign-agent**

#### **Home Agent**

- The home-agent is a router attached to the home network.
- The home-agent acts on behalf of mobile-host when a remote-host sends a packet to mobile-host.
- The home-agent receives and delivers packets sent by the remote-host to the foreign-agent.

#### **Foreign Agent**

- The foreign-agent is a router attached to the foreign network.
- The foreign-agent receives and delivers packets sent by the home-agent to the mobile-host.
- The mobile-host can also act as a foreign-agent i.e. mobile-host and foreign-agent can be the same.
- However, to do this, a mobile-host must be able to receive a care-of address by itself.
- In addition, the mobile-host needs the necessary software to allow it to communicate with the home- agent and to have two addresses: i) its home address and ii) its care-of address.
- This dual addressing must be transparent to the application programs.

#### **Collocated Care-of-Address**

- When the mobile-host and the foreign-agent are the same, the care-of-address is called a collocated care-of-address.
- **Advantage:**
  - mobile-host can move to any network w/o worrying about availability of a foreign-agent.
- **Disadvantage:**
  - The mobile-host needs extra software to act as its own foreign-agent.

### 2.11.3.3 Three Phases

- To communicate with a remote-host, a mobile-host goes through 3 phases as shown in Figure 2.89.

**Agent Discovery:** involves the mobile-host, the foreign-agent, and the home-agent.

**Registration:** involves the mobile-host, the foreign-agent, and the home-agent.

**Data Transfer:** Here, the remote-host is also involved.

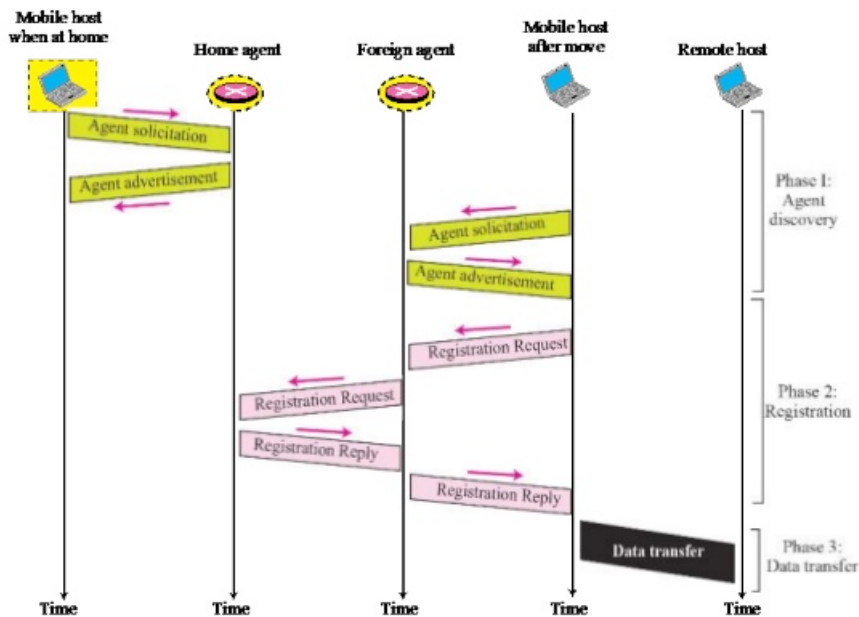


Figure 2.89 Remote Host and Mobile Host Communication

#### Agent Discovery

- Agent discovery consists of two sub-phases:

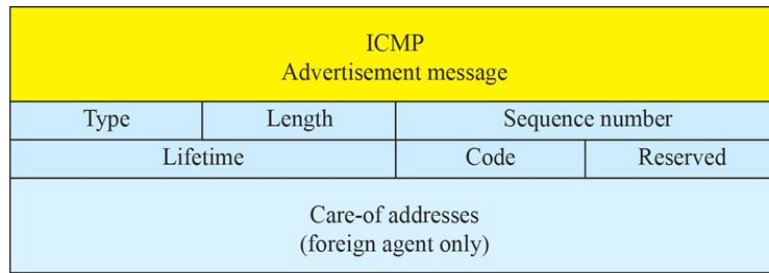
A mobile-host must discover (learn the address of) a home-agent before it leaves its home network.

A mobile-host must also discover a foreign-agent after it has moved to a foreign network.

- This discovery consists of learning the care-of address as well as the foreign-agent's address.
- Two types of messages are used: i) advertisement and ii) solicitation.

#### Agent Advertisement

- When a router advertises its presence on a network using an ICMP router advertisement, it can append an agent advertisement to the packet if it acts as an agent.



**Figure 2.90 Agent Advertisement**

- Various fields of agent advertisement are (Figure 2.90)
  - (i) **Type**
    - This field is set to 16.
  - (ii) **Length**
    - This field defines the total length of the extension message.
  - (iii) **Sequence Number**
    - This field holds the message number.
    - The recipient can use the sequence number to determine if a message is lost.
  - (iv) **Lifetime**
    - This field defines the number of seconds that the agent will accept requests.
    - If the value is a string of 1s, the lifetime is infinite.
  - (v) **Code**
    - This field is a flag in which each bit is set (1) or unset (0) (Table 11.1).
  - (vi) **Care-of Addresses**
    - This field contains a list of addresses available for use as care-of addresses.
    - The mobile-host can choose one of these addresses.
    - The selection of this care-of address is announced in the registration request.

**Table 11.1 Code Bits**

Bit	Meaning
0	Registration required. No collocated care-of address.
1	Agent is busy and does not accept registration at this moment.
2	Agent acts as a home agent.
3	Agent acts as a foreign agent.
4	Agent uses minimal encapsulation.
5	Agent uses generic routing encapsulation (GRE).
6	Agent supports header compression.
7	Unused (0).

**Table 11.1 Code Bits**



### Agent Solicitation

- When a mobile-host has moved to a new network and has not received agent advertisements, it can initiate an agent solicitation.
- It can use the ICMP solicitation message to inform an agent that it needs assistance

#### 2.11.3.4 Registration

- After a mobile-host has moved to a foreign network and discovered the foreign-agent, it must register.
- Four aspects of registration:
  - The mobile-host must register itself with the foreign-agent.
  - The mobile-host must register itself with its home-agent. This is normally done by the foreign-agent on behalf of the mobile-host.
  - The mobile-host must renew registration if it has expired.
  - The mobile-host must cancel its registration (deregistration) when it returns home.

##### 2.11.3.4.1 Request & Reply

- To register with the foreign-agent and the home-agent, the mobile-host uses a registration request and a registration reply.

#### Registration Request

- A registration request is sent from the mobile-host to the foreign-agent
  - to register its care-of address and
  - to announce its home address and home-agent address.
- Foreign-agent, after receiving and registering the request, relays the message to the home-agent.
- The home-agent now knows the address of the foreign-agent because the IP packet that is used for relaying has the IP address of the foreign-agent as the source address.

Type	Flag	Lifetime
Home address		
Home agent address		
Care-of address		
Identification		
Extensions ...		

**Figure 2.91 Registration request format**

Various fields of Registration request are (Figure 2.91)

#### **Type**

- This field defines the type of message.

- For a request message the value of this field is 1.

**Flag**

- This field defines forwarding information.
- The value of each bit can be set or unset (Table 11.2).

<i>Bit</i>	<i>Meaning</i>
0	Mobile host requests that home agent retain its prior care-of address.
1	Mobile host requests that home agent tunnel any broadcast message.
2	Mobile host is using collocated care-of address.
3	Mobile host requests that home agent use minimal encapsulation.
4	Mobile host requests generic routing encapsulation (GRE).
5	Mobile host requests header compression.
6–7	Reserved bits.

**Table 11.2 Registration request flag field bits****Lifetime**

- This field defines the number of seconds the registration is valid.  
If the field is a string of 0s, the request message is asking for deregistration.  
This field If the field is a string of 1s, the lifetime is infinite.

**Home Address**

- This field contains the permanent (first) address of the mobile-host.

**Home Agent Address**

- This field contains the address of the home-agent.

**Care-of-Address**

- This field is the temporary (second) address of the mobile-host.

**Identification**

- This field contains a 64-bit number that is inserted into the request by the mobile-host.
- This field matches a request with a reply.

**Extensions**

- This field is used for authentication.
- This field allows a home-agent to authenticate the mobile agent.

**Registration Reply**

- A registration reply is sent from home-agent to foreign-agent and then relayed to the mobile-host.
- The reply confirms or denies the registration request (Figure 2.92).

- The fields are similar to registration request with the 3 exceptions:  
The value of the type field is 3.  
The code field replaces the flag field and shows the result of the registration request (acceptance or denial).  
The care-of address field is not needed.

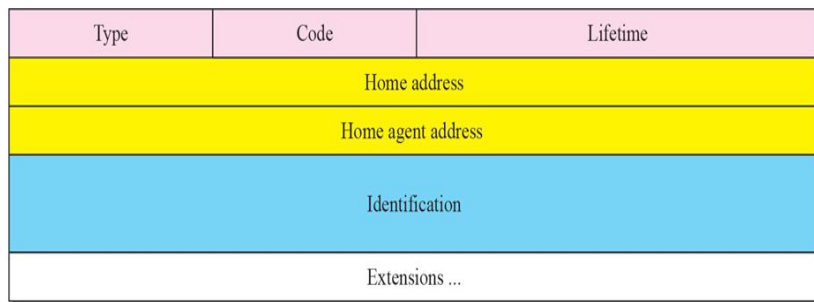


Figure 2.92 Registration reply format

#### 2.11.3.4.2 Data Transfer

- After agent discovery & registration, a mobile-host can communicate with a remote-host.

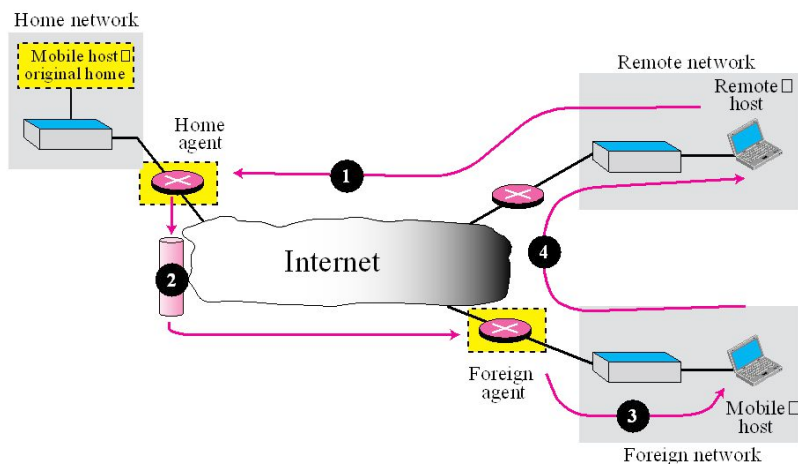


Figure 2.93 Data Transfer

Here we have 4 cases (Figure 2.93);

##### (i) From Remote Host to Home Agent

- When a remote-host wants to send a packet to the mobile-host, the remote-host uses
  - address of itself as the source address and
  - home address of the mobile-host as the destination address.
- In other words, the remote-host sends a packet as though the mobile-host is at its home network.

- The packet is intercepted by the home-agent, which pretends it is the mobile-host.
- This is done using the proxy ARP technique (Path 1 of Figure 2.93)

**(ii) *From Home Agent to Foreign Agent***

- After receiving the packet, the home-agent sends the packet to the foreign-agent, using the tunneling concept.
- The home-agent encapsulates the whole IP packet inside another IP packet using its address as the source and the foreign-agent's address as the destination. (Path 2 of Figure2.93).

**(iii) *From Foreign Agent to Mobile Host***

- When the foreign-agent receives the packet, it removes the original packet.
- However, since the destination address is the home address of the mobile-host, the foreign-agent consults a registry table to find the care-of address of the mobile-host. (Otherwise, the would just be sent back to the home network.)
- The packet is then sent to the care-of address (Path 3 of Figure2.93).

**(iv) *From Mobile Host to Remote Host***

- When a mobile-host wants to send a packet to a remote-host (for example, a response to the packet it has received), it sends as it does normally.
- The mobile-host prepares a packet with its home address as the source, and the address of the remote-host as the destination.
- Although the packet comes from the foreign network, it has the home address of the mobile-host (Path 4 of Figure 2.93).

**2.11.3.4.3 *Inefficiency in Mobile IP***

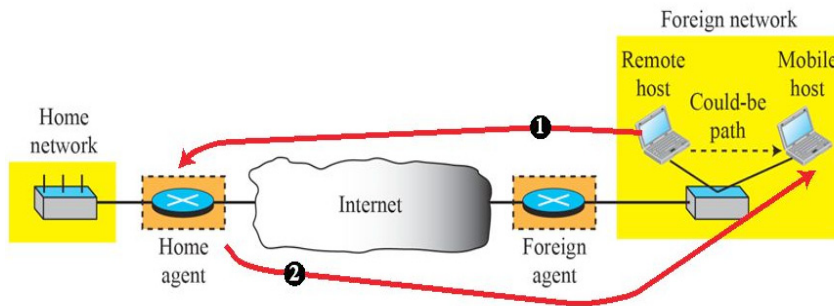
- Communication involving mobile IP can be inefficient.
- The inefficiency can be severe or moderate.
  - (i) The severe case is called double crossing or 2X.
  - (ii) The moderate case is called triangle routing or dog-leg routing.

***Double Crossing***

- Double crossing occurs when a remote-host communicates with a mobile-host that has moved to the same network (or site) as the remote-host.
- When the mobile-host sends a packet to the remote-host, there is no inefficiency; the communication is local.
- However, when remote-host sends a packet to mobile-host, the packet crosses the Internet twice.
- Since a computer usually communicates with other local computers (principle of locality), the inefficiency from double crossing is significant.

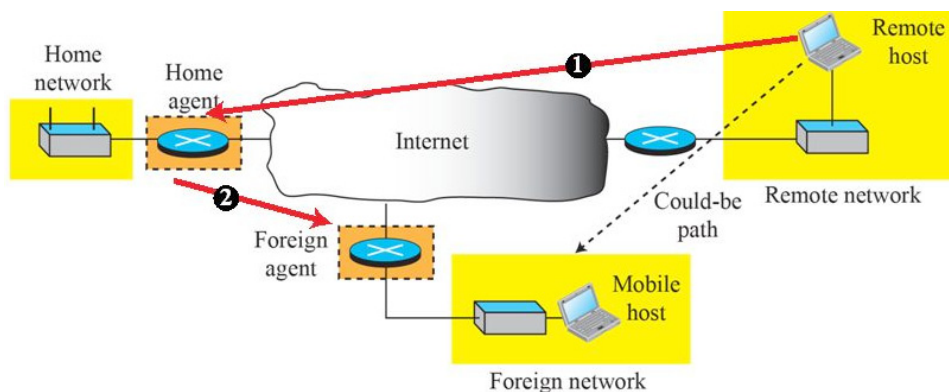
### Triangle Routing

- Triangle routing occurs when the remote-host communicates with a mobile-host that is not attached to the same network (or site) as the mobile-host.
- When the mobile-host sends a packet to the remote-host, there is no inefficiency.



**Figure 2.94 Double Crossing**

- However, when the remote-host sends a packet to the mobile-host, the packet goes from the remote-host to the home-agent and then to the mobile-host.
- The packet travels the two sides of a triangle, instead of just one side (Figure 2.95).



**Figure 2.95 Triangular Routing**

### Solution

- One solution to inefficiency is for the remote-host to bind the care-of address to the home address of a mobile-host.
- For example, when a home-agent receives the first packet for a mobile-host, it forwards the packet to the foreign-agent; it could also send an update binding packet to the remote-host so that future packets to this host could be sent to the care-of address.
- The remote-host can keep this information in a cache.
- The problem with this strategy is that the cache entry becomes outdated once the mobile-host moves.
- In this case, the home-agent needs to send a warning packet to the remote-host to inform it of the change.











# ROUTING

---

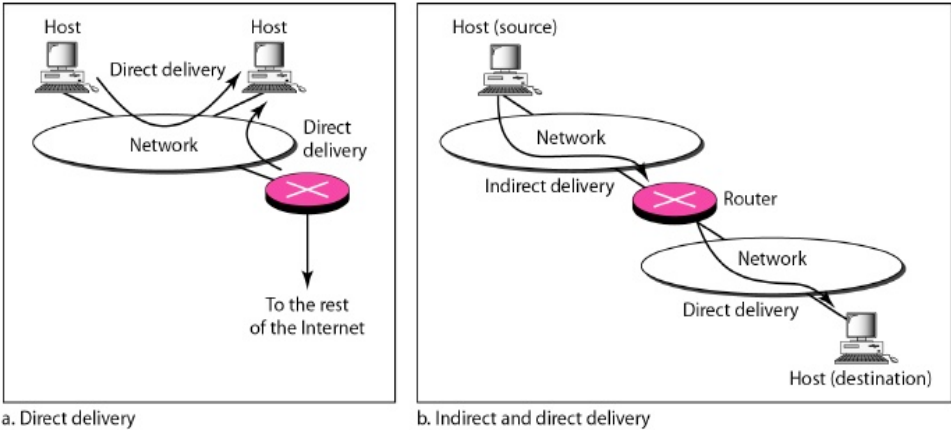
## 3.1 ROUTING

In today’s networking world, it is important to know about the delivery, forwarding, and routing of IP packets to their final destinations. Delivery refers to the way a packet is handled by the underlying networks under the control of the network layer.

Forwarding refers to the way a packet is delivered to the next station. Routing refers to the way routing tables are created to help in forwarding. Routing protocols are used to continuously update the routing tables that are consulted for forwarding and routing.

### 31.1 Delivery

The network layer supervises the handling of the packets by the underlying physical networks. This handling is referred as the delivery of a packet. The delivery of a packet to its final destination can be done in one of the following methods.



**Figure 3.1 Direct and indirect delivery**

- (i) **Direct Delivery:** Direct delivery occurs when the source and destination of the packet are located on the same physical network or when the delivery is between the last router and the destination host.
- (ii) **Indirect Delivery:** If the destination host is not on the same network as the deliverer, the packet is delivered indirectly. The packet goes from router to router until it reaches the

one connected to the same physical network as its final destination. It always involves one direct delivery but zero or more indirect deliveries.

### 3.1.2 Forwarding

Forwarding is the technique to place the packet in its route to its destination. Forwarding requires a host or a router to have a routing table. When a host has a packet to send or when a router has received a packet to be forwarded, it looks at this table to find the route to the final destination.

This solution is impossible because the number of entries needed in the routing table would make table lookups inefficient.

#### 3.1.2.1 Forwarding Techniques

The following techniques can make the size of the routing table manageable and also handle security issues.

##### (i) Next-Hop Method versus Route Method

- It is used to reduce the contents of a routing table.
- The routing table holds only the address of the next hop instead of information about the complete route (route method).
- The entries of a routing table must be consistent with one another.

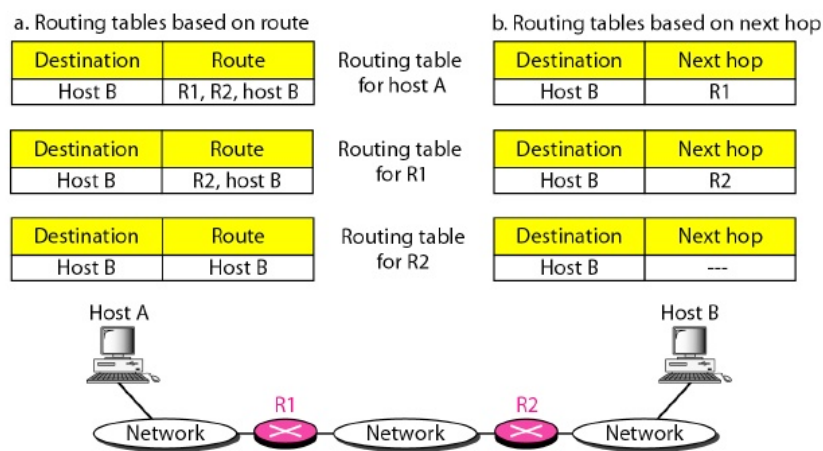


Figure 3.2 Next-Hop Method - Routing Tables

##### (ii) Network-Specific Method versus Host-Specific Method

- It reduces the routing table and simplifies the searching process.
- Instead of having an entry for every destination host connected to the same physical network, only one entry is used to define the address of the destination network itself.
- That means all hosts connected to the same network are treated as one single entity.
- It is used for checking the route and providing security.

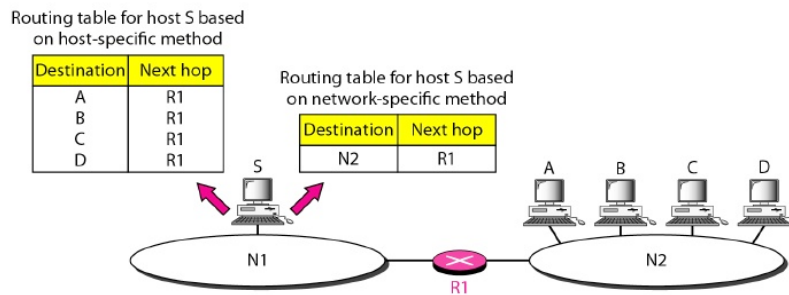


Figure 3.3 Host-specific versus network-specific method

(iii) Default Method

- It is used to simplify the routing table.
- In figure 3.4, host A is connected to a network with two routers.

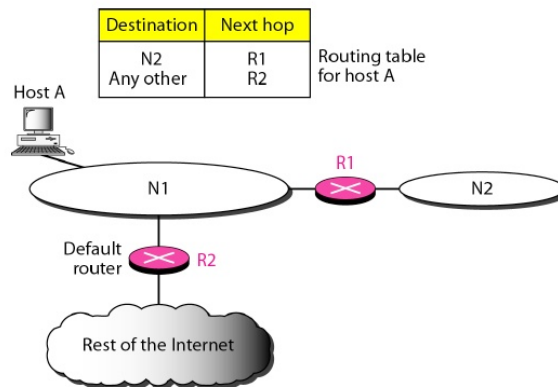


Figure 3.4 Default method

- Router R1 routes the packets to hosts connected to network N2, for the rest of the Internet, router R2 is used.
- So instead of listing all networks in the entire Internet, host A can just have one entry called the default (normally defined as network address 0.0.0.0).

3.1.2.2 Forwarding Process

- In classless addressing, the routing table needs to be searched based on the network address (first address in the block).
- But the destination address in the packet gives no clue about the network address.
- To solve the problem, the mask (/n) is added in the table.
- In classless addressing, we need at least four columns in a routing table (Mask, Network address, Next-hop address and interface).

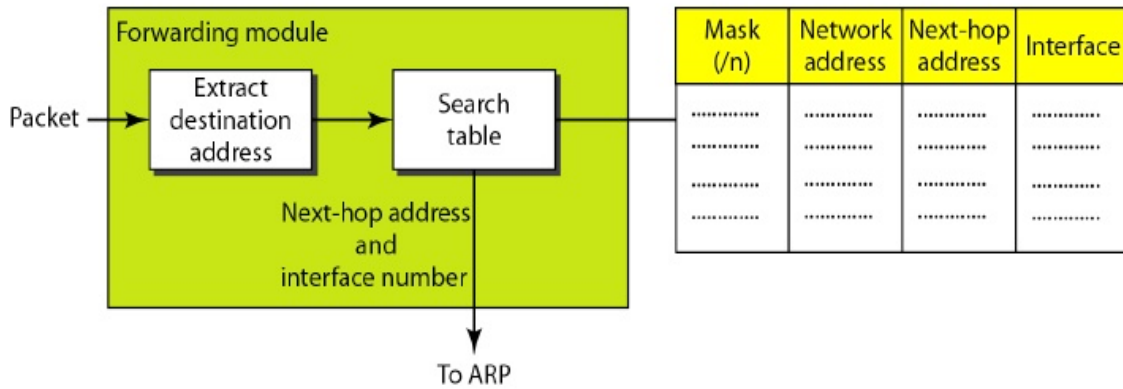


Figure 3.5 Simplified forwarding module in classless address

**Example 1**

Make a routing table for router R1, using the configuration in Figure 3.6.

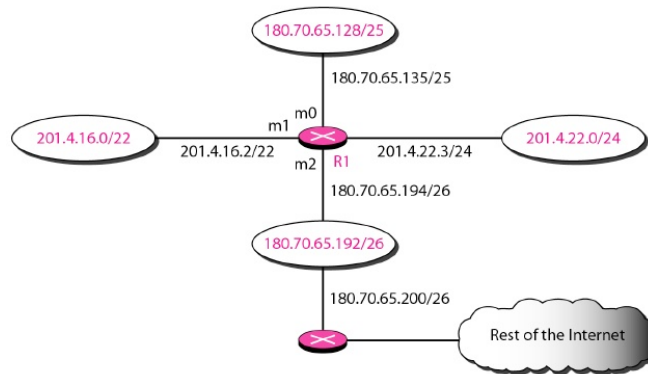


Figure 3.6 Configuration for Example 1

**Solution:**

Mask	Network	Next Hop	Interface
/26	180.70.65.192	--	m2
/25	180.70.65.128	--	m0
/24	201.4.22.0	--	m3
/22	201.4.16.0	...	m1
Any	Any	180.70.65.200	m2

Table 3.1 Routing table for router R1 in Figure 3.6

**(i) Hierarchical Routing**

- To solve the problem of extremely large routing tables, we can create a hierarchy in the routing tables.

- If the routing table has a sense of hierarchy, the routing table can decrease in size.

**(ii) Geographical Routing**

- It is used to decrease the size of the routing table.
- It divides the entire address space into a few large blocks.

**3.1.2.3 Routing Table**

- A routing table needs a minimum of 4 columns: mask, destination network address, next hop address and interface.
- When a packet arrives, the router applies the mask to the destination address it receives (one-by-one until a match is found) in order to find the corresponding destination network address.
- So, the mask serves as essential tool to match destination address in routing table and the address it receives.
- If found, the packet is sent out from the corresponding interface in the table. If not found, the packet is delivered to the default interface which carries the packet to default router.
- The routing table can be either static or dynamic.

**(i) Static Routing Table**

- The administrator enters the information (route for each destination) into the table manually.
- When a table is created, it cannot update automatically.
- The table must be manually altered by the administrator.
- It is suitable for a small internet an experimental internet for troubleshooting.

**(ii) Dynamic Routing Table**

- Routing table information is created by using one of the dynamic routing protocols such as RIP, OSPF, or BGP.
- Whenever there is a change in the Internet (shutdown of a router or breaking of a link), the dynamic routing protocols update all the tables in the routers automatically.
- It is suitable for a big internet.

**Format**

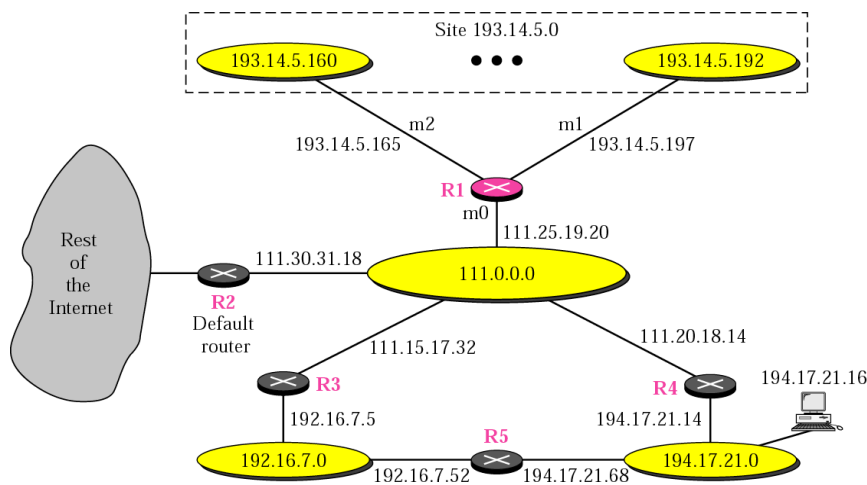
A routing table for classless addressing has a minimum of four columns, which are listed below. Some of the routers may have more columns.

Mask	Network address	Next-hop address	Interface	Flags	Reference count	Use
-----	-----	-----	-----	-----	-----	-----

*Figure 3.7 Common fields in a routing table*

- (i) **Mask:** This field defines the mask applied for the entry.
- (ii) **Network address:** It defines the network address to which the packet is finally delivered. In the case of host-specific routing, this field defines the address of the destination host.
- (iii) **Next-hop address:** It defines the address of the next-hop router to which the packet is delivered.
- (iv) **Interface:** It shows the name of the interface.
- (v) **Flags:** It defines up to five flags. Flags are on/off switches that signify either presence or absence. The five flags are;
- 1) **U (up):** The U flag indicates the router is up and running. If this flag is not present, it means that the router is down. The packet cannot be forwarded and is discarded.
  - 2) **G (gateway):** The G flag means that the destination is in another network. The packet is delivered to the next-hop router for delivery. When this flag is missing, it means the destination is in this network (direct delivery).
  - 3) **H (host-specific):** The H flag indicates that the entry in the network address field is a host-specific address. When it is missing, it means that the address is only the network address of the destination.
  - 4) **D (added by redirection):** The D flag indicates that routing information for this destination has been added to the host routing table by a redirection message from ICMP.
  - 5) **M (modified by redirection):** The M flag indicates that the routing information for this destination has been modified by a redirection message from ICMP.
- (vi) **Reference count:** This field gives the number of users of this route at the moment.
- (vii) **Use:** This field shows the number of packets transmitted through this router for the corresponding destination.

### 3.1.3 Configuration for Routing Example



	Mask	Dest	Next Hop	I.
Standard delivery	255.0.0.0	111.0.0.0	--	m0
	255.255.255.224	193.14.5.160	-	m2
Host-specific	255.255.255.224	193.14.5.192	-	m1
	255.255.255.255	194.17.21.16	111.20.18.14	m0
Network-specific	255.255.255.0	192.16.7.0	111.15.17.32	m0
	255.255.255.0	194.17.21.0	111.20.18.14	m0
Default	0.0.0.0	0.0.0.0	111.30.31.18	m0

*Figure 3.8 Routing configuration and routing table*

## 3.2. UNICAST ROUTING

An internet is a combination of networks connected by routers. When a packet goes from a source to a destination, it will pass through many routers until it reaches the router attached to destination network.

A router consults a routing table when a packet is ready to be forwarded. The routing table specifies the optimum path for the packet and can be either static or dynamic. Dynamic routing is more popular.

A static routing table contains information entered manually. Static table does not change frequently.

Dynamic table is updated automatically when there is a change somewhere in the network. That is, when a route is down or a better route has been created. A dynamic routing table is updated using one of the dynamic routing protocols such as RIP, OSPF, or BGP.

Routing protocols is a combination of rules/procedures that lets routers in the internet inform one another when changes occur. The change may occur based on sharing/combining information between routers at different networks.

### Unicast Routing

Unicast routing is a 1-to-1 relationship (one source and one destination). In unicast routing, when a router receives a packet, it forwards the packet through only one of its ports as defined in the routing table. The router may discard the packet if it cannot find the destination address. Here we have to answer the following questions;

- (1) In dynamic routing, how does the router decide to which network should it pass the packet next?
- (2) What routing algorithm is the routing based on?
- (3) Which of the available pathways is the best/optimum path? How is it measured?

#### 3.2.1 Metrics

A metric is a cost assigned for passing through a network and the total metric of a particular route is equal to the sum of the metrics of networks that comprise the route.



Simple protocols such as Routing Information Protocol (RIP) treat all networks equally. That means the cost of passing each network is the same as one hop count per network.

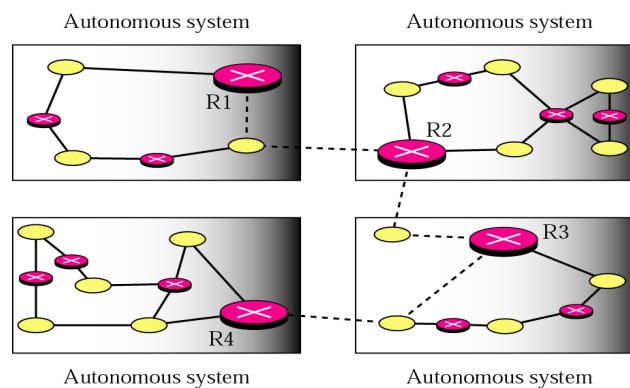
Other protocols like OSPF, allow the administrator to assign a cost for passing through a network based on the type of service required. A route through a network can have different costs (metrics).

### 3.2.2 Routing Architecture in the Internet

Nobody owns the whole Internet. However, parts of the Internet are owned and administered by commercial and public organisations (such as ISPs, universities, governmental offices, research institutes, companies etc.).

The Internet can be divided into an Autonomous Systems (AS). The ASs is independently administered by individual organisations. Each administrative authority uses its own routing protocol within the AS to exchange routing information among AS.

An AS is a group of networks and routers under the authority of a single administrator.



**Figure 3.9 Autonomous systems**

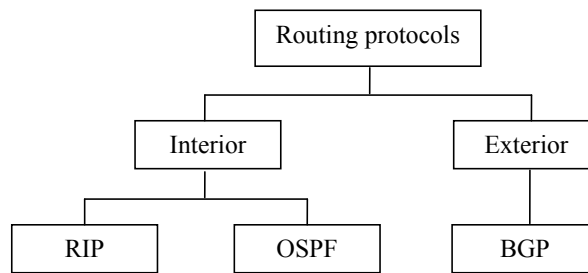
- Routing inside an AS is referred to as intra-domain (interior) routing whereas routing between ASs is referred to as inter-domain (exterior) routing.
- Each AS can choose one or more interior routing protocols inside an AS.
- Only one exterior routing protocol is usually chosen to handle routing between ASs.
- To know the next 'path' (or router) a packet should be pass-on, the decision is based on some optimisation rule/protocol.

### 3.2.3 Routing Protocols

Several intra-domain and inter-domain routing protocols are in use. The two most popular intra-domain routing protocols are;

- (i) Distance vector routing protocol
- (ii) Link state routing protocol

The most popular inter-domain routing protocol is a path vector routing protocol. Figure 3.10 shows the taxonomy of unicast routing protocols.



**Figure 3.10 Popular routing protocols**

- Routing Information Protocol (RIP) is an implementation of the distance vector protocol.
- Open Shortest Path First (OSPF) is an implementation of the link state protocol.
- Border Gateway Protocol (BGP) is an implementation of the path vector protocol.

### 3.3. ROUTING INFORMATION PROTOCOL (RIP)

RIP is based on distance vector routing, which uses the Bellman-Ford algorithm for calculating the routing table. The Routing Information Protocol (RIP) is an intra-domain routing protocol used inside an autonomous system. It is a very simple protocol based on distance vector routing.

- RIP treats all network equals. That means the cost of passing through a network is the same (one hop count per network).
- Each router/node maintains a vector (table) of minimum distances to every node.
- The least-cost route between any nodes is the route with the minimum number of hop-count.
- The hop-count is the number of networks that a packet encounters to reach its destination. Path costs are based on number of hops.
- In distance vector routing, each router periodically shares its knowledge about the entire internet with its neighbour.
- Each router keeps a routing table that has one entry for each destination network of which the router is aware.
- The entry consists of Destination Network Address/id, Hop-Count and Next-Router.

#### RIP implementation

RIP implementations distance vector routing directly with the following considerations;

- (1) In an autonomous system, we are dealing with routers and networks (links). The routers have routing tables and the networks do not.
- (2) The destination in a routing table is a network, which means the first column defines a network address.
- (3) The metric used by RIP is very simple. The distance is defined as the number of links (networks) to reach the destination. For this reason, the metric in RIP is called a hop count.

- (4) Infinity is defined as 16, which means that any route in an autonomous system using RIP cannot have more than 15 hops.
- (5) The next-node column defines the address of the router to which the packet is to be sent to reach its destination.

### 3.3.1 Distance Vector Routing

In distance vector routing, the least-cost route between any two nodes is the route with minimum distance. Each node maintains a vector (table) of minimum distances to every node. The table at each node also guides the packets to the desired node by using the next-hop routing.

In distance vector routing, each router periodically shares its knowledge about the entire network with its neighbor. The three keys to understand how this algorithm works as follows;

- (1) Sharing knowledge about the entire AS: Each router shares its knowledge about the entire AS with neighbours. It sends whatever it has.
- (2) Sharing only with immediate neighbours: Each router sends whatever knowledge it has thru its entire interface.
- (3) Sharing at regular intervals: Each router sends its information about the whole network to its neighbours at fixed intervals. For example at every 30 sec.

#### 3.3.1.1 Sharing information

The whole idea of distance vector routing is the sharing of information between neighbors. Figure 3.11 shows the Example of an Internet. In this figure 3.11,

- (i) Clouds represents the LAN's
- (ii) Boxes labeled with A, B... represents routers and their relationships to the neighboring routers.
- (iii) Cost is based on hop count

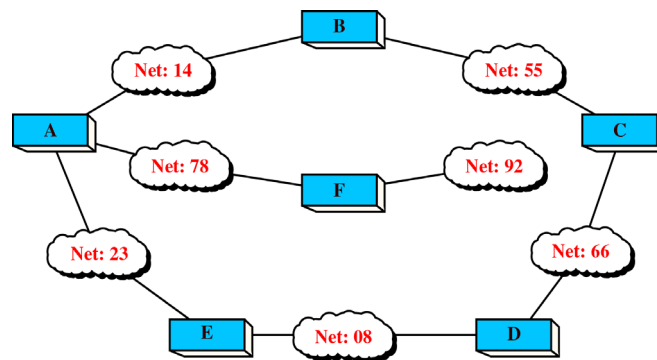
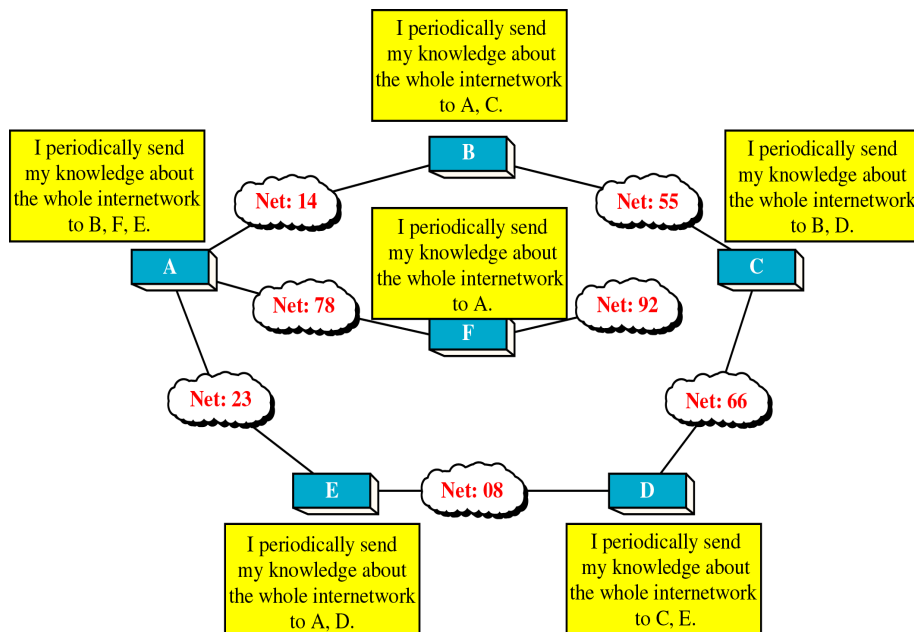


Figure 3.11 Example of an Internet

In distance vector routing a cost of one unit is assigned for every link. The efficiency of transmission is a function, composed by the number of links required to reach a destination. In distance vector routing the cost is based on hop count.

Figure 3.12 shows the concept of distance vector routing. Each router sends its information about the network only to its immediate neighbors.



**Figure 3.12** The concept of distance vector routing

A route sends its knowledge to its neighbors. The neighbors add this knowledge to their own knowledge and send the whole table to their own neighbors. In this way, the first router gets its own information back with new additional information about its neighbors and other neighbors.

Each of these neighbors adds its knowledge and sends the updated table on to its own neighbors, and so on.

### 3.3.1.2 Routing table

Let us see, how each router gets its initial knowledge about the internetwork and how it uses the shared information to update the knowledge.

#### Creating the routing table

At the initial stage, a router’s knowledge of the internetwork is sparse. Every router knows that, it is connected to some number of LANs. The router knows the ID of each station in the LAN, because the router is also one of the stations in that LAN.

The port ID and the network Id are used by the router to find the logical address of the station. With the help of logical address, router can find the LAN into which the station is attached. Based on this information, a router can construct its initial routing table. Figure 3.13 shows the distance vector routing table.

Network ID	Cost	Next Hop
-----	-----	-----
-----	-----	-----
-----	-----	-----
-----	-----	-----

**Figure 3.13** Distance vector routing table

The routing table contains the following three fields. They are Network ID, Cost and Next Hop.

- (i) Network ID – Final destination of the packet.
- (ii) Cost – Number of hops the packet must make to get there.
- (iii) Next hop – The next router to which the packet must be delivered.

Figure 3.14 shows the routing table distribution in distance vector routing. In this figure 3.14, all the routers are have their original routing tables.

The third column is empty, because the only destination network identified are those attached to the current router. Next routers have not been identified here, so no multiple hop destinations. The basic tables are sent out to neighbors.

For example, A sends its routing table to routers B, F and E. B sends its routing table to routers C and A, and so on.

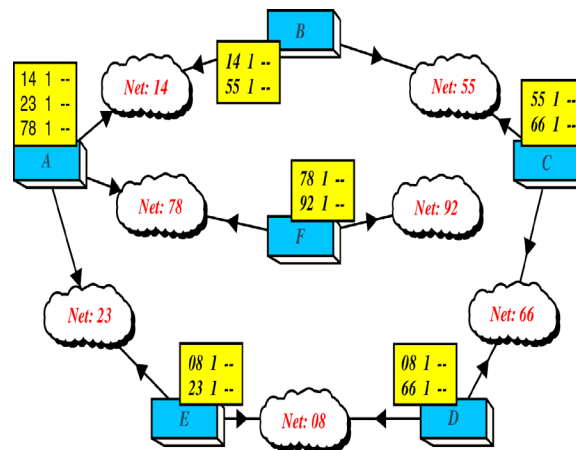


Figure 3.14 Routing table distribution in distance vector routing

### Updating the table

When A receives a routing table from B, it uses the information to update its own table. It says to itself that, B has sent a table that shows how it is packets can get to network 55 and 14.

A knows that B is its neighbor, so packets from A can reach B in one hop. If A adds one hop to all of the costs shown in B's table, the sum will be A's cost for reaching the networks 55 and 14.

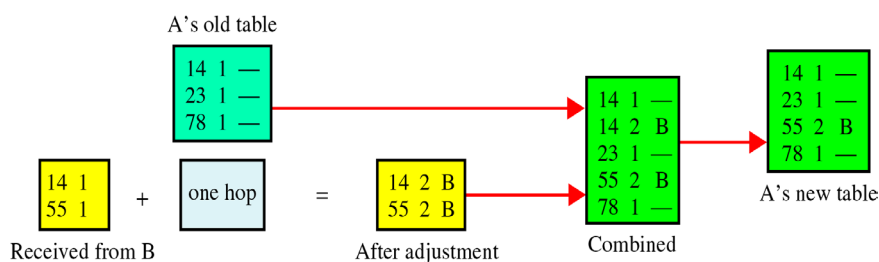


Figure 3.15 Updating Routing Table for Router A

Now A combines this new table with its own table to create a new comprehensive table for A. This combined table may contain duplicate data for some destination networks. Now the router A finds and purges the duplicate entries based on the cost. It always keeps the entries with lowest cost.

This process continues for all routers. Every router receives information from neighbors and updates its routing table.

### Updating Algorithm

The router first adds one hop to the hop count field for each advertised route. The router then applies the following rules to each advertised route.

- (a) If the advertised domain is not in the routing table, the router should add the advertised information to the table.
- (b) If the advertised destination is in the routing table,
  - (i) If the next hop field is the same, the router should replace the entry in the table with the advertised one. Note that even if the advertised hop count is larger, the advertised entry should replace the entry in the table because the new information invalidates the old.
  - (ii) If the next hop field is not the same,
    - If the advertised hop count is smaller, the router should replace the entry in the table with the new one.
    - If the advertised hop count is not smaller, the router should do nothing.

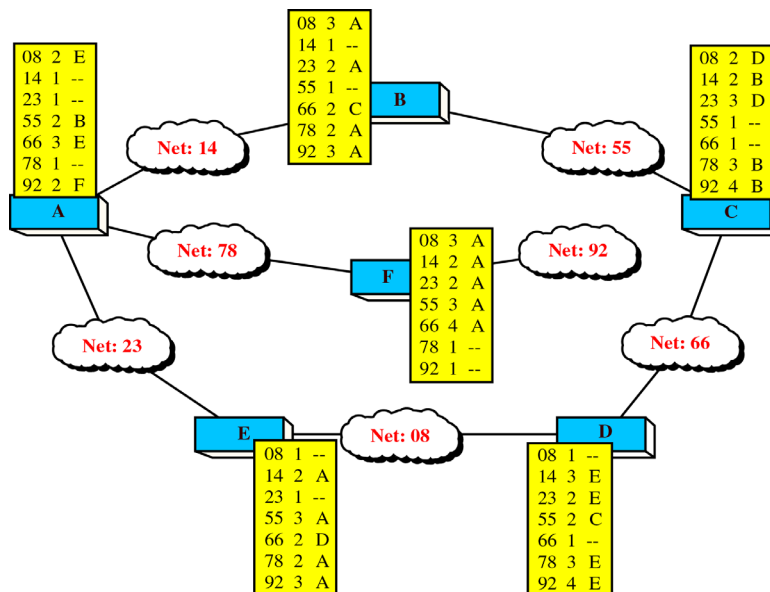
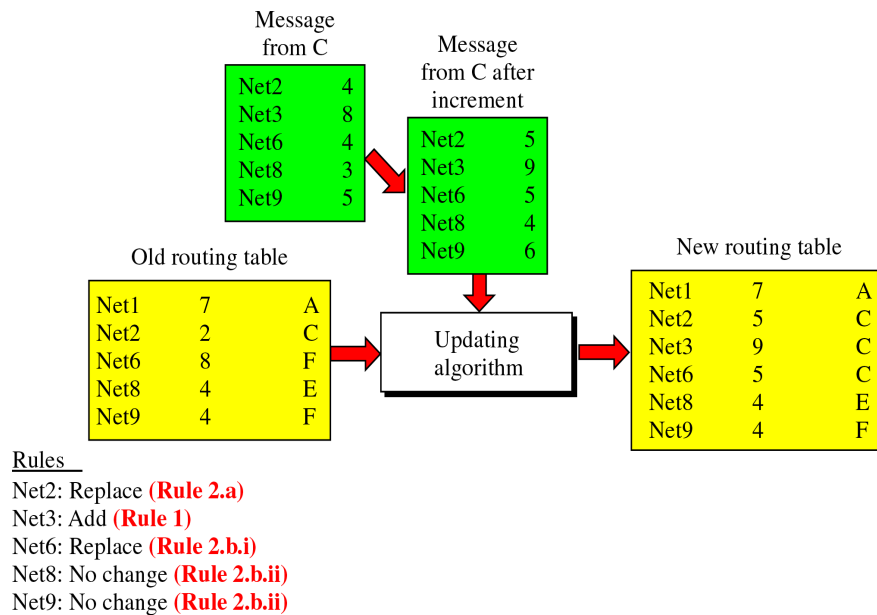


Figure 3.16 Final Routing Tables

#### 3.3.1.3 Example

Figure 3.17 shows an example of updating the routing table based on the updating algorithm.



Note that there is no news about Net1 in the advertised message, so none of the rules apply to this entry.

**Figure 3.17 Example of routing table updating**

### 3.4. OPEN SHORTEST PATH FIRST PROTOCOL (OSPF)

The Open Shortest Path First or OSPF protocol is an inter domain routing protocol based on link state routing. Its domain is also an autonomous system. OSPF uses link state routing to update the routing table in an area. OSPF divides an AS into different areas (depending on their type).

Unlike RIP, OSPF treats the entire network differently based on the types, cost (metric) and condition of each link (To define the state of a link). OSPF allows the administrator to assign a cost for passing through a network based on the type of service required (Minimum delay, maximum throughput).

Each router should have the exact topology of the AS network at every moment. The topology is a graph consisting of nodes and edges. Each router needs to advertise to the neighbourhood of every other router involved in an Area. This process is called flooding.

#### Areas

- To handle routing efficiently, OSPF divides an autonomous system into areas.
- An area is a collection of networks, hosts, and routers all contained within an autonomous system.
- An autonomous system can be divided into many different areas. All networks inside an area must be connected.
- Routers inside an area flood the area with routing information.
- At the border of an area, special routers called area border routers summarize the information about the area and send it to other areas.

## Backbone

- Backbone is a special area which is located inside an autonomous system.
- The backbone serves as a primary area and the other areas as secondary areas.
- The connectivity between a backbone and an area is broken, a virtual link between routers must be created by an administrator to allow continuity of the functions of the backbone as the primary area.
- Each area has an area identification.
- The area identification of the backbone is zero.

## Types of Links

In OSPF, a connection is called a link. Four types of links have been defined for an OSPF protocol. They are as follows;

- (i) **Point-to-point link:** It connects two routers without any other host or router in between.
- (ii) **Transient link:** It is a network with several routers attached to it. The data can enter through any of the routers and leave through any router.
- (iii) **Stub link:** It is a network that is connected to only one router. The data packets enter the network through this single router and leave the network through this same router.
- (iv) **Virtual link:** When the link between two routers is broken, the administration may create a virtual link between them, using a longer path that probably goes through several routers.

### 3.4.1 Link State Routing (LSR)

In link state routing, each routers shares its knowledge of its neighbors with every other router in the network. However, in LSR, the link-state packet (LSP) defines the best known network topology (of an area) is sent to every routers (of other area) after it is constructed locally.

Whereas RIP slowly converges to final routing list based information received from immediate neighbours. In LSR the node can use Dijkstra's algorithm to build a routing table.

From the received LSPs and knowledge of entire topology, a router can then calculate the shortest path between itself and each network.

#### *Three keys to understand how this algorithm works*

- (1) **Sharing knowledge about the neighbourhood:** Each router sends the state of its neighbourhood to every other router in the area.
- (2) **Sharing with every other router:** Through process of flooding each router sends the state of its neighbourhood thru all its output ports and each neighbour sends to every other neighbour and so on until all routers received same full information eventually.
- (3) **Sharing when there is a change.** Each router shares its state of its neighbour only when there is a change; contrasting DVR results in lower traffic.



### 3.4.1.1 Information sharing

The first step in link state routing is information sharing. Each router sends its knowledge about its neighborhood to every other router in the internetwork.

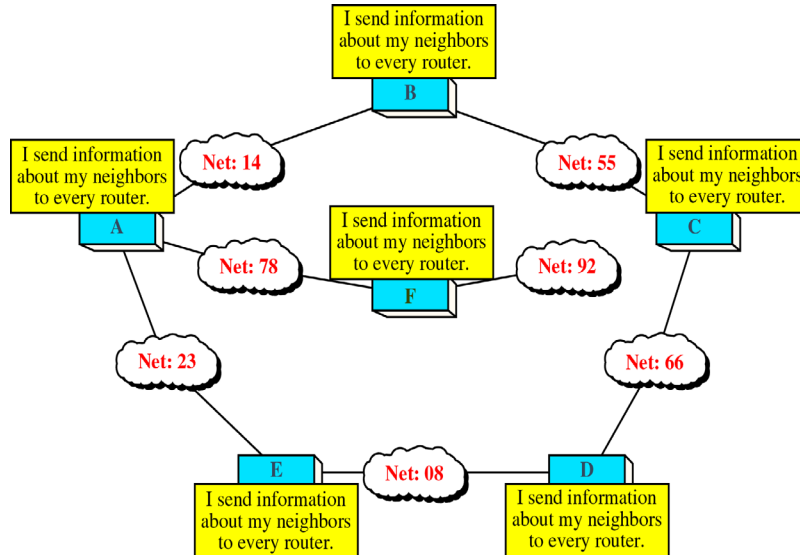


Figure 3.18 Concept of LSR

### 3.4.1.2 Metrics

The OSPF protocol allows the administrator to assign a cost, called the metric, to each route. The metric can be based on a type of service (minimum delay, maximum throughput, and so on).

#### Packet cost

- LSR is the lowest-cost algorithm.
- Cost is a weighted value based on a variety of factors (security level, traffic).
- Cost is applied only by the routers not by any other stations or networks.
- Cost is applied as a packet leaves the router rather than as it enters.

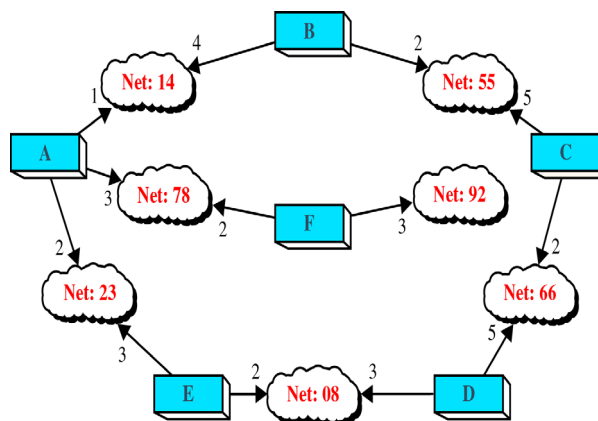


Figure 3.19 Cost in link state routing

**Link State Packet (LSP)**

- When a router floods the network with information about its neighbourhood, it is said to be advertising.
- Advertising can be done with the help of short packet called Link State Packet (LSP).
- Link State Packet (LSP) usually contains the following four fields;
  - (i) The Id of the advertiser
  - (ii) The id of the destination network
  - (iii)The cost
  - (iv)The id of the neighbourhood router

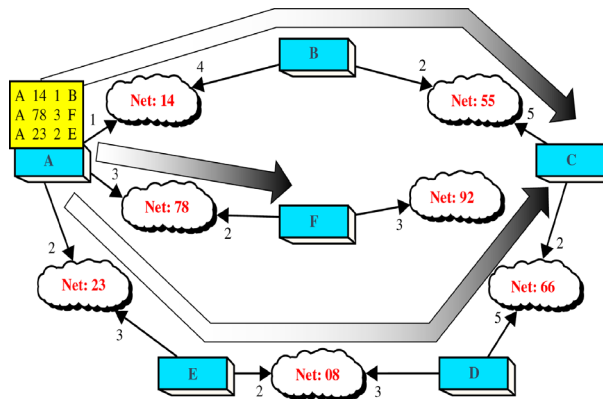
Advertiser	Network	Cost	Neighbor
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____

**Figure 3.20 Link state packet**

**3.4.1.3 Building Routing Tables**

In link state routing, four sets of actions are required to ensure that each node has the routing table showing the least-cost node to every other node. They are;

- (i) Creation of the link state packet (LSP).
- (ii) Distribution of LSPs to every other router, called flooding, in an efficient and reliable way.
- (iii)Formation of a shortest path tree for each node.
- (iv)Calculation of a routing table based on the shortest path tree.



**Figure 3.21 Flooding of A's LSP**

**Link State Database**

- Every router receives LSP and puts the information into a link state database.
- Advertising can be done with the help of short packet called Link State Packet (LSP).

Advertiser	Network	Cost	Neighbor
A	14	1	B
A	78	3	F
A	23	2	E
B	14	4	A
B	55	2	C
C	55	5	B
C	66	2	D
D	66	5	C
D	08	3	E
E	23	3	A
E	08	2	D
F	78	2	A
F	92	3	-

Figure 3.22 Link state database

- Link State Packet (LSP) usually contains four fields:
  - The Id of the advertiser
  - The id of the destination network
  - The cost
  - The id of the neighbourhood router

### 3.4.2 The Dijkstra’s Algorithm

Each route applies the Dijkstra’s Algorithm to calculate its routing table. The Dijkstra’s algorithm calculates the shortest path between two points on a network using a graph.

A graph is made up of nodes and arcs. Nodes are of two types: Networks and Routers. Arcs are the connection between a router and a network. Cost is applied only to the arc from router to network. The cost of the arc from network to router is always zero.

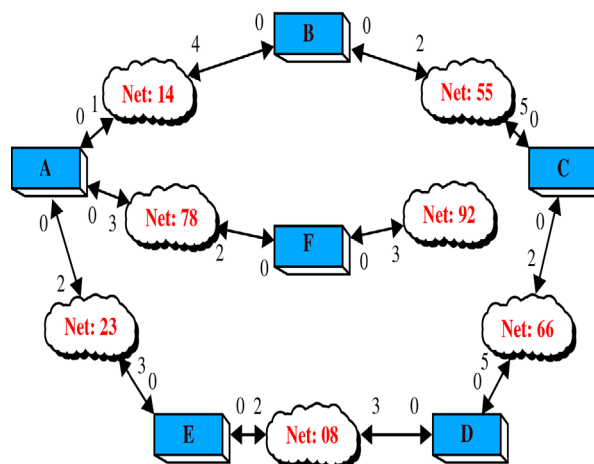


Figure 3.23 Costs in the Dijkstra’s Algorithm

### 3.4.2.1 Shortest path tree

The Dijkstra's algorithm follows four steps to discover the shortest path tree for each router.

- (1) Start with the local node (router): the root of the tree.
- (2) Assign a cost of 0 to this node and make it the first permanent node.

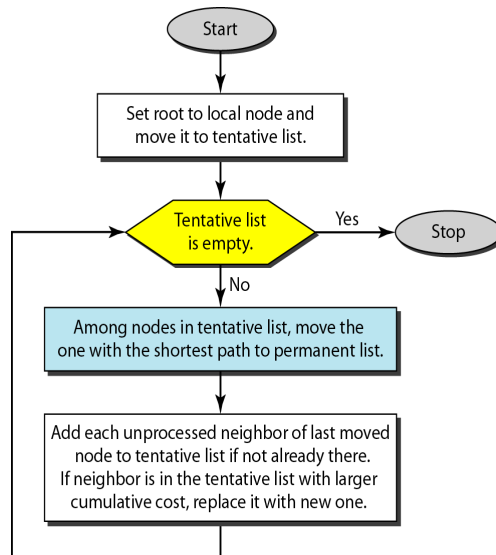
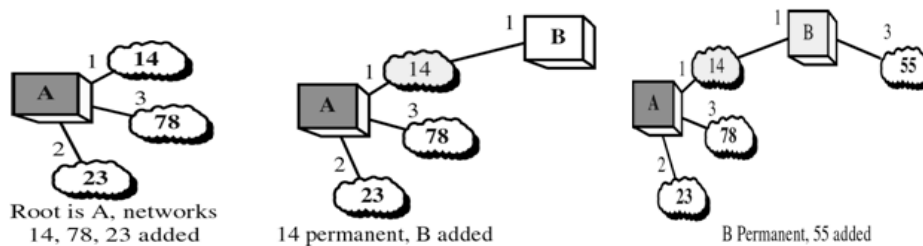


Figure 3.24 Dijkstra's algorithm

- (3) Examine each neighbor node of the node that was the last permanent node.
- (4) Assign a cumulative cost to each node and make it tentative.
- (5) Among the list of tentative nodes,
  - (a) Find the node with the smallest cumulative cost and make it permanent.
  - (b) If a node can be reached from more than one direction
  - (c) Select the direction with the shortest cumulative cost.
- (6) Repeat steps 3 to 5 until every node becomes permanent.

### 3.4.2.2 Shortest path calculation

Figure 3.25 shows the steps of the Dijkstra's algorithm applied by node A of our sample internet.



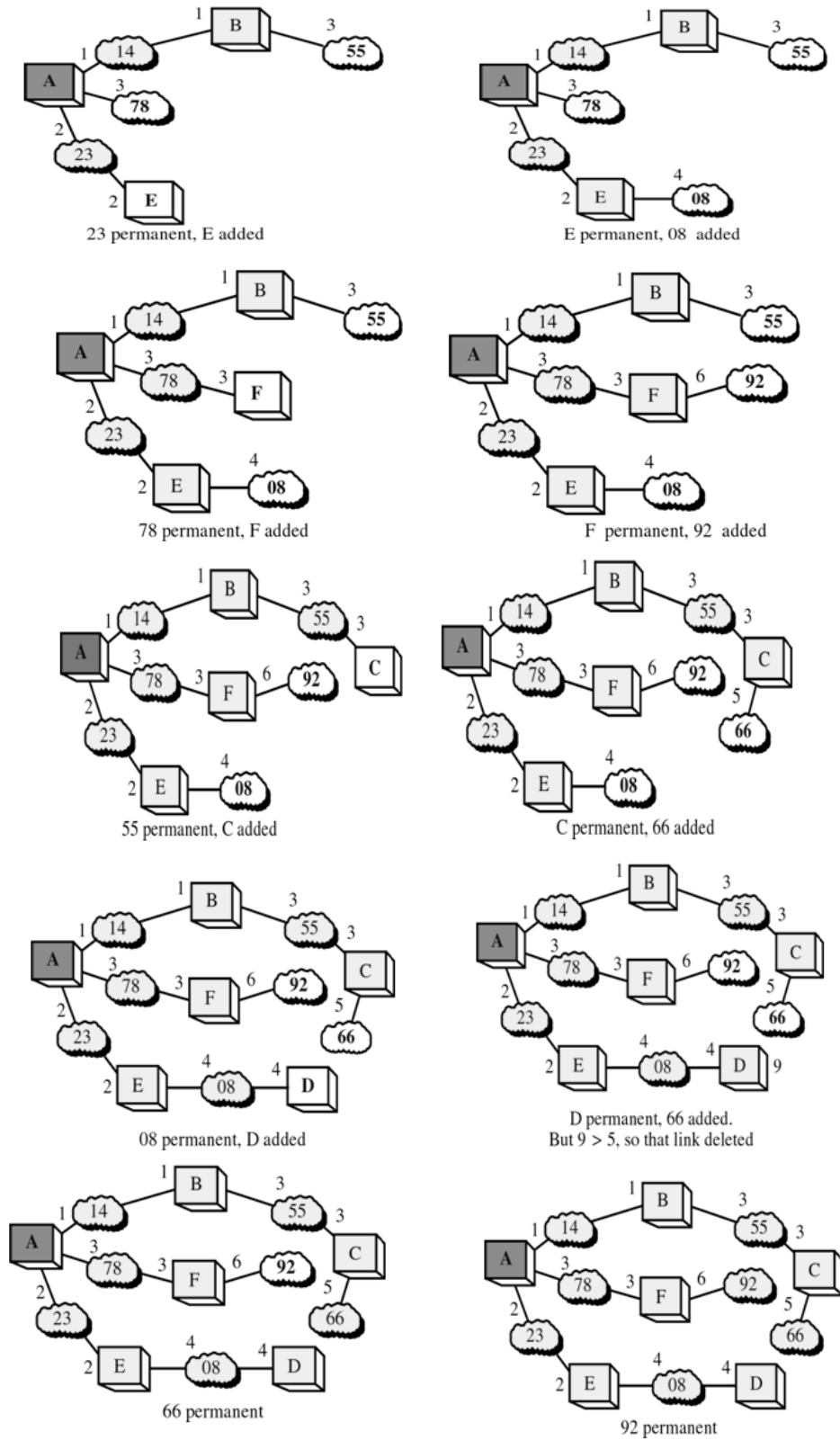


Figure 3.25 Shortest path tree for router A

### 3.4.2.3 Routing table

Each router uses the shortest path tree to construct its routing table. Each router uses the same algorithm and the same link state database to calculate its own shortest path tree and routing table. These are different for each router.

Net	Cost	Next router
08	4	E
14	1	–
23	2	–
55	3	B
66	5	B
78	3	–
92	6	F

Figure 3.26 Routing table for router A

## 3.5 GLOBAL INTERNET

Global Internet is not just a random interconnection of Ethernets, but instead it takes on a shape that reflects the fact that it interconnects many different organizations. Figure 3.27 gives a simple representation of the state of the Internet.

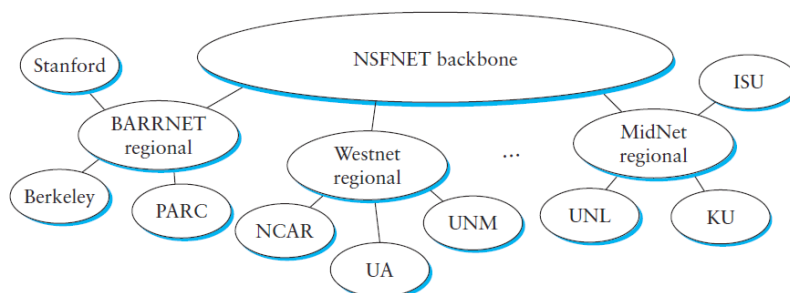


Figure 3.27 The tree structure of the Internet

This topology consists of “end user” sites (e.g., Stanford University) that connect to “service provider” networks (e.g., BARRNET was a provider network that served sites in the San Francisco Bay Area).

In figure 3.27, the regional networks were connected by a nationwide backbone, which was funded by the National Science Foundation (NSF). This backbone is called as the NSFNET backbone.

The provider networks are built from a large number of point-to-point links that connect to routers. Each end user site is not a typical single network, instead consists of multiple physical networks connected by routers and bridges. In global Internetworking, we must answer the following questions.

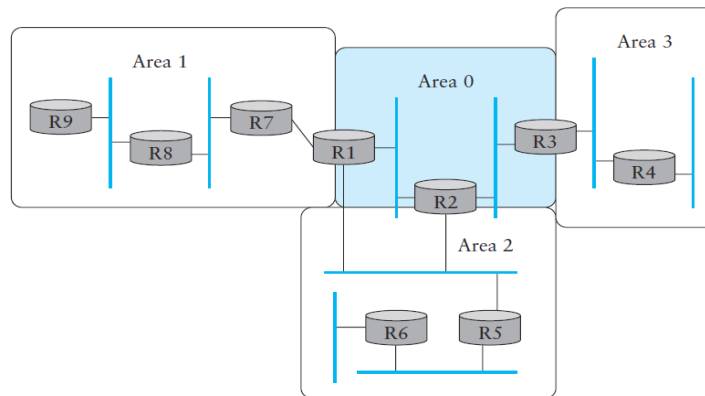
- (i) How to connect a heterogeneous collection of networks to create an internetwork?

- (ii) How to use the simple hierarchy of the IP address, to make a scalable routing in an internet?

### 3.5.1 Routing Areas

An area is a set of routers that are administratively configured to exchange link state information with each other. There is a special area called the backbone area, also known as area 0. Figure 3.28 shows that, how a routing domain is divided into areas.

- The routers R1, R2, and R3 are members of the backbone area. They are also members of at least one non-backbone area;
- R1 is actually a member of both area 1 and area 2.
- A router that is a member of both the backbone area and a non-backbone area is an area border router (ABR).
- The routers that are at the edge of an AS, are referred to as AS border routers.



*Figure 3.28 A domain divided into areas*

#### *Routing within a single area*

- All the routers in the area send link-state advertisements to each other, and develop a complete, consistent map of the area.
- The link-state advertisements of routers do not leave the area in which they are originated.
- This has the effect of making the flooding and route calculation processes considerably more scalable.
- For example, router R4 in area 3 will never see a link-state advertisement from router R8 in area 1.
- The routers will know nothing about the detailed topology of areas other than its own.

#### *Routing among various area*

Now we have to answer the following question how does a router in one area determine the right next hop for a packet destined to a network in another area?

The path of a packet, that has to travel from one non-backbone area to another as being split into three parts.

- (i) First, it travels from its source network to the backbone area.
- (ii) Then it crosses the backbone.
- (iii) Then it travels from backbone to destination network.
  - To make this work, the area border routers summarize routing information that they have learned from one area and make it available in their advertisements to other areas.
  - For example, R1 receives link-state advertisements from all the routers in area 1 and can thus determine the cost of reaching any network in area 1.
  - When R1 sends link-state advertisements into area 0, it advertises the costs of reaching the networks in area 1 much as if all those networks were directly connected to R1. It enables all the area 0 routers to learn the cost to reach all networks in area 1.
  - The area border routers then summarize this information and advertise it into the non-backbone areas. Thus, all routers learn how to reach all networks in the domain.

Note that in area 2, there are two ABRs. So, we have to select any one of the ABR to reach the backbone.

### ***Scalability and optimality***

- When dividing a domain into areas, the network administrator makes a transaction between scalability and optimality of routing.
- The use of areas forces all packets traveling from one area to another to go via the backbone area, even if a shorter path might have been available.

For example, even if R4 and R5 were directly connected, packets would not flow between them because they are in different non-backbone areas. It turns out that the need for scalability is often more important than the need to use the absolute shortest path.

### **3.5.2 Border Gateway Protocol (BGP)**

BGP is an inter-domain routing protocol using path vector routing. It was introduced in 1989 and has gone through four versions.

#### ***Path vector routing (PVR)***

BGP uses path vector routing to update the routing table in an area. DVR and LSR are not suitable candidates for inter-AS routing because of the following reasons.

- (i) ***DVR***: There are occasions in which the route with the smallest hop count is not the preferred route; non-secure path although the shortest route is taken.
- (ii) ***LSR***: internet is too big for this routing method to require each router to have a huge link state database. It takes very long time to calculate the routing table.



PVR defines the exact paths as an ordered list of ASs that a packet should travel thru to reach the destination in its routing table. Security and Political issues involved: more desired to avoid 'unsaved' paths/routes/ASs than to take a shorter route.

The AS boundary router, that participate in PVR advertise the routes of the networks in their own AS to neighbour AS boundary routers. Solve the count-to-infinity problem.

### ***Path Vector Routing Policy***

- Policy routing can be easily implemented through path vector routing.
- When a router receives a message from its neighbour, the speaker node or AS boundary router can check the path with its approved list of ASs.
- If one of the ASs listed in the path is against its policy, the router can ignore that path entirely and that destination.
- For any unapproved paths, the router does not update its routing table with this path, and it does not send the PV message to its neighbours.
- This means that the routing table in path vector routing are not based on the smallest hop count (as in distance vector routing) or the minimum delay metric (as in open shortest path first routing); they are based on the policy imposed on the router by the administrator.
- The path was presented as a list of ASs, but is in fact, a list of attributes. Each attribute gives some information about the path. The list of attributes helps the receiving router make a better decision when applying its policy (Well-known & Optional).

### ***3.5.2.1 Types of Autonomous Systems***

The Internet is divided into hierarchical domains called autonomous systems. The autonomous systems can be divided into three categories: stub, multi homed, and transit.

#### ***(i) Stub AS***

- A stub AS has only one connection to another AS.
- The inter domain data traffic in a stub AS can be either created or terminated in the AS.
- The hosts in the AS can send data to other ASs.
- The hosts in the AS can receive data coming from hosts in other ASs.
- Data traffic cannot pass through a stub AS.
- A stub AS is either a source or a sink.
- Example: A small corporation or a small local ISP.

#### ***(ii) Multihomed AS***

- A multihomed AS has more than one connection to other ASs.
- It can act as a source or sink for data traffic.
- It can receive data traffic from more than one AS.

- It can send data traffic to more than one AS.
- It does not allow data coming from one AS and going to another AS to pass through (No transient traffic).
- Example: A large corporation is connected to more than one regional or national AS.

### **(iii) Transit AS**

- A transit AS is a multihomed AS that also allows transient traffic.
- Example: National and international ISPs (Internet backbones).

### **3.5.2.2 Path Attributes**

The path was presented as a list of autonomous systems (a list of attributes). Each attribute gives some information about the path. The list of attributes helps the receiving router make a more-informed decision when applying its policy. Attributes are divided into two broad categories;

- (1) **Well known:** Recognized by every BGP router.
- (2) **Optional:** Need not be recognized by every BGP router.

#### **Well-known attributes**

It can be recognized by every BGP router. Well-known attributes are themselves divided into two categories;

- (i) **Mandatory:** A well-known mandatory attribute is one that must appear in the description of a route.
- (ii) **Discretionary:** A well-known discretionary attribute is one that must be recognized by each router, but is not required to be included in every update message.

#### **Examples:**

- **ORIGIN:** This defines the source of the routing information (RIP, OSPF, and so on).
- **AS\_PATH:** This defines the list of autonomous systems through which the destination can be reached.
- **NEXT-HOP:** It defines the next router to which the data packet should be sent.

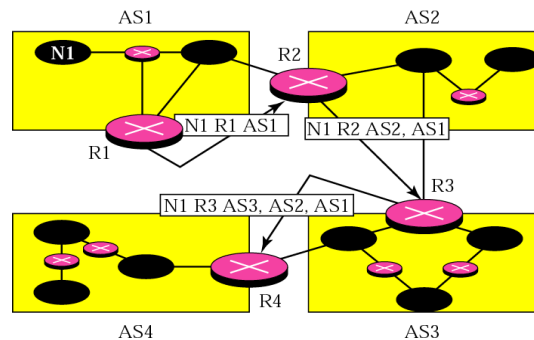
#### **Optional attributes**

It need not be recognized by every BGP router. The optional attributes can also be subdivided into two categories;

- (i) **Transitive:** An optional transitive attribute must be passed to the next router by the router that has not implemented this attribute.
- (ii) **Non-transitive:** An optional non-transitive attribute must be discarded if the receiving router has not implemented it.

### **3.5.2.3 Path Vector Packets**

Each AS has its 'speaker' router/node that acts on behalves of the AS. Only speaker router can communicate with other speaker routers.



**Figure 3.29 Path vector packets**

R1 sends a path vector message advertising its reachability of N1. R2 receives the message, updates its routing table and after adding its AS to the path and inserting itself as next router, send message to R3.

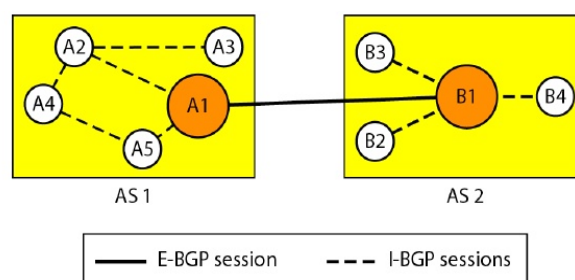
R3 receives the message, updates its routing table, make changes and sends the message to R4.

#### 3.5.2.4 BGP Sessions

A session is a connection that is established between two BGP routers only for the sake of exchanging routing information. To create a reliable environment, BGP uses the services of TCP.

When a TCP connection is created for BGP, it can last for a long time, until something unusual happens. Because of this, the BGP sessions are sometimes referred to as semi-permanent connections. BGP can have two types of sessions;

- (1) **External BGP (E-BGP) sessions:** The E-BGP session is used to exchange information between two speaker nodes belonging to two different autonomous systems.
- (2) **Internal BGP (I-BGP) sessions:** The I-BGP session is used to exchange routing information between two routers inside an autonomous system.



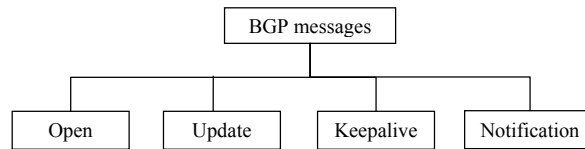
**Figure 3.30 Internal and external BGP sessions**

In figure 3.30,

- The session established between AS1 and AS2 is an E-BGP session.
- The two speaker routers exchange information they know about networks in the Internet.
- These two routers need to collect information from other routers in the autonomous systems.
- This is done using I-BGP sessions.

### 3.5.3 Types of BGP messages

BGP uses the following four types of messages. Figure 3.31 shows the taxonomy of BGP messages.



*Figure 3.31 BGP messages*

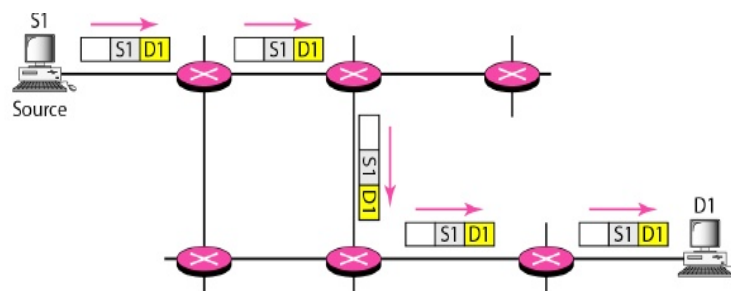
- (a) **Open:** To create a relationship, a router running BGP opens a connection with a neighbouring AS and sends an open message.
- (b) **Keep-alive:** If the neighbour accepted, it responds with a Keep-alive message to establish relationship between the two routers.
- (c) **Update:** It is the main process of BGP protocol. It is used by a router to withdraw destination that have been advertised previously and announce a new route to a new destination or do both. (Withdraw several but advertise only one).
- (d) **Notification:** sent by a router whenever an error condition is detected or router wants to close the connection (down).

### 3.6 MULTICASTING - ADDRESSES

A message can be transmitted in any of the following ways.

#### (i) Unicasting

- In this communication, there is one source and one destination.
- The relationship between the source and the destination is one-to-one.
- The IP datagram contains the unicast addresses assigned to the hosts (both the source and destination addresses).

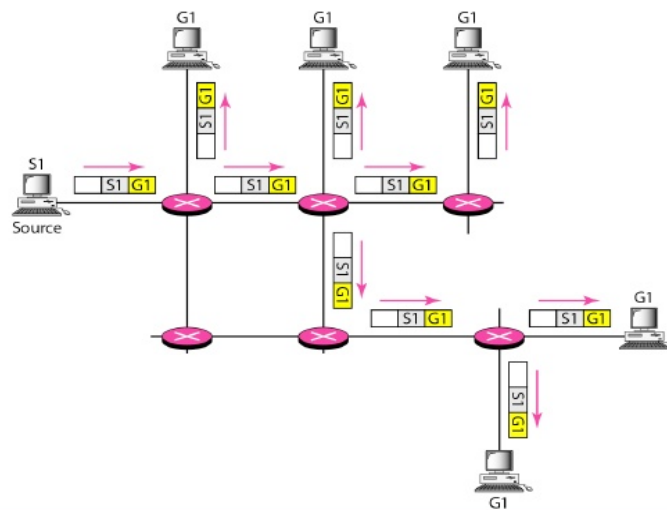


*Figure 3.31 Unicasting*

- In figure 3.31, a unicast packet starts from the source S1 and passes through routers to reach the destination D1.
- In unicasting, when a router receives a packet, it forwards the packet through only one of its interfaces as defined in the routing table.
- The router may discard the packet if it cannot find the destination address in its routing table.

**(ii) Multicasting**

- In multicast communication, there is one source and a group of destinations.
- The relationship is one-to-many.
- The source address is a unicast address, and the destination address is a group address, which defines one or more destinations.
- The group address identifies the members of the group.
- A multicast packet starts from the source S1 and goes to all destinations that belong to group G1.
- When a router receives a packet, it may forward it through several of its interfaces.

**Figure 3.32 Multicasting****(iii) Broadcasting**

- The relationship between the source and the destination is one-to-all.
- There is only one source, all the other hosts are the destinations.
- It produces the huge amount of traffic and requires huge amount of bandwidth.

**3.6.1 Multicasting Versus Multiple Unicasting**

Figure 3.33 and below table shows the differences between multicasting and multiple unicasting.

MULTICASTING	MULTIPLE UNICASTING
One single packet from the source	Several packets start from the source
The packet is duplicated by the Routers	The packet is not duplicated by the routers
The destination address in each packet is the same for all duplicates	The destination address in each packet is different for all packets

**Figure 3.33 Multicasting versus multiple unicasting**

## Emulation of Multicasting with Unicasting

Sending document for translation over the Internet in unencrypted HTML format is called emulation.

The mechanisms used for multicasting differs, when it can be emulated with unicasting. There are two obvious reasons for this.

- (1) Multicasting is more efficient than multiple unicasting because of less bandwidth requirements.
- (2) In multiple unicasting, the packets are created by the source with a relative delay between packets. In multicasting, there is no delay because only one packet is created by the source.

### 3.6.2 Applications

Multicasting has many applications today such as access to distributed databases, information dissemination, teleconferencing, and distance learning.

#### (1) *Access to Distributed Databases*

- Most of the large databases today are distributed.
- That is, the information is stored in more than one location, usually at the time of production.
- The user who needs to access the database does not know the location of the information.
- A user's request is multicast to all the database locations, and the location that has the information responds.

#### (2) *Information Dissemination*

- Businesses often need to send information to their customers.
- If the nature of the information is the same for each customer, it can be multicast.
- That is, a business can send one message that can reach many customers.
- Example: A software update can be sent to all purchasers of a particular software package.

#### (3) *Dissemination of News*

- News can be easily disseminated through multicasting.
- One single message can be sent to those interested in a particular topic.

#### (4) *Teleconferencing*

- The individuals attending a teleconference, all need to receive the same information at the same time.
- Temporary or permanent groups can be formed for this purpose.

#### (5) *Distance Learning*

- One growing area in the use of multicasting is distance learning.
- Lessons taught by one single professor can be received by a specific group of students.

### 3.7. MULTICAST ROUTING

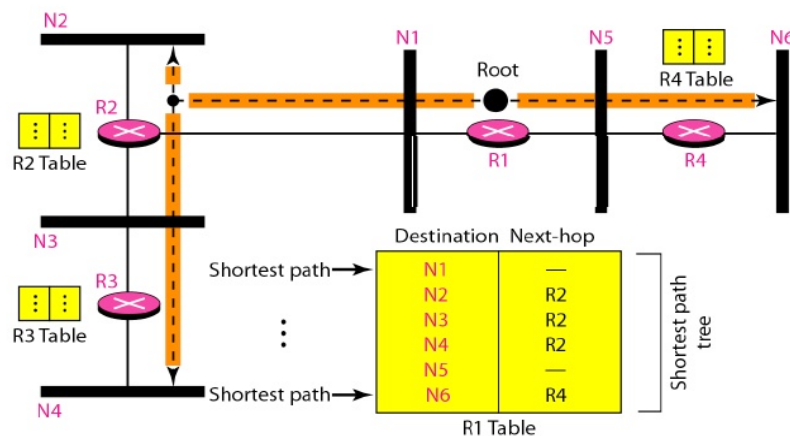
The idea of optimal routing is used commonly in all multicast protocols. The optimal inter domain routing is used to find the shortest path tree.

#### *Optimal Routing: Shortest Path Trees*

- The root of the tree is the source, and the leaves are the potential destinations.
- The path from the root to each destination is the shortest path.
- The number of trees and the formation of the trees in unicast and multicast routing are different.

#### 3.7.1 Unicast Routing

- In unicast routing, when a router receives a packet to forward, it needs to find the shortest path to the destination of the packet.
- The router consults its routing table for that particular destination.
- The next-hop entry corresponding to the destination is the start of the shortest path.
- The router knows the shortest path for each destination, which means that the router has a shortest path tree to optimally reach all destinations.
- In unicast routing, each line of the routing table is a shortest path and the whole routing table is a shortest path tree.
- In unicast routing, each router needs only one shortest path tree to forward a packet and each router has its own shortest path tree.



**Figure 3.34 Shortest path tree in unicast routing**

The figure 3.34 shows the details of the routing table and the shortest path tree for router R1. Each line in the routing table corresponds to one path from the root to the corresponding network. The whole table represents the shortest path tree. In unicast routing, each router in the domain has a table that defines a shortest path tree to possible destinations.

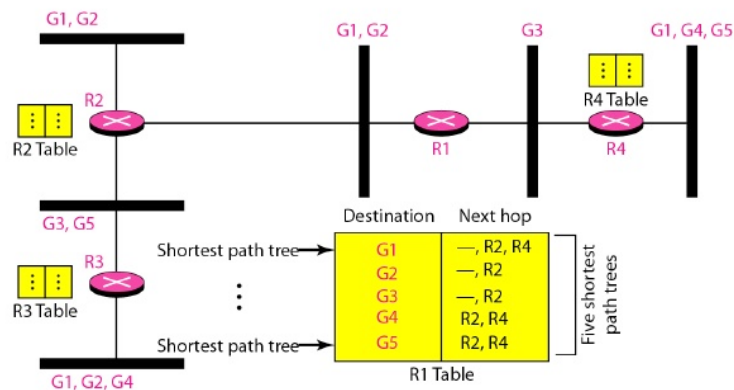
#### 3.7.2 Multicast Routing

- A multicast packet may have destinations in more than one network.

- Forwarding of a single packet to members of a group requires a shortest path tree.
- In multicast routing, each involved router needs to construct a shortest path tree for each group.
- If we have n groups, we may need n shortest path trees.
- Two approaches have been used to solve the problem: Source-based trees and Group-shared trees.

**(1) Source-Based Tree**

In this approach, each router needs to have one shortest path tree for each group. The shortest path tree for a group defines the next hop for each network that has loyal member(s) for that group.



**Figure 3.35 Source-based tree approach**

In figure 3.35, we assume that we have only five groups in the domain: G1, G2, G3, G4, and G5. At the moment G1 has loyal members in four networks, G2 in three, G3 in two, G4 in two, and G5 in two. We have shown the names of the groups with loyal members on each network.

The figure 3.35 also shows the multicast routing table for router R1. There is one shortest path tree for each group. Therefore there are five shortest path trees for five groups.

If router R1 receives a packet with destination address G1, it needs to send a copy of the packet to the attached network. That means a copy to router R2 and a copy to router R4 so that all members of G1 can receive a copy.

**(2) Group-Shared Tree**

To avoid the complexity in source-based tree routing table management, the group-based tree approach was developed.

- In the group-shared tree approach, one designated router called the center core or rendezvous router is used.
- It takes the responsibility of distributing multicast traffic.
- The core has m shortest path trees in its routing table.
- The rest of the routers in the domain have none.



- If a router receives a multicast packet, it encapsulates the packet in a unicast packet and sends it to the core router.
- The core router removes the multicast packet from its capsule, and consults its routing table to route the packet.

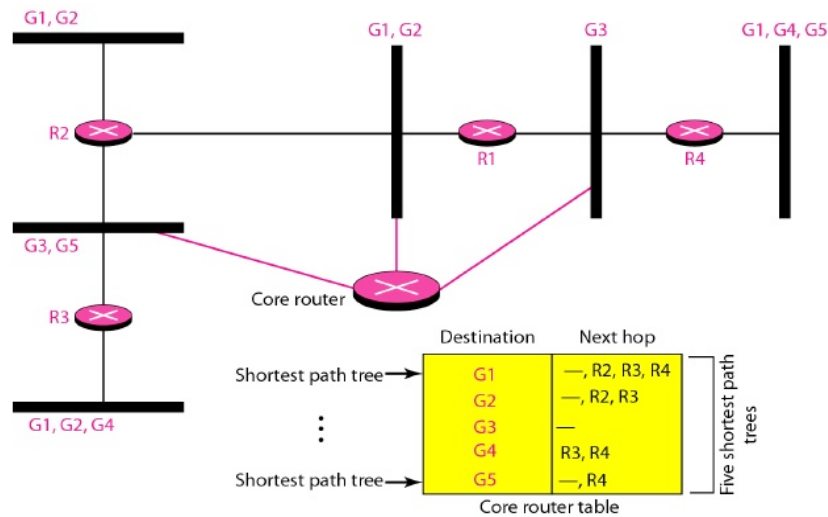


Figure 3.36 Group-shared tree approach

### 3.8. ROUTING PROTOCOLS

Several multicast routing protocols have emerged. Some of these protocols are extensions of unicast routing protocols. The figure 3.37 shows the taxonomy of these protocols.

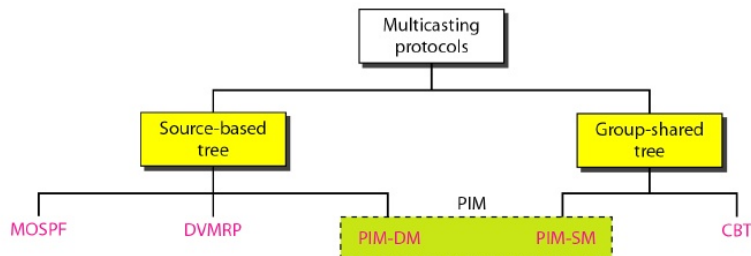


Figure 3.37 Taxonomy of common multicast protocols

#### 3.8.1 Multicast Link State Routing

In unicast routing each router creates a shortest path tree by using Dijkstra's algorithm. The routing table is a translation of the shortest path tree.

Multicast link state routing is a direct extension of unicast routing and it uses a source-based tree approach. In multicast routing, a node needs to revise the interpretation of state (what groups are active on this link).

A node advertises every group which has any loyal member on the link. The information about the group comes from IGMP. Each router is running IGMP to find out the membership status.

When a router receives all the LSPs, it creates  $n$  ( $n$  is the number of groups) topologies, from which  $n$  shortest path trees are made by using Dijkstra's algorithm. So each router has a routing table that represents as many shortest path trees as there are groups.

### ***Disadvantage***

The only problem with this protocol is the time and space needed to create and save the many shortest path trees.

The solution is to create the trees only when needed. When a router receives a packet with a multicast destination address, it runs the Dijkstra's algorithm to calculate the shortest path tree for that group.

The result can be cached in case there are additional packets for that destination.

#### ***3.8.1.1 Multicast Open Shortest Path First (MOSPF)***

- It is an extension of the OSPF protocol.
- It uses multicast link state routing to create source-based trees.
- The protocol requires a new link state update packet to associate the unicast address of a host with the group address.
- This packet is called the group-membership LSA.
- The routing table includes the entries only for the hosts that belong to a particular group.
- To improve the efficiency, the router calculates the shortest path trees on demand (when it receives the first multicast packet).
- The tree can be saved in cache memory for future use by the same source/group pair.
- MOSPF is a data-driven protocol.

#### **3.8.2 Multicast Distance Vector Routing (DVMRP)**

Multicast distance vector routing uses source-based trees, but the router never actually makes a routing table. When a router receives a multicast packet, it forwards the packet as though it is consulting a routing table.

We can say that the shortest path tree is temporary. After its use (after a packet is forwarded) the table is destroyed.

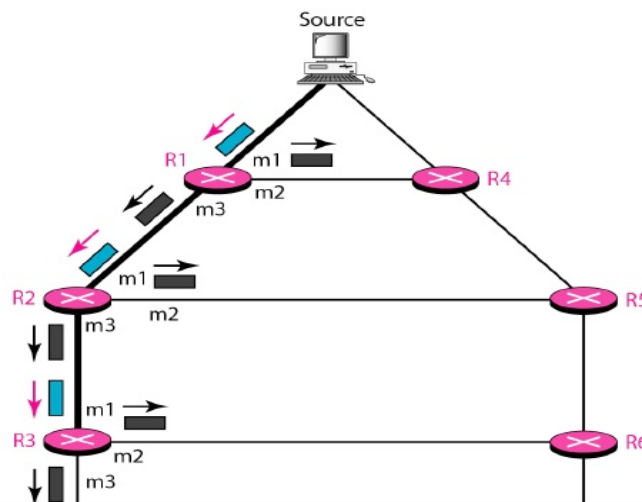
To accomplish this, the multicast distance vector algorithm uses a process based on four decision-making strategies.

##### ***3.8.2.1 Flooding***

- When a router receives a packet, without looking at the destination group address, sends it out from every link except the one from which it was received.
- Flooding accomplishes the first goal of multicasting: every network with active members receives the packet.
- Flooding broadcasts packets, but creates loops in the systems.

### 3.8.2.2 Reverse Path Forwarding (RPF)

- RPF is a modified flooding strategy.
- To prevent loops, only one copy is forwarded, the other copies are dropped.
- In RPF, a router forwards only one copy, that has traveled the shortest path from the source to the router.
- To find this copy, RPF uses the unicast routing table.
- The router receives a packet and extracts the source address (a unicast address).
- It consults its unicast routing table as though it wants to send a packet to the source address.
- The routing table tells the router the next hop.
- If the multicast packet has just come from the hop defined in the table, the packet has traveled the shortest path from the source to the router because the shortest path is reciprocal in unicast distance vector routing protocols.
- If the path from A to B is the shortest, then it is also the shortest from B to A.
- The router forwards the packet if it has traveled from the shortest path; it discards it otherwise.
- This strategy prevents loops because there is always one shortest path from the source to the router.



**Figure 3.38 Reverse path forwarding (RPF)**

Figure 3.38 shows the part of a domain and a source for reverse path forwarding. The shortest path tree as calculated by routers R1, R2, and R3 is shown by a thick line.

When R1 receives a packet from the source through the interface m1, it consults its routing table and finds that the shortest path from R1 to the source is through interface m1. The packet is forwarded.

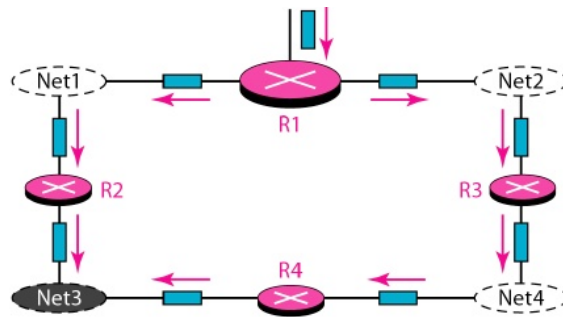
But, if a copy of the packet has arrived through interface m2, it is discarded. That means, m2 does not define the shortest path from R1 to the source.

**Problem with RPF**

- RPF does not guarantee that each network receives only one copy; a network may receive two or more copies.
- The reason is that RPF is not based on the destination address (a group address); forwarding is based on the source address.
- Figure 3.39 is illustrating this problem.

In figure 3.39, each router sends out one copy from each interface. But Net3 in figure 3.39 receives two copies of the packet. There is duplication because a tree has not been made. Instead of a tree, we have a graph.

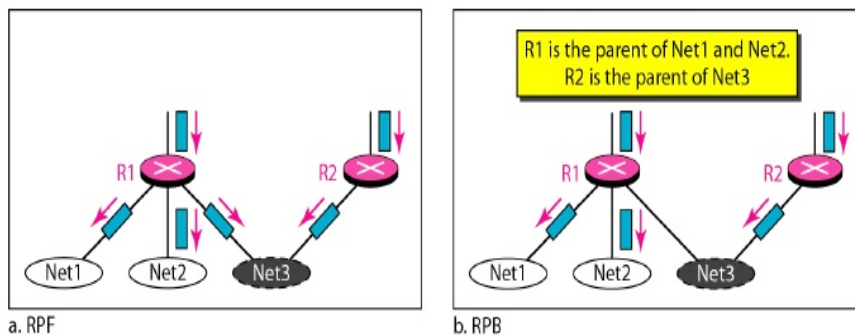
Net3 has two parents: routers R2 and R4. To eliminate duplication, we must define only one parent router for each network. That means, network can receive a multicast packet from a particular source only through a designated parent router.



**Figure 3.39 Problem with RPF**

**3.8.2.3 Reverse Path Broadcasting (RPB)**

- RPF guarantees that each network receives a copy of the multicast packet without formation of loops.
- For each source, the router sends the packet only out of those interfaces for which it is the designated parent. This policy is called reverse path broadcasting (RPB).
- RPB guarantees that the packet reaches every network and that every network receives only one copy.
- The figure 3.40 shows the difference between RPF and RPB.



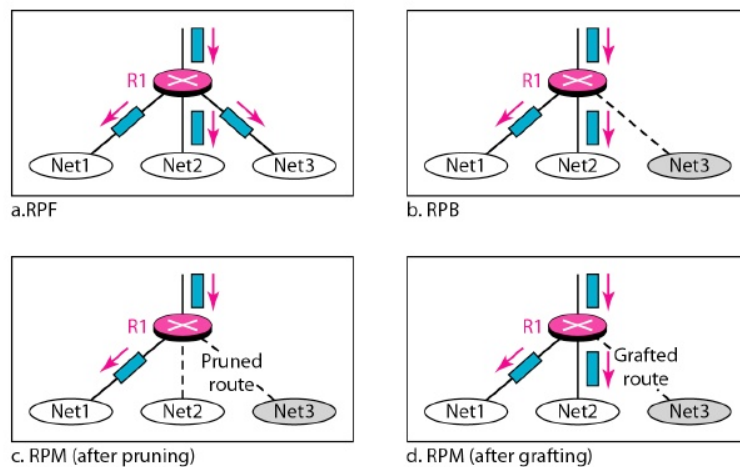
**Figure 3.40 RPF versus RPB**

**Problem with RPF**

- RPB does not multicast the packet, it broadcasts it.
- This is not efficient. To increase efficiency, the multicast packet must reach only those networks that have active members for that particular group.

**3.8.2.4 Reverse Path Multicasting (RPM)**

The reverse path multicasting (RPM) is used to convert broadcasting to multicasting. The protocol uses two procedures, pruning and grafting to achieve the multicasting. The figure 3.41 shows the idea of pruning and grafting.



**Figure 3.41 RPF, RPB, and RPM**

**Pruning**

- The designated parent router of each network is responsible for holding the membership information.
- This is done through the IGMP protocol.
- The process starts when a router connected to a network finds that there is no interest in a multicast packet.
- The router sends a prune message to the upstream router so that it can exclude the corresponding interface.
- That is, the upstream router can stop sending multicast messages for this group through that interface.
- Now if this router receives prune messages from all downstream routers, it, in turn, sends a prune message to its upstream router.

**Grafting**

What if a leaf router (a router at the bottom of the tree) has sent a prune message but suddenly realizes, through IGMP, that one of its networks is again interested in receiving the multicast packet?

It can send a graft message. The graft message forces the upstream router to resume sending the multicast messages.

RPM adds pruning and grafting to RPB to create a multicast shortest path tree that supports dynamic membership changes.

### ***DVMRP***

The Distance Vector Multicast Routing Protocol (DVMRP) is an implementation of multicast distance vector routing. It is a source-based routing protocol, based on RIP.

### **3.8.3 The Core-Based Tree (CBT)**

- This is a group-shared protocol that uses a core as the root of the tree.
- The autonomous system is divided into regions, and a core (center router or rendezvous router) is chosen for each region.

#### ***Formation of the Tree***

After the rendezvous point is selected, every router is informed of the unicast address of the selected router. Each router then sends a unicast join message (similar to a grafting message) to show that it wants to join the group.

This message passes through all routers that are located between the sender and the rendezvous router. Each intermediate router extracts the necessary information from the message. They are,

- (i) The unicast address of the sender.
- (ii) The interface through which the packet has arrived.

After extracting the information, it forwards the message to the next router in the path. When the rendezvous router has received all join messages from every member of the group, the tree is formed.

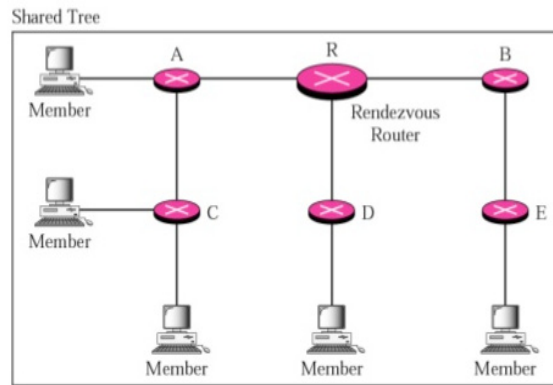
Now every router knows its upstream router (the router that leads to the root) and the downstream router (the router that leads to the leaf).

If a router wants to leave the group, it sends a leave message to its upstream router. The upstream router removes the link to that router from the tree and forwards the message to its upstream router, and so on. Figure 3.42 shows a group-shared tree with its rendezvous router.

The Core-Based Tree (CBT) is a group-shared tree, center-based protocol. One of the routers in the tree is called the core. A packet is sent from the source to members of the group following this procedure.

- (i) The source, which may or may not be part of the tree, encapsulates the multicast packet inside a unicast packet with the unicast destination address of the core and sends it to the core. This part of delivery is done using a unicast address; the only recipient is the core router.

- (ii) The core decapsulates the unicast packet and forwards it to all interested interfaces.
- (iii) Each router that receives the multicast packet, in turn, forwards it to all interested interfaces.



*Figure 3.42 Group-shared tree with rendezvous router*

### 3.8.4 Protocol independent multicast (PIM)

PIM is the name given to two independent multicast routing protocols: Protocol Independent Multicast-Dense Mode (PIM-DM) and Protocol Independent Multicast-Sparse Mode (PIM-SM). Both protocols are unicast protocol-dependent.

#### *PIM-DM*

- PIM-DM is used in a dense multicast environment, such as a LAN.
- PIM-DM is used when there is a possibility that each router is involved in multicasting.
- PIM-DM is a source-based tree routing protocol that uses RPF, pruning and grafting strategies for multicasting.
- It's operates like DVMRP, but it does not depend on a specific unicasting protocol.
- It assumes that the autonomous system is using a unicast protocol and each router has a table that can find the outgoing interface that has an optimal path to a destination.
- This unicast protocol can be a distance vector protocol (RIP) or link state protocol (OSPF).

#### *PIM-SM*

- PIM-SM is used in a sparse multicast environment such as a WAN.
- PIM-SM is used when there is a possibility that each router is involved in multicasting.
- PIM-SM is a group-shared tree routing protocol that has a rendezvous point (RP) as the source of the tree.
- It operates CBT, but it does not require acknowledgment from a join message.
- In addition, it creates a backup set of RPs for each region to cover RP failures.

### 3.8.5 MBONE

Multimedia and real-time communications have increased the need for multicasting in the Internet. Only a small fraction of multicast routers are available in the Internet.

So, a multicast router may not find another multicast router in the neighborhood to forward the multicast packet. This problem can be solved in the following ways;

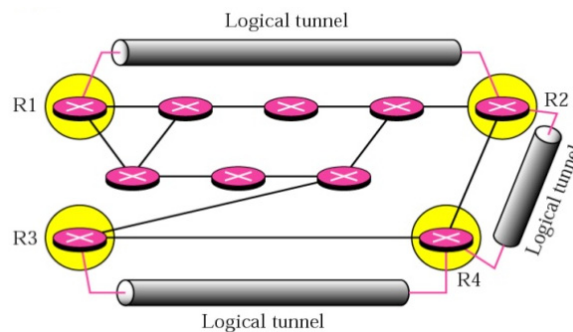
- (1) Adding more and more multicast routers.
- (2) Tunneling.

#### *Tunneling*

Figure 3.43 illustrates the concept of tunneling. The multicast routers are seen as a group of routers on top of unicast routers. The multicast routers may not be connected directly, but they are connected logically.

The routers enclosed in the shaded circles only are capable of multicasting, because these routers are isolated by themselves.

Tunneling is used to implement the logical connection (Direct connection) between all the routers by building a multicast backbone (MBONE).



**Figure 3.43 Logical tunneling**

### 3.9. IPV6

The network layer protocol in the TCP/IP protocol suite is currently IPv4. IPv4 provides the host-to-host communication between systems in the Internet. IPv4 has some deficiencies that make it unsuitable for the fast-growing Internet.

- (i) Address depletion is still a long-term problem in the Internet.
- (ii) The Internet must accommodate real-time audio and video transmission. This type of transmission requires minimum delay strategies and reservation of resources not provided in the IPv4 design.
- (iii) The Internet must accommodate encryption and authentication of data for some applications. No encryption or authentication is provided by IPv4.

To overcome these deficiencies, IPv6 (Internetworking Protocol, version 6) was proposed and is now a standard. IPv6 is also known as IPng (Internetworking Protocol, next generation).



### 3.9.1 Advantages of IPv6

IPv6 has some advantages over IPv4 that can be summarized as follows:

- (1) **Larger address space:** An IPv6 address is 128 bits long. Compared with the 32-bit address of IPv4, this is a huge (296) increase in the address space.
- (2) **Better header format:** IPv6 uses a new header format in which options are separated from the base header and inserted, when needed, between the base header and the upper-layer data. This simplifies and speeds up the routing process.
- (3) **New options:** IPv6 has new options to allow for additional functionalities.
- (4) **Allowance for extension:** IPv6 is designed to allow the extension of the protocol if required by new technologies or applications.
- (5) **Support for resource allocation:** In IPv6, the type-of-service field has been removed, but flow label has been added to enable the source to request special handling of the packet.
- (6) **Support for more security:** The encryption and authentication options in IPv6 provide confidentiality and integrity of the packet.

### 3.9.2 Packet Format

The IPv6 packet is composed of a mandatory base header followed by the payload. The payload consists of two parts;

- (i) Optional extension headers
- (ii) Data from an upper layer

The base header occupies 40 bytes, whereas the extension headers and data from the upper layer contain up to 65,535 bytes of information. Figure 3.44 shows the IPv6 packet format.

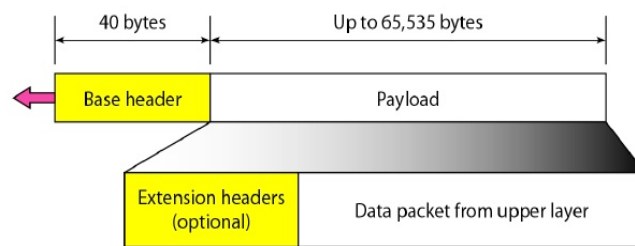


Figure 3.44 IPv6 datagram header and payload

#### Base Header

Figure 3.45 shows the base header with its eight fields. These fields are as follows;

- (1) **Version:** A 4-bit field defines the version number of the IP. For IPv6, the value is 6.
- (2) **Priority:** The 4-bit priority field defines the priority of the packet with respect to traffic congestion.

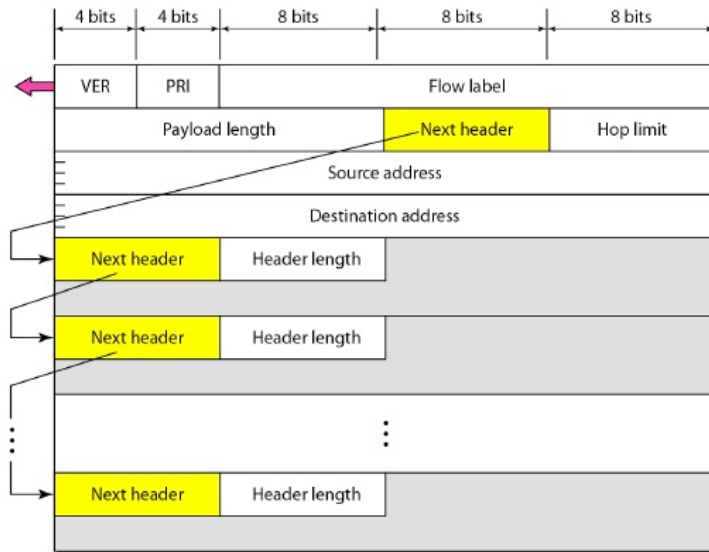


Figure 3.45 Format of an IPv6 datagram

- (3) **Flow label:** The flow label is a 3-byte (24-bit) field that is designed to provide special handling for a particular flow of data.
- (4) **Payload length:** The 2-byte payload length field defines the length of the IP datagram excluding the base header.
- (5) **Next header:** The next header is an 8-bit field defining the header that follows the base header in the datagram. The table 3.3 shows the values of next headers.
- (6) **Hop limit:** This 8-bit hop limit field serves the same purpose as the TTL field in IPv4.
- (7) **Source address:** The source address field is a 16-byte (128-bit) Internet address that identifies the original source of the datagram.
- (8) **Destination address:** The destination address field is a 16-byte (128-bit) Internet address that usually identifies the final destination of the datagram.

Code	Next header
0	Hop-by-hop option
2	ICMP
6	TCP
17	UDP
43	Source routing
44	Fragmentation
50	Encrypted security payload
51	Authentication
59	Null (no next header)
60	Destination option

Table 3.3 Next header codes for IPv6

### 3.9.3 Fields of IPv6 Base Header

#### *Priority*

- The priority field of the IPv6 packet defines the priority of each packet with respect to other packets from the same source.
- For example, if one of two consecutive datagrams must be discarded due to congestion, the datagram with the lower **packet priority** will be discarded.
- IPv6 divides traffic into two broad categories;
  - (i) Congestion-controlled
  - (ii) Noncongestion-controlled

#### *(i) Congestion-Controlled Traffic*

- If a source adapts itself to traffic slowdown when there is congestion, the traffic is referred to as congestion-controlled traffic.
- In congestion-controlled traffic, it is understood that packets may arrive delayed, lost, or out of order.

Priority	Meaning
0	No specific traffic
1	Background data
2	Unattended data traffic
3	Reserved
4	Attended bulk data traffic
5	Reserved
6	Interactive traffic
7	Control traffic

**Table 3.4 Priorities for congestion-controlled traffic**

- Congestion-controlled data are assigned priorities from 0 to 7, which are listed in table 3.4. A priority of 0 is the lowest; a priority of 7 is the highest.
- Example: TCP (Using the sliding window protocol).

The priority descriptions are as follows;

- (1) No specific traffic:** A priority of 0 is assigned to a packet when the process does not define a priority.
- (2) Background data:** This group defines data that are usually delivered in the background. Delivery of the news is a good example.
- (3) Unattended data traffic:** If the user is not waiting for the data to be received, the packet will be given a priority of 2. E-mail belongs to this group.
- (4) Attended bulk data traffic:** A protocol that transfers data while the user is waiting to receive the data (possibly with delay) is given a priority of 4. FTP and HTTP belong to this group.

- (5) **Interactive traffic:** Protocols such as TELNET that need user interaction are assigned the second-highest priority (6) in this group.
- (6) **Control traffic:** Control traffic is given the highest priority (7). Routing protocols such as OSPF and RIP and management protocols such as SNMP have this priority.
- (7) **Noncongestion-Controlled Traffic:** This refers to a type of traffic that expects minimum delay. Discarding of packets is not desirable. Retransmission in most cases is impossible. Real-time audio and video are examples of this type of traffic.

**(ii) Noncongestion-Controlled Traffic**

- Priority numbers from 8 to 15 are assigned to Noncongestion-controlled traffic.
- The priorities are usually based on how much the quality of received data is affected by the discarding of packets.
- Data containing less redundancy can be given a higher priority (15).
- Data containing more redundancy are given a lower priority (8).

**Flow Label**

- A sequence of packets, sent from a particular source to a particular destination needs special handling by routers is called a flow of packets.
- The combination of the source address and the value of the flow label uniquely define a flow of packets.
- To a router, a flow is a sequence of packets that share the same characteristics like traveling the same path, using the same resources, having the same kind of security, and so on.
- A router that supports the handling of flow labels has a flow label table.
- The table has an entry for each active flow label; each entry defines the services required by the corresponding flow label.
- When the router receives a packet, it consults its flow label table to find the corresponding entry for the flow label value defined in the packet.

### 3.9.4 Comparison Between IPv4 and IPv6

The list given below compares IPv4 and IPv6 headers.

- (i) The header length field is eliminated in IPv6 because the length of the header is fixed in this version.
- (ii) The service type field is eliminated in IPv6. The priority and flow label fields together take over the function of the service type field.
- (iii) The total length field is eliminated in IPv6 and replaced by the payload length field.
- (iv) The identification, flag, and offset fields are eliminated from the base header in IPv6. They are included in the fragmentation extension header.
- (v) The TTL field is called hop limit in IPv6.

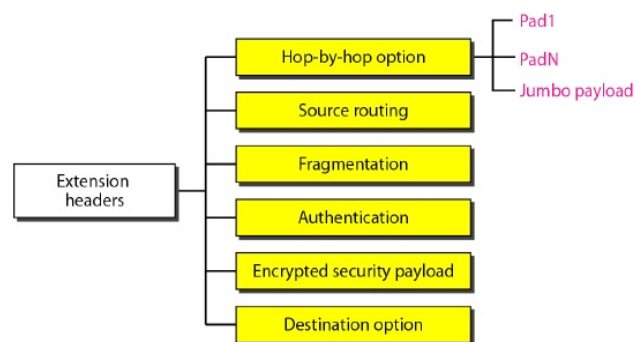
- (vi) The protocol field is replaced by the next header field.
- (vii) The header checksum is eliminated because the checksum is provided by upper-layer protocols. It is therefore not needed at this level.
- (viii) The option field in IPv4 is implemented as extension headers in IPv6.
- (ix) Below table shows the comparison between IPv4 options and IPv6 extension headers.

The list given below shows the comparison between IPv4 options and IPv6 extension headers

- (i) The no-operation and end-of-option in IPv4 are replaced by pad1 and pad N options in IPv6.
- (ii) The record route option is not implemented in IPv6 because it was not used.
- (iii) The timestamp option is not implemented because it was not used.
- (iv) The source route option is called the source route extension header in IPv6.
- (v) The fragmentation field in the base header section of IPv4 has moved to the fragmentation extension header in IPv6.
- (vi) The authentication extension header is new in IPv6.
- (vii) The encrypted security payload extension header is new in IPv6.

### 3.9.5 Extension Headers

- The length of the base header is fixed at 40 bytes.
- To give greater functionality to the IP datagram, the base header can be followed by up to six extension headers.
- Six types of extension headers have been defined for IPv6.
- Figure 3.46 shows the IPv6 extension header.



**Figure 3.46** Extension header types

Six types of IPv6 extension headers are listed below;

- (i) **Hop-by-Hop Option:** It is used when the source needs to pass information to all routers visited by the datagram.
- (ii) **Fragmentation:** In IPv6, only the original source can fragment the data. A source must use a path MTU discovery technique to find the smallest MTU supported by any network on the path. The source then fragments using this knowledge.

- (iii) **Authentication:** The authentication extension header is used to validate the message sender and ensures the integrity of data.
- (iv) **Encrypted Security Payload:** The encrypted security payload (ESP) is an extension that provides confidentiality and guards against eavesdropping.
- (v) **Destination Option:** The destination option is used when the source needs to pass information to the destination only. Intermediate routers are not permitted to access this information.

### 3.9.6 IPv6 Addresses

IPv6 addresses are used to solve the following issues faced by the IPv4 protocol. They are;

- (i) Address depletion problem for the Internet.
- (ii) Lack of accommodation for real-time audio and video transmission.
- (iii) Encryption and authentication of data for some applications.
  - An IPv6 address consists of 16 bytes (octets).
  - An IPv6 address is 128 bits long.

#### Hexadecimal Colon Notation

To make addresses more readable, IPv6 specifies hexadecimal colon notation. In this notation, 128 bits is divided into eight sections, each 2 bytes in length.

Two bytes in hexadecimal notation requires four hexadecimal digits. Therefore, the address consists of 32 hexadecimal digits, with every four digits separated by a colon, as shown in figure 3.47.

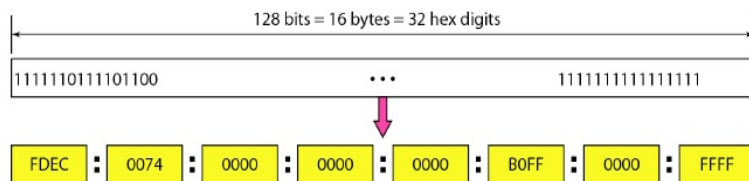


Figure 3.47 IPv6 address in binary and hexadecimal colon notation

#### Abbreviation

The hexadecimal format of the IP address is very long and many of the digits are zeros.

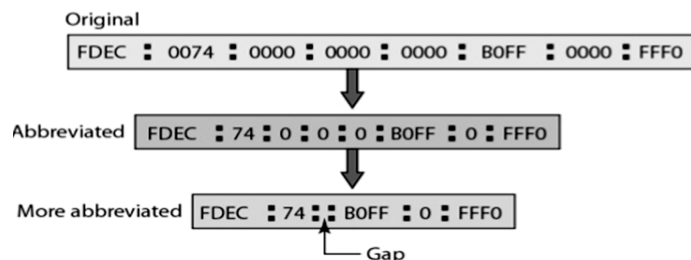


Figure 3.48 Abbreviated IPv6 addresses

In this case, we can abbreviate the address. The leading zeros of a section (four digits between two colons) can be omitted. Only the leading zeros can be dropped, not the trailing zeros.

### *Address Space*

- IPv6 has a much larger address space. That is 2<sup>128</sup> addresses are available.
- The IPv6 address is divided into several categories.
- A few leftmost bits, called the type prefix are used to define the address category.
- The type prefix is variable in length, but it is unique.

<b>TYPE PREFIX</b>	<b>TYPE</b>
0000 0000	Reserved
0000 0001	Unassigned
0000 001	ISO network address
0000 010	IPX (Novell) network address
0000 011	Unassigned
0000 1	Unassigned
0001	Reserved
001	Reserved
010	Provider-based unicast address
011	Unassigned
100	Geographic-based unicast address
101	Unassigned
110	Unassigned
1110	Unassigned
1111 0	Unassigned
1111 10	Unassigned
1111 110	Unassigned
1111 1110 0	Unassigned
1111 1110 10	Link local addresses
1111 1110 11	Site local addresses
1111 1111	Multicast addresses

*Table 3.5 Type prefixes for IPv6 addresses*

## **TRANSPORT LAYER**

---

### **4.1. 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.

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 ensures that the whole message arrives intact and in order. It takes care of both error control and flow control at the source – to – destination level rather than across a single link.

The transport layer header must include a type of address called a service-point address in the OSI model and port number or port addresses in the Internet and TCP/IP protocol suite.

A transport layer protocol can be either connectionless or connection-oriented.

- (i) A connectionless transport layer treats each segment as an independent packet and delivers it to the transport layer at the destination machine.
- (ii) A connection-oriented transport layer makes a connection with the transport layer at the destination machine first before delivering the packets. After all the data is transferred, the connection is terminated.

At the transport layer, a message is normally divided into transmittable segments. A connectionless protocol, such as UDP, treats each segment separately. A connection oriented protocol, such as TCP and SCTP, creates a relationship between the segments using sequence numbers.

Flow and error control at this layer is performed end to end rather than across a single link.

### **4.2. USER DATAGRAM PROTOCOL (UDP)**

The User Datagram Protocol (UDP) is a connectionless, unreliable transport protocol. It provides process-to-process communication. It performs very limited error checking.

UDP is a very simple protocol with minimum overhead. For example, sending a small message by using UDP takes much less interaction between the sender and receiver.

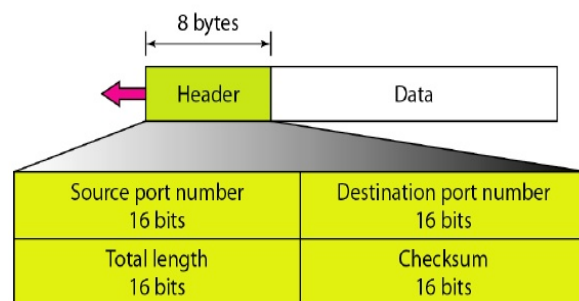


Port	Protocol	Description
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	User	Active users
13	Daytime	Returns the date and time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
53	Nameserver	Domain name service
67	BOOTPs	Server port to download bootstrap information
68	BOOTPc	Client port to download bootstrap information
69	TFTP	Trivial file transfer protocol
111	RPC	Remote procedure call
123	NTP	Network time protocol
161	SNMP	Simple network management protocol
162	SNMP	Simple network management protocol (trap)

**Table 4.1 Well-known ports used with UDP**

### 4.2.1 User Datagram

In UDP, the packets are called user datagrams. Datagrams have a fixed-size header of 8 bytes. The figure 4.1 shows the datagram format of UDP.



**Figure 4.1 User datagram format**

The fields are as follows:

#### **i) Source port number**

- The port number used by the process running on the source host.
- It is 16 bits long.
- The port number can range from 0 to 65,535.
- If the source host is the client, the port number is an ephemeral port number requested by the process and chosen by the UDP software.
- If the source host is the server, the port number is a well-known port number.

**ii) Destination port number**

- It is the port number used by the process running on the destination host.
- It is 16 bits long.
- If the destination host is the server, the port number is a well-known port number.
- If the destination host is the client, the port number is an ephemeral port number.

**iii) Length**

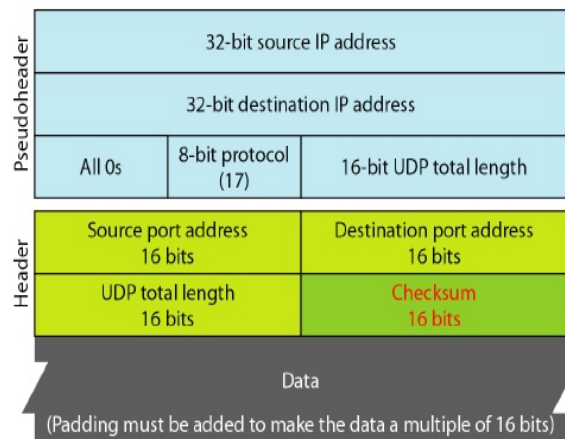
- It defines the total length of the user datagram (header plus data).
- This is a 16-bit field (0 to 65,535 bytes).
- That defines the total length of the user datagram.
- A user datagram is encapsulated in an IP datagram, So

$$\text{UDP length} = \text{IP length} - \text{IP header's length.}$$

**iv) Checksum**

- Used to detect errors over the entire user datagram (header plus data).
- The checksum includes three sections: a pseudo header, the UDP header, and the data coming from the application layer.

The pseudo header is the part of the header of the IP packet in which the user datagram is to be encapsulated with some fields filled with 0s. The above figure shows the format of the UDP pseudo header. With the help of the pseudo header we can verify that a given message has been delivered between the correct two end points. The value of the protocol field for UDP is 17.



**Figure 4.2 Pseudo header for checksum calculation**

**4.2.2 UDP Operation**

UDP uses concepts common to the transport layer.

**(i) Connectionless Services**

- UDP provides a connectionless service.

- Each user datagram sent by UDP is treated as an independent datagram.
- The user datagrams are not numbered.
- There is no connection establishment and no connection termination.
- Each user datagram can travel on a different path.

**(ii) Flow and Error Control**

- UDP is a very simple, unreliable transport protocol.
- There is no flow control and no window mechanism.
- The receiver may overflow with incoming messages.
- There is no error control mechanism in UDP (Except the checksum at the receiver).
- The sender does not know if a message has been lost or duplicated.
- When the receiver detects an error through the checksum, the user datagram is silently discarded.

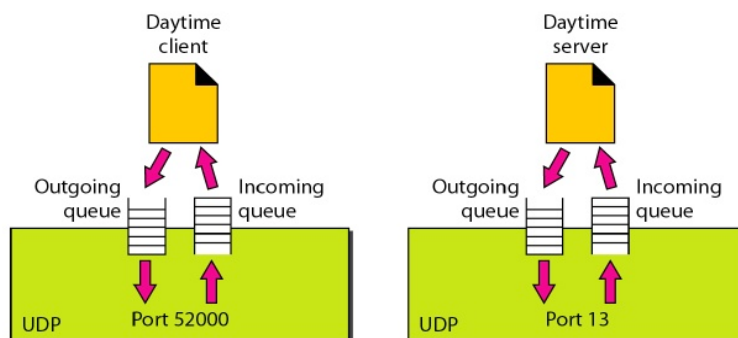
**(iii) Encapsulation and Decapsulation**

- To send a message from one process to another, the UDP protocol encapsulates and decapsulates messages in an IP datagram.

**(iv) Queuing**

In UDP, queues are associated with ports.

- When a message arrives, the protocol appends the message to the end of the queue.
- If the queue is full, the message will be discarded.
- There is no flow control mechanism that tells the sender to slow down.
- When an application process wants to receive the message, one is removed from the front of the queue.
- If the queue is empty, the process blocks until a message becomes available.
- It ensures the correctness of the message by the use of a checksum.
- UDP computes its checksum over the UDP header, the contents of the message body and something called pseudo header.



**Figure 4.3 Queues in UDP**



### 4.2.3 Applications of UDP

- (i) UDP is suitable for a simple request-response communication.
- (ii) It is suitable for a process with internal flow and error control mechanisms (For example, the Trivial File Transfer Protocol (TFTP)).
- (iii) It is a suitable transport protocol for multicasting.
- (iv) It is used for management processes such as SNMP.
- (v) It is used for some route updating protocols such as Routing Information Protocol.

#### *Disadvantages of UDP*

- (i) It is not usually used for a process such as FTP that needs to send bulk data.
- (ii) It is suitable for a process with little concern for flow and error control.

## 4.3 RELIABLE BYTE STREAM (TCP)

- TCP is a transport protocol which offers a reliable, connection-oriented, byte-stream service.
- TCP is used to avoid the missing or reordered data.
- TCP guarantees the reliable, in-order delivery of a stream of bytes.
- It is a full-duplex protocol (Each TCP connection supports a pair of byte streams, one flowing in each direction).
- It also includes a flow-control mechanism for each of these byte streams.
- Flow control mechanism allows the receiver to limit how much data the sender can transmit at a given time.
- TCP supports a demultiplexing mechanism that allows processing multiple application programs simultaneously.
- TCP also implements a highly tuned congestion-control mechanism.
- Congestion-control mechanism is used to decide how fast TCP sends data, how to keep the sender from overrunning the receiver and the sender from overloading the network.

### 4.3.1 Port Numbers

TCP provides process-to-process communication using port numbers. The table 4.2 lists some well-known port numbers used by TCP. If the application is going to both TCP and UDP, then the same port number will be assigned to that application for these two protocols.

Port	Protocol	Description
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	User	Active users

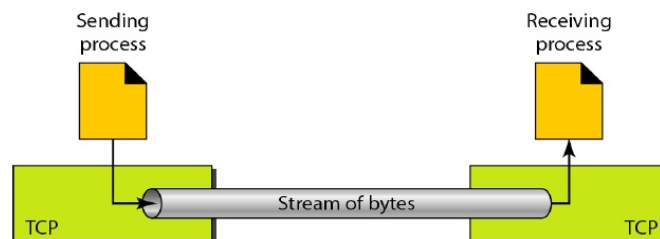
13	Daytime	Returns the date and time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
20	FIP.Data	File transfer protocol (data connection)
21	FIP.Control	File transfer protocol (control connection)
23	TELNET	Tenninal network
25	SMTP	Simple mail transfer protocol
53	DNS	Domain name server
67	BOOTP	Bootstrap protocol
79	Finger	Finger
80	HTTP	Hypertext transfer protocol
111	RPC	Remote procedure call

*Table 4.2 Well-known ports used by TCP*

## 4.3.2 TCP Services

### 4.3.2.1 Stream Delivery Service

- TCP is a stream-oriented protocol.
- TCP allows the sending process to deliver the data as a stream of bytes and allows the receiving process to obtain data as a stream of bytes.
- In TCP, two processes can be connected by an imaginary tube.
- The imaginary tube is used to carry the data across the Internet.
- In the figure given below the sending process produces (writes to) the stream of bytes, and the receiving process consumes (reads from) them.



*Figure 4.4 Stream delivery*

### *Sending and Receiving Buffers*

TCP is in the need of buffers for storage, because the sending and the receiving processes may not write or read data at the same speed. TCP is uses two buffers, the sending buffer and the receiving buffer, one for each direction.

Buffers can be implemented by using a circular array of I-byte locations. The figure 4.5 shows the data movement in one direction.

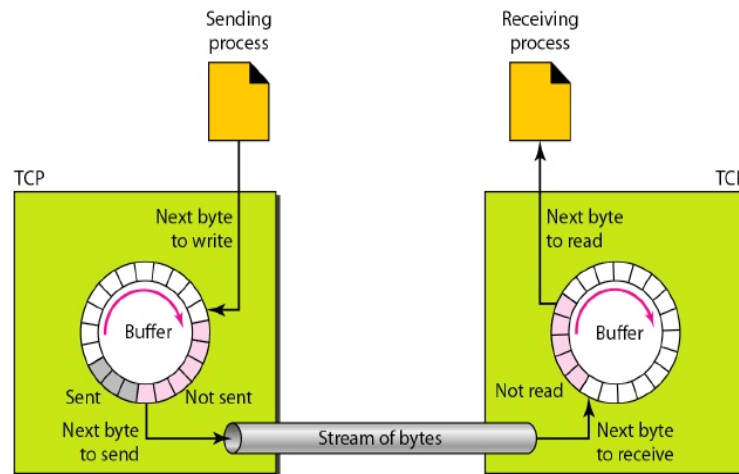


Figure 4.5 Sending and receiving buffers

### ***Sending Site***

At the sending site, the buffer has three types of chambers.

- (i) The white section contains empty chambers that can be filled by the sending process (producer).
- (ii) The gray area holds bytes that have been sent but not yet acknowledged. TCP keeps these bytes in the buffer until it receives an acknowledgment.
- (iii) The colored area contains bytes to be sent by the sending TCP.

### ***Receiving site***

At the receiver site, the circular buffer is divided into two areas.

- (i) The white area contains empty chambers to be filled by bytes received from the network.
- (ii) The colored sections contain received bytes that can be read by the receiving process.

### ***Segments***


- TCP groups a number of bytes together into a packet called a segment.
- TCP adds a header to each segment and delivers the segment to the IP layer for transmission.

#### ***4.3.2.2 Full-Duplex Communication***

- TCP offers full-duplex service, in which data can flow in both directions at the same time.
- Each TCP has a sending and receiving buffer, and segments move in both directions.

#### ***4.3.2.3 Connection-Oriented Service***

TCP is a connection-oriented protocol. The following steps have to be taken when the sending process wants to send and receive data from the receiving process.

- 
- (i) The two TCPs establish a connection between them.
  - (ii) Data are exchanged in both directions.
  - (iii) The connection is terminated.

#### **4.3.2.4 Reliable Service**

- TCP is a reliable transport protocol.
- It uses an acknowledgment mechanism to check the safe and sound arrival of data.

### **4.3.3 TCP Features**

#### **4.3.3.1 Numbering System**

The bytes of data being transferred in each connection are numbered by TCP. The numbering starts with a randomly generated number. There are two fields called the sequence number and the acknowledgment number used by the TCP software to keep track of the segments being transmitted or received. These two fields refer to the byte number and not the segment number.

##### ***Byte Number***

- TCP numbers all data bytes that are transmitted in a connection.
- Numbering is independent in each direction.
- When TCP receives bytes of data from a process, it stores them in the sending buffer and numbers them.
- The numbering does not necessarily start from 0.
- TCP generates a random number between 0 and 232 - 1 for the number of the first byte.

##### ***Sequence Number***

- After the bytes have been numbered, TCP assigns a sequence number to each segment that is being sent.
- The sequence number for each segment is the number of the first byte carried in that segment.

##### ***Example***

**Suppose a TCP connection is transferring a file of 5000 bytes. The first byte is numbered 10,001. What are the sequence numbers for each segment if data are sent in five segments, each carrying 1000 bytes?**

##### ***Solution:***

The following shows the sequence number for each segment:

Segment 1	→	Sequence Number:	10,001 (range: 10,001 to 11,000)
Segment 2	→	sequence Number:	11,001 (range: 11,001 to 12,000)
Segment 3	→	Sequence Number:	12,001 (range: 12,001 to 13,000)
Segment 4	→	Sequence Number:	13,001 (range: 13,001 to 14,000)

Segment 5 → Sequence Number: 14,001 (range: 14,001 to 15,000)

The value in the sequence number field of a segment defines the number of the first data byte contained in that segment.

**Acknowledgment field**

- The value of the acknowledgment field in a segment defines the number of the next byte a party expects to receive.
- The acknowledgment number is cumulative.

**4.3.3.2 Flow Control**

- TCP provides the mechanism for flow control.
- The receiver of the data controls the amount of data that are to be sent by the sender.
- This is done to prevent the receiver from being overwhelmed with data.
- The numbering system allows TCP to use a byte-oriented flow control.

**4.3.3.3 Error Control**

- TCP implements an error control mechanism to provide reliable service.
- Error control mechanism considers a segment as the unit of data for error detection (loss or corrupted segments).
- Error control is byte-oriented.

**4.3.3.4 Congestion Control**

- TCP also provides the mechanism to control the congestion in the network.
- The amount of data sent by a sender is not only controlled by the receiver (flow control), but is also determined by the level of congestion in the network.

**4.3.4 TCP SEGMENT**

A packet in TCP is called a segment. The format of a segment is shown in the below figure.

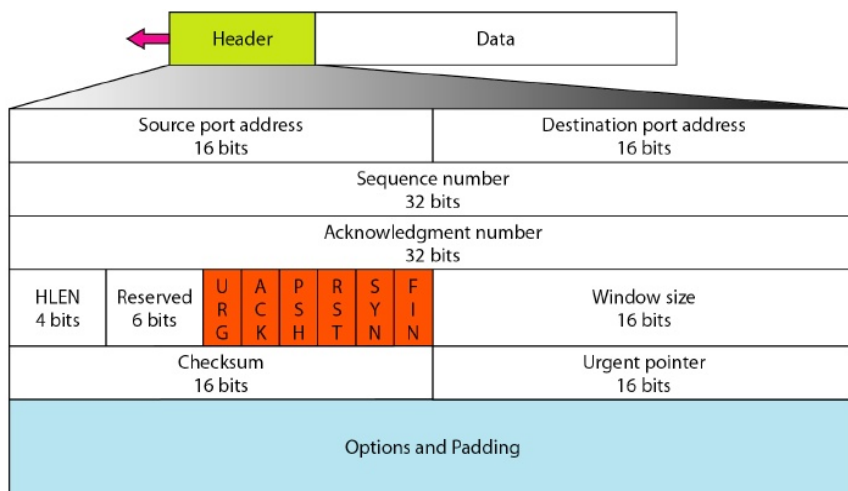
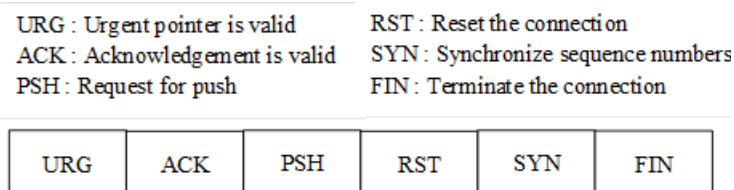


Figure 4.6 TCP segment format



The segment consists of a 20 to 60 byte header, followed by data. The header is 20 bytes if there are no options and up to 60 bytes if it contains options. The header contains the following sub fields.

- (i) **Source port address:** This is a 16-bit field that defines the port number of the application which is sending the segment.
- (ii) **Destination port address:** This is a 16-bit field that defines the port number of the application program which is receiving the segment.
- (iii) **Sequence number:** This 32-bit field defines the number assigned to the first byte of data contained in this segment.
- (iv) **Acknowledgment number:** This 32-bit field defines the byte number that the receiver of the segment is expecting to receive from the other party. If the receiver has successfully received byte number  $x$ , then it defines  $x + 1$  as the acknowledgment number.
- (v) **Header length:** This 4-bit field indicates the number of 4-byte words in the TCP header. The length of the header can be between 20 and 60 bytes.
- (vi) **Reserved:** This is a 6-bit field reserved for future use.
- (vii) **Control:** This field defines 6 different control bits or flags as shown in the below figure. One or more of these bits can be set at a time.



**Figure 4.7 Control field**

Control bits are used to enable flow control, connection establishment and termination, connection abortion, and the mode of data transfer in TCP. The table 4.3 describes each control bits in detail.

Flag	Description
URG	The value of the urgent pointer field is valid
ACK	The value of the acknowledgement field is valid
PSH	Push the data
RST	Reset the connection
SYN	Synchronize sequence numbers during connection
FIN	Terminate the connection

**Table 4.3 Description of flags in the control field**

- (viii) **Window size:** This is a 16 bit field, defines the size of the window. The maximum size of the window is 65,535 bytes and is determined by the receiver. The sender must obey the dictation of the receiver in this case.
- (ix) **Checksum:** This 16-bit field contains the checksum is used for error detection and correction during the data transmission.
- (x) **Urgent pointer:** This 16-bit field is valid only if the urgent flag is set. It is used when the segment contains urgent data. It must be added to the sequence number to obtain the number of the last urgent byte in the data section of the segment.
- (xi) **Options:** There can be up to 40 bytes of optional information in the TCP header.

#### 4.4. TCP CONNECTION MANAGEMENT

TCP is connection-oriented transport protocol. It establishes a virtual path between the source and destination. All the segments belonging to a message are sent over this virtual path. The virtual path facilitates the acknowledgment process as well as retransmission of damaged or lost frames.

TCP uses the services of IP to deliver individual segments to the receiver, but it controls the connection itself. If a segment is lost or corrupted, it is retransmitted. If a segment arrives out of order, TCP holds it until the missing segments arrive.

The connection-oriented transmission requires three phases. They are;

- (i) Connection establishment
- (ii) Data transfer
- (iii) Connection termination

##### 4.4.1 Connection Establishment

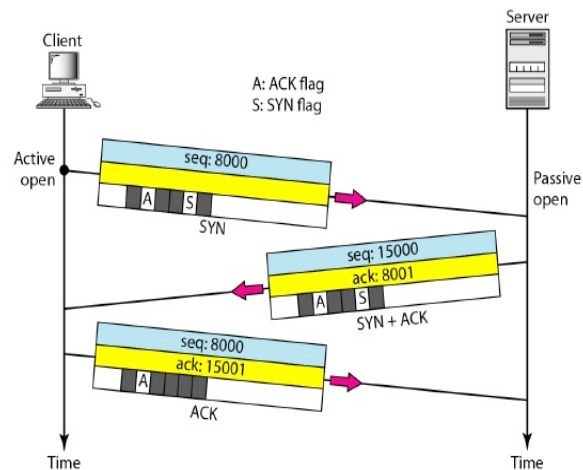
TCP transmits data in full-duplex mode. For transferring the data, each party must initialize communication and get approval from the other party.

##### *Three-Way Handshaking*

The connection establishment in TCP is called three-way handshaking. In our example, a client application program wants to make a connection with server application program using TCP protocol.

The process starts with the server. The server program tells its TCP that it is ready to accept a connection. This is called a request for a **passive open**.

The client program issues a request for an **active open**. A client that wishes to connect to an open server tells its TCP that it needs to be connected to that particular server. TCP can now start the three-way handshaking process as shown in the below figure.



**Figure 4.8** Connection establishment using three-way handshaking

The three steps involved in the connection establishment are as follows.

- (1) The client sends the first segment, a SYN segment. This contains the source and the destination port numbers. This segment also contains the client Initialization Sequence Number (ISN). The ISN is used for numbering the bytes of the data sent from the client to the server.
- (2) The server sends the second segment, a SYN +ACK segment, with 2 flag bits set: SYN and ACK. This segment has a dual purpose. First it acknowledges the receipt of the first segment by using the ACK flag and acknowledgement number field. Second, the segment is used as the initialization segment for the server. It contains the initialization sequence number used to numbering the bytes sent from the server to the client.
- (3) The client sends the third segment. This is an ACK segment. It acknowledges the receipt of the second segment with the ACK flag and acknowledgment number field. The client must define the server widow size here.

#### 4.4.2 Data Transfer

After connection is established, bidirectional data transfer can take place. The client and server can both send data and acknowledgments. The acknowledgment is piggybacked with the data.

In the figure 4.9 after connection is established the client sends 2000 bytes of data in two segments. The server then sends 2000 bytes in one segment. The client sends one more segment. The first three segments carry both data and acknowledgment, but the last segment carries only an acknowledgment because there are no more data to be sent.

The data segments sent by the client have the PSH flag set so that the server TCP knows to deliver data to the server process as soon as they are received.

When the receiving TCP receives a segment with the URG bit set, it extracts the urgent data from the segment, using the value of the urgent pointer, and delivers them, out of order, to the receiving application program.

Buffer is used at the sending TCP and the receiving TCP side for providing the synchronization between the sender and the client.

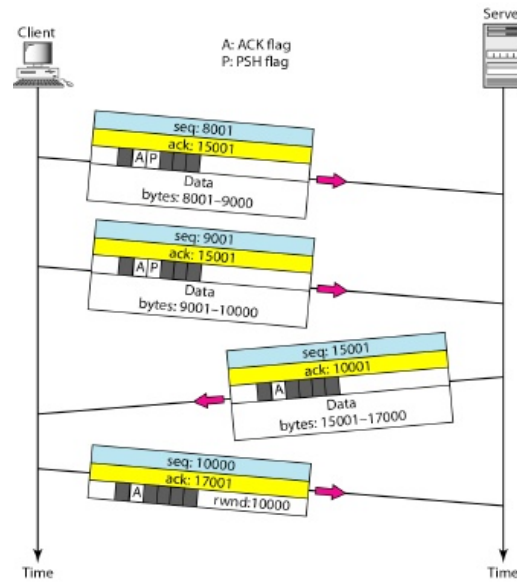


Figure 4.9 Data transfer

### 4.4.3 Connection Termination

- The connection can be closed either by the client or server which is involved in exchanging data.
- The connection termination is usually initiated by the client.
- When connection in one direction is terminated, the other party can continue sending data in the other direction.
- It can be done in two options:
  - (i) Three-way handshaking
  - (ii) Four-way handshaking with a half-close option.

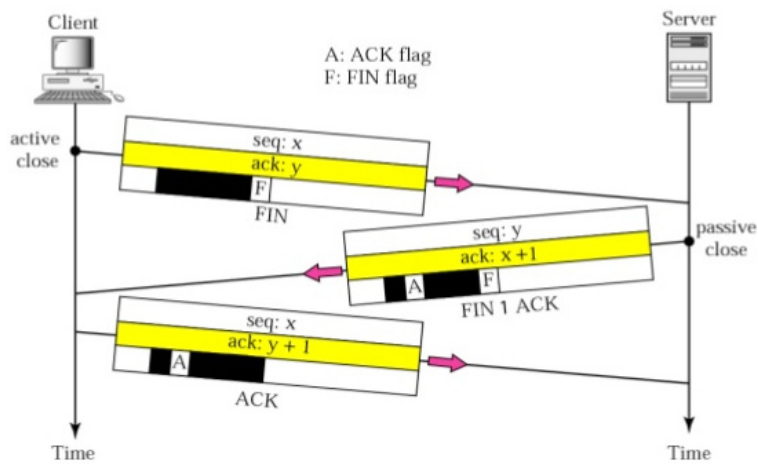


Figure 4.10 Connection termination using three-way handshaking

### Three-Way Handshaking

The following steps are needed to close the connection in both directions. The following figure explains the same.

- (1) The Client TCP sends the first segment, a FIN segment in which the FIN flag is set.
- (2) After receiving the FIN segment, the server TCP sends the second segment, a FIN +ACK segment, to confirm the receipt of the FIN segment from the client.
- (3) The server TCP can continue server client direction. When it not having any data to send, then it sends the third segment called FIN segment.
- (4) The client TCP sends the last segment, an ACK segment, to confirm the receipt of the FIN segment from the TCP server.

### Half-Close

In TCP, one end can stop sending data while still receiving data. This is called a half-close. Either end can issue a half-close; it is normally initiated by the client. It can occur when the server needs all the data before processing can begin. Below figure is an example for half-close.

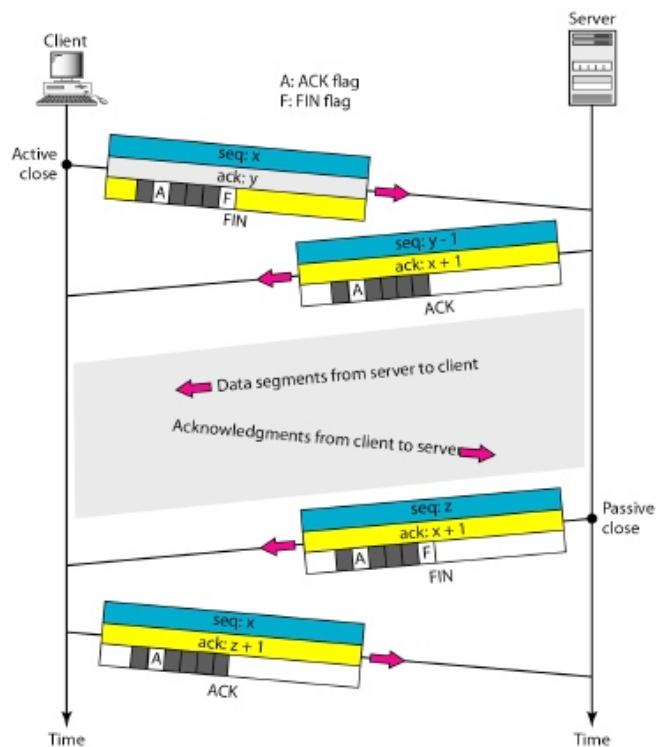


Figure 4.11 Half-close

- The client half-closes the connection by sending a FIN segment.
- The server accepts the half-close by sending the ACK segment.
- The data transfer from the client to the server stops.
- The server can still send data.

- When the server has sent all the processed data, it sends a FIN segment, which is acknowledged by an ACK from the client.
- After half-closing of the connection, data can travel from the server to the client and acknowledgments can travel from the client to the server.
- The client cannot send any more data to the server.

### Connection Resetting

TCP may request the connection resetting. Resetting means that the current connection is destroyed. This can be done in the following three cases;

- (1) The TCP on one side has requested a connection to a non-existent port. The TCP on the other side may send a segment with its RST bit set to annul the request.
- (2) One TCP may want to abort the connection due to an abnormal situation. It can send an RST segment to close the connection.
- (3) The TCO on one side may discover that the TCP on the other side has been idle for a long time. It may send an RST segment to destroy the connection.

### 4.4.4 State Transition Diagram

State transition diagram is used to keep track of the entire events happening during connection establishment, connection termination and data transfer. The TCP software can be implemented as a finite state machine.

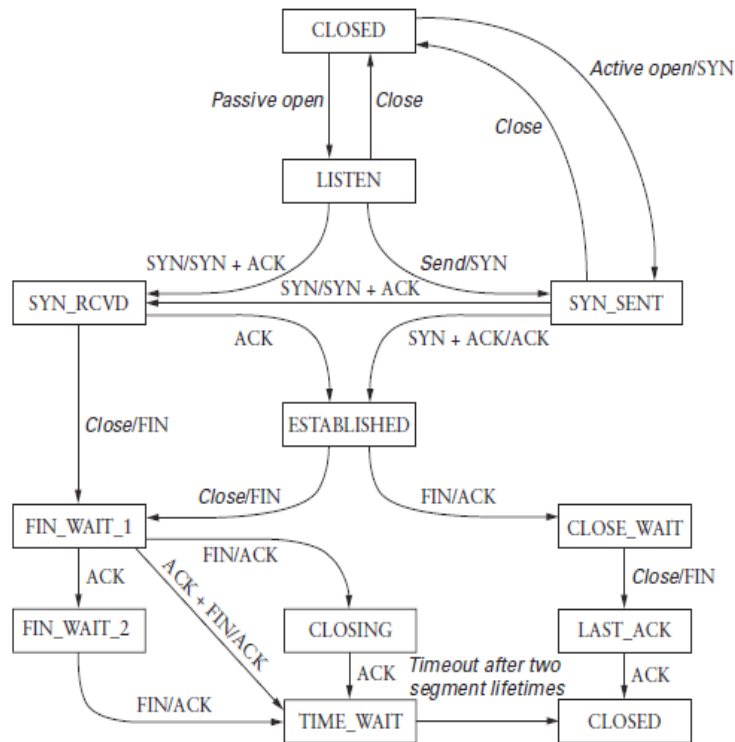
A finite state machine is a machine that goes through a limited number of states. At any moment the machine is in one of the state. The table 4.4 shows the different states of the TCP finite state machine.

State	Description
CLOSED	No connection is active or pending
LISTEN	The server is waiting for an incoming call
SYN RCVD	A connection request has arrived; wait for ACK
SYN SENT	The application has started to open a connection
ESTABLISHED	The normal data transfer state
FIN WAIT 1	The application has said it is finished
FIN WAIT 2	The other side has agreed to release
TIMED WAIT	Wait for all packets to die off
CLOSING	Both sides have tried to close simultaneously
CLOSE WAIT	The other side has initiated a release
LAST ACK	Wait for all packets to die off

**Table 4.4** The states used in the TCP connection management finite state machine

The states are shown using ovals. The transition from one state to another is shown using the directed lines. Each string has two strings separated by a slash. The first string is an input, what TCP receives. The second string is an output, what TCP sends. The dotted line in the figure represents

the server activities and the solid lines represent the client activities. The figure 4.12 shows the state transition diagram for both client and server.




**Figure 4.12 TCP state transition diagram.**

### Client Diagram

The client can be in one of the following states;

CLOSED, SYN\_SENT, ESTABLISHED, FIN\_WAIT\_1, FIN\_WAIT\_2, AND TIME\_WAIT.

- (1) The client TCP starts in the CLOSED state.
- (2) In this state, the client TCP can receive an active open request from the client application program. It sends a SYN segment to the server TCP and goes to the SYN\_SENT state.
- (3) In this state, the client TCP can receive a SYN + ACK segment from other TCP. It sends an ACK segment to other TCP and goes to the ESTABLISHED state. This is the data transfer state. The client remains in this state as long as it is sending and receiving data.
- (4) In this state, the client TCP can receive a close request from the client application program. It sends a FIN segment to other TCP and goes to the FIN\_WAIT\_1 state.
- (5) While in this state, the client TCP waits to receive an ACK from the server TCP. When the ACK is received, it goes to the FIN\_WAIT\_2 state. It does not send anything. Now the connection is closed in one direction.

- 
- (6) The client remains in this state, waiting for the server to close the connection from other side. If the client receives a FIN segment from other end, it sends an ACK segment and goes to the TIME\_WAIT state.
  - (7) When the client is in this state, it starts a timer and waits until this timer goes off. The value of this timer is set to double the maximum life time estimated for a segment. The client remains in this state until all the duplicate packets from the other ends are going to be discarded. After the time-out, the client goes to the CLOSED state.

### ***Server diagram***

The server in TCP is in one of the following states;

CLOSED, LISTEN, SYN\_RCVD, ESTABLISHED, CLOSE\_WAIT, and LAST\_ACK.

The server TCP starts in the CLOSED state.

- (1) In this state, the server TCP can receive a passive open request from the server application program. It goes to the LISTEN state.
- (2) In this state, the server TCP can receive a SYN segment from the client TCP. It sends a SYN + ACK segment to the client TCP and then goes to the SYN\_RCVD state.
- (3) In this state, the server TCP can receive an ACK segment from the client TCP. It goes to the ESTABLISHED state. This is the data transfer state. The server remains in this state as long as it is receiving and sending data.
- (4) In this state, the server TCO can receive a FIN segment from the client. Here the client may wish to close the connection. It can send the ACK segment to the client and goes to the CLOSE\_WAIT state.
- (5) In this state, the server waits until it receives a close request from the server program. It then sends a FIN segment to the client and goes to the LAST\_ACK state.
- (6) While in this state, the server waits for the last ACK segment. If then goes to the CLOSED state.

## **4.5 FLOW CONTROL**

Flow control defines the amount of data a source can send before receiving an acknowledgement from the destination.

In the earlier stage, a transport layer protocol can send 1 byte of data and waits for an acknowledgement before sending the next byte. This is an extremely slow process. Because, the source is in idle state while it is waits for an acknowledgement.

To overcome this problem, a transport layer protocol can send all the data it has without worrying about an acknowledgement. This method can speed up the process, but it may overwhelm the receiver.

Flow control balances the rate a producer creates data with the rate a consumer can use the data. TCP separates flow control from error control. We temporarily assume that the logical channel



between the sending and receiving TCP is error-free. The figure 4.13 shows unidirectional data transfer between a sender and a receiver. To solve this problem, TCP has introduced the concept of sliding windows.

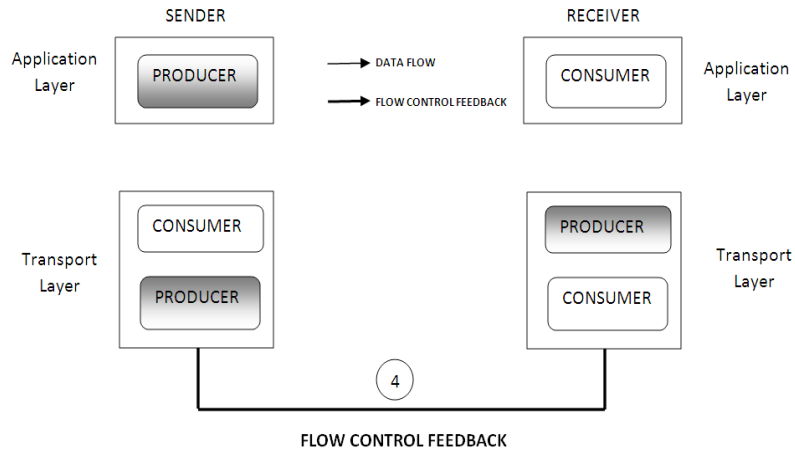


Figure 4.13 TCP/IP protocol suites

### 4.5.1 Windows in TCP

TCP uses two windows (send window and receive window) for each direction of data transfer, which means four windows for a bidirectional communication. To make the discussion simple, we make an assumption that communication is only unidirectional; the bidirectional communication can be inferred using two unidirectional communications with piggybacking.

#### Send Window

The sender window has the size less than or equal to the size of the receiver window. This window includes the bytes already sent and not acknowledged and those that can be sent. The figure 4.14 shows the sender buffer with the sender window.

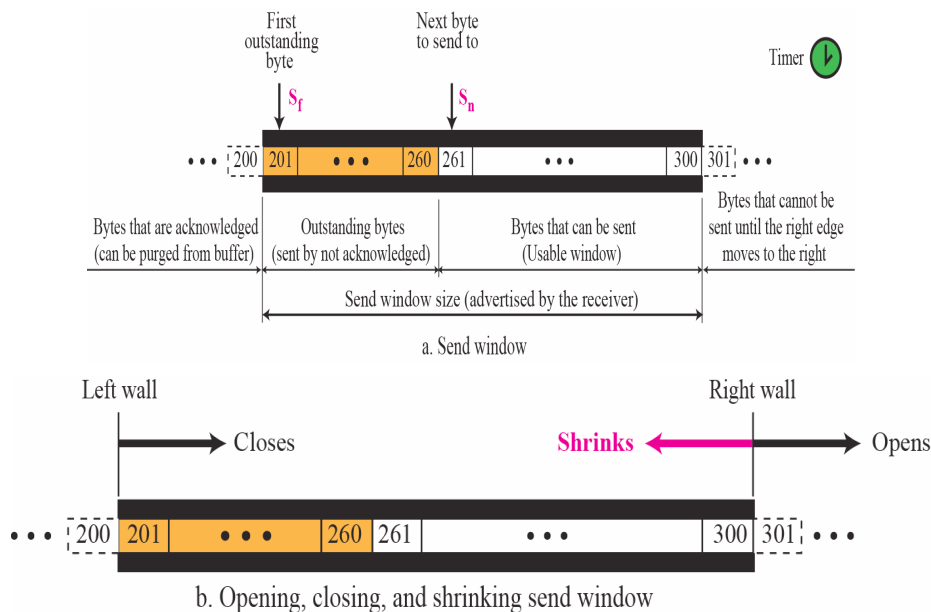
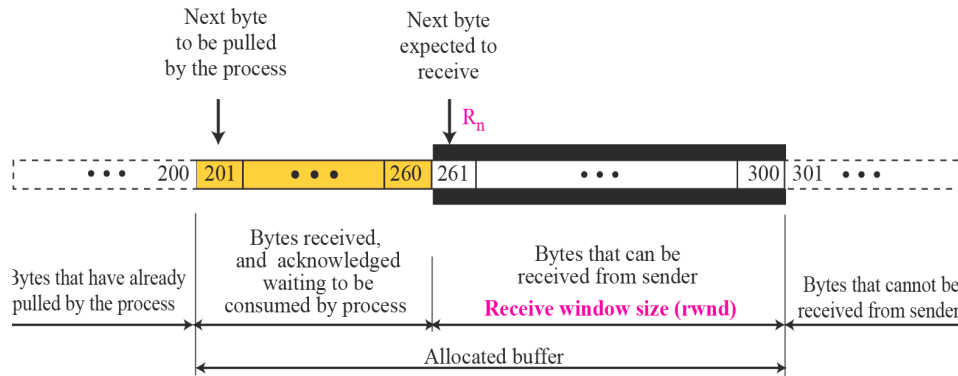


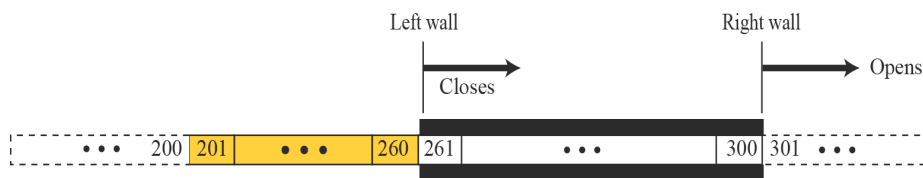
Figure 4.14 Send window in TCP

### Receive Window

The receiver window size is fixed by the receiver itself. This window shows the next byte which will be consumed by the receiver. The figure 4.15 shows the receiver buffer with the receiver window.



a. Receive window and allocated buffer



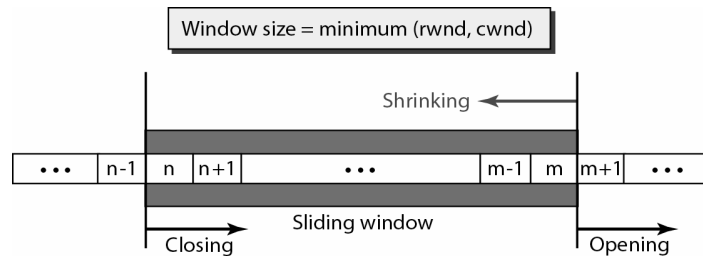
b. Opening and closing of receive window

**Figure 4.15 Receive windows in TCP**

### 4.5.2 Sliding Window Protocol

The sliding window of TCP is byte-oriented and of variable size. The figure 4.16 shows the sliding window in TCP.

A sliding window is used to make transmission more efficient as well as to control the flow of data so that the destination does not become overwhelmed with data. TCP sliding windows are byte-oriented.



**Figure 4.16 Sliding window**

The window spans a portion of the buffer containing bytes received from the process. The bytes inside the window are the bytes that can be in transit. That means, they can be sent without worrying about acknowledgment. The imaginary window has two walls: one left and one right.

The window can be opened, closed, or shrunk. These three activities are in the control of the receiver, not by the sender. The sender must obey the commands of the receiver.

- (i) Opening a window means moving the right wall to the right. This allows more new bytes in the buffer that are eligible for sending.
- (ii) Closing the window means moving the left wall to the right. This means that some of the transmitted bytes have been acknowledged and the sender need not worry about them.
- (iii) Shrinking the window means moving the right wall to the left. This is strongly discouraged and not allowed in some implementations. It means revoking the eligibility of some bytes for sending. This is a problem if the sender has already sent these bytes.

The size of the window at one end is determined by the lesser of two values: receiver window (rwnd) or congestion window (cwnd).

- (i) **rwnd**: It is the number of bytes the other end can accept before its buffer overflows and data are discarded.
- (ii) **cwnd**: It is a value determined by the network to avoid congestion.

### Example 1

**What is the value of the receiver window (rwnd) for host A if the receiver, host B, and has a buffer size of 5000 bytes and 1000 bytes of received and unprocessed data?**

#### Solution:

$$\begin{aligned} \text{The value of rwnd} &= 5000 - 1000 \\ &= 4000. \end{aligned}$$

So, host B can receive only 4000 bytes of data before overflowing its buffer.

Now, host B advertises this value in its next segment to A.

### Example 2

**What is the size of the window for host A if the value of rwnd is 3000 bytes and the value of cwnd is 3500 bytes?**

#### Solution:

We know that, the size of the window is the smaller of rwnd and cwnd.

So, window size for the host A = 3000 bytes.

### Example 3

**In the below figure the sender has sent bytes up to 202. We assume that cwnd is 20. The receiver has sent an acknowledgment number of 200 with an rwnd of 9 bytes. The size of the sender window is the minimum of rwnd and cwnd, or 9 bytes. Bytes 200 to 202 are sent, but not acknowledged. Bytes 203 to 208 can be sent without worrying about acknowledgment. Bytes 209 and above cannot be sent.**

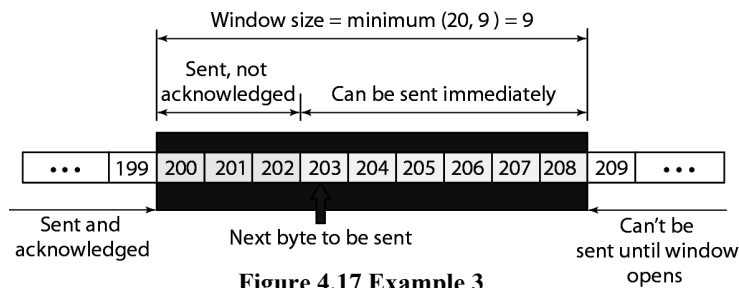


Figure 4.17 Example 3

### 4.5.3 Shrinking the Sender Window

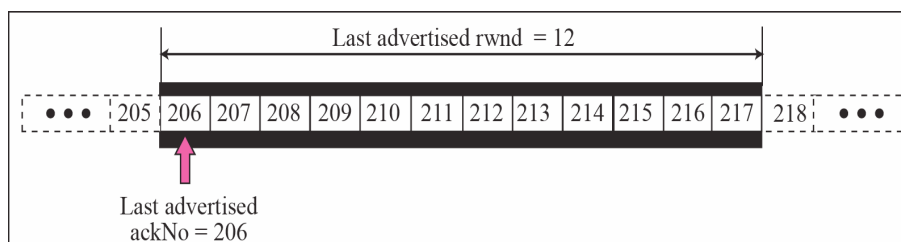
The figure 4.18 shows the reason for the mandate in window shrinking. Part (a) of the figure shows values of last acknowledgment and rwnd. Part (b) shows the situation in which the sender has sent bytes 206 to 214. Bytes 206 to 209 are acknowledged and purged.

The new advertisement, defines the new value of rwnd as 4, in which  $210 + 4 < 206 + 12$ . When the send window shrinks, it creates a problem: byte 214 which has been already sent is outside the window.

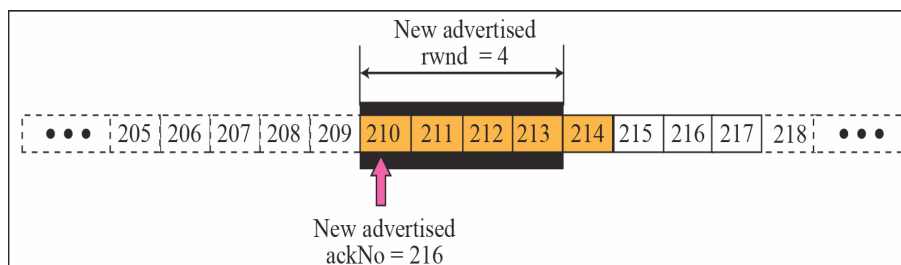
The relation discussed before forces the receiver to maintain the right-hand wall of the window to be as shown in part (a) because the receiver does not know which of the bytes 210 to 217 has already been sent.

One way to prevent this situation is to let the receiver postpone its feedback until enough buffer locations are available in its window.

In other words, the receiver should wait until more bytes are consumed by its process.



a. The window after the last advertisement



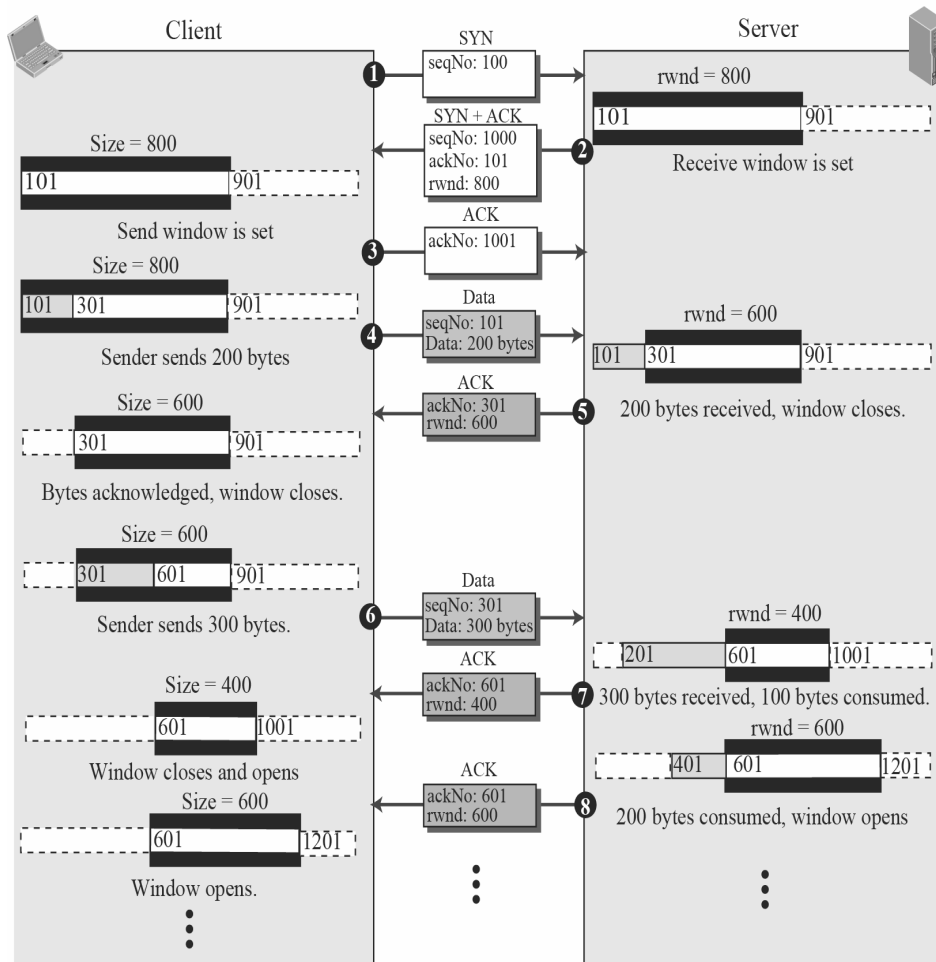
b. The window after the new advertisement; window has shrunk

Figure 4.18 Shrinking the sender window

**Important points about TCP sliding windows**

- The size of the window is the lesser of *rwnd* and *cwnd*.
- The source does not have to send a full window's worth of data.
- The window can be opened or closed by the receiver, but should not be shrunk.
- The destination can send an acknowledgment at any time as long as it does not result in a shrinking window.
- The receiver can temporarily shut down the window. The sender can always send a segment of 1 byte after the window is shut down.

Note: We assume only unidirectional communication from client to server. Therefore, only one window at each side is shown.



**Figure 4.19** An example of flow control

**4.5.4 Silly Window Syndrome**

A serious problem can arise in the sliding window operation, either the sending application program creates data slowly or the receiving application process consumes the data slowly or both. This problem is called as a silly window syndrome.

←—————→

***Syndrome created by the Sender***

- Sending application program creates data slowly (e.g. 1 byte at a time)
- Wait and collect data to send in a larger block
- How long should the sending TCP wait?
- Solution: Nagle’s algorithm
- Nagle’s algorithm takes into account (1) the speed of the application program that creates the data, and (2) the speed of the network that transports the data

***Syndrome created by the Receiver***

- Receiving application program consumes data slowly (e.g. 1 byte at a time)
- The receiving TCP announces a window size of 1 byte. The sending TCP sends only 1 byte.
- ***Solution 1:*** Clark’s solution
  - i) Sending an ACK but announcing a window size of zero until there is enough space to accommodate a segment of max. size or until half of the buffer is empty
- ***Solution 2:*** Delayed Acknowledgement
  - i) The receiver waits until there is decent amount of space in its incoming buffer before acknowledging the arrived segments
  - ii) The delayed acknowledgement prevents the sending TCP from sliding its window. It also reduces traffic.
- ***Disadvantage:*** it may force the sender to retransmit the unacknowledged segments
- ***To balance:*** should not be delayed by more than 500ms

**4.5.5 Adaptive Flow Control**

The sender and the receiver side buffers are of finite size. The size can be denoted as follows;

- (i) MaxSendBuffer
- (ii) MaxRcvdBuffer

We might know about the number of bytes being buffered at the sender and the receiver. But we don’t know where those bytes are actually stored. To avoid overflowing its buffer, the receiver side TCP must keep,

$$\text{LastByteRcvd} - \text{LastByteRead} \leq \text{MaxRcvdBuffer}$$

Therefore, TCP introduces a window called an Advertised Window to represent the amount of free space remaining in its buffer.

$$\text{AdvertisedWindow} = \text{MaxRcvdBuffer} - ((\text{NextByteExpected} - 1) - \text{LastByteRead})$$

The advertised window potentially shrinks by incrementing the LastByteRcvd (It means moves to right).

If the local process is reading the data as fast as it arrives, then the advertised window stays open. In this case,

$$\text{AdvertisedWindow} = \text{MaxRcvdBuffer}$$

If the receiving process falls behind, then the advertised window grows smaller with every segment that arrives, until it eventually goes to 0.

TCP on the sender side must adhere to the advertised window it gets from the receiver. It means at any given time, the sender side must ensure that

$$\text{LastByteSent} - \text{LastByteAked} \leq \text{AdvertisedWindow}$$

The effective window size is used to define the amount of data can be send by the sender. The effective Window size must be greater than 0 before the sender can send more data. The sender can compute the effective window size as follows;

$$\text{EffectiveWindow} = \text{AdvertisedWindow} - (\text{LastByteSent} - \text{LastByteAked})$$

The sender side must make sure that, the local application process does not overflow the send buffer. That is,

$$\text{LastByteWritten} - \text{LastByteAked} \leq \text{MaxSendBuffer}$$

If the sending process attempts to write 'X' number of bytes to TCP as,

$$(\text{LastByteWritten} - \text{LastByteAked}) + X > \text{MaxSendBuffer}$$

Then TCP blocks the sending process and does not allow it to generate more data. When the receiver side has advertised the window size of 0, the sender is not permitted to send any more data. TCP on the receiver side does not suddenly send non data segments. It sends the non data segments in response to an arriving data segments only.

## 4.6. RETRANSMISSION

TCP guarantees the reliable delivery of data. So it retransmits each segment if an ACK is not received in a certain period of time. TCP sets this timeout as a function of the RTT it expects between the two ends of the connection. In today's world the following things are not possible to calculate the RTT.

- (i) The range of possible RTTs between any pair of hosts in the Internet
- (ii) The variation in RTT between the same two hosts over time
- (iii) Choosing an appropriate timeout value is not that easy.

To overcome this problem, TCP uses an adaptive retransmission mechanism.

### 4.6.1 Original Algorithm

This is a simple algorithm for computing a timeout value between a pair of hosts. This algorithm can be used by any end-to-end protocol. The basic ideas behind this algorithm are;

- (i) To keep a running average of the RTT
- (ii) Based on the running average, to compute the timeout as a function of this RTT.

Whenever TCP sends a data segment, it records the time. When an ACK for that segment arrives, TCP reads the time again. TCP takes the difference between these two times as a SampleRTT.

With the help of SampleRTT, an EstimatedRTT can be computed as follows. It is a weighted average between the previous estimate and this new sample.

$$\text{EstimatedRTT} = \alpha \times \text{EstimatedRTT} + (1 - \alpha) \times \text{SampleRTT}$$

- The parameter  $\alpha$  is selected to smooth the EstimatedRTT.
- A small  $\alpha \rightarrow$  follow changes in the RTT but influenced by temporary fluctuations.
- A large  $\alpha \rightarrow$  stable but not quick enough to adapt to real changes.
- Recommended range of  $\alpha \rightarrow$  between 0.8 and 0.9.

TCP then uses EstimatedRTT to compute the timeout value as follows.

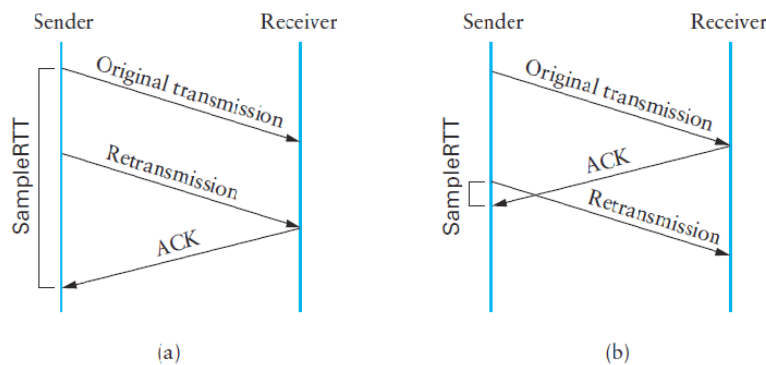
$$\text{TimeOut} = 2 \times \text{EstimatedRTT}$$

The problems addressed in the simple algorithm are;

- (1) ACK does not really acknowledge a transmission. It actually acknowledges the receipt of data.
- (2) Whenever a segment is retransmitted and then an ACK arrives at the sender, it is impossible to determine if this ACK should be associated with the first or the second transmission of the segment. This may produce confusions in the measurement of sample RTT.

Consider the figure 4.20 as an example to address the issues in a simple algorithm. Here,

- (i) In figure (a), if we assume that the ACK is for the original transmission but it was really for the second, then the SampleRTT is too large.
- (ii) In figure (b), if we assume that the ACK is for the second transmission but it was actually for the first, then the SampleRTT is too small.



**Figure 4.20** The ACK with (a) original transmission versus (b) retransmission.



### 4.6.2 Karn/Partridge Algorithm

Karn/Partridge algorithm was developed to overcome the problems in the simple algorithm. The simple solution is,

- Whenever TCP retransmits a segment, it stops taking samples of the RTT.
- It only measures SampleRTT for segments that have been sent only once.

The Karn/Partridge algorithm also includes a second small change to TCP's timeout mechanism.

- Each time TCP retransmits, it sets the next timeout to be twice the last timeout.
- It is not calculated based on the last EstimatedRTT.

The Karn and Partridge proposed that TCP use exponential backoff. The motivation for using exponential backoff is simple:

- Congestion is the most likely cause of lost segments (The TCP source should not react too aggressively to a timeout).

### 4.6.3 Jacobson/Karels Algorithm

The Karn/Partridge algorithm was designed to fix some of the causes of the congestion, and although it was an improvement, the congestion was not eliminated. To overcome these issues, Jacobson and Karels proposed a more drastic change to TCP to battle congestion. The proposal is related to,

- (i) Deciding when to time out and retransmit a segment.
- (ii) How the timeout mechanism is related to congestion.
- (iii) How the network goes overloaded by unnecessarily retransmit a segment.
- (iv) An accurate timeout value is needed to imply congestion, which triggers a congestion-control mechanism.

In the Jacobson/Karels timeout computation the variance of the sample RTTs was taken into account. If the variation among samples is small, then

- The EstimatedRTT can be better trusted.
- There is no reason for multiplying this estimate by 2 to compute the timeout.

A large variance in the samples suggests that the timeout value should not be too tightly coupled to the EstimatedRTT.

In the new approach, the sender measures a new SampleRTT as before. It then folds this new sample into the timeout calculation as follows:

$$\mathbf{Difference = SampleRTT - EstimatedRTT}$$

$$\mathbf{EstimatedRTT = EstimatedRTT + (\delta \times Difference)}$$

$$\mathbf{Deviation = Deviation + \delta(|Difference| - Deviation)}$$

where  $\delta$  is a fraction between 0 and 1.

TCP then computes the timeout value as a function of both EstimatedRTT and Deviation as follows:

$$\text{TimeOut} = \mu \times \text{EstimatedRTT} + \varphi \times \text{Deviation}$$

where,  $\mu$  is typically set to 1 and  $\varphi$  is set to 4.

When the variance is small, TimeOut is close to EstimatedRTT. A large variance causes the Deviation term to dominate the calculation.

#### 4.6.4 Implementation

There are two things to be noted regarding the implementation of timeouts in TCP. They are;

- (1) It is possible to implement the calculation for EstimatedRTT and Deviation without using floating-point arithmetic. The whole calculation is scaled by  $2n$ , with  $\delta$  selected to be  $1/2n$ . This allows us to do integer arithmetic. The corresponding code is given below,

```

{
  SampleRTT -= (EstimatedRTT >> 3);
  EstimatedRTT += SampleRTT;
  if (SampleRTT < 0)
    SampleRTT = -SampleRTT;
  SampleRTT -= (Deviation >> 3);
  Deviation += SampleRTT;
  TimeOut = (EstimatedRTT >> 3) + (Deviation >> 1);
}

```

- 2) The Jacobson/Karels algorithm is only as good as the clock used to read the current time.

## 4.7. TCP CONGESTION CONTROL

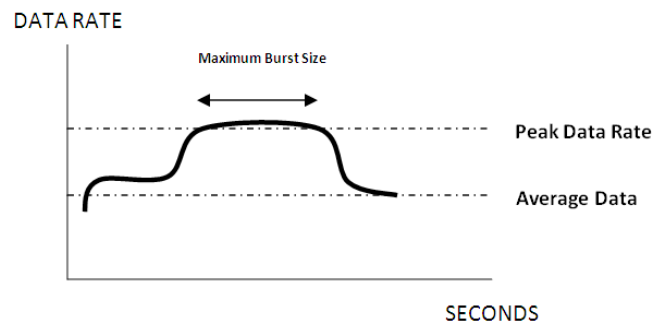
Congestion control and quality of service are closely bound together, because both of them are directly proportional to each other.

### 4.7.1 Data Traffic

The main focus of congestion control and quality of service is data traffic. In congestion control we try to avoid traffic congestion.

#### *Traffic Descriptor*

Traffic descriptors are qualitative values that represent a data flow. The figure 4.21 shows a traffic flow with some of these values.



*Figure 4.21 Traffic descriptors*

### ***Average Data Rate***

The average data rate is the number of bits sent during a period of time, divided by the number of seconds in that period. It is used to indicate the average bandwidth needed by the traffic. We use the following equation:

$$\text{Average data rate} = \text{amount of data} / \text{time}$$

### ***Peak Data Rate***

The peak data rate defines the maximum data rate of the traffic. It indicates the peak bandwidth that the network needs for traffic.

### ***Maximum Burst Size***

The maximum burst size normally refers to the maximum length of time the traffic is generated at the peak rate. It can be ignored if the duration of the peak value is very short.

### ***Effective Bandwidth***

The effective bandwidth is the bandwidth that the network needs to allocate for the flow of traffic. The effective bandwidth is a function of three values:

- (i) Average data rate
- (ii) Peak data rate
- (iii) Maximum burst size.

### ***Traffic Profiles***

A data flow can have one of the following traffic profiles;

- (i) Constant bit rate (CBR) or Fixed Rate
- (ii) Variable bit rate (VBR)
- (iii) Bursty

#### ***(i) Constant Bit Rate***

- A data rate that does not change.

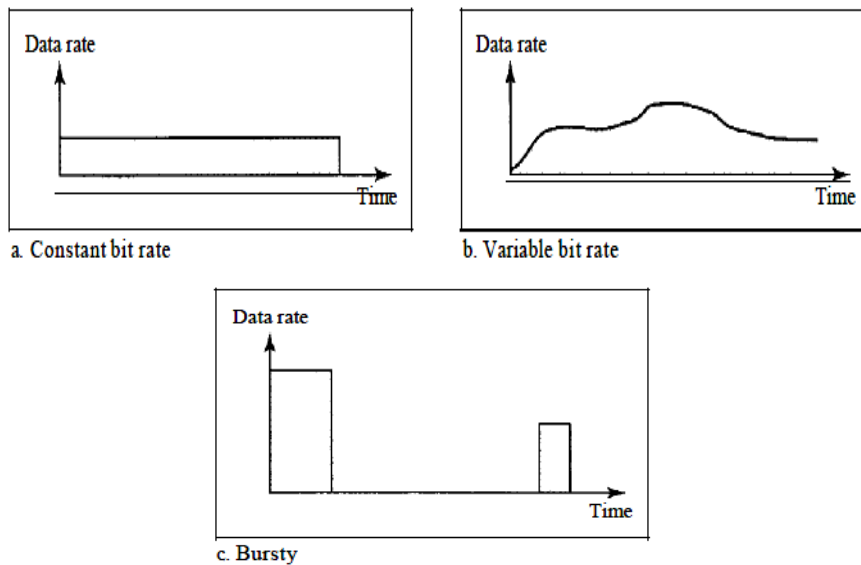
- The average data rate and the peak data rate are the same.
- The maximum burst size is not applicable.
- The network knows in advance how much bandwidth to allocate for data flow.

**(ii) Variable Bit Rate**

- The rate of the data flow changes in time (smooth changes).
- The average data rate and the peak data rate are different.
- The maximum burst size is usually a small value.
- This type of traffic is more difficult to handle

**(iii) Bursty**

- The data rate changes suddenly in a very short time.
- The average bit rate and the peak bit rate are very different values.
- The maximum burst size is significant.
- This is the most difficult type of traffic for a network to handle.
- Bursty traffic is one of the main causes of congestion in a network.



**Figure 4.22 Three traffic profiles**

**4.7.2 Congestion**

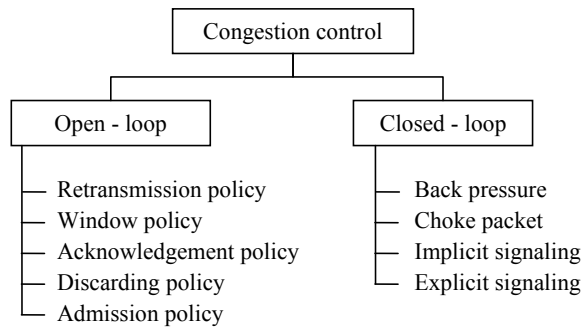
Congestion in a network may occur if the load on the network (the number of packets sent to the network) is greater than the capacity of the network (the number of packets a network can handle).

Congestion control refers to the mechanisms and techniques to control the congestion and keep the load below the capacity. Congestion control involves two factors that measure the performance of a network: delay and throughput.

### 4.7.3 Congestion Control

Congestion control refers to techniques and mechanisms that can either prevent congestion, before it happens, or remove congestion, after it has happened. In general, we can divide congestion control mechanisms into two broad categories:

- (1) Open-loop congestion control (prevention)
- (2) Closed-loop congestion control (removal)



**Figure 4.23 Congestion control categories**

#### 4.7.3.1 Open-Loop Congestion Control

In open-loop congestion control, policies are applied to prevent congestion before it happens. Congestion control is handled by either the source or the destination. The following policies are used to prevent congestion.

##### (i) Retransmission Policy

- Retransmission is sometimes unavoidable.
- If the sender feels that a sent packet is lost or corrupted, the packet needs to be retransmitted.
- Retransmission may increase congestion in the network.
- The retransmission policy and the retransmission timers must be designed to optimize efficiency and at the same time prevent congestion.

##### (ii) Window Policy

- The type of window at the sender may affect congestion.
- The Selective Repeat window is better than the Go-Back-N window for congestion control.
- The Selective Repeat window tries to send the specific packets that have been lost or corrupted.

##### (iii) Acknowledgment Policy

- The acknowledgment policy imposed by the receiver may affect congestion.
- If the receiver does not acknowledge every packet it receives, it may slow down the sender and help prevent congestion.

- A receiver may send an acknowledgment only if it has a packet to be sent or a special timer expires.
- A receiver may decide to acknowledge only N packets at a time.
- Sending fewer acknowledgments means imposing fewer loads on the network.

**(iv) Discarding Policy**

- A good discarding policy by the routers may prevent congestion.

**(v) Admission Policy**

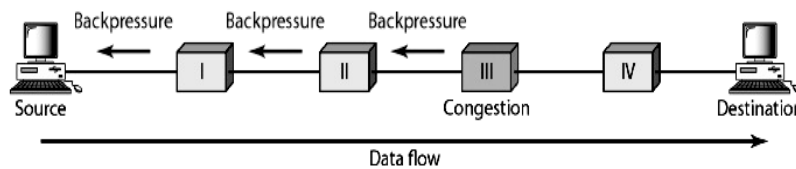
- It is a quality-of-service mechanism used to prevent congestion.
- Switches in a flow first check the resource requirement of a flow before admitting it to the network.

**4.7.3.2 Closed-Loop Congestion Control**

Closed-loop congestion control mechanisms try to alleviate congestion after it happens. Some of the mechanisms have been listed below.

**(i) Backpressure**

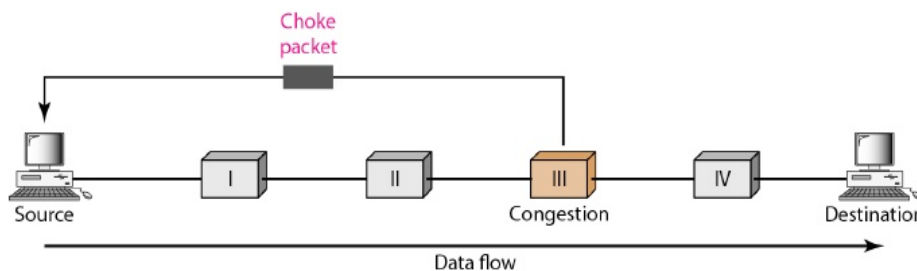
- It refers to a congestion control mechanism in which a congested node stops receiving data from the immediate upstream node or nodes.
- Backpressure is a node-to-node congestion control that starts with a node and propagates, in the opposite direction of data flow, to the source.



**Figure 4.24 Backpressure method for alleviating congestion**

**(ii) Choke Packet**

- A choke packet is a packet sent by a node to the source to inform it of congestion.
- In this method, the warning is from the router, which has encountered congestion, to the source station directly.
- The intermediate nodes through which the packet has traveled are not warned.



**Figure 4.25 Choke packet**

**(iii) Implicit Signaling**

- In implicit signaling, there is no communication between the congested node or nodes and the source.
- The source guesses that there is a congestion somewhere in the network from other symptoms.
- For example, the delay in receiving an acknowledgment is interpreted as congestion in the network; the source should slow down.

**(iv) Explicit Signaling**

- The node that experiences congestion can explicitly send a signal to the source or destination.
- In explicit signaling method, the signal is included in the packets that carry data.
- Explicit Signaling can occur in either the forward or the backward direction.

**(v) Forward Signaling**

- A bit can be set in a packet moving in the direction of the congestion.
- This bit can warn the destination that there is congestion.
- The receiver in this case can use policies, such as slowing down the acknowledgments, to alleviate the congestion.

**(vi) Backward Signaling**

- A bit can be set in a packet moving in the direction opposite to the congestion.
- This bit can warn the source that there is congestion and that it needs to slow down to avoid the discarding of packets.

**4.7.4 An Example for Congestion Control**

To better understand the concept of congestion control, let us give two examples: one in TCP and the other in Frame Relay.

**4.7.4.1 Congestion Control in TCP**

We now show how TCP uses congestion control to avoid congestion or alleviate congestion in the network.

***Congestion Window***

Normally, the sender window size is determined by the available buffer space in the receiver (rwnd). In addition to the receiver, the network is a second entity that determines the size of the sender's window. Today, the sender's window size is determined not only by the receiver but also by congestion in the network. The sender has two pieces of information;

- (i) The receiver-advertised window size
- (ii) The congestion window size

The actual size of the window is the minimum of these two. That is,

$$\text{Actual window size} = \text{minimum (rwnd, cwnd)}$$

**Congestion Policy**

The policy for handling TCP congestion is based on three phases;

- (i) Slow start
- (ii) Congestion avoidance
- (iii) Congestion detection

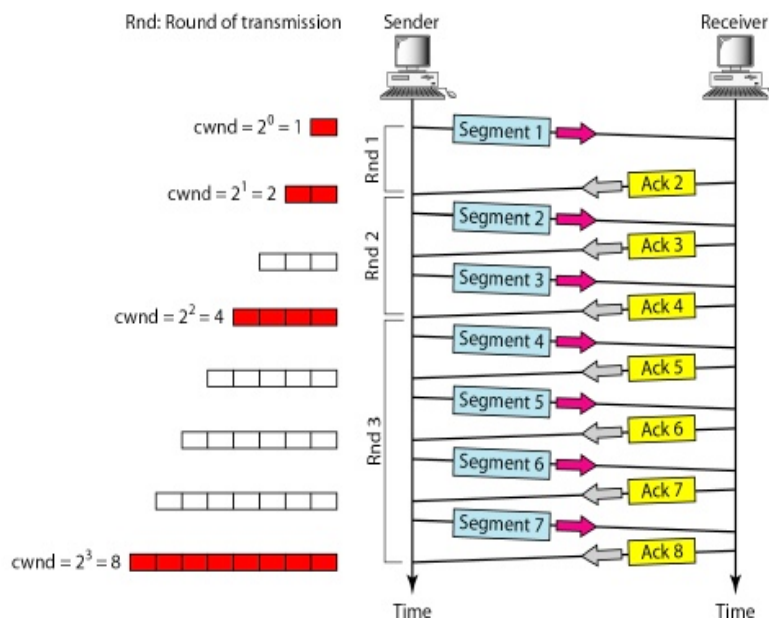
**(i) Slow-start phase**

- The sender starts with a very slow rate of transmission, but increases the rate rapidly to reach a threshold.
- When the threshold is reached, the data rate is reduced to avoid congestion.
- If congestion is detected, the sender goes back to the slow-start or congestion avoidance phase based.

**Slow Start: Exponential Increase**

- The idea behind this algorithm is that, the size of the congestion window (cwnd) starts with a maximum segment size (MSS).
- The size of the window increases one MSS each time an acknowledgment is received.
- The window starts slowly, but grows exponentially ( $2^n$ ).

In the figure 4.26, we have used segment numbers instead of byte numbers (Each segment contains only 1 byte). The sender starts with  $cwnd = 1$  MSS. This means that the sender can send only one segment. After receipt of the acknowledgment for segment 1, the size of the congestion window is increased by 1, which means that  $cwnd$  is now 2. Now two more segments can be sent. When each acknowledgment is received, the size of the window is increased by 1 MSS. When all seven segments are acknowledged,  $cwnd = 8$ .

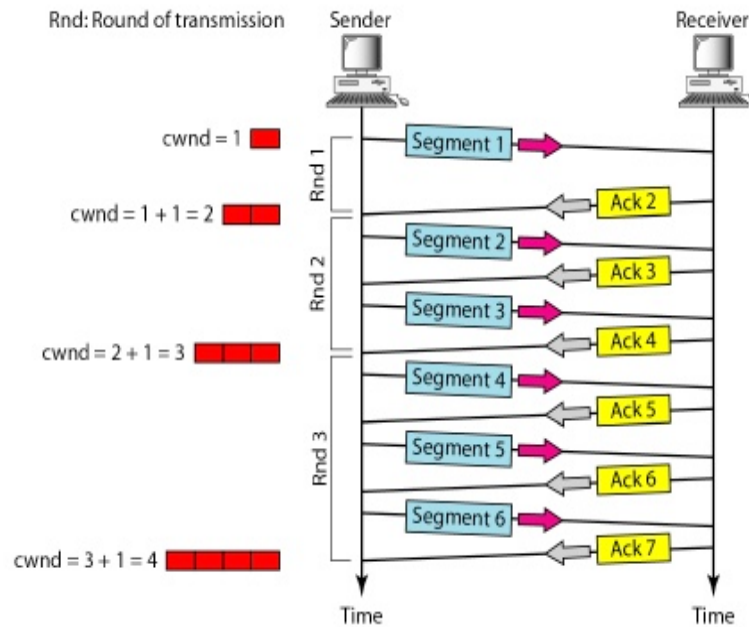


**Figure 4.26 Slow start, exponential increase**



**(ii) Congestion Avoidance (Additive Increase)**

- TCP defines another algorithm called congestion avoidance, which undergoes an additive increase instead of an exponential one to avoid congestion.



**Figure 4.27 Congestion avoidance, additive increase**

- The additive phase begins, when the size of the congestion window reaches the slow-start threshold.
- In this algorithm, each time the whole window of segments is acknowledged (one round), the size of the congestion window is increased by 1.

In this case, after the sender has received acknowledgments for a complete window size of segments, the size of the window is increased by one segment. In the above figure, the size of cwnd is increased additively.

**(iii) Congestion Detection (Multiplicative Decrease)**

If congestion occurs, the congestion window size must be decreased. The sender can guess that, congestion has occurred by the need to retransmit a segment. The retransmission can occur in one of two cases;

- When a timer times out
- When three ACKs are received.

In both cases, the size of the threshold is dropped to one-half, a multiplicative decrease. TCP implementations have two reactions:

- If a time-out occurs, there is a stronger possibility of congestion; a segment has probably been dropped in the network, and there is no news about the sent segments.

In this case TCP do the following things;

- (a) It sets the value of the threshold to one-half of the current window size.
  - (b) It sets cwnd to the size of one segment.
  - (c) It starts the slow-start phase again.
- (2) If three ACKs are received, there is a weaker possibility of congestion; a segment may have been dropped, but some segments after that may have arrived safely since three ACKs are received. This is called fast transmission and fast recovery.

In this case TCP will do the following things;

- (a) It sets the value of the threshold to one-half of the current window size.
- (b) It sets cwnd to the value of the threshold.
- (c) It starts the congestion avoidance phase.

#### 4.7.4.2 Congestion Control in Frame Relay

Congestion in a Frame Relay network decreases throughput and increases delay. The main goals of the frame relay protocol are the high throughput and low delay. Frame Relay allows the user to transmit bursty data.

##### Congestion Avoidance

To avoid the congestion, Frame Relay protocol uses 2 bits in the frame to explicitly warn the source and the destination of the presence of congestion. They are;

- (i) BECN
- (ii) FECN.

##### (i) Backward Explicit Congestion Notification (BECN)

- The BECN bit warns the sender of congestion in the network.
- Two methods are used to warn the sender;
  - (1) The switch can use response frames from the receiver.
  - (2) The switch can use a predefined connection to send special frames.
- The sender can respond to this warning by simply reducing the data rate.

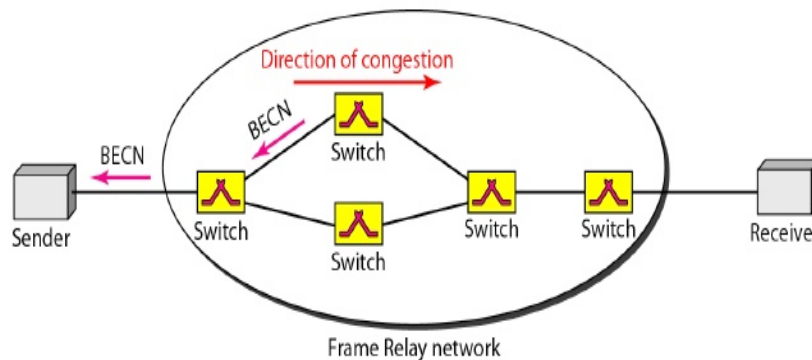
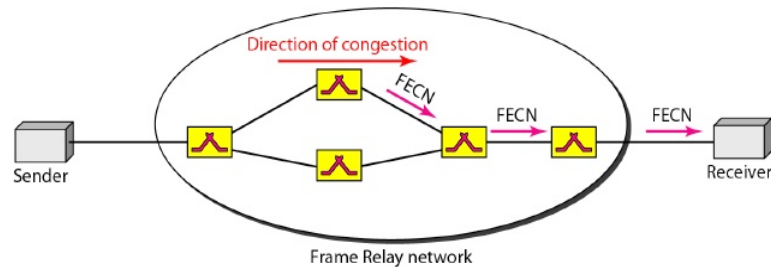


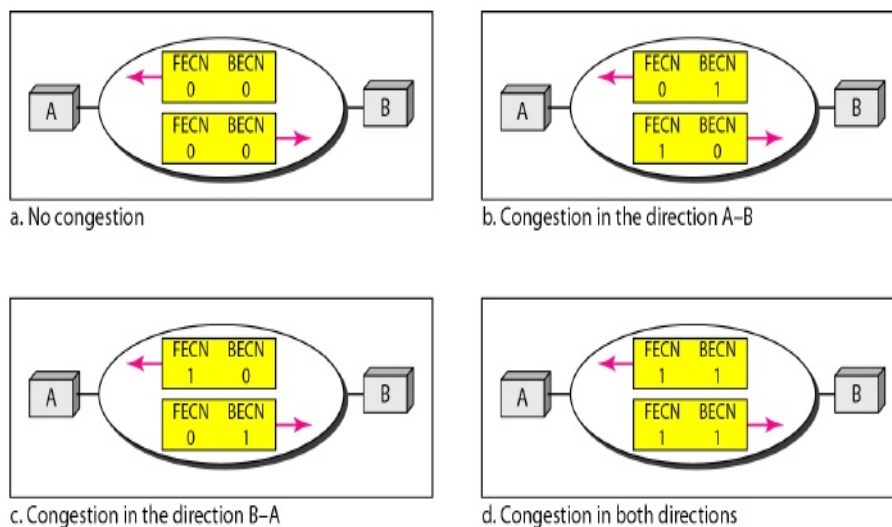
Figure 4.28 Use of BECN

**(ii) Forward Explicit Congestion Notification (FECN)**

- The FECN bit is used to warn the receiver of congestion in the network.
- The receiver can respond to this warning by communicating with the sender.
- The receiver can delay the acknowledgment, thus forcing the sender to slow down.

**Figure 4.29 Use of FECN**

When two endpoints are communicating using a Frame Relay network, four situations may occur with regard to congestion. Below figure shows these four situations and the values of FECN and BECN.

**Figure 4.30 Four cases of congestion****4.8 CONGESTION AVOIDANCE**

Congestion avoidance means, to predict when congestion is about to happen and then to reduce the rate at which hosts send data just before packets start being discarded.

In TCP, three different congestion-avoidance mechanisms are used. The first two mechanisms put a small amount of additional functionality into the router to assist the end node in the anticipation of congestion. The third mechanism attempts to avoid congestion purely from the end nodes.

### 4.8.1 DEC BIT

- It was the first congestion avoidance mechanism developed for the Digital Network Architecture (DNA).
- DNA is a connectionless network with a connection-oriented transport protocol.
- This mechanism can be applied for TCP and IP.

#### *Working principle*

- The responsibility for congestion control is evenly split between the routers and the end nodes.
- Each router monitors the load it is experiencing and explicitly notifies the end nodes when congestion is about to occur.
- This notification is done by a binary congestion bit in the packets that flow through the router.
- This binary congestion bit is named as DEC bit.
- The destination host then copies this congestion bit into the ACK and sends it back to the source.
- After the reception of ACK from the receiver, the source adjusts its sending rate to avoid congestion.

### 4.8.2 Random Early Detection (RED)

- Random early detection is a second congestion avoidance mechanism.
- RED is similar to the DEC bit scheme.
- In RED, each router is programmed to monitor its own queue length, and when it detects that congestion is imminent, to notify the source to adjust its congestion window.
- RED differs from the DEC bit scheme in two major ways.
  - (i) Instead of sending a congestion notification message explicitly to the source, RED implicitly notifies the source of congestion by dropping one of its packets. The source is effectively notified by the subsequent timeout or duplicate ACK.
  - (ii) The second difference between RED and DEC bit is, how RED decides when to drop a packet and what packet it decides to drop. RED is not waiting for the FIFO queue to become completely full and then be forced to drop each arriving packet. Instead, RED drops each arriving packet with some drop probability whenever the queue length exceeds some drop level. This idea is called early **random drop**.

#### *4.8.2.1 The RED algorithm*

The RED algorithm defines the details of how to monitor the queue length and when to drop a packet. The RED algorithm is originally proposed by Floyd and Jacobson.

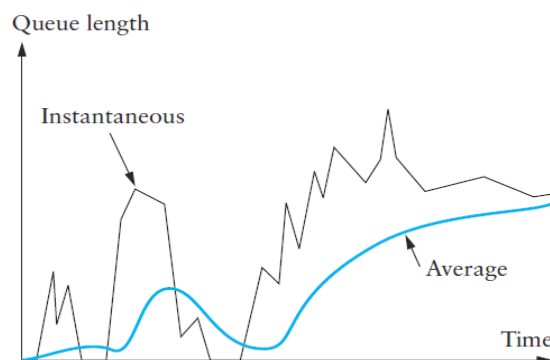
First, RED computes an average queue length using a weighted running average. That is, AvgLen is computed as,

$$\text{AvgLen} = (1 - \text{Weight}) \times \text{AvgLen} + \text{Weight} \times \text{SampleLen}$$

Where,  $0 < \text{Weight} < 1$

SampleLen is the length of the queue when a sample measurement is made. In software implementations, the queue length is measured every time a new packet arrives at the gateway. In hardware implementations, the queue length is measured at fixed interval.

- An average queue is used to capture the notion of congestion accurately.
- If a queue is spending most of its time empty, then it can not be concluded that the router is congested and tell the hosts to slow down.
- The weighted running average calculation tries to detect long-lived congestion.



**Figure 4.31** Weighted running average queue length.

In the above figure, weighted running average is indicated in the right-hand portion of figure.

The RED has two queue length thresholds that trigger certain activity. They are;

- (i) MinThreshold
- (ii) MaxThreshold.

When a packet arrives at the gateway, RED compares the current AvgLen with these two thresholds, according to the following rules:

- (1) If the average queue length is smaller than the lower threshold, no action is taken.

**if AvgLen  $\leq$  MinThreshold**

- **queue the packet**

- (2) If the average queue length is between the two thresholds, then the newly arriving packet is dropped with some probability P.

**if MinThreshold < AvgLen < MaxThreshold**

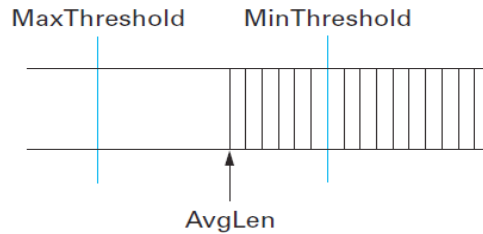
- **calculate probability P**
- **drop the arriving packet with probability P**

- (3) If the average queue length is larger than the upper threshold, then the packet is always dropped.

**if  $\text{MaxThreshold} \leq \text{AvgLen}$**

- **drop the arriving packet**

Rule 2 is depicted in the below figure 6.16. The approximate relationship between P and AvgLen is shown here.



**Figure 4.32 RED thresholds on a FIFO queue.**

### 4.8.3 Source-Based Congestion Avoidance

Source-Based Congestion Avoidance is used for detecting the initial stages of congestion, before losses occur from the end hosts. Various algorithms listed below are used for Source-Based Congestion Avoidance.

**Algorithm 1**

- The congestion window normally increases.
- Every two round-trip delays, the algorithm checks that if the current RTT is greater than the average of the minimum and maximum RTTs.
- If it is, then the algorithm decreases the congestion window by one-eighth.

**Algorithm 2**

- This algorithm works similar to algorithm 1.
- Unlike the algorithm 1, the window is adjusted once every two round-trip delays based on the product,
 
$$(\text{CurrentWindow} - \text{OldWindow}) \times (\text{CurrentRTT} - \text{OldRTT})$$
- If the result is positive, the source decreases the window size by one-eighth.
- If the result is negative or zero, the source increases the window by one maximum packet size.

**Algorithm 3**

- For every RTT, it increases the window size is increased by one packet and compares the throughput achieved to the throughput when the window was one packet smaller.
- If the difference is less than one-half, the algorithm decreases the window by one packet.
- The throughput calculates by dividing the number of bytes outstanding in the network by the RTT.

**Algorithm 4**

- This algorithm works based on the changes in the throughput rate (changes in the sending rate).
- It calculates throughput based on the change in the slope of the throughput.
- It compares the measured throughput rate with an expected throughput rate.
- The algorithm is also called as TCP Vegas.
- It is not widely deployed in the Internet.

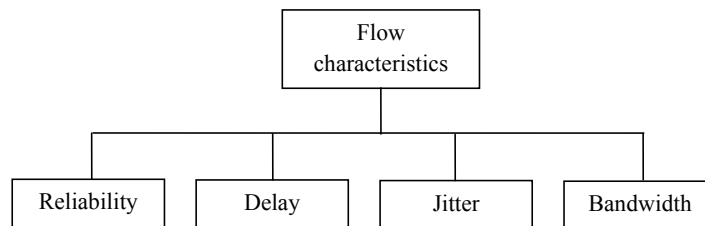
**4.9. QUALITY OF SERVICE**

Quality of service (QoS) can be defined as something a flow seeks to attain. It is a set of attributes related to the performance of the connection.

**4.9.1 Flow Characteristics**

Four types of characteristics are attributed to a flow. They are;

- (i) Reliability
- (ii) Delay
- (iii) Jitter
- (iv) Bandwidth



**Figure 4.33 Flow characteristics**

**(i) Reliability**

- Reliability is a characteristic needed by the data flow.
- Loss of packet and acknowledgement may happen, if the reliability is lack, which leads to retransmission.
- Electronic mail, file transfer, and Internet access have reliable transmissions than telephony or audio conferencing.

**(ii) Delay**

- It is a delay between the source and destination during the data transmission.
- An application can accept the delay in different degrees.
- Minimum delay is acceptable in telephony, audio conferencing, video conferencing, and remote login.
- Delay is not considered as a major fact in file transfer or e-mail.

**(iii) Jitter**

- Jitter is defined as the variation in delay for packets belonging to the same flow.
- High jitter means the difference between delays is large.
- Low jitter means the variation is small.
- If the jitter is high, some action is needed in order to use the received data.

**Example:**

*Case 1: If four packets depart at times 0, 1, 2, 3 and arrive at 20, 21, 22, 23, all have the same delay, 20 units of time.*

*Case 2: If the above four packets arrive at 21, 23, 21, and 28, they will have different delays: 21, 22, 19, and 24.*

*In audio and video, the first case is completely acceptable and the second case is not.*

**(iv) Bandwidth**

- Bandwidth refers to the capacity of the channel.
- Different applications need different bandwidths.
- In video conferencing, we need to send millions of bits per second.
- In an e-mail service it may not reach even a million.

**Flow Classes**

- Based on the flow characteristics, we can classify flows into groups.
- Each group is having a similar level of characteristics.
- This categorization is not formal or universal.

**4.9.2 Techniques to Improve Qos**

The following four techniques can be used to improve the quality of service. They are;

- (i) Scheduling
- (ii) Traffic shaping
- (iii) Admission control
- (iv) Resource reservation

**4.9.2.1 Scheduling**

Packets from different flows arrive at a switch or router for processing. A good scheduling technique treats the different flows in a fair and appropriate manner. Three types of scheduling techniques are designed to improve the quality of service. They are;

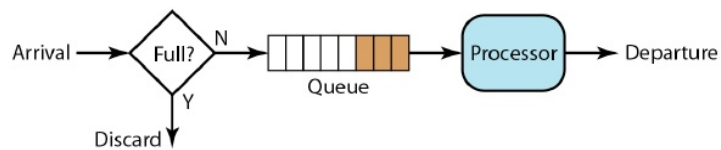
- (1) FIFO queuing
- (2) Priority queuing



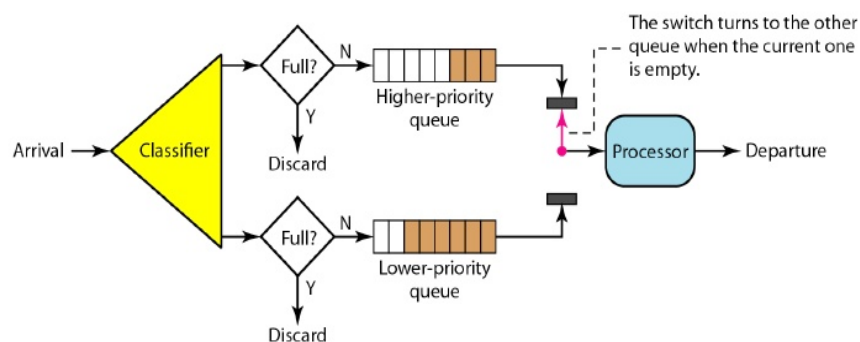
## (3) Weighted fair queuing

**(1) FIFO Queuing**

- In first-in, first-out (FIFO) queuing, packets wait in a buffer (queue) until the node (router or switch) is ready to process them.
- If the average arrival rate is higher than the average processing rate, the queue will fill up and new packets will be discarded.
- The figure 4.34 shows a conceptual view of a FIFO queue.

**Figure 4.34 FIFO queue****(2) Priority Queuing**

- In priority queuing, packets are first assigned to a priority class.
- Each priority class has its own queue.
- The packets in the highest-priority queue are processed first.
- The packets in the lowest-priority queue are processed last.
- The system does not stop serving a queue until it is empty.
- **Advantage:** A priority queue can provide better QoS than the FIFO queue.
- **Disadvantage:** If there is a continuous flow in a high-priority queue, the packets in the lower-priority queues will never have a chance to be processed (called starvation).
- The figure 4.34 shows priority queuing with two priority levels.

**Figure 4.35 Priority queuing****(3) Weighted Fair Queuing**

- This is a better scheduling method.
- Here, the packets are assigned to different classes and admitted to different queues.

- The queues are weighted based on the priority of the queues (higher priority means a higher weight).
- The system processes packets in each queue in a round-robin fashion.
- The number of packets selected from each queue is based on the corresponding weight.
- For example, if the weights are 3, 2, and 1, three packets are processed from the first queue, two from the second queue, and one from the third queue.

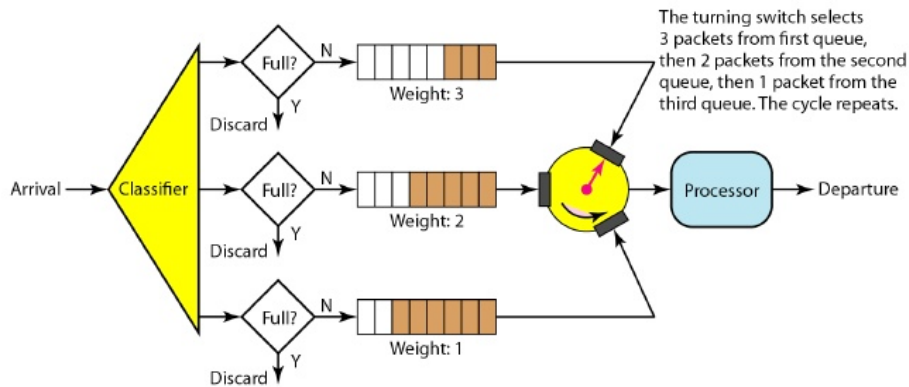


Figure 4.36 Weighted fair queuing

#### 4.9.2.2 Traffic Shaping

Traffic shaping is a mechanism to control the amount and the rate of the traffic sent to the network. Two techniques are used to shape the traffic. They are;

- (1) Leaky bucket
- (2) Token bucket

##### (1) Leaky Bucket

- Basic idea behind this technique is, the input rate can vary but the output rate remains constant.
- In networking, a leaky bucket technique can be used to smooth out bursty traffic.

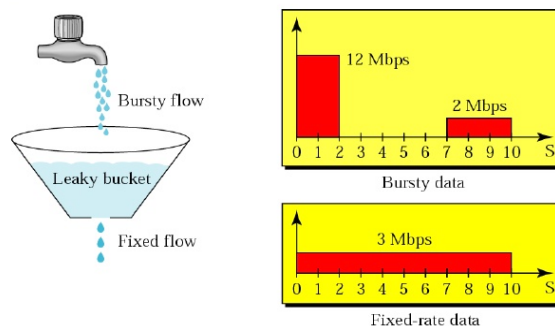


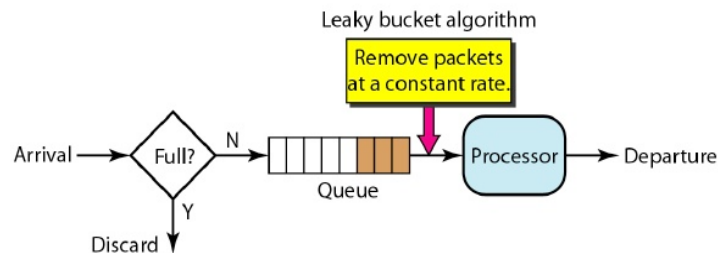
Figure 4.37 Leaky bucket

- Bursty chunks are stored in the bucket and sent out at an average rate.

- The figure 4.37 shows a leaky bucket and its effects.
- Leaky bucket may prevent congestion.

### **Leaky bucket implementation**

- A FIFO queue holds the packets.
- If the traffic consists of fixed-size packets the process removes a fixed number of packets from the queue at each tick of the clock.



**Figure 4.38 Leaky bucket implementation**

- If the traffic consists of variable-length packets, the fixed output rate must be based on the number of bytes or bits. The following is an algorithm for variable-length packets;
  - (i) Initialize a counter to  $n$  at the tick of the clock.
  - (ii) If  $n$  is greater than the size of the packet, send the packet and decrement the counter by the packet size. Repeat this step until  $n$  is smaller than the packet size.
  - (iii) Reset the counter and go to step 1.
- A leaky bucket algorithm shapes bursty traffic into fixed-rate traffic by averaging the data rate.
- It may drop the packets if the bucket is full.

### **Drawbacks of Leaky Bucket Algorithm**

- The leaky bucket is very restrictive.
- It does not credit an idle host.
- For example, if a host is not sending for a while, its bucket becomes empty.
- Now if the host has bursty data, the leaky bucket allows only an average rate.
- The time when the host was idle is not taken into account.

### **2) Token Bucket**

- The token bucket algorithm allows idle hosts to accumulate credit for the future in the form of tokens.
- For each tick of the clock, the system sends  $n$  tokens to the bucket (For example, if  $n$  is 100 and the host is idle for 100 ticks, the bucket collects 10,000 tokens).

- The system removes one token for every byte (cell) of data sent.
- The host can send bursty data as long as the bucket is not empty (Now the host can consume all these tokens in one tick with 10,000 cells, or the host takes 1000 ticks with 10 cells per tick).

### Token Bucket Implementation

- The token bucket can easily be implemented with a counter.
- The token is initialized to zero.
- Each time a token is added, the counter is incremented by 1.
- Each time a unit of data is sent, the counter is decremented by 1.
- When the counter is zero, the host cannot send data.
- The token bucket allows bursty traffic at a regulated maximum rate.

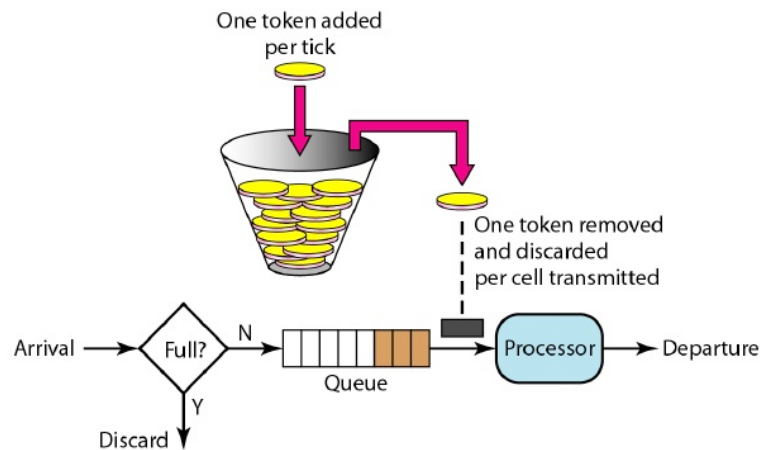


Figure 4.39 Token bucket

### Combining Token Bucket and Leaky Bucket

- The two techniques can be combined to credit an idle host and at the same time regulate the traffic.
- The leaky bucket is applied after the token bucket.
- The rate of the leaky bucket needs to be higher than the rate of tokens dropped in the bucket.

### 4.9.2.3 Resource Reservation

- A flow of data needs resources such as a buffer, bandwidth, CPU time, and so on.
- The quality of service is improved if these resources are reserved beforehand.
- The QoS model called Integrated Services, which depends heavily on resource reservation to improve the quality of service.

#### 4.9.2.4 *Admission Control*

- Admission control refers to the mechanism used by a router, or a switch, to accept or reject a flow based on flow specifications.
- Before a router accepts a flow for processing, it checks the flow specifications to see if its capacity (in terms of bandwidth, buffer size, CPU speed, etc.) and its previous commitments to other flows can handle the new flow.

#### 4.9.3 *Integrated Services*

Two models have been designed to provide quality of service in the Internet. They are;

(i) Integrated Services

(ii) Differentiated Services.

- Both models emphasize the use of quality of service at the network (IP) and data link.
- IP was originally designed for best-effort delivery.
- Integrated Services are called as IntServ.
- It is a flow-based QoS model (A user needs to create a flow, a kind of virtual circuit, from the source to the destination and inform all routers of the resource requirement).

#### *Signaling*

- To implement a flow-based model over a connectionless protocol, we must use a signaling protocol.
- A signaling protocol can run over IP and provides the signaling mechanism for making a reservation.
- This signaling protocol is called Resource Reservation Protocol (**RSVP**).

#### *Flow Specification*

- When a source makes a reservation, it needs to define a flow specification.
- A flow specification has two parts: (i) **Rspec** (resource specification) and (ii) **Tspec** (traffic specification).
- Rspec defines the resource that the flow needs to reserve (buffer, bandwidth, etc.).
- Tspec defines the traffic characterization of the flow.

#### *Admission*

- After a router receives the flow specification, it decides to admit or deny the service.
- The decision is based on the previous commitments of the router and the current availability of the resource.

#### *Service Classes*

Two classes of services have been defined for Integrated Services:

(i) Guaranteed service

(ii) Controlled-load service.

**(i) Guaranteed Service Class**

- It is designed for real-time traffic that needs a minimum end-to-end delay.
- The end-to-end delay is the sum of the delays in the routers, the propagation delay in the media, and the setup mechanism.
- The guaranteed services are quantitative services, in which the amount of end-to-end delay and the data rate must be defined by the application.

**(ii) Controlled-Load Service Class**

- It is designed for applications that can accept some delays.
- Example applications: File transfer, e-mail, and Internet access.
- The controlled load service is a qualitative type of service in that the application requests the possibility of low-loss or no-loss packets.

**RSVP**

The Resource Reservation Protocol (RSVP) is a signaling protocol to help IP for create a flow and consequently make a resource reservation.

**Multicast Trees**

- RSVP is designed for multicasting.
- It can be also used for unicasting because unicasting is just a special case of multicasting with only one member in the multicast group.

**Receiver-Based Reservation**

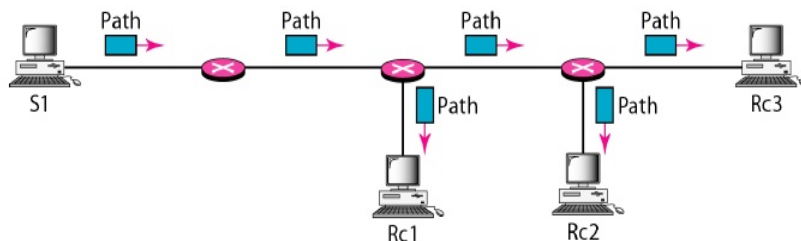
- In RSVP, reservation is done by the receivers, not the sender.
- For example, in multicast routing protocols, the receivers, not the sender, make a decision to join or leave a multicast group.

**RSVP Messages**

RSVP has several types of messages. Two of them are;

- (i) Path
- (ii) Resv

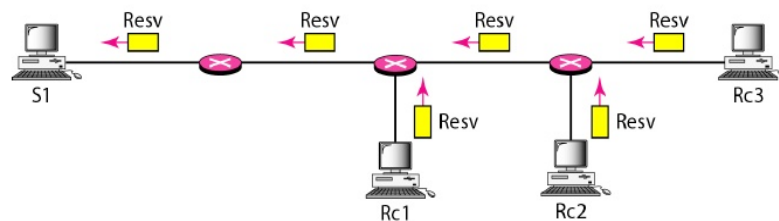
**(i) Path Messages:** A Path message travels from the sender and reaches all receivers in the multicast path. Path message stores the necessary information for the receivers. A Path message is sent in a multicast environment. A new message is created when the path diverges.



**Figure 4.40 Path messages**

**(ii) Resv Messages**

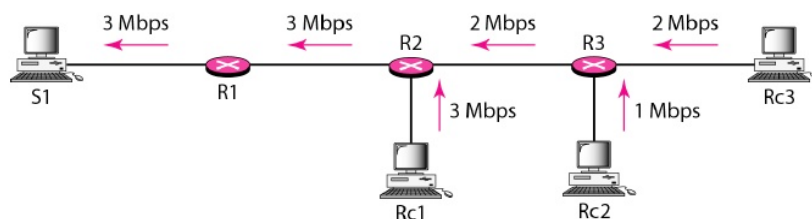
After a receiver has received a Path message, it sends a Resv message. The Resv message travels toward the sender and makes a resource reservation on the routers that support RSVP. If a router does not support RSVP on the path, it routes the packet based on the best-effort delivery methods.



**Figure 4.41 Resv messages**

**Reservation merging**

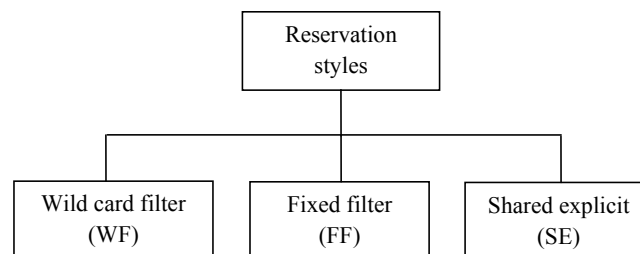
In RSVP, the resources are not reserved for each receiver in a flow, the reservation is merged. In the figure 4.42, RC3 requests a 2Mbps bandwidth while RC2 requests 1 Mbps bandwidth. Router R3, which need to make a bandwidth reservation, merges the two requests. The reservation is made for 2Mbps, the larger of the two. Because a 2 Mbps input reservation can handle both requests.



**Figure 4.42 Reservation merging**

**Reservation Styles**

When there is more than one flow, the router needs to make a reservation to accommodate all of them. RSVP defines three types of reservation styles, as shown in below figure.



**Figure 4.43 Reservation styles**

- (i) **Wild Card Filter Style:** The router creates a single reservation for all senders. The reservation is based on the largest request.



(ii) **Fixed Filter Style:** The router creates a distinct reservation for each flow. This means that if there are n flows, n different reservations are made.

(iii) **Shared Explicit Style:** The router creates a single reservation which can be shared by a set of flows.

**Soft State**

The reservation information (state) stored in every node for a flow needs to be refreshed periodically. It is referred to as a soft state. The default interval for refreshing is currently 30 s.

**Problems with Integrated Services**

The two problems mentioned below with Integrated Services may prevent its full implementation in the Internet. They are;

- (i) Scalability
- (ii) Service-Type Limitation

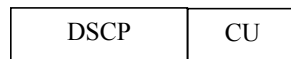
**4.9.4 Differentiated Services (DS)**

Differentiated Services (Diffserv) was introduced to handle the shortcomings of Integrated Services. Differentiated Services is a class-based QoS model designed for IP. Two fundamental changes were made;

- (1) The routers do not have to store information about flows. This solves the scalability problem. The applications, or hosts, define the type of service they need each time they send a packet.
- (2) The per-flow service is changed to per-class service. The router routes the packet based on the class of service defined in the packet, not the flow. This solves the service-type limitation problem. We can define different types of classes based on the needs of applications.

**DS Field**

- In Diffserv, each packet contains a field called the DS field.
- The value of this field is set at the boundary of the network by the host or the first router designated as the boundary router.
- The DS field contains two subfields: DSCP and CU.
- The DSCP (Differentiated Services Code Point) is a 6-bit subfield that defines the per-hop behavior (PHB).
- The 2-bit CU (currently unused) subfield is not currently used.



**Figure 4.44 DS field**

**Per-Hop Behavior**

The Diffserv model defines three PHBs. They are;

- (i) DE PHB (default PHB): Provides best-effort delivery.



(ii) EF PHB (expedited forwarding PHB): Provides Low loss, Low latency, Ensured bandwidth.

(iii) AF PHB (assured forwarding PHB): Delivers the packet with a high assurance.

### Traffic Conditioner

To implement Diffserv, the OS node uses traffic conditioners such as meters, markers, shapers and droppers.

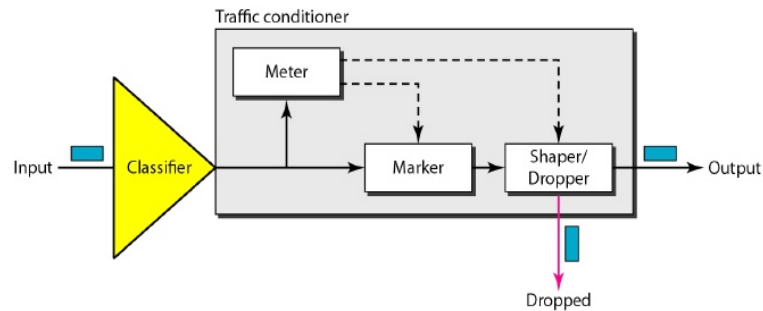


Figure 4.45 Traffic conditioner

- (i) **Meters:** It checks that, if the incoming flow matches the negotiated traffic profile. The meter can use the tools such as a token bucket to check the profile.
- (ii) **Marker:** A marker can remark a packet or down-mark a packet based on information received from the meter. Down marking occurs if the flow does not match the profile. A marker does not up-mark a packet.
- (iii) **Shaper:** A shaper uses the information received from the meter to reshape the traffic if it is not compliant with the negotiated profile.
- (iv) **Dropper:** A dropper, which works as a shaper with no buffer, discards packets if the flow severely violates the negotiated profile.

## 4.9.5 QoS in Switched Networks

QoS can be used in two switched networks: Frame Relay and ATM. These two networks are virtual-circuit networks that need a signaling protocol such as RSVP.

### 4.9.5.1 QoS in Frame Relay

Four different attributes to control traffic have been devised in Frame Relay. They are;

- (i) Access rate
- (ii) Committed burst size
- (iii) Committed information rate (CIR)
- (iv) Excess burst size
  - These are set during the negotiation between the user and the network.
  - For PVC connections, they are negotiated once.

- For SVC connections, they are negotiated for each connection during connection setup.
  - (i) **Access Rate:** It is defined for every connection. The access rate actually depends on the bandwidth of the channel connecting the user to the network. The user can never exceed this rate.
  - (ii) **Committed Burst Size:** Frame Relay defines a committed burst size  $B_e$ . This is the maximum number of bits in a predefined time that the network is committed to transfer without discarding any frame.
  - (iii) **Committed Information Rate:** It defines an average rate in bits per second. If the user follows this rate continuously, the network is committed to deliver the frames.
  - (iv) **Excess Burst Size:** This is the maximum number of bits in excess of  $B_e$  that a user can send during a predefined time. The network is committed to transfer these bits if there is no congestion.

#### 4.9.5.2 QoS in ATM

The QoS in ATM is based on the following three things.

- (i) Class
- (ii) User-related attributes
- (iii) Network-related attributes

##### (i) Classes

The ATM Forum defines four service classes: CBR, VBR, ABR, and UBR.

- **CBR (constant-bit-rate):** It is designed for customers who need real-time audio or video services.
- **VBR (variable-bit-rate):** It is divided into two subclasses: real-time (VBR-RT) and non-real-time (VBR-NRT). VBR-RT is designed for those users who need real-time services and use compression techniques to create a variable bit rate. VBR-NRT is designed for those users who do not need real-time services but use compression techniques to create a variable bit rate.

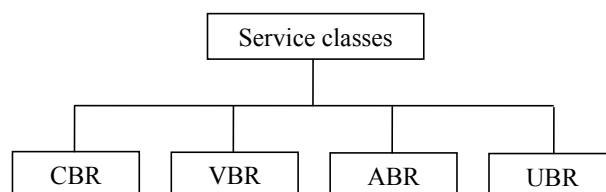
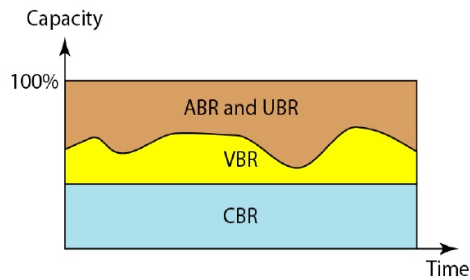


Figure 4.46 Service classes

- **ABR (available-bit-rate):** It delivers cells at a minimum rate. If more network capacity is available, this minimum rate can be exceeded. ABR is suitable for bursty transmission.
- **UBR (unspecified-bit-rate):** It is a best-effort delivery service that does not guarantee anything.



**Figure 4.47 Relationship of service classes Vs total capacity of the network**

### **(ii) User-Related Attributes**

User-related attributes are used to define, how fast the user wants to send data. These are negotiated at the time of contract between a user and a network. The following are some user-related attributes.

- **SCR (Sustained Cell Rate):** The average cell rate over a long time interval. The actual cell rate may be lower or higher than this value, but the average should be equal to or less than the SCR.
- **PCR (Peak Cell Rate):** The sender's maximum cell rate.
- **MCR (Minimum Cell Rate):** It defines the minimum cell rate acceptable to the sender.
- **CVDT (Cell Variation Delay Tolerance):** It is a measure of the variation in cell transmission times.

### **(iii) Network-Related Attributes**

The network-related attributes are those that define characteristics of the network. The following are some network-related attributes.

- **CLR (Cell Loss Ratio):** It defines the fraction of cells lost during transmission.
- **CTD (Cell Transfer Delay):** It is the average time needed for a cell to travel from source to destination. The maximum CTD and the minimum CTD are also considered as attributes.
- **CDV (Cell Delay Variation):** It is the difference between the CTD maximum and the CTD minimum.
- **CER (Cell Error Ratio):** It defines the fraction of the cells delivered in error.