RGPVNOTES.IN

Program : **B.Tech**

Subject Name: **Computer Networks**

Subject Code: **CS-602**

Semester: **6th**

Department of Computer Science and Engineering
CS602 Computer Networks
Subject Notes: UNIT-I

Syllabus: **Computer Network: Definitions, goals, components, Architecture, Classifications & Types. Layered Architecture: Protocol hierarchy, Design Issues, Interfaces and Services, Connection Oriented & Connectionless Services, Service primitives, Design issues & its functionality. ISO OSI Reference Model: Principle, Model, Descriptions of various layers and its comparison with TCP/IP. Principals of physical layer: Media, Bandwidth, Data rate and Modulations.**

Computer Network: Definition

A **computer network** is a set of **computers** connected together for the purpose of sharing resources. The most common resource shared today is connection to the Internet. Other shared resources can include a printer or a file server.

#Goals

- Several machines can share printers, tape drives, etc.
- Reduced cost
- Resource and load sharing
- Programs do not need to run on a single machine
- High reliability
- If a machine goes down, another can take over
- Mail and communication

#Components

A data communications system has five components.

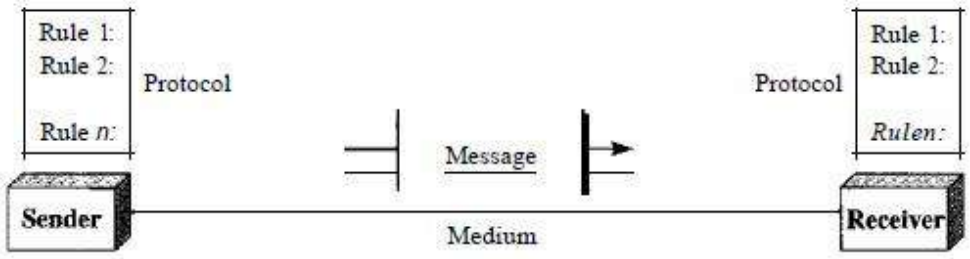


Fig. 1.1 Computer Network: Components

1. Message. The message is the information (data) to be communicated. Popular forms of information include text, numbers, pictures, audio, and video.
2. Sender. The sender is the device that sends the data message. It can be a computer, workstation, telephone handset, video camera, and so on.
3. Receiver. The receiver is the device that receives the message. It can be a computer, workstation, telephone handset, television, and so on.
4. Transmission medium. The transmission medium is the physical path by which a message travels from sender to receiver. Some examples of transmission media include twisted-pair wire, coaxial cable, fiber-optic cable, and radio waves
5. Protocol. A protocol is a set of rules that govern data communications. It represents an agreement between the communicating devices. Without a protocol, two devices may be connected but not communicating.

#Architecture

Network architecture is the design of a communications network. It is a framework for the specification of a network's physical components and their functional organization and configuration.
In telecommunication, the specification of a network architecture may also include a detailed description of products and services delivered via a communications network, as well as detailed rate and billing structures under which services are compensated. The network architecture of the Internet is predominantly expressed by its use of the Internet Protocol Suite, rather than a specific model for interconnecting networks or nodes in the network, or the usage of specific types of hardware link

#Computer Network's: Classifications & Types.

There are three types of network classification

1) LAN (Local area network)
2) MAN (Metropolitan Area network)
3) WAN (Wide area network)

Fig. 1.2 Computer Network: Classifications

1) Local area network (LAN)

LAN is a group of the computers placed in the same room, same floor, or the same building so they are connected to each other to form a single network to share their resources such as disk drives, data, CPU, modem etc. LAN is limited to some geographical area less than 2 km. Most of LAN is used widely is an Ethernet system of the bus topology.

Characteristics of LAN

LAN connects the computer in a single building, block and they are working in any limited area less than 2 km.
Media access control methods in a LAN, the bus-based Ethernet and token ring.

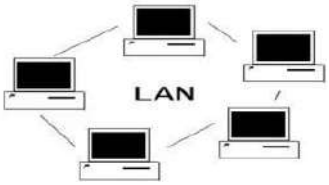


Fig. 1.3 Local area network

2) Metropolitan Area network (MAN)

The metropolitan area network is a large computer network that expands a Metropolitan area or campus. Its geographic area between a WAN and LAN, its expand round 50km devices used are modem and wire/cable.

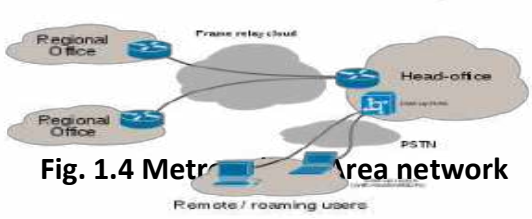


Fig. 1.4 Metr Area network

Characteristics of MAN

1) Its covers the towns and cities (50km)
2) MAN is used by the communication medium for optical fibre cables, it also used for other media.

3) Wide area Network (WAN)

The wide area network is a network which connects the countries, cities or the continents, it is a public communications links. The most popular example of a WAN is the internet. WAN is used to connect LAN so the users and the computer in the one location can communicate with each other.

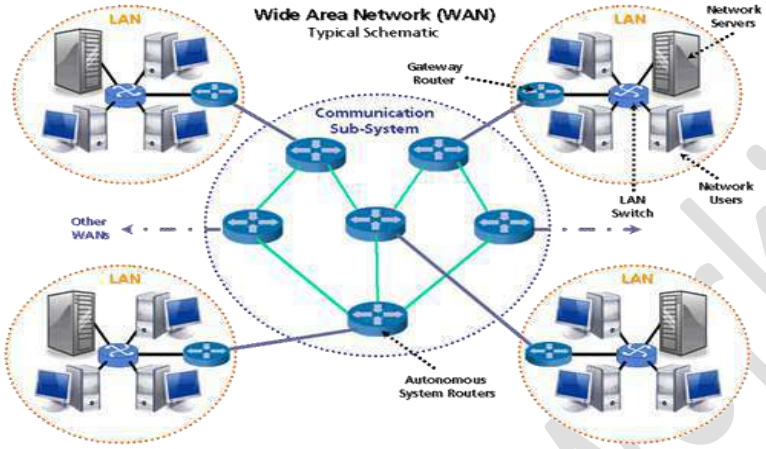


Fig. 1.5 Wide area Network

Characteristics of WAN

1) Its covers the large distances (More than 100 KM).
2) Communication medium used are satellite, telephones which are connected by the routers.

#Layered Architecture:

Protocol hierarchy: - To tackle with the design complexity most of the networks are organize as a set of layers or levels. The fundamental idea of layered architecture is to divide the design into small pieces. The layering provides modularity to the network design. The main duty of each layer is to provide offer services to higher layers, and provide abstraction. The main benefits of layered architecture are modularity and clear interfaces.

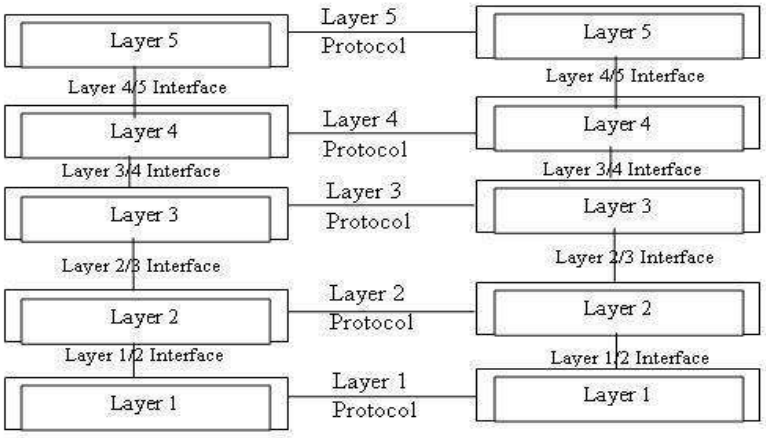


Fig. 1.6 Five Layered Network

#Design Issues:

Layered architecture in computer network design

Layered architectures have several advantages. Some of them are,

- Modularity and clear interface
- Provide flexibility to modify network services

- Ensure independence of layers
- Management of network architecture is easy
- Each layer can be analysed and tested independent of other layers

#Interfaces and Services:

The benefits to layering networking protocol specifications are many including:

Interoperability - Layering promotes greater interoperability between devices from different manufacturers and even between different generations of the same type of device from the same manufacturer.

Greater Compatibility - One of the greatest of all the benefits of using a hierarchal or layered approach to networking and communications protocols is the greater compatibility between devices, systems and networks that this delivers.

Better Flexibility - Layering and the greater compatibility that it delivers goes a long way to improving the flexibility. Particularly in terms of options and choices.

Increased Life Expectancy - Increased product working life expectancies as backwards compatibility is made considerably easier. Devices from different technology generations can co-exist thus the older units do not get discarded immediately newer technologies are adopted.

Scalability - Experience has shown that a layered or hierarchal approach to networking protocol design and implementation scales better than the horizontal approach.

Mobility - Greater mobility is more readily delivered whenever we adopt the layered and segmented strategies into our architectural design

Cost Effective Quality - The layered approach has proven time again and again to be the most economical way of developing and implementing any system be they are small, simple, large or complex makes no difference. This ease of development and implementation translates to greater efficiency and effectiveness which in turn translates into greater economic rationalization and cheaper products while not compromising quality.

Standardization and Certification - The layered approach to networking protocol specifications facilitates a more streamlined and simplified standardization and certification process; particularly from an "industry" point of view. This is due to the clearer and more distinct definition and demarcation of what functions occur at each layer when the layered approach is taken.

Rapid Application Development (RAD) - Workloads can be evenly distributed which means that multiple activities can be conducted in parallel thereby reducing the time taken to develop, debug, optimize and package new technologies ready for production implementation.

#Connection Oriented & Connectionless Services, Service primitives, Design issues & its functionality

- **Connection-oriented**

There is a sequence of operation to be followed by the users of connection-oriented service. They are:

1. Connection is established
2. Information is sent
3. Connection is released

In connection-oriented service we must establish a connection before starting the communication. When connection is established we send the message or the information. Then we release the connection.

Connection oriented service is more reliable than connectionless service. Example of connection oriented is TCP (Transmission Control Protocol) protocol.

- **Connectionless**

It is similar to postal services, as it carries the full address where the message (letter) is to be carried. Each message is routed independently from source to destination. The order of message sent can be different from the order received.

In connectionless the data is transferred in one direction from source to destination without checking that destination is still there or not or if it prepared to accept the message. Authentication is not needed in this. Example of Connectionless service is UDP (User Datagram Protocol) protocol.

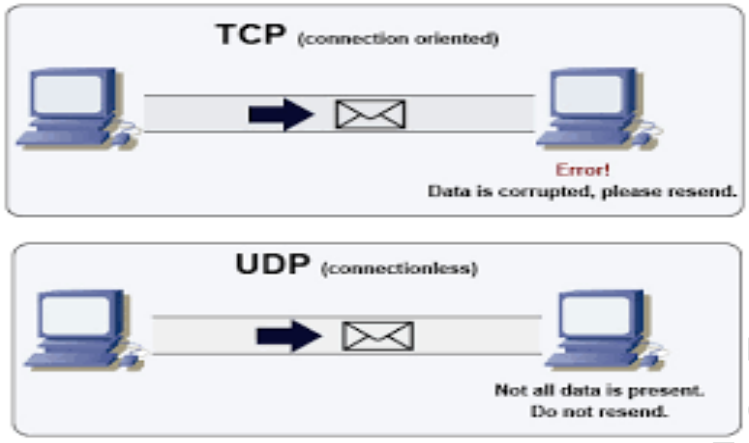


Fig. 1.7 Connection Oriented & Connectionless Services

#Service Primitives

Connection Oriented Service Primitives

LISTEN	Block waiting for an incoming connection
CONNECTION	Establish a connection with a waiting peer
RECEIVE	Block waiting for an incoming message
SEND	Sending a message to the peer
DISCONNECT	Terminate a connection

Connectionless Service Primitives

UNIDATA	This primitive sends a packet of data
FACILITY, REPORT	Primitive for enquiring about the performance of the network, like delivery statistics.

Design issues & its functionality

- **Justifying a Network**: - Some applications may be best satisfied by individual point to point connections to handle very specific communication requirements.
- **Scope**: - The scope of the network is viewed as bounded on one side by the offerings of the common carriers who provide communication facilities from which the network is built and on the other side by the application on which it is interconnected.
- Manageability:-
- **Network Architecture**: - While designing the network architecture, network may be a single homogeneous mesh comprised of a single type of node and a single type of link. Network architecture might be hierarchical network with one type link riding on another.
- **Switch Mode**: - For data transmission, different types of switching methods are possible. These are packet switching, circuit switching and hybrid switching.
- **Node Placement and sizing**: - A fundamental problem in the topological optimization of a network is the selection of the network node sites and where to place multiplexers, hubs and switch.
- **Link Topology and sizing**: - It involves selecting the specific links interconnecting nodes. At the highest level, that is where the architecture of the network is derived. Thus a hierarchy that include a backbone as well as LAN'S may be defined. It is possible to permit the backbone to be a mesh while LAN is constrained to be trees.
- **Routing**: - It involves selecting paths for each requirements. At higher level, this involves selecting the routing procedure itself.

#ISO-OSI Reference Model

#Principles of OSI Reference Model

The OSI reference model has 7 layers. The principles that were applied to arrive at the seven layers can be briefly summarized as follows:

1. A layer should be created where a different abstraction is needed.
2. Each layer should perform a well-defined function.
3. The function of each layer should be chosen with an eye toward defining internationally standardized protocols.
4. The layer boundaries should be chosen to minimize the information flow across the interfaces.
5. The number of layers should be large enough that distinct functions need not be thrown together in the same layer out of necessity and small enough that architecture does not become unwieldly.

Feature of OSI Model:

1. Big picture of communication over network is understandable through this OSI model.
2. We see how hardware and software work together.
3. We can understand new technologies as they are developed.
4. Troubleshooting is easier by separate networks.
5. Can be used to compare basic functional relationships on different networks.

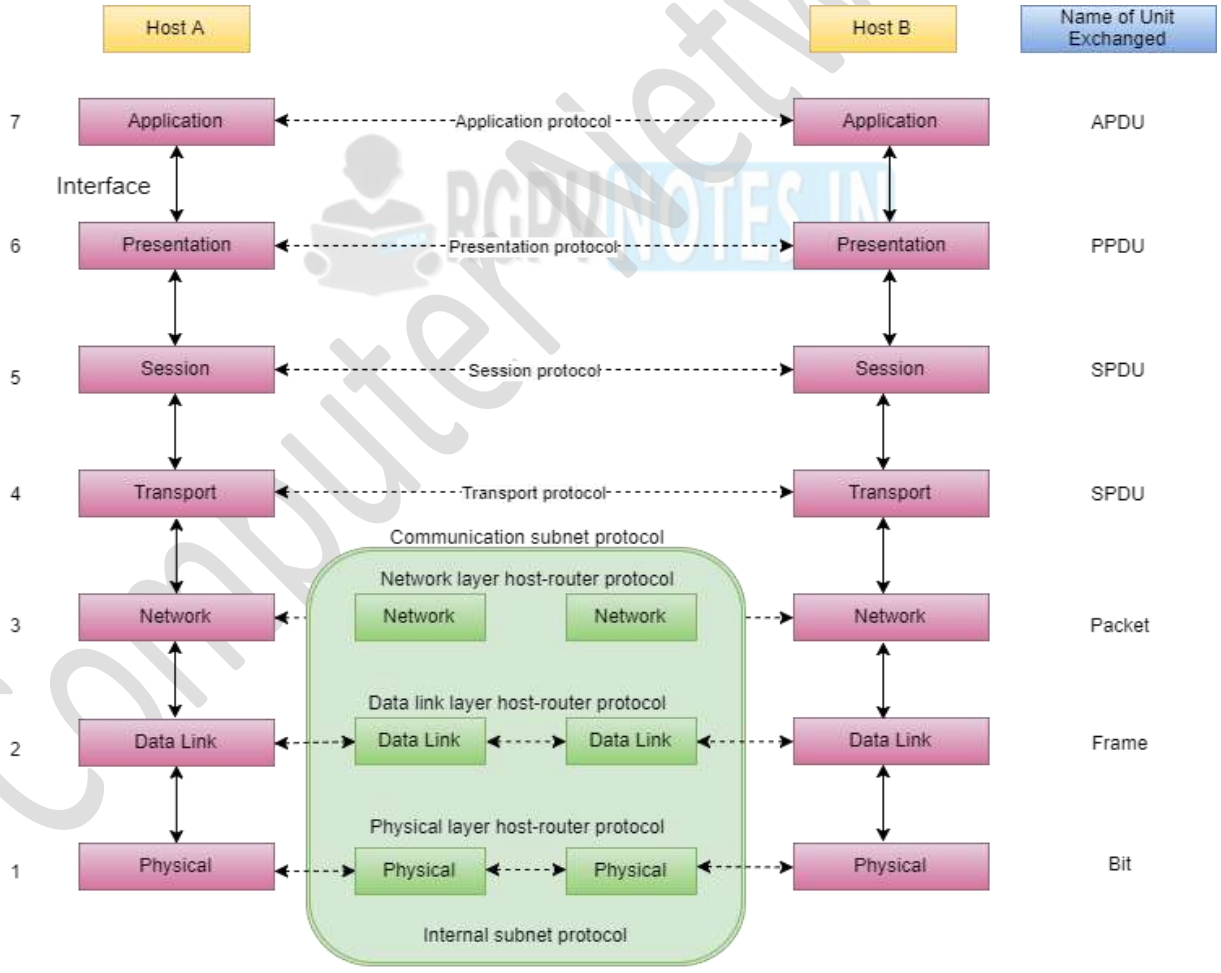


Fig. 1.8 OSI Reference Model

#Description of Different Layers:

Layer 1: The Physical Layer:

1. It is the lowest layer of the OSI Model.

2. It activates, maintains and deactivates the physical connection.
3. It is responsible for transmission and reception of the unstructured raw data over network.
4. Voltages and data rates needed for transmission is defined in the physical layer.
5. It converts the digital/ analog bits into electrical signal or optical signals.
6. Data encoding is also done in this layer.

Layer 2: Data Link Layer:

1. Data link layer synchronizes the information which is to be transmitted over the physical layer.
2. The main function of this layer is to make sure data transfer is error free from one node to another, over the physical layer.
3. Transmitting and receiving data frames sequentially is managed by this layer.
4. This layer sends and expects acknowledgements for frames received and sent respectively. Resending of non-acknowledgement received frames is also handled by this layer.
5. This layer establishes a logical layer between two nodes and also manages the Frame traffic control over the network. It signals the transmitting node to stop, when the frame buffers are full.

Layer 3: The Network Layer:

1. It routes the signal through different channels from one node to other.
2. It acts as a network controller. It manages the Subnet traffic.
3. It decides by which route data should take.
4. It divides the outgoing messages into packets and assembles the incoming packets into messages for higher levels.

Layer 4: Transport Layer:

1. It decides if data transmission should be on parallel path or single path.
2. Functions such as Multiplexing, Segmenting or Splitting on the data are done by this layer
3. It receives messages from the Session layer above it, converts the message into smaller units and passes it on to the Network layer.
4. Transport layer can be very complex, depending upon the network requirements.

Layer 5: The Session Layer:

1. Session layer manages and synchronize the conversation between two different applications.
2. Transfer of data from source to destination session layer streams of data are marked and are resynchronized properly, so that the ends of the messages are not cut prematurely, and data loss is avoided.

Layer 6: The Presentation Layer:

1. Presentation layer takes care that the data is sent in such a way that the receiver will understand the information (data) and will be able to use the data.
2. While receiving the data, presentation layer transforms the data to be ready for the application layer.
3. Languages (syntax) can be different of the two communicating systems. Under this condition presentation layer plays a role of translator.
4. It performs Data compression, Data encryption, Data conversion etc.

Layer 7: Application Layer:

1. It is the topmost layer.
2. Transferring of files disturbing the results to the user is also done in this layer. Mail services, directory services, network resource etc are services provided by application layer.
3. This layer mainly holds application programs to act upon the received and to be sent data.

Merits of OSI reference model:

1. OSI model distinguishes well between the services, interfaces and protocols.
2. Protocols of OSI model are very well hidden.
3. Protocols can be replaced by new protocols as technology changes.
4. Supports connection-oriented services as well as connectionless service.

Demerits of OSI reference model:

1. Model was devised before the invention of protocols.
2. Fitting of protocols is tedious task.
3. It is just used as a reference model.

#TCP/IP Reference Model:

The TCP/IP reference model was developed prior to OSI model. The major design goals of this model were,

1. To connect multiple networks together so that they appear as a single network.
2. To survive after partial subnet hardware failures.
3. To provide a flexible architecture.

Unlike OSI reference model, TCP/IP reference model has only 4 layers. They are,

1. Host-to-Network Layer
2. Internet Layer
3. Transport Layer
4. Application Layer

Layer 1: Host-to-network Layer

1. Lowest layer of the all.
2. Protocol is used to connect to the host, so that the packets can be sent over it.
3. Varies from host to host and network to network.

Layer 2: Internet layer

1. Selection of a packet switching network which is based on a connectionless internetwork layer is called a internet layer.
2. It is the layer which holds the whole architecture together.
3. It helps the packet to travel independently to the destination.
4. Order in which packets are received is different from the way they are sent.
5. IP (Internet Protocol) is used in this layer.
6. The various functions performed by the Internet Layer are:
    - Delivering IP packets
    - Performing routing
    - Avoiding congestion

Layer 3: Transport Layer

1. It decides if data transmission should be on parallel path or single path.
2. Functions such as multiplexing, segmenting or splitting on the data is done by transport layer.
3. The applications can read and write to the transport layer.
4. Transport layer adds header information to the data.
5. Transport layer breaks the message (data) into small units so that they are handled more efficiently by the network layer.
6. Transport layer also arrange the packets to be sent, in sequence.

Layer 4: Application Layer

The TCP/IP specifications described a lot of applications that were at the top of the protocol stack. Some of them were TELNET, FTP, SMTP, DNS etc.

1. TELNET is a two-way communication protocol which allows connecting to a remote machine and run applications on it.
2. FTP (File Transfer Protocol) is a protocol, that allows File transfer amongst computer users connected over a network. It is reliable, simple and efficient.
3. SMTP (Simple Mail Transport Protocol) is a protocol, which is used to transport electronic mail between a source and destination, directed via a route.
4. DNS (Domain Name Server) resolves an IP address into a textual address for Hosts connected over a network.

5. It allows peer entities to carry conversation.
6. It defines two end-to-end protocols: TCP and UDP
    - **TCP (Transmission Control Protocol):** It is a reliable connection-oriented protocol which handles byte-stream from source to destination without error and flow control.
    - **UDP (User-Datagram Protocol):** It is an unreliable connection-less protocol that does not want TCPs, sequencing and flow control. Example: One-shot request-reply kind of service.

Merits of TCP/IP model

1. It operated independently.
2. It is scalable.
3. Client/server architecture.
4. Supports number of routing protocols.
5. Can be used to establish a connection between two computers.

Demerits of TCP/IP

1. In this, the transport layer does not guarantee delivery of packets.
2. The model cannot be used in any other application.
3. Replacing protocol is not easy.
4. It has not clearly separated its services, interfaces and protocols.

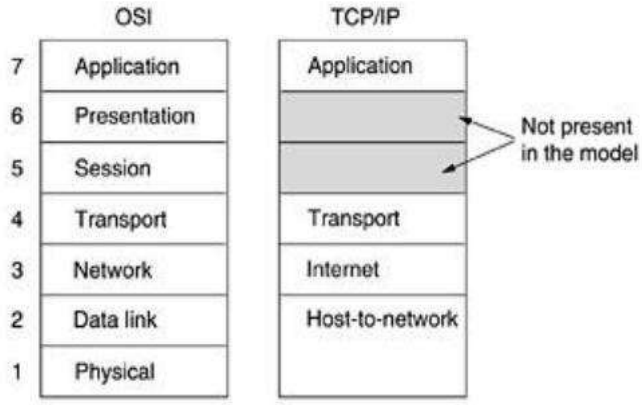


Fig 1.9 The TCP/IP reference model.

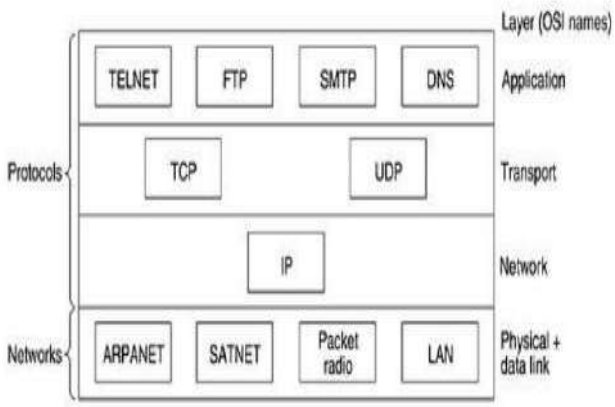


Fig 1.10 Protocols in the TCP/IP model initially.

#Comparison of the OSI and TCP/IP Reference Models:

OSI (Open System Interconnection)	TCP/IP (Transmission Control Protocol / Internet Protocol)
1. OSI is a generic, protocol independent standard, acting as a communication gateway between the network and end user.	1. TCP/IP model is based on standard protocols around which the Internet has developed. It is a communication protocol, which allows connection of hosts over a network.
2. In OSI model the transport layer guarantees the delivery of packets.	2. In TCP/IP model the transport layer does not guarantees delivery of packets. Still the TCP/IP model is more reliable.
3. Follows vertical approach.	3. Follows horizontal approach.
4. OSI model has a separate Presentation layer and Session layer.	4. TCP/IP does not have a separate Presentation layer or Session layer.
5. Transport Layer is Connection Oriented.	5. Transport Layer is both Connection Oriented and Connection less.

6. Network Layer is both Connection Oriented and Connection less.	6. Network Layer is Connection less.
7. OSI is a reference model around which the networks are built. Generally, it is used as a guidance tool.	7. TCP/IP model is, in a way implementation of the OSI model.
8. Network layer of OSI model provides both connection oriented and connectionless service.	8. The Network layer in TCP/IP model provides connectionless service.
9. OSI model has a problem of fitting the protocols into the model.	9. TCP/IP model does not fit any protocol
10. Protocols are hidden in OSI model and are easily replaced as the technology changes.	10. In TCP/IP replacing protocol is not easy.
11. OSI model defines services, interfaces and protocols very clearly and makes clear distinction between them. It is protocol independent.	11. In TCP/IP, services, interfaces and protocols are not clearly separated. It is also protocol dependent.
12. It has 7 layers	12. It has 4 layers

Principals of physical layer: **Physical** components are the electronic hardware devices, media, and other connectors that transmit and carry the signals to represent the bits. ... Codes are groupings of bits used to provide a predictable pattern that can be recognized by both the sender and the received.The physical components are the electronic hardware devices, media, and other connectors that transmit and carry the signals to represent the bits. Hardware components such as network adapters (NICs), interfaces and connectors, cable materials, and cable designs are all specified in standards associated with the physical layer. The various ports and interfaces on a Cisco 1941 router are also examples of physical components with specific connectors and pinouts resulting from standards.

Encoding

Encoding or line encoding is a method of converting a stream of data bits into a predefined "code". Codes are groupings of bits used to provide a predictable pattern that can be recognized by both the sender and the received. In the case of networking, encoding is a pattern of voltage or current used to represent bits; the 0s and 1s.

Signaling

The physical layer must generate the electrical, optical, or wireless signals that represent the "1" and "0" on the media. The method of representing the bits is called the signaling method. The physical layer standards must define what type of signal represents a "1" and what type of signal represents a "0". This can be as simple as a change in the level of an electrical signal or optical pulse. For example, a long pulse might represent a 1, whereas a short pulse represents a 0.

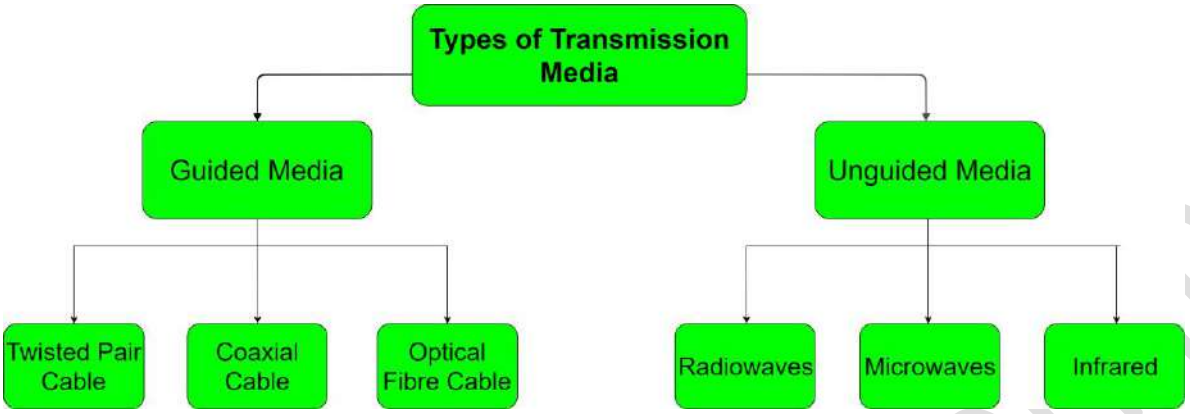
Media, Bandwidth, Data rate and Modulations

Media:-

Network media refers to the communication channels used to interconnect nodes on a computer network. Typical examples of network media include copper coaxial cable, copper twisted pair

cables and optical fiber cables used in wired networks, and radio waves used in wireless data communications networks.

In data communication terminology, a transmission medium is a physical path between the transmitter and the receiver i.e it is the channel through which data is sent from one place to another.



Bandwidth

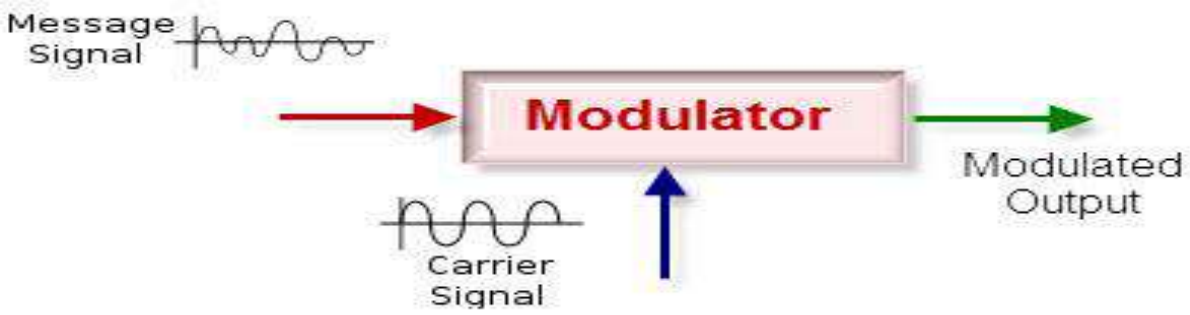
Bandwidth describes the maximum data transfer rate of a network or Internet connection. It measures how much data can be sent over a specific connection in a given amount of time. For example, a gigabit Ethernet connection has a bandwidth of 1,000 Mbps (125 megabytes per second). An Internet connection via cable modem may provide 25 Mbps of bandwidth. Bandwidth also refers to a range of frequencies used to transmit a signal. This type of bandwidth is measured in hertz and is often referenced in signal processing applications.

Data rate

The data rate is a term to denote the transmission speed, or the number of bits per second transferred. The useful data rate for the user is usually less than the actual data rate transported on the network. One reason for this is that additional bits are transferred for e.g signalling, the address, the recovery of timing information at the receiver or error correction to compensate for possible transmission errors. In telecommunications, it is common use to express the data rate in bits per seconds (bit/s), see bit rate. In data communication, the data rate is often expressed in bytes per second (B/s).

Modulation

Modulation plays a key role in communication system to encode information digitally in analog world. It is very important to modulate the signals before sending them to the receiver section for larger distance transfer, accurate data transfer and low-noise data reception.



Modulation is a process of changing the characteristics of the wave to be transmitted by superimposing the message signal on the high frequency signal. In this process video, voice and other data signals modify high frequency signals – also known as carrier wave. This carrier wave can be DC or AC or pulse chain depending on the application used. Usually high frequency sine wave is used as a carrier wave signal.These modulation techniques are classified into two major types: analog and digital or pulse modulation. Prior to discussing further about the different types of modulation techniques, let us understand the importance of modulation.

Note: Bandwidth and data rate are related by the modulation format. Different modulation formats will require different bandwidths for the same data rate. For FM modulation, the bandwidth is approximately 2*(df + fm) where df is the maximum frequency deviation and fm is the frequency of the message

RGPVNOTES.IN

We hope you find these notes useful.

You can get previous year question papers at https://qp.rgpvnotes.in .

If you have any queries or you want to submit your study notes please write us at rgpvnotes.in@gmail.com

LIKE & FOLLOW US ON FACEBOOK
facebook.com/rgpvnotes.in

RGPVNOTES.IN

Program : **B.Tech**

Subject Name: **Computer Networks**

Subject Code: **CS-602**

Semester: **6th**

Department of Computer Science and Engineering
CS602 Computer Networks
Subject Notes: UNIT-II

Syllabus: **Data Link Layer: Need, Services Provided, Framing, Flow Control, Error control. Data Link Layer Protocol: Elementary &Sliding Window protocol: 1-bit, Go-Back-N, Selective Repeat, Hybrid ARQ. Protocol verification: Finite State Machine Models & Petri net models. ARP/RARP/GARP**

DATA LINK LAYER: NEED

Data Link Layer is second layer of OSI Layered Model. This layer is one of the most complicated layers and has complex functionalities and liabilities. Data link layer hides the details of underlying hardware and represents itself to upper layer as the medium to communicate.

Data link layer works between two hosts which are directly connected in some sense. This direct connection could be point to point or broadcast. Data link layer is responsible for converting data stream to signals bit by bit and to send that over the underlying hardware.

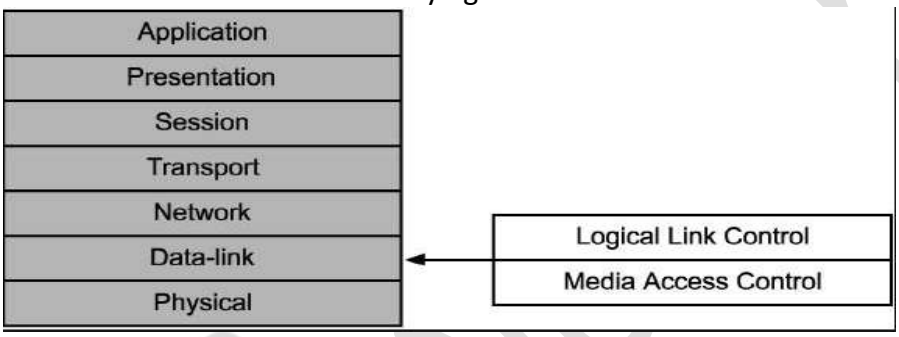


Fig. 2.1 Seven Layer Architecture

Data link layer has two sub-layers:

- Logical Link Control: It deals with protocols, flow-control, and error control
- Media Access Control: It deals with actual control of media

DATA LINK LAYER: SERVICE PROVIDED

- Encapsulation of network layer data packets into frames.
- Frame synchronization.
- Error Control
- Flow control, in addition to the one provided on the transport layer.
- LAN switching (packet switching) including MAC filtering and spanning tree protocol
- Data packet queuing or scheduling
- Store-and-forward switching or cut-through switching

DATA LINK LAYER: FRAMING

Since the physical layer merely accepts and transmits a stream of bits without any regard to meaning or structure, it is up to the data link layer to create and recognize frame boundaries. This can be accomplished by attaching special bit patterns to the beginning and end of the frame. If these bit patterns can accidentally occur in data, special care must be taken to make sure these patterns are not incorrectly interpreted as frame delimiters. The four framing methods that are widely used are

- Character count
- Starting and ending characters, with character stuffing
- Starting and ending flags, with bit stuffing

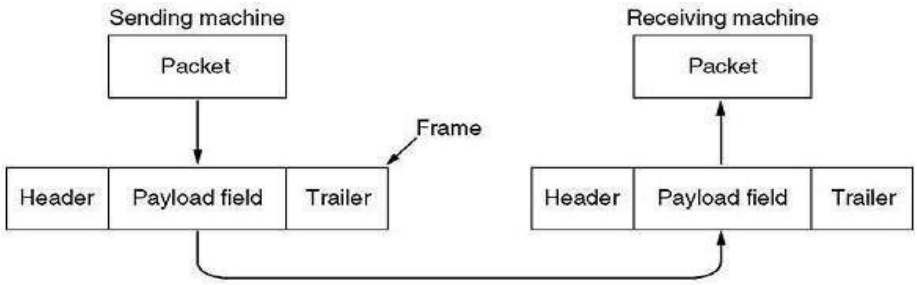


Fig. 2.2 Data Link Layer: Framing

Character Count

This method uses a field in the header to specify the number of characters in the frame. When the data link layer at the destination sees the character count, it knows how many characters follow, and hence where the end of the frame is. The disadvantage is that if the count is garbled by a transmission error, the destination will lose synchronization and will be unable to locate the start of the next frame. So, this method is rarely used.

Character stuffing

In the second method, each frame starts with the ASCII character sequence DLE STX and ends with the sequence DLE ETX. This method overcomes the drawbacks of the character count method. However, character stuffing is closely associated with 8-bit characters and this is a major hurdle in transmitting arbitrary sized characters.

Bit stuffing

The third method allows data frames to contain an arbitrary number of bits and allows character codes with an arbitrary number of bits per character. At the start and end of each frame is a flag byte consisting of the special bit pattern 01111110. Whenever the sender's data link layer encounters five consecutive 1s in the data, it automatically stuffs a zero bit into the outgoing bit stream. This technique is called bit stuffing

Physical layer coding violations

The final framing method is physical layer coding violations and is applicable to networks in which the encoding on the physical medium contains some redundancy. In such cases normally, a 1 bit is a high-low pair and a 0 bit is a low-high pair. The combinations of low-low and high-high which are not used for data may be used for marking frame boundaries.

DATALINK LAYER: FLOW CONTROL

Flow control coordinates that amount of data that can be sent before receiving acknowledgement.

- It is one of the most important duties of the data link layer.
- Flow control tells the sender how much data to send.
- It makes the sender wait for some sort of an acknowledgment (ACK) before continuing to send more data.
- Flow Control Techniques: Stop-and-wait, and Sliding Window

DATA LINK LAYER: ERROR CONTROL

Error control in the data link layer is based on ARQ (automatic repeat request), which is the retransmission of data.

- The term error control refers to methods of error detection and retransmission.
- Anytime an error is detected in an exchange, specified frames are retransmitted. This process is called ARQ.

To ensure reliable communication, there needs to exist flow control (managing the amount of data the sender sends), and error control (that data arrives at the destination error free).

- Flow and error control needs to be done at several layers.
- For node-to-node links, flow and error control is carried out in the data-link layer.

- For end-point to end-point, flow and error control is carried out in the transport layer.

There may be three types of errors:

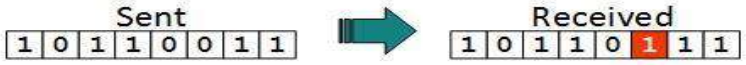


Fig. 2.3 Single bit error

In a frame, there is only one bit, anywhere though, which is corrupt.

Fig. 2.4 Multiple bits error

Frame is received with more than one bit in corrupted state.

Fig. 2.5 Burst error

Frame contains more than1 consecutive bits corrupted.

DATA LINK LAYER PROTOCOL

The basic function of the layer is to transmit frames over a physical communication link. Transmission may be half duplex or full duplex. To ensure that frames are delivered free of errors to the destination station (IMP) a number of requirements are placed on a data link protocol. The protocol (control mechanism) should be capable of performing:

1. The identification of a frame (i.e. recognises the first and last bits of a frame).
2. The transmission of frames of any length up to a given maximum. Any bit pattern is permitted in a frame.
3. The detection of transmission errors.
4. The retransmission of frames which were damaged by errors.
5. The assurance that no frames were lost.
6. In a multidrop configuration some mechanism must be used for preventing conflicts caused by simultaneous transmission by many stations.
7. The detection of failure or abnormal situations for control and monitoring purposes.

It should be noted that as far as layer 2 is concerned a host message is pure data, every single bit of which is to be delivered to the other host. The frame header pertains to layer 2 and is never given to the host.

Elementary Data Link Protocols

- Data are transmitted in one direction only
- The transmitting (Tx) and receiving (Rx) hosts are always ready
- Processing time can be ignored
- Infinite buffer space is available
- No errors occur; i.e. no damaged frames and no lost frames (perfect channel)

Sliding Window protocol:

A sliding window protocol is a feature of packet-based data transmission protocols. Sliding window protocols are used where reliable in-order delivery of packets is required, such as in the Data Link Layer (OSI model) as well as in the Transmission Control Protocol (TCP).

The Sliding Window ARQ has three techniques

1. 1-bit
2. Go- Back N
3. Selective Repeat

1-bit

One-bit sliding window protocol is also called Stop-And-Wait protocol. In this protocol, the sender sends out one frame, waits for acknowledgment before sending next frame, thus the name Stop-And-Wait.

Problem with Stop-And-Wait protocol is that it is very inefficient. At any one moment, only in frame is in transition. The sender will have to wait at least one round trip time before sending next. The waiting can be long for a slow network such as satellite link.

Stop and Wait Protocol

Characteristics

- Used in Connection-oriented communication.
- It offers error and flow control
- It is used in Data Link and Transport Layers
- Stop and Wait ARQ mainly implements Sliding Window Protocol concept with Window Size 1

Useful Terms:

- **Propagation Delay:** Amount of time taken by a packet to make a physical journey from one router to another router.

  Propagation Delay = (Distance between routers) / (Velocity of propagation)
- RoundTripTime (RTT) = 2* Propagation Delay
- TimeOut (TO) = 2* RTT
- Time To Live (TTL) = 2* TimeOut. (Maximum TTL is 180 seconds)

Simple Stop and Wait

Sender:

Rule 1) Send one data packet at a time.

Rule 2) Send next packet only after receiving acknowledgement for previous.

Receiver:

Rule 1) Send acknowledgement after receiving and consuming of data packet.

Rule 2) after consuming packet acknowledgement need to be sent (Flow Control)

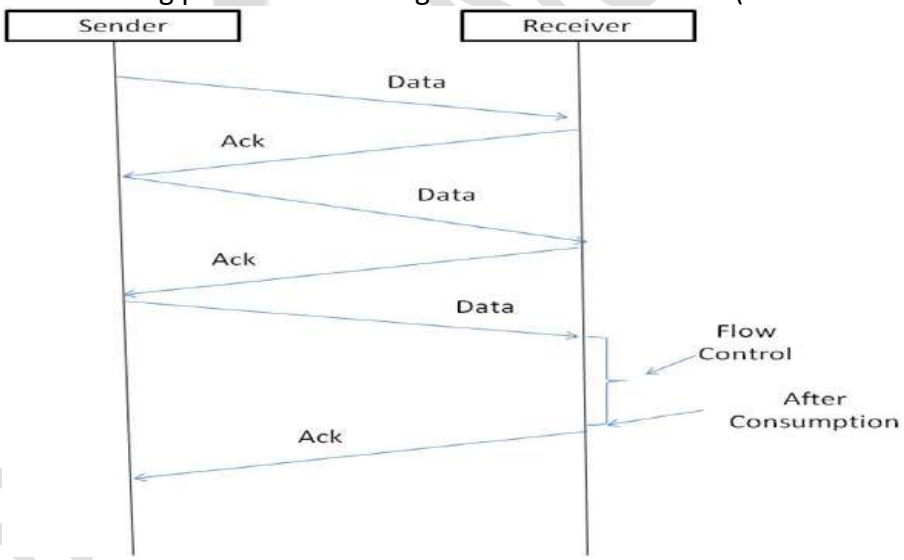


Fig. 2.6 Stop and Wait

Problems:

1. Lost Data

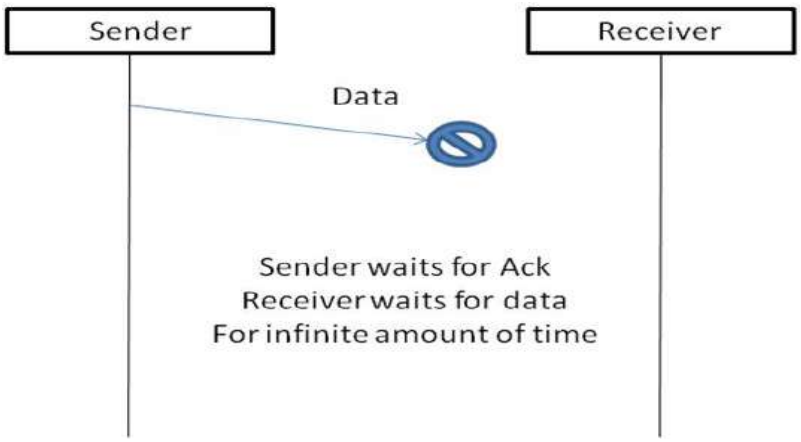


Fig. 2.7 Stop and Wait- Lost Data

2. Lost Acknowledgement:

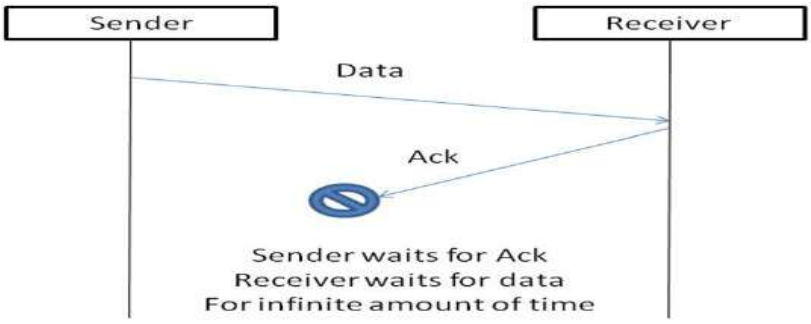


Fig. 2.8 Stop and Wait- Lost Acknowledgement

3. Delayed Acknowledgement/Data: After timeout on sender side, a long-delayed acknowledgement might be wrongly considered as acknowledgement of some other recent packet.

Stop and Wait ARQ (Automatic Repeat Request)

Above 3 problems are resolved by Stop and Wait ARQ (Automatic Repeat Request) that does both error control and flow control.

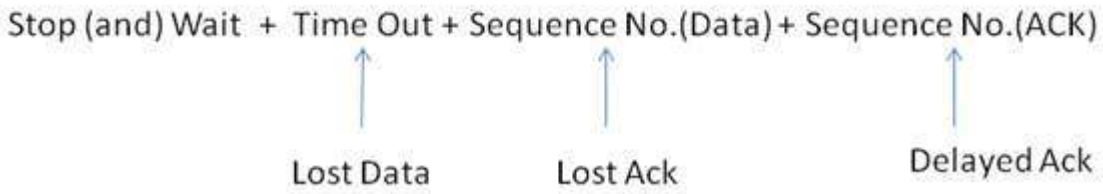


Fig. 2.9 Stop and Wait ARQ (Automatic Repeat Request)

1. Time Out:

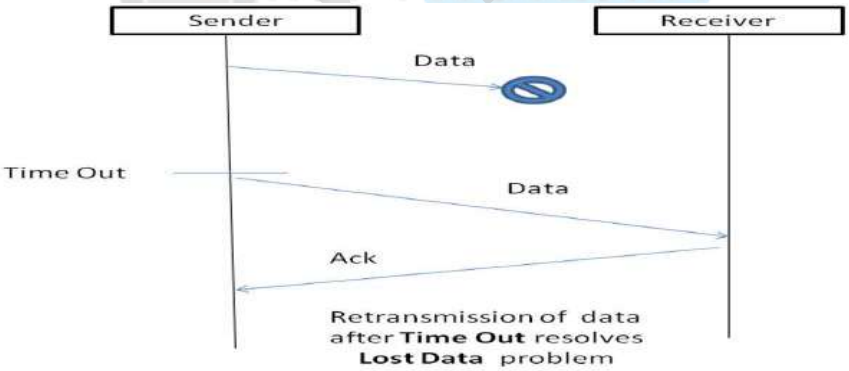


Fig. 2.10 Stop and Wait ARQ-Time Out

2. Sequence Number (Data)

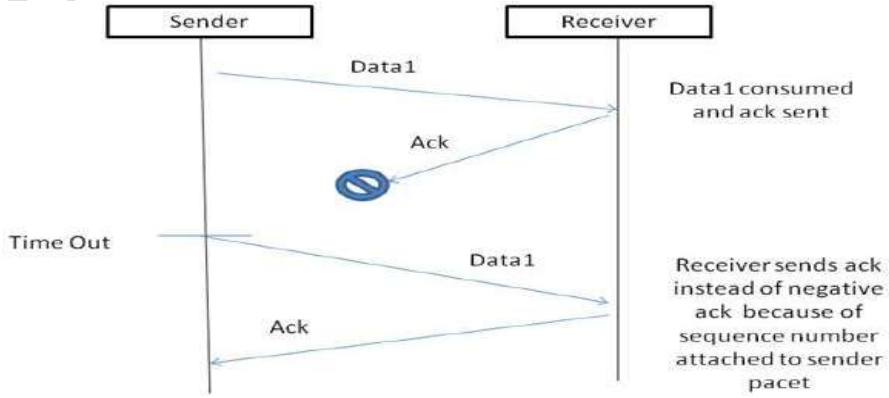


Fig. 2.11 Stop and Wait ARQ-ACK Lost

3. Delayed Acknowledgement:

This is resolved by introducing sequence number for acknowledgement also.

Working of Stop and Wait ARQ:

1) Sender A sends a data frame or packet with sequence number 0.

2) Receiver B, after receiving data frame, sends and acknowledgement with sequence number 1 (sequence number of next expected data frame or packet)

There is only one-bit sequence number that implies that both sender and receiver have buffer for one frame or packet only.

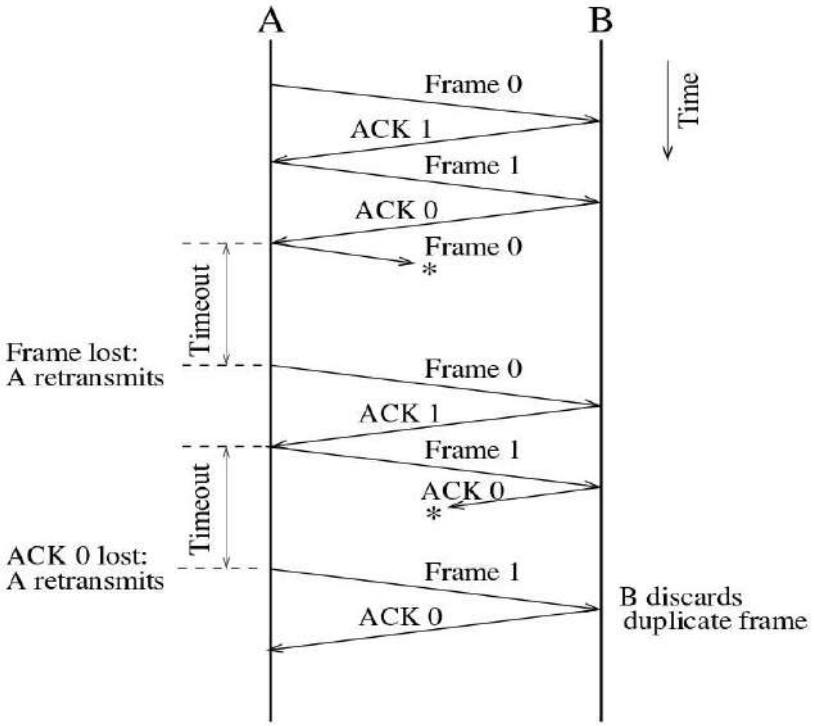


Fig. 2.12 Working of Stop and Wait ARQ

Characteristics of Stop and Wait ARQ:

- It uses link between sender and receiver as half duplex link
- Throughput = 1 Data packet/frame per RTT
- If Bandwidth*Delay product is very high, then stop and wait protocol is not so useful. The sender has to keep waiting for acknowledgements before sending the processed next packet.
- It is an example for "**Closed Loop OR connection oriented** "protocols
- It is a special category of SWP where its window size is 1
- Irrespective of number of packets sender is having stop and wait protocol requires only 2 sequences numbers 0 and 1

The Stop and Wait ARQ solves main three problems, but may cause big performance issues as sender always waits for acknowledgement even if it has next packet ready to send. Consider a situation where you have a high bandwidth connection and propagation delay is also high (you are connected to some server in some other country though a high-speed connection). To solve this problem, we can send more than one packet at a time with a larger sequence numbers. We will be discussing these protocols in next articles.

So, Stop and Wait ARQ may work fine where propagation delay is very less for example LAN connections, but performs badly for distant connections like satellite connection.

Go- Back N protocol

Go-Back-N protocol is a sliding window protocol. It is a mechanism to detect and control the error in datalink layer. During transmission of frames between sender and receiver, if a frame is damaged, lost, or an acknowledgement is lost then the action performed by sender and receiver.

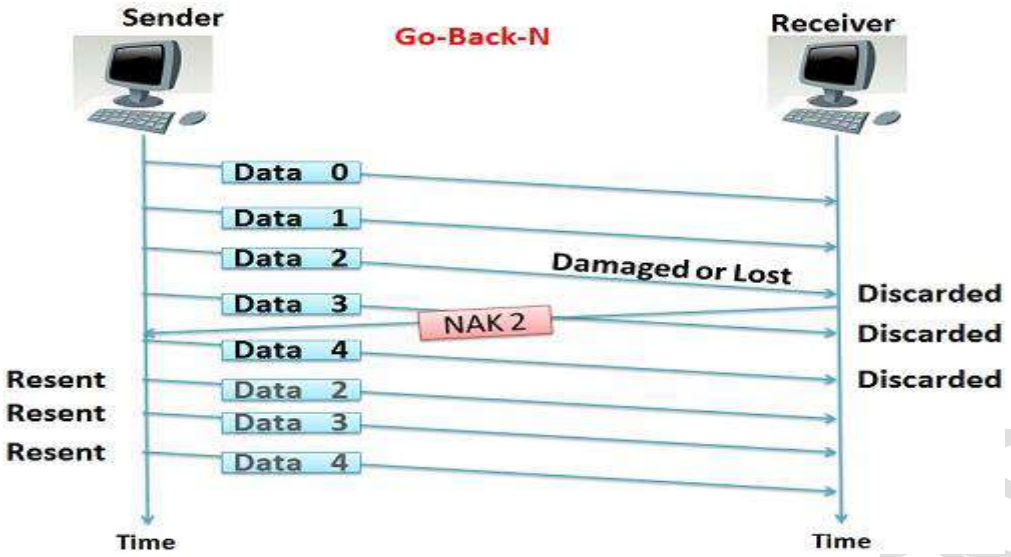


Fig. 2.13 Go- Back N protocol

Selective Repeat protocol

Selective repeat is also the sliding window protocol which detects or corrects the error occurred in datalink layer. The selective repeat protocol retransmits only that frame which is damaged or lost. In selective repeat protocol, the retransmitted framed is received out of sequence. The selective repeat protocol can perform following actions

- The receiver is capable of sorting the frame in a proper sequence, as it receives the retransmitted frame whose sequence is out of order of the receiving frame.
- The sender must be capable of searching the frame for which the NAK has been received.
- The receiver must contain the buffer to store all the previously received frame on hold till the retransmitted frame is sorted and placed in a proper sequence.
- The ACK number, like NAK number, refers to the frame which is lost or damaged.
- It requires the less window size as compared to go-back-n protocol.

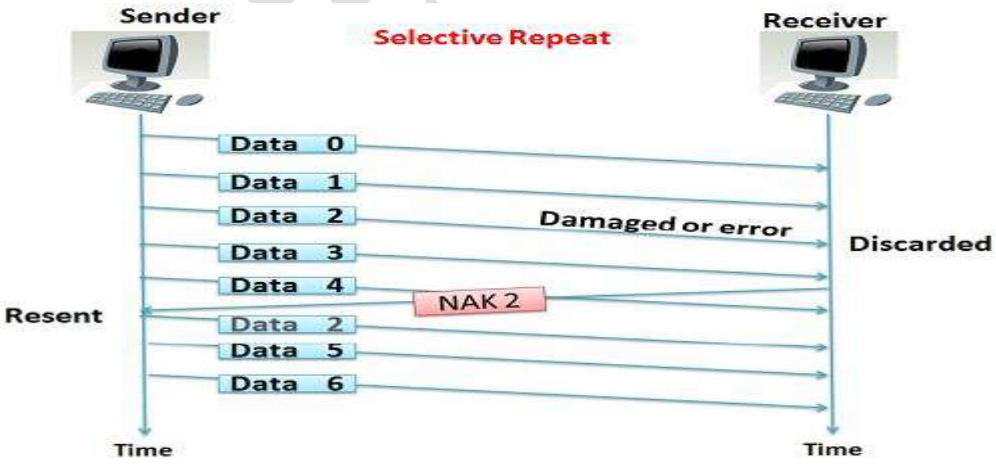


Fig. 2.14 Selective Repeat protocol

HYBRID ARQ

The HARQ is the use of conventional ARQ along with an Error Correction technique called 'Soft Combining', which no longer discards the received bad data (with error).

With the 'Soft Combining' data packets that are not properly decoded are not discarded anymore. The received signal is stored in a 'buffer', and will be combined with next retransmission.

That is, two or more packets received each one with insufficient SNR to allow individual decoding can be combined in such a way that the total signal can be decoded!

The following image explains this procedure. The transmitter sends a package [1]. The package [1] arrives, and is 'OK'. If the package [1] is 'OK' then the receiver sends an 'ACK'.

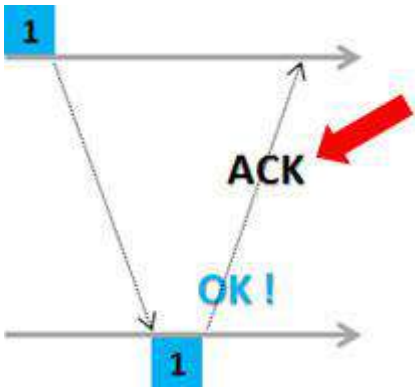


Fig. 2.15 Transmitter sends a packet-1

The transmission continues, and is sent a package [2]. The package [2] arrives, but let's consider now that it arrives with errors. If the package [2] arrives with errors, the receiver sends a 'NACK'.

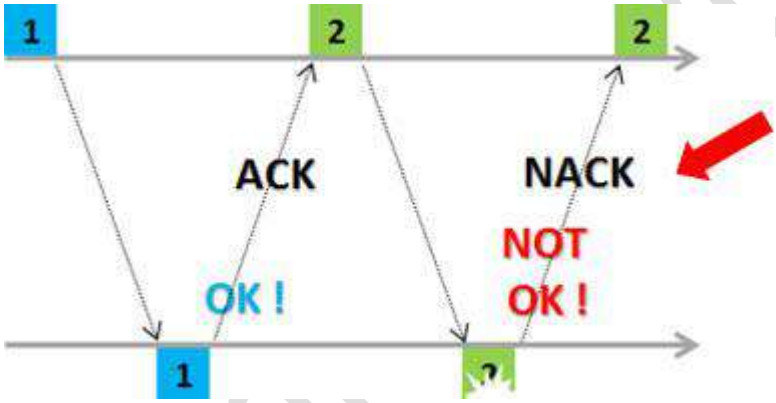


Fig. 2.16 Transmitter sends a packet-2

Only now this package [2] (bad) is not thrown away, as it is done in conventional ARQ. Now it is stored in a 'buffer'.

Fig. 2.17 Receiver buffers a packet-2

Continuing, the transmitter sends another package [2.1] that also (let's consider) arrives with errors.

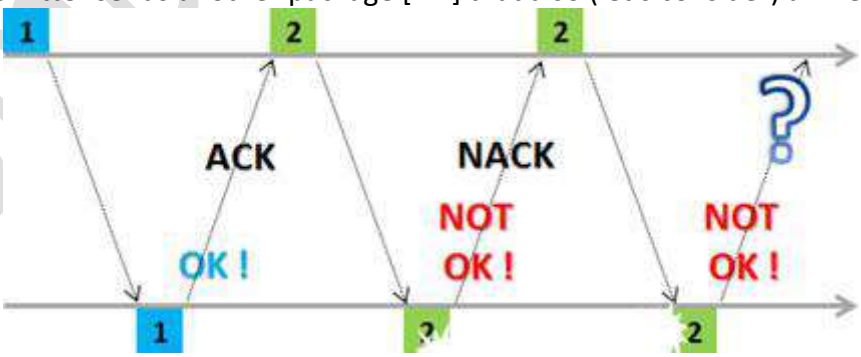


Fig. 2.18 Transmitter sends another packet-2

We have then in a buffer: bad package [2], and another package [2.1] which is also bad.
Does by adding (combining) these two packages ([2] + [2.1]) we have the complete information?
Yes. So, we send an 'ACK'.

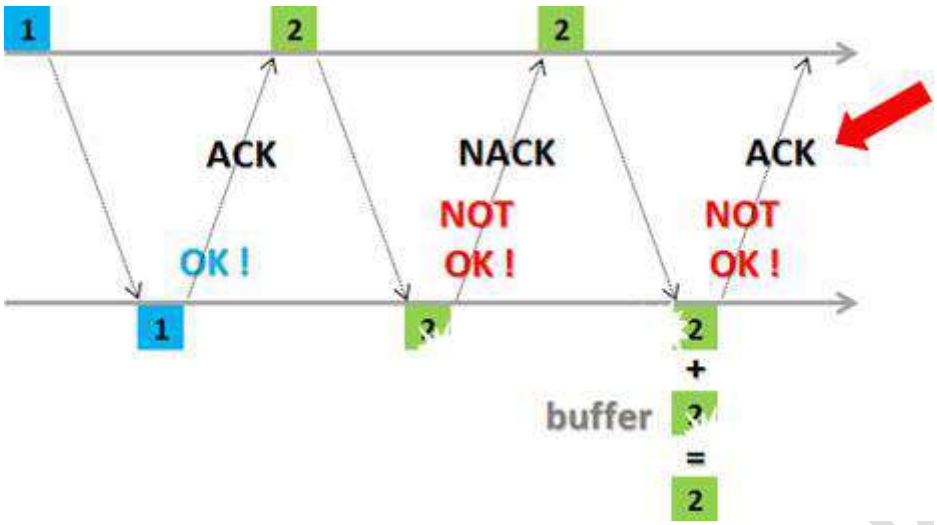


Fig. 2.19 Receiver combining buffers a packet-2 and another packet-2

But if the combination of these two packages still does not give us the complete information, the process must continue - and another 'NACK' is sent.

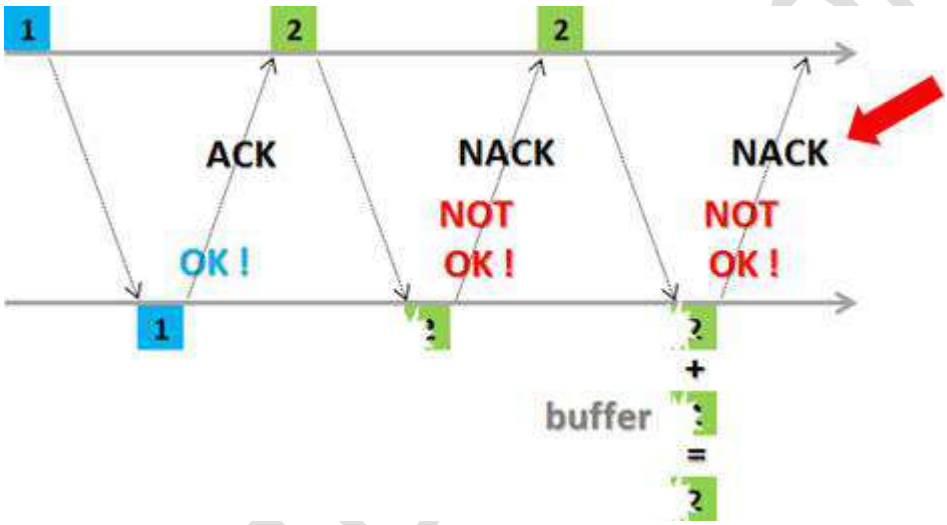


Fig. 2.20 Receiver sends NACK

And there we have another retransmission. Now the transmitter sends a third package [2.2]. Let's consider that now it is 'OK', and the receiver sends an 'ACK'.

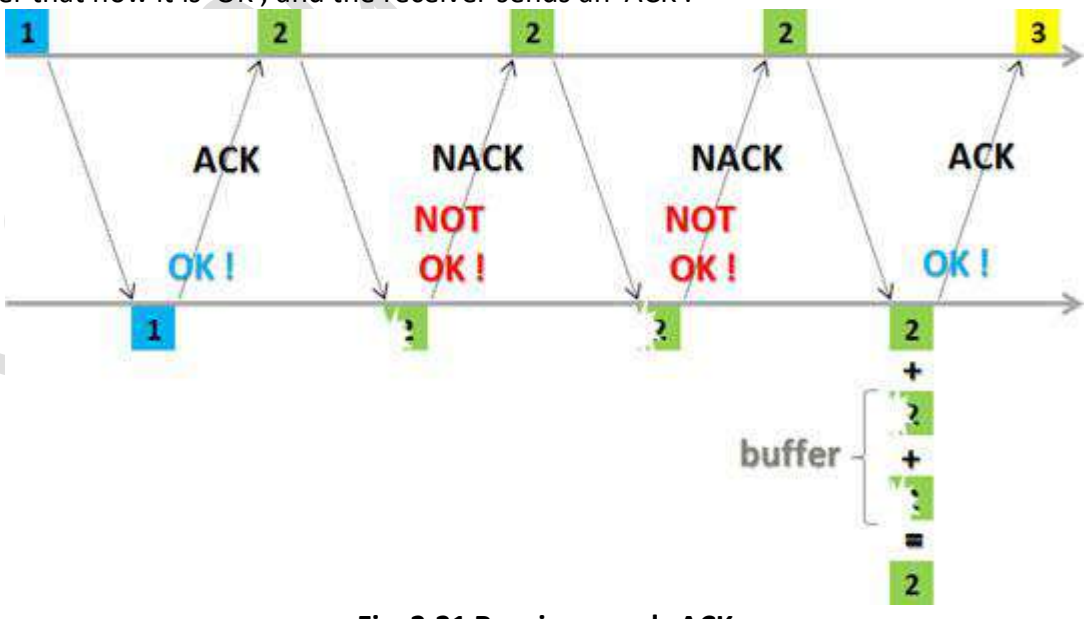


Fig. 2.21 Receiver sends ACK

Here we can see the following: along with the received package [2.2], the receiver also has packages [2] and [2.1] that have not been dropped and are stored in the buffer.

In our example, we see that the package arrived 2 times 'wrong'. And what is the limit of these retransmissions? Up to 4. IE, we can have up to 4 retransmissions in each process. This is the maximum number supported by 'buffer'.

BIT ORIENTED PROTOCOLS

A bit-oriented protocol is a communications protocol that sees the transmitted data as an opaque stream of bits with no semantics, or meaning. Control codes are defined in terms of bit sequences instead of characters. Bit oriented protocol can transfer data frames regardless of frame contents. It can also be stated as "bit stuffing" this technique allows the data frames to contain an arbitrary number of bits and allows character codes with arbitrary number of bits per character.

SDLC

Synchronous Data Link Control (SDLC) supports a variety of link types and topologies. It can be used with point-to-point and multipoint links, bounded and unbounded media, half-duplex and full-duplex transmission facilities, and circuit-switched and packet-switched networks.

SDLC identifies two types of network nodes: primary and secondary. Primary nodes control the operation of other stations, called secondary. The primary polls the secondary in a predetermined order and secondary can then transmit if they have outgoing data. The primary also sets up and tears down links and manages the link while it is operational. Secondary nodes are controlled by a primary, which means that secondary can send information to the primary only if the primary grants permission.

SDLC primaries and secondary can be connected in four basic configurations:

- Point-to-point---Involves only two nodes, one primary and one secondary.
- Multipoint---Involves one primary and multiple secondary.
- Loop---Involves a loop topology, with the primary connected to the first and last secondary. Intermediate secondary pass messages through one another as they respond to the requests of the primary.
- Hub go-ahead---Involves an inbound and an outbound channel. The primary uses the outbound channel to communicate with the secondary. The secondary use the inbound channel to communicate with the primary. The inbound channel is daisy-chained back to the primary through each secondary.

SDLC Frame Format

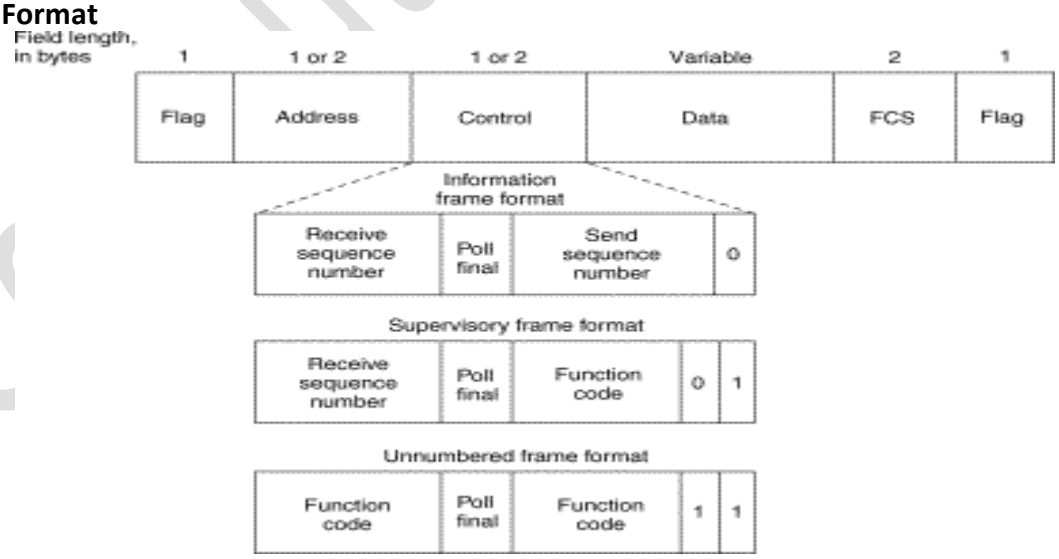


Fig. 2.22 SDLC Frame Format

- **Flag**---Initiates and terminates error checking.
- **Address**---Contains the SDLC address of the secondary station, which indicates whether the frame comes from the primary or secondary. This address can contain a specific address, a group address, or a broadcast address. A primary is either a communication source or a destination, which eliminates the need to include the address of the primary.
- **Control**---Employs three different formats, depending on the type of SDLC frame used:

1. **Information (I) frame:** Carries upper-layer information and some control information. This frame sends and receives sequence numbers, and the poll final (P/F) bit performs flow and error control. The send-sequence number refers to the number of the frame to be sent next. The receive-sequence number provides the number of the frame to be received next. Both sender and receiver maintain send- and receive-sequence numbers.
   A primary station uses the P/F bit to tell the secondary whether it requires an immediate response. A secondary station uses the P/F bit to tell the primary whether the current frame is the last in its current response.
2. **Supervisory (S) frame:** Provides control information. An S frame can request and suspend transmission, reports on status, and acknowledge receipt of I frames. S frames do not have an information field.
3. **Unnumbered (U) frame:** Supports control purposes and is not sequenced. A U frame can be used to initialize secondary. Depending on the function of the U frame, its control field is 1 or 2 bytes. Some U frames have an information field.

- **Data**---Contains path information unit (PIU) or exchange identification (XID) information.
- **Frame Check Sequence (FCS)** ---Precedes the ending flag delimiter and is usually a cyclic redundancy check (CRC) calculation remainder. The CRC calculation is redone in the receiver. If the result differs from the value in the original frame, an error is assumed.

HDLC

High-Level Data Link Control (HDLC) is a bit-oriented code-transparent synchronous data link layer protocol. HDLC provides both connection-oriented and connectionless service. HDLC can be used for point to multipoint connections, but is now used almost exclusively to connect one device to another, using what is known as Asynchronous Balanced Mode (ABM). The original master-slave modes Normal Response Mode (NRM) and Asynchronous Response Mode (ARM) are rarely used.

FRAMING

HDLC frames can be transmitted over synchronous or asynchronous serial communication links. Those links have no mechanism to mark the beginning or end of a frame, so the beginning and end of each frame has to be identified. This is done by using a frame delimiter, or flag, which is a unique sequence of bits that is guaranteed not to be seen inside a frame. This sequence is '01111110', or, in hexadecimal notation, 0x7E. Each frame begins and ends with a frame delimiter. A frame delimiter at the end of a frame may also mark the start of the next frame. A sequence of 7 or more consecutive 1-bits within a frame will cause the frame to be aborted.

When no frames are being transmitted on a simplex or full-duplex synchronous link, a frame delimiter is continuously transmitted on the link. Using the standard NRZI encoding from bits to line levels (0 bit = transition, 1 bit = no transition), this generates one of two continuous waveforms, depending on the initial state:

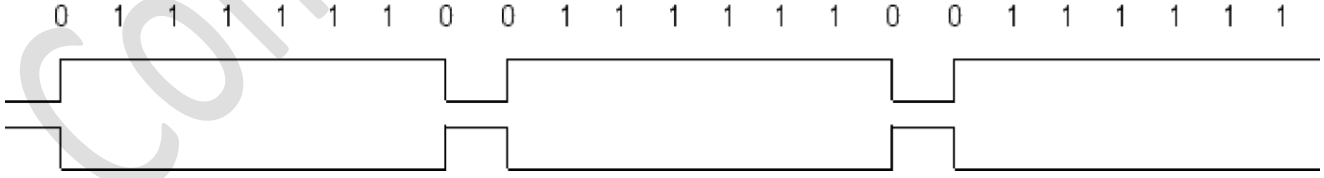


Fig. 2.23 HDLC Framing

This is used by modems to train and synchronize their clocks via phase-locked loops. Some protocols allow the 0-bit at the end of a frame delimiter to be shared with the start of the next frame delimiter, i.e. '011111101111110'.

Frame structure

The contents of an HDLC frame are shown in the following table:

Flag	Address	Control	Information	FCS	Flag
8 bits	8 or more bits	8 or 16 bits	Variable length, 0 or more bits	16 or 32 bits	8 bits

Fig. 2.24 HDLC Frame structure

Note that the end flag of one frame may be (but does not have to be) the beginning (start) flag of the next frame.

Data is usually sent in multiples of 8 bits, but only some variants require this; others theoretically permit data alignments on other than 8-bit boundaries.

There are three fundamental types of HDLC frames.

- Information frames, or I-frames, transport user data from the network layer. In addition, they can also include flow and error control information piggybacked on data.
- Supervisory Frames, or S-frames, are used for flow and error control whenever piggybacking is impossible or inappropriate, such as when a station does not have data to send. S-frames do not have information fields.
- Unnumbered frames, or U-frames, are used for various miscellaneous purposes, including link management. Some U-frames contain an information field, depending on the type.

BISYNC

Binary Synchronous Communication (BSC or Bisync) is an IBM character-oriented, half duplex link protocol, announced in 1967 after the introduction of System/360. It replaced the synchronous transmit-receive (STR) protocol used with second generation computers. The intent was that common link management rules could be used with three different character encodings for messages. Six-bit Transcode looked backwards to older systems.

BISYNC establishes rules for transmitting binary-coded data between a terminal and a host computer's BISYNC port. While BISYNC is a half-duplex protocol, it will synchronize in both directions on a full-duplex channel. BISYNC supports both point-to-point (over leased or dial-up lines) and multipoint transmissions. Each message must be acknowledged, adding to its overhead.

BISYNC is character oriented, meaning that groups of bits (bytes) are the main elements of transmission, rather than a stream of bits. The BISYNC frame is pictured next. It starts with two sync characters that the receiver and transmitter use for synchronizing. This is followed by a start of header (SOH) command, and then the header. Following this are the start of text (STX) command and the text. Finally, an end of text (EOT) command and a cyclic redundancy check (CRC) end the frame. The CRC provides error detection and correction.

Fig. 2.25 BISYNC

Most of the bisynchronous protocols, of which there are many, provide only half-duplex transmission and require an acknowledgment for every block of transmitted data. Some do provide full-duplex transmission and bit-oriented operation.

BISYNC has largely been replaced by the more powerful SDLC (Synchronous Data Link Control).

LAP AND LAPB

Link Access Procedure (LAP) protocols are Data Link layer protocols for framing and transmitting data across point-to-point links. LAP was originally derived from HDLC (High-Level Data Link Control), but was later updated and renamed LAPB (LAP Balanced).

LAPB is the data link protocol for X.25.LAPB is a bit-oriented protocol derived from HDLC that ensures that frames are error free and in the right sequence. It can be used as a Data Link Layer protocol implementing the connection-mode data link service in the OSI Reference Model as defined by ITU-T Recommendation X.222.

LAPB is used to manage communication and packet framing between data terminal equipment (DTE) and the data circuit-terminating equipment (DCE) devices in the X.25 protocol stack. LAPB is essentially HDLC in Asynchronous Balanced Mode (ABM). LAPB sessions can be established by either the DTE or DCE. The station initiating the call is determined to be the primary, and the responding station is the secondary.

Frame types

- **I-Frames (Information frames):** Carries upper-layer information and some control information. I-frame functions include sequencing, flow control, and error detection and recovery. I-frames carry send and receive sequence numbers.
- **S-Frames (Supervisory Frames):** Carries control information. S-frame functions include requesting and suspending transmissions, reporting on status, and acknowledging the receipt of I-frames. S-frames carry only receive sequence numbers.
- **U-Frames (Unnumbered Frames):** carries control information. U-frame functions include link setup and disconnection, as well as error reporting. U-frames carry no sequence numbers

Frame format

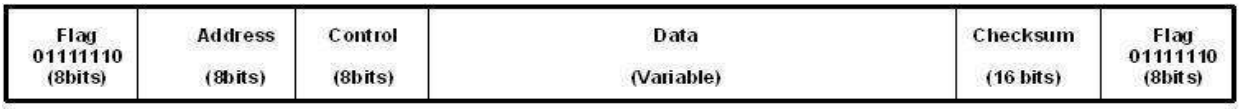


Fig. 2.26 Frame format

Flag – The value of the flag is always 0x7E. In order to ensure that the bit pattern of the frame delimiter flag does not appear in the data field of the frame (and therefore cause frame misalignment), a technique known as Bit stuffing is used by both the transmitter and the receiver.

Address field – In LAPB, this field has no meaning since the protocol works in a point to point mode and the DTE network address is represented in the layer 3 packets. This byte is therefore put to a different use; it separates the link commands from the responses and can have only two values: 0x01 and 0x03. 01 identifies frames containing commands from DTE to DCE and responses to these commands from DCE to DTE. 03 are used for frames containing commands from DCE to DTE and for responses from DTE to DCE.

Control field – it serves to identify the type of the frame. In addition, it includes sequence numbers, control features and error tracking according to the frame type.

Modes of operation

LAPB works in the Asynchronous Balanced Mode (ABM). This mode is balanced (i.e., no master/slave relationship) and is signified by the SABM (E)/SM frame. Each station may initialize, supervise, recover from errors, and send frames at any time. The DTE and DCE are treated as equals.

FCS – The Frame Check Sequence enables a high level of physical error control by allowing the integrity of the transmitted frame data to be checked.

Window size – LAPB supports an extended window size (modulo 128 and modulo 32768) where the maximum number of outstanding frames for acknowledgment is raised from 7 (modulo 8) to 127 (modulo 128) and 32767 (modulo 32768).

Protocol operation

LAPB has no master/slave node relationships. The sender uses the Poll bit in command frames to insist on an immediate response. In the response frame this same bit becomes the receivers Final bit. The receiver always turns on the Final bit in its response to a command from the sender with the Poll bit set. The P/F bit is generally used when either end becomes unsure about proper frame sequencing because of a possible missing acknowledgment, and it is necessary to re-establish a point of reference. It is also used to trigger an acknowledgment of outstanding I-frames.

Protocol verification:

Finite State Machine Models

A **finite-state machine (FSM)** or **finite-state automaton** (plural: automata), or simply a **state machine**, is a mathematical model of computation used to design both computer programs and sequential logic circuits. It is conceived as an abstract machine that can be in one of a finite number of states. The machine is in only one state at a time; the state it is in at any given time is called the current state. It can change from one state to another when initiated by a triggering event or condition; this is called a transition. A particular FSM is defined by a list of its states, and the triggering condition for each transition.

Finite-state machines can model a large number of problems, among which are electronic design automation, communication protocol design, language parsing and other engineering applications. In

biology and artificial intelligence research, state machines or hierarchies of state machines have been used to describe neurological systems. In linguistics, they are used to describe simple parts of the grammars of natural languages.

The FSM Consist of

- **States** are those instants that the protocol machine is waiting for, the next event to happen e.g. waiting for ACK.
- **Transitions** occur when some event happens. E.g. when a frame is sent, when a frame is arriving, when timer goes off, when an interrupt occurs.
- **Initial State** gives description of the system i.e. when it starts running.
- A **deadlock** is a situation in which the protocol can make no more forward progress, there exists a set of states from which there is no exit and no progress can be made.

How to know a protocol really works → specifies and verify protocol using, e.g. finite state machine

–Each protocol machine (sender or receiver) is at a specific state at every time instant

–Each state has zero or more possible transitions to other states

–One particular state is initial state: from initial state, some or possibly all other states may be reachable by a sequence of transitions.

• Simplex stop and wait ARQ protocol:

–State SRC: S = 0, 1 → which frame sender is sending;

R = 0, 1 → which frame receiver is expecting;

C = 0, 1, A (ACK), – (empty) → channel state, i.e. what is in channel

There are 9 transitions

Transition	Who runs?	Frame Accepted	Frame Emitted	To Network Layer
0	–	Frame lost	Frame lost	
1	R	0	A	–
2	S	A	1	Yes
3	R	1	A	–
4	S	A	0	Yes
5	R	0	A	–
6	R	1	A	No
7	S	Time out	0	No
8	S	Time out	1	–

Table 2.1 List of Transitions

–Initial state (000): sender has just sent frame 0, receiver is expecting frame 0, and frame 0 is currently in channel

–Transition 0: States channel losing its content.

–Transition 1: consists of channel correctly delivering packet 0 to receiver, and receiver expecting frame 1 and emitting ACK 0. Also receiver delivering packet 0 to the network layer.

–During normal operation, transitions 1,2,3,4 are repeated in order over and over: in each cycle, two frames are delivered, bringing sender back to initial state.

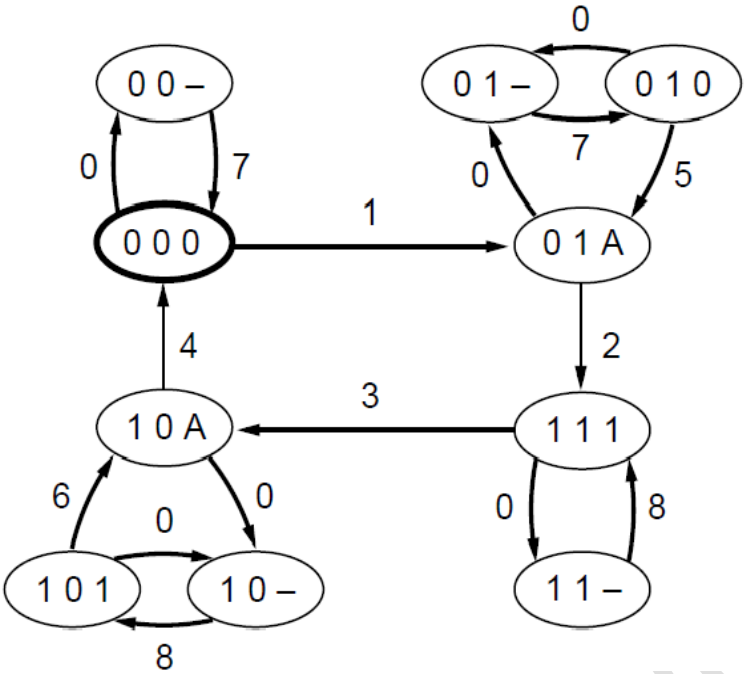


Fig 2.27 FSM for Stop and Wait Protocol (Half Duplex)

Petri Net

Petri Net(PN) is an abstract model to show the interaction between asynchronous processes. It is only one of the many ways to represent these interactions. Asynchronous means that the designer doesn't know when the processes start and in which sequence they'll take place. A common manner to visualize the concepts is with the use of places, tokens, transitions and arcs. We refer to the basics of Petri Net for a first introduction in notations. We want to mention that a transition can only fire when there are tokens in every input-place. When it fires, one token is taken from every input-place and every output-place from the transition gets an (extra) token.

The Basics:

A Petri Net is a collection of directed arcs connecting places and transitions. Places may hold tokens. The state or marking of a net is its assignment of tokens to places. Here is a simple net containing all components of a Petri Net:

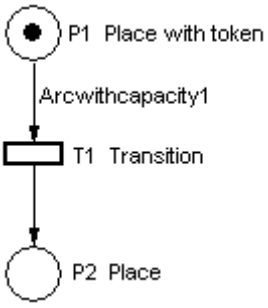


Fig 2.28 Petri Net Model

Arcs have capacity 1 by default; if other than 1, the capacity is marked on the arc. Places have infinite capacity by default, and transitions have no capacity, and cannot store tokens at all. With the rule that arcs can only connect places to transitions and vice versa, we have all we need to begin using Petri Nets. A few other features and considerations will be added as we need them.

A transition is enabled when the number of tokens in each of its input places is at least equal to the arc weight going from the place to the transition. An enabled transition may fire at any time. When fired, the tokens in the input places are moved to output places, according to arc weights and place capacities. This results in a new marking of the net, a state description of all places.

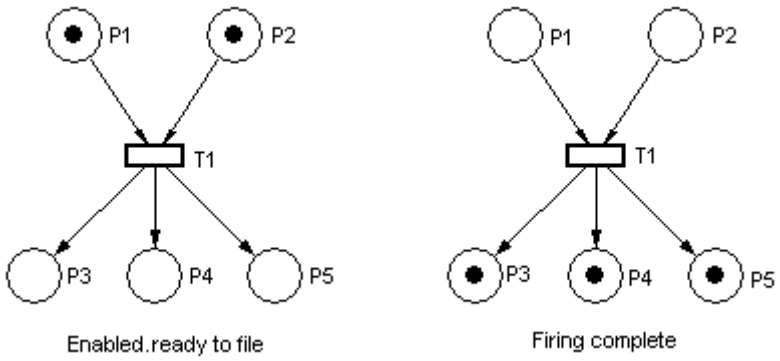


Fig 2.29

When arcs have different weights, we have what might at first seem confusing behaviour. Here is a similar net, ready to fire:

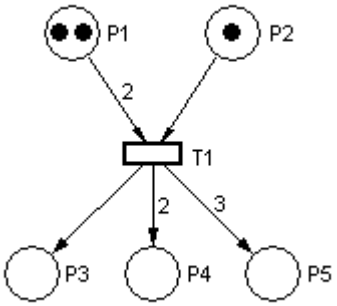


Fig 2.30

and here it is after firing:

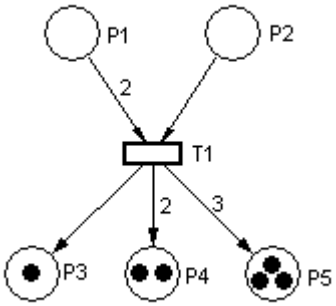


Fig 2.31

When a transition fires, it takes the tokens that enabled it from the input places; it then distributes tokens to output places according to arc weights. If the arc weights are all the same, it appears that tokens are moved across the transition. If they differ, however, it appears that tokens may disappear or be created. That, in fact, is what happens; think of the transition as removing its enabling tokens and producing output tokens according to arc weight.

A special kind of arc, the inhibitor arc, is used to reverse the logic of an input place. With an inhibitor arc, the absence of a token in the input place enables, not the presence:

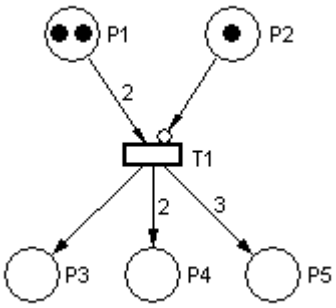


Fig 2.32

This transition cannot fire, because the token in P2 inhibits it.

Tokens can play the following roles:

A physical object: a robot;

- An information object: a message between two robots;
- A collection of objects: the people mover;
- An indicator of a state: the state in which a robot is: defender/attacker;
- An indicator of a condition: a token indicates whether a certain condition is fulfilled (ex. Soccer game starts when the referee gives the signal).

Transitions can play the following roles:

- An event: start a thread, the switching of a machine from normal to safe mode;
- A transformation of an object: a robot that changes his role, see further;
- A transport of an object: the ball is passed between the robots.

An arc connects only places and transitions and indicates the direction in which the token travels.

Petri net Link https://www.youtube.com/watch?v=EmYVZuczJ6k
Finite State Machine Models https://www.youtube.com/watch?v=hJIST1cEf6A
SDLC AND HDLC https://www.youtube.com/watch?v=_fwVTFO-u4g

ARP:

ARP or Address Resolution Protocol is a simple communications protocol used primarily today in IP and Ethernet networks. It's main purpose is to discover and associate IP addresses to physical MAC hardware addresses. ARP is used to find the MAC address of device on a network using only the IP address. The ARP protocol will make a broadcast out to the network asking for the MAC address of the destination IP address. The machine with the IP address will respond with its MAC address. The communication then drops to the link layer for physical to physical data communication between computers. ARP's job is to basically discover and associate IP addresses to physical MAC addresses.

RARP:

RARP (Reverse ARP) is a legacy protocol that has now been replaced by BOOTP and later by DHCP. Its purpose was for diskless workstations (i.e no ability to store an IP address) to discover what their own IP address was - based on their MAC address. At the point of boot, the workstation would send a request requesting its IP, a RARP server would then respond with the appropriate IP. For example:

RARP Request: What is my IP address (MAC address is within Ethernet header)? RARP Response: Your IP address is 192.168.1.11.

The main problem with RARP was that:

- The RARP server needed to be populated with the MAC to IP mappings.
- No additional data (DNS, NTP) could be sent other than the IP address.

- It only operates within a broadcast domain.

RARP was, therefore, superseded by BOOTP. However, BOOTP still required a static mapping to be defined (MAC to IP). DHCP was then built upon BOOTP with the ability to use a pool of addresses.

GARP:

In more advanced networking situations you may run across something known as Gratuitous ARP (GARP). A gratuitous arp something that is often performed by a computer when it is first booted up. When a NIC's is first powered on, it will do what's known as a gratuitous ARP and automatically ARP out it's MAC address to the entire network. This allows any switches to know the location of the physical devices and DHCP servers to know where to send an IP address if needed and requested. Gratuitous ARP is also used by many high availability routing and load balancing devices. Routers or load balancers are often configured in an HA (high availability) pair to provide optimum reliability and maximum uptime. Usually these devices will be configured in an Active/Standby pair. One device will be active while the second will be sleeping waiting for the active device to fail. Think of it as an understudy for the lead role in a movie. If the leading lady gets sick, the understudy will gladly and quickly take her place in the lime light.

When a failure occurs, the standby device will assert itself as the new active device and issue a gratuitous ARP out to the network instructing all other devices to send traffic to it's MAC address instead of the failed device.

RGPVNOTES.IN

We hope you find these notes useful.

You can get previous year question papers at https://qp.rgpvnotes.in .

If you have any queries or you want to submit your study notes please write us at rgpvnotes.in@gmail.com

LIKE & FOLLOW US ON FACEBOOK
facebook.com/rgpvnotes.in

RGPVNOTES.IN

Program : **B.Tech**

Subject Name: **Computer Networks**

Subject Code: **CS-602**

Semester: **6th**

Department of Computer Science and Engineering
CS602 Computer Networks
Subject Notes: UNIT-III

***Syllabus*: MAC Sub layer: MAC Addressing, Binary Exponential Back-off (BEB) Algorithm, Distributed Random Access Schemes/Contention Schemes: for Data Services (ALOHA and Slotted-ALOHA), for Local-Area Networks (CSMA, CSMA/CD, CSMA/CA), Collision Free Protocols: Basic Bit Map, BRAP, Binary Count Down, MLMA Limited Contention Protocols: Adaptive Tree Walk, Performance Measuring Metrics. IEEE Standards 802 series & their variant.**

MAC Sublayer

In the seven-layer OSI model of computer networking, media access control (MAC) data communication protocol is a sublayer of the data link layer (layer 2). The MAC sublayer provides addressing and channel access control mechanisms that make it possible for several terminals or network nodes to communicate within a multiple access network that incorporates a shared medium, e.g. Ethernet. The hardware that implements the MAC is referred to as a media access controller.

The MAC sublayer acts as an interface between the logical link control (LLC) sublayer and the network's physical layer. The MAC layer emulates a full-duplex logical communication channel in a multi-point network. This channel may provide unicast, multicast or broadcast communication service.

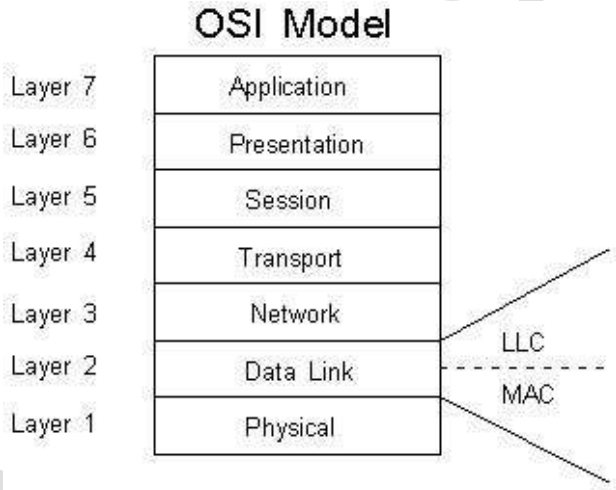


Fig.3.1 MAC Sub Layer

MAC Addressing (Media Access Control address)

In a local area network (LAN) or other network, the MAC (Media Access Control) address is your computer's unique hardware number.
In a local area network (LAN) or other network, the MAC (Media Access Control) address is your computer's unique hardware number. (On an Ethernet LAN, it's the same as your Ethernet address.) When you're connected to the Internet from your computer (or host as the Internet protocol thinks of it), a correspondence table relates your IP address to your computer's physical (MAC) address on the LAN.

What Is a MAC Address?

The MAC address is a unique value associated with a network adapter. MAC addresses are also known as hardware addresses or physical addresses. They uniquely identify an adapter on a LAN.
MAC addresses are 12-digit hexadecimal numbers (48 bits in length). By convention, MAC addresses are usually written in one of the following two formats:
MM:MM:MM:SS:SS:SS
MM-MM-MM-SS-SS-SS

The first half of a MAC address contains the ID number of the adapter manufacturer. These IDs are regulated by an Internet standards body. The second half of a MAC address represents the serial number assigned to the adapter by the manufacturer.
In the example, 00:A0:C9:14:C8:29 The prefix 00A0C9 indicates the manufacturer is Intel Corporation.

Why MAC Addresses?
Recall that TCP/IP and other mainstream networking architectures generally adopt the OSI model. In this model, network functionality is subdivided into layers. MAC addresses function at the data link layer (layer 2 in the OSI model). They allow computers to uniquely identify themselves on a network at this relatively low level.
MAC vs. IP Addressing
Whereas MAC addressing works at the data link layer, IP addressing functions at the network layer (layer 3). It's a slight oversimplification, but one can think of IP addressing as supporting the software implementation and MAC addresses as supporting the hardware implementation of the network stack. The MAC address generally remains fixed and follows the network device, but the IP address changes as the network device moves from one network to another
IP networks maintain a mapping between the IP address of a device and its MAC address. This mapping is known as the ARP cache or ARP table. ARP, the Address Resolution Protocol, supports the logic for obtaining this mapping and keeping the cache up to date.
DHCP also usually relies on MAC addresses to manage the unique assignment of IP addresses to devices.

Link MAC Address: https://www.youtube.com/watch?v=W52Wt1LDweQ

Binary Exponential Back-off (BEB) Algorithm
In a variety of computer networks, binary exponential back off or truncated binary exponential back off refers to an algorithm used to space out repeated retransmissions of the same block of data, often as part of network congestion avoidance.
Examples are the retransmission of frames in carrier sense multiple access with collision avoidance (CSMA/CA) and carrier sense multiple access with collision detection(CSMA/CD) networks, where this algorithm is part of the channel access method used to send data on these networks. In Ethernet networks, the algorithm is commonly used to schedule retransmissions after collisions. The retransmission is delayed by an amount of time derived from the slot time and the number of attempts to retransmit.
After c collisions, a random number of slot times between 0 and 2c - 1 is chosen. For the first collision, each sender will wait 0 or 1 slot times. After the second collision, the senders will wait anywhere from 0 to 3 slot times (inclusive). After the third collision, the senders will wait anywhere from 0 to 7 slot times (inclusive), and so forth. As the number of retransmission attempts increases, the number of possibilities for delay increases exponentially.

Link: https://www.youtube.com/watch?v=WeGNeUHYv5g

Distributed Random Access Schemes/Contention Schemes: for Data Services (ALOHA and Slotted ALOHA)
ALOHA: ALOHA is a system for coordinating and arbitrating access to a shared communication Networks channel. The original system used for ground-based radio broadcasting, but the system has been implemented in satellite communication systems.
A shared communication system like ALOHA requires a method of handling collisions that occur when two or more systems attempt to transmit on the channel at the same time. In the ALOHA system, a node transmits whenever data is available to send. If another node transmits at the same time, a collision occurs, and the frames that were transmitted are lost. However, a node can listen to broadcasts on the medium, even its own, and determine whether the frames were transmitted.

Aloha means "Hello". Aloha is a multiple access protocol at the datalink layer and proposes how multiple terminals access the medium without interference or collision. A scientest developed a protocol that would increase the capacity of aloha two-fold. The Slotted Aloha protocol involves dividing the time interval into discrete slots and each slot interval corresponds to the time period of one frame. This method requires synchronization between the sending nodes to prevent collisions.

There are two different versions

Types of ALOHA:

(i) Pure ALOHA

(ii) Slotted ALOHA

(i) Pure ALOHA

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

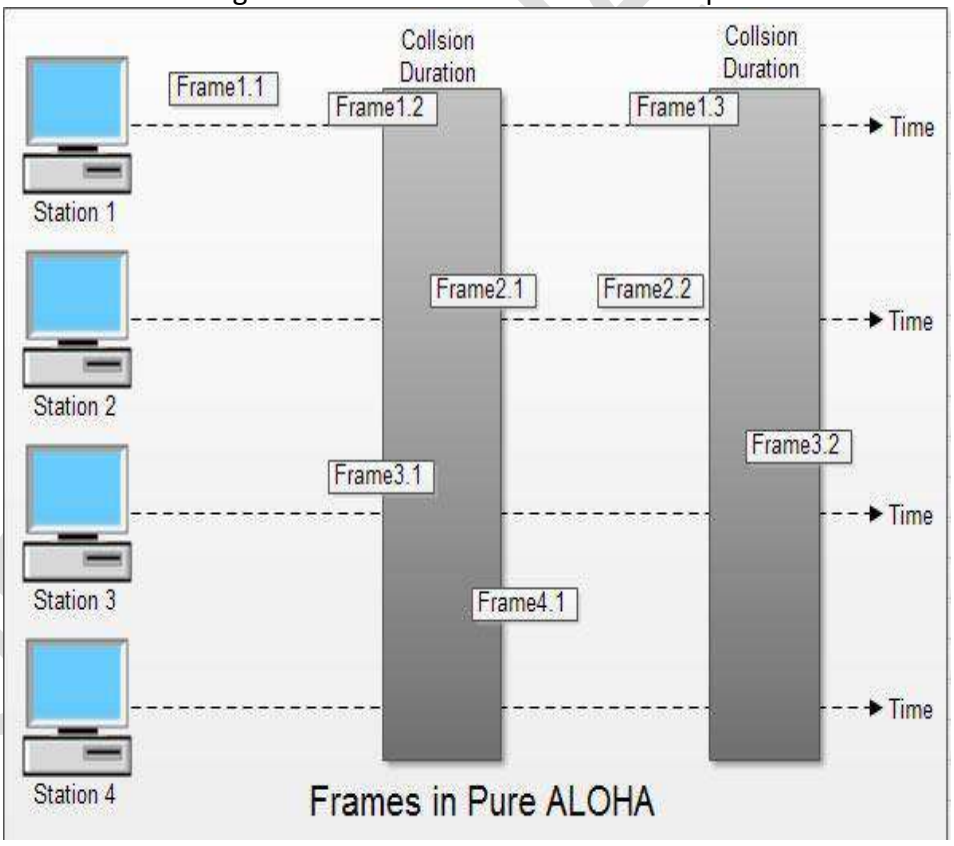


Fig 3.2 Pure ALOHA

(ii) Slotted ALOHA

- Slotted ALOHA was invented to improve the efficiency of pure ALOHA as chances of collision in pure ALOHA are very high.
- In slotted ALOHA, the time of the shared channel is divided into discrete intervals called slots.
- The stations can send a frame only at the beginning of the slot and only one frame is sent in each slot.
- In slotted ALOHA, if any station is not able to place the frame onto the channel at the beginning of the slot i.e. it misses the time slot then the station has to wait until the beginning of the next time slot.

- In slotted ALOHA, there is still a possibility of collision if two stations try to send at the beginning of the same time slot.
- Slotted ALOHA still has an edge over pure ALOHA as chances of collision are reduced to one-half.

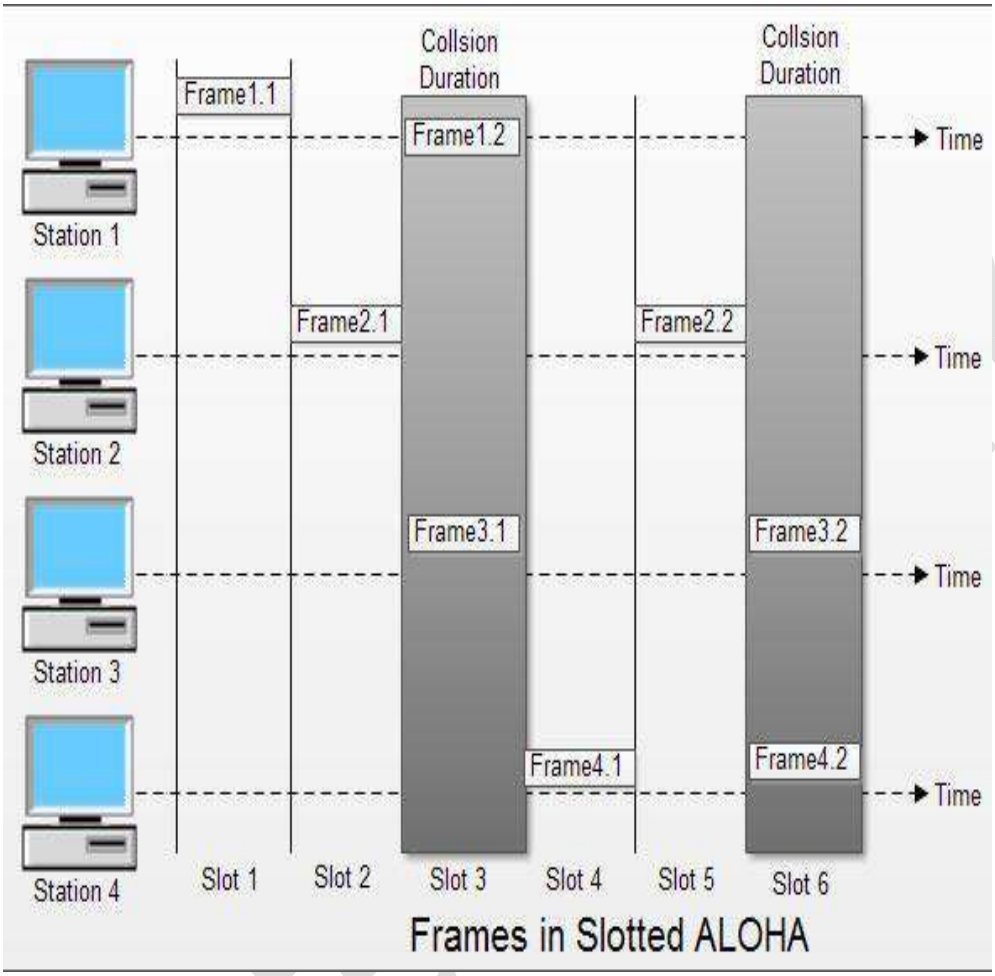


Fig 3.3 Slotted ALOHA

Link ALOHA: https://www.youtube.com/watch?v=c39k2clZU74

For Local-Area Networks (CSMA, CSMA/CD, CSMA/CA)

Carrier sense multiple access (CSMA) is a probabilistic media access control (MAC) protocol in which a node verifies the absence of other traffic before transmitting on a shared transmission medium, such as an electrical bus, or a band of the electromagnetic spectrum.

Carrier sense means that a transmitter uses feedback from a receiver to determine whether another transmission is in progress before initiating a transmission. That is, it tries to detect the presence of a carrier wave from another station before attempting to transmit. If a carrier is sensed, the station waits for the transmission in progress to finish before initiating its own transmission. In other words, CSMA is based on the principle "sense before transmit" or "listen before talk".

Multiple access means that multiple stations send and receive on the medium. Transmissions by one node are generally received by all other stations connected to the medium.

CSMA protocol was developed to overcome the problem found in ALOHA i.e. to minimize the chances of collision, so as to improve the performance. CSMA protocol is based on the principle of 'carrier sense'. The station senses the carrier or channel before transmitting a frame. It means the station checks the state of channel, whether it is idle or busy.

Even though devices attempt to sense whether the network is in use, there is a good chance that two stations will attempt to access it at the same time. On large networks, the transmission time between one end of the cable and another is enough that one station may access the cable even though another has already just accessed it.

The chances of collision still exist because of propagation delay. The frame transmitted by one station takes some time to reach other stations. In the meantime, other stations may sense the channel to be idle and transmit their frames. This results in the collision.

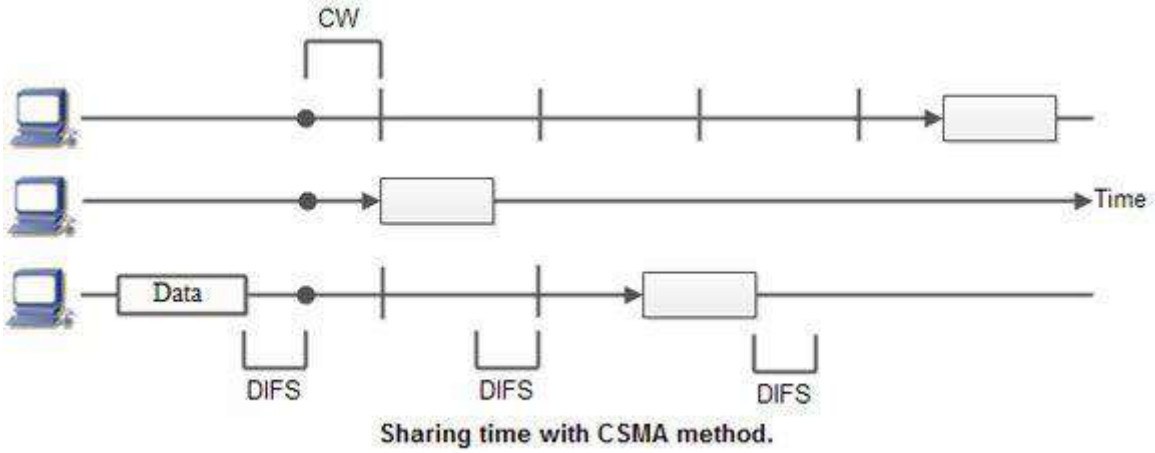


Fig 3.4 CSMA

There Are Three Different Type of CSMA Protocols

(I) 1-persistent CSMA
(ii) Non- Persistent CSMA
(iii) p-persistent CSMA

(i) 1-persistent CSMA

- In this method, station that wants to transmit data continuously senses the channel to check whether the channel is idle or busy.
- If the channel is busy, the station waits until it becomes idle.
- When the station detects an idle-channel, it immediately transmits the frame with probability 1. Hence it is called 1-persistent CSMA.
- This method has the highest chance of collision because two or more stations may find channel to be idle at the same time and transmit their frames.
- When the collision occurs, the stations wait a random amount of time and start allover again.

Drawback of I-persistent

The propagation delay time greatly affects this protocol. Let us suppose, just after the station 1 begins its transmission, station 2 also became ready to send its data and senses the channel. If the station 1 signal has not yet reached station 2, station 2 will sense the channel to be idle and will begin its transmission. This will result in collision.

Even if propagation delay time is zero, collision will still occur. If two stations became .ready in the middle of third station's transmission, both stations will wait until the transmission of first station ends and then both will begin their transmission exactly simultaneously. This will also result in collision.

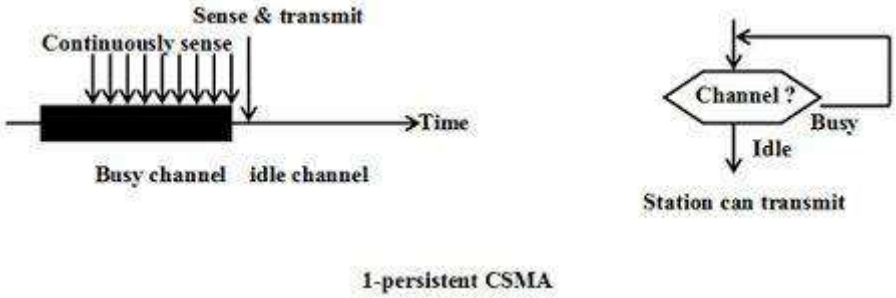


Fig 3.5 1-persistent CSMA

(ii) Non-persistent CSMA

- In this scheme, if a station wants to transmit a frame and it finds that the channel is busy (some other station is transmitting) then it will wait for fixed interval oftime.

- After this time, it again checks the status of the channel and if the channel is.free it will transmit.
- A station that has a frame to send senses the channel.
- If the channel is idle, it sends immediately.
- If the channel is busy, it waits a random amount of time and then senses the channel again.
- In non-persistent CSMA the station does not continuously sense the channel for the purpose of capturing it when it detects the end of previous transmission.

Advantage of non-persistent

- It reduces the chance of collision because the stations wait a random amount of time. It is unlikely that two or more stations will wait for same amount of time and will retransmit at the same time.

Disadvantage of non-persistent

- It reduces the efficiency of network because the channel remains idle when there may be stations with frames to send. This is due to the fact that the stations wait a random amount of time after the collision.

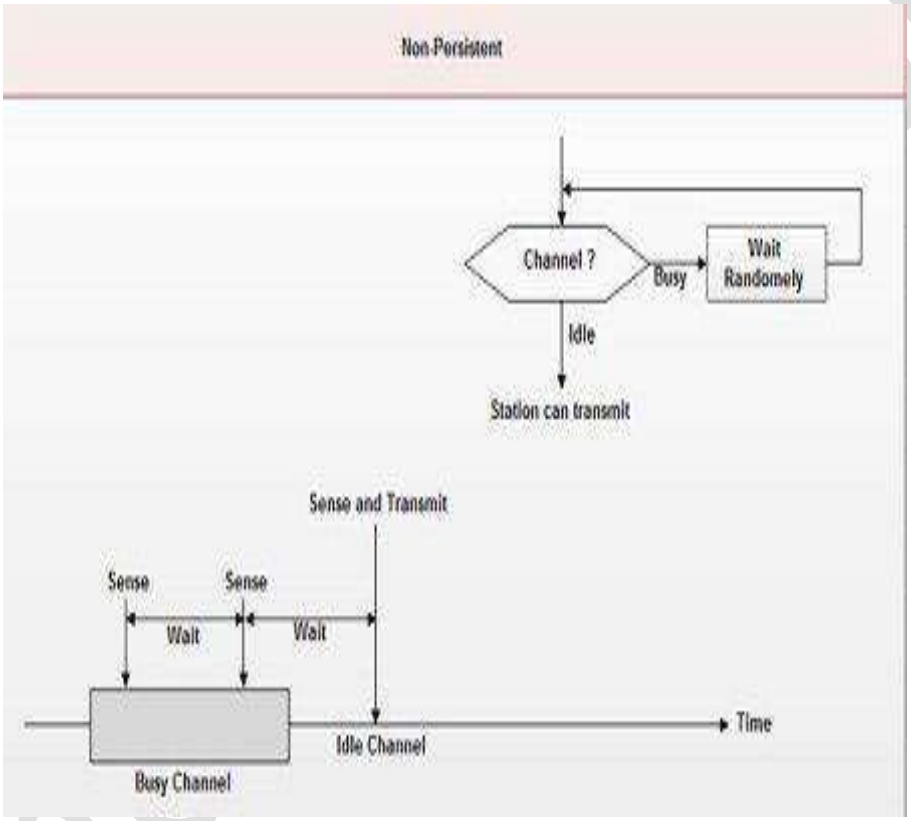


Fig 3.6 Non-persistent CSMA

(iii) p-persistent CSMA

- This method is used when channel has time slots such that the time slot duration is equal to or greater than the maximum propagation delay time.
- Whenever a station becomes ready to send, it senses the channel.
- If channel is busy, station waits until next slot.
- If channel is idle, it transmits with a probability p.
- With the probability q=1-p, the station then waits for the beginning of the next time slot.
- If the next slot is also idle, it either transmits or waits again with probabilities p and q.
- This process is repeated till either frame has been transmitted or another station has begun transmitting.

- In case of the transmission by another station, the station acts as though a collision has occurred and it waits a random amount of time and starts again.

Advantage of p-persistent

- It reduces the chance of collision and improves the efficiency of the network.

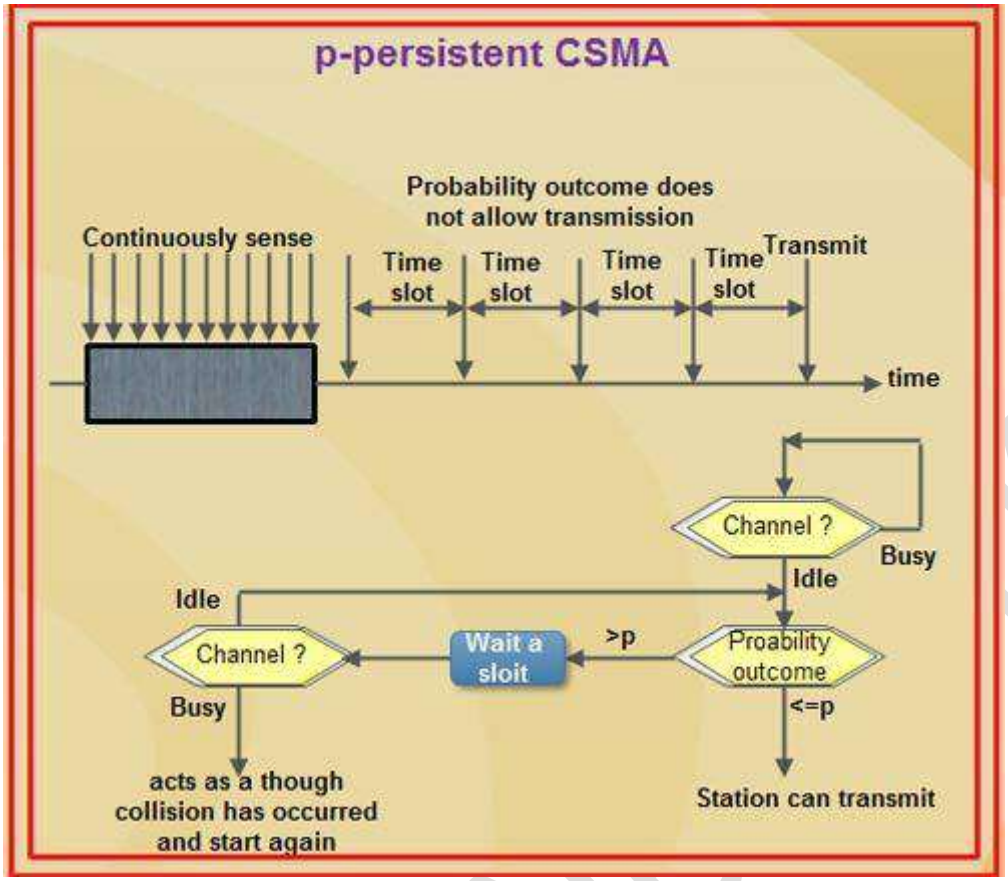


Fig 3.7 p-persistent CSMA

CSMA/CD - Carrier Sense Multiple Access / Collision Detection

To reduce the impact of collisions on the network performance, Ethernet uses an algorithm called CSMA with Collision Detection (CSMA / CD): CSMA/CD is a protocol in which the station senses the carrier or channel before transmitting frame just as in persistent and non-persistent CSMA. If the channel is busy, the station waits. it listens at the same time on communication media to ensure that there is no collision with a packet sent by another station. In a collision, the issuer immediately cancel the sending of the package. This allows to limit the duration of collisions: we do not waste time to send a packet complete if it detects a collision. After a collision, the transmitter waits again silence and again, he continued his hold for a random number; but this time the random number is nearly double the previous one: it is this called back-off (that is to say, the "decline") exponential. In fact, the window collision is simply doubled (unless it has already reached a maximum). From a packet is transmitted successfully, the window will return to its original size.

Again, this is what we do naturally in a meeting room if many people speak exactly the same time, they are realizing account immediately (as they listen at the same time they speak), and they interrupt without completing their sentence. After a while, one of them speaks again. If a new collision occurs, the two are interrupted again and tend to wait a little longer before speaking again.

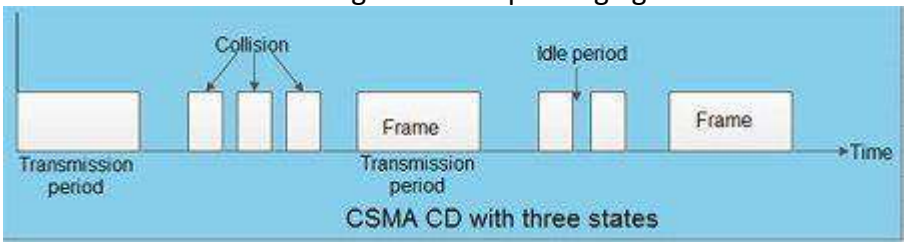


Fig 3.8 CSMA/CD

CSMA/CA - Carrier Sense Multiple Access/Collision Avoidance

CSMA/CA protocol is used in wireless networks because they cannot detect the collision so the only solution is collision avoidance.

• CSMA/CA avoids the collisions using three basic techniques.

(i) Interframe space
(ii) Contention window
(iii) Acknowledgements

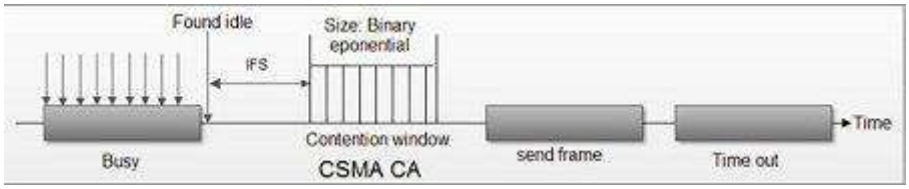


Fig 3.9 CSMA/CA

Comparision between all with an BAR Chart

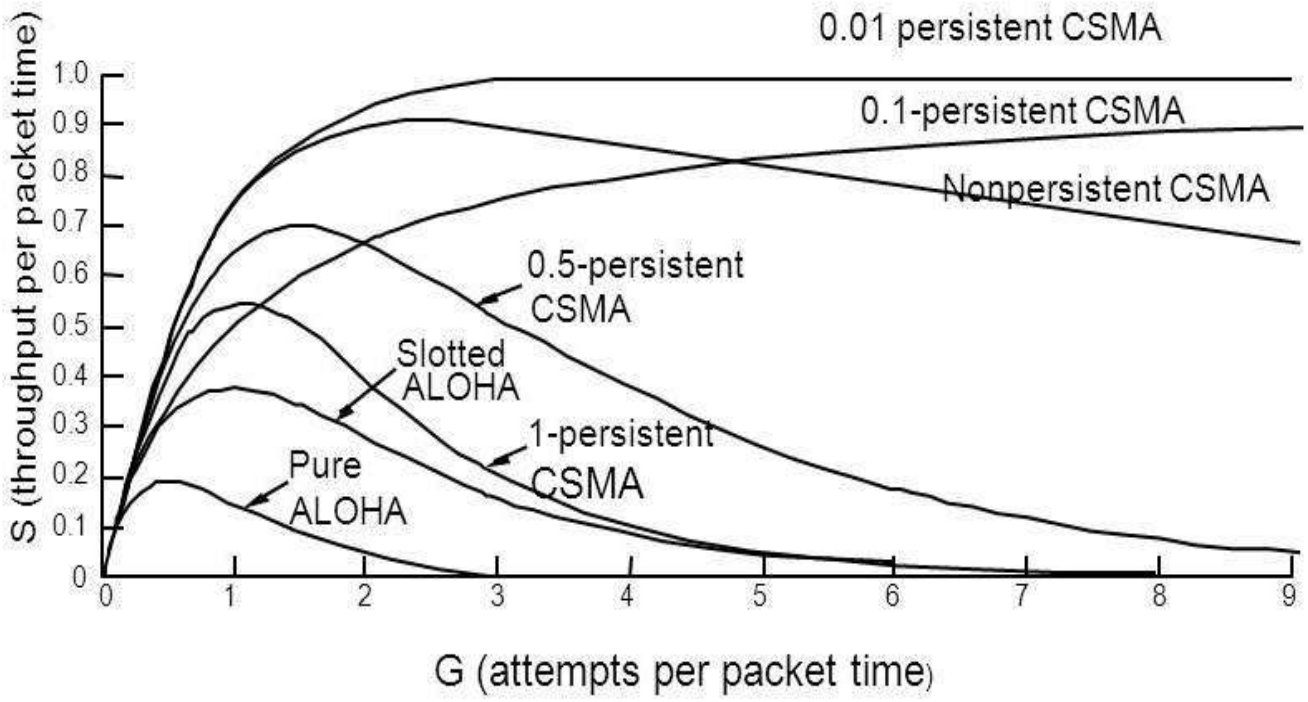


Fig 3.10 Comparision between all with an BAR Chart

1. Interframe Space (IFS)

- Whenever the channel is found idle, the station does not transmit immediately. It waits for a period of time called interframe space (IFS).
- When channel is sensed to be idle, it may be possible that same distant station may have already started transmitting and the signal of that distant station has not yet reached other stations.
- Therefore the purpose of IFS time is to allow this transmitted signal to reach other stations.
- If after this IFS time, the channel is still idle, the station can send, but it still needs to wait a time equal to contention time.
- IFS variable can also be used to define the priority of a station or a frame.

2. Contention Window

- Contention window is an amount of time divided into slots.
- A station that is ready to send chooses a random number of slots as its wait time.
- The number of slots in the window changes according to the binary exponential back-off strategy. It means that it is set of one slot the first time and then doubles each time the station cannot detect an idle channel after the IFS time.
- This is very similar to the p-persistent method except that a random outcome defines the number of slots taken by the waiting station.
- In contention window the station needs to sense the channel after each time slot.

• If the station finds the channel busy, it does not restart the process. It just stops the timer & restarts it when the channel is sensed as idle.

3. Acknowledgement

• Despite all the precautions, collisions may occur and destroy the data.

• The positive acknowledgment and the time-out timer can help guarantee that receiver has received the frame.

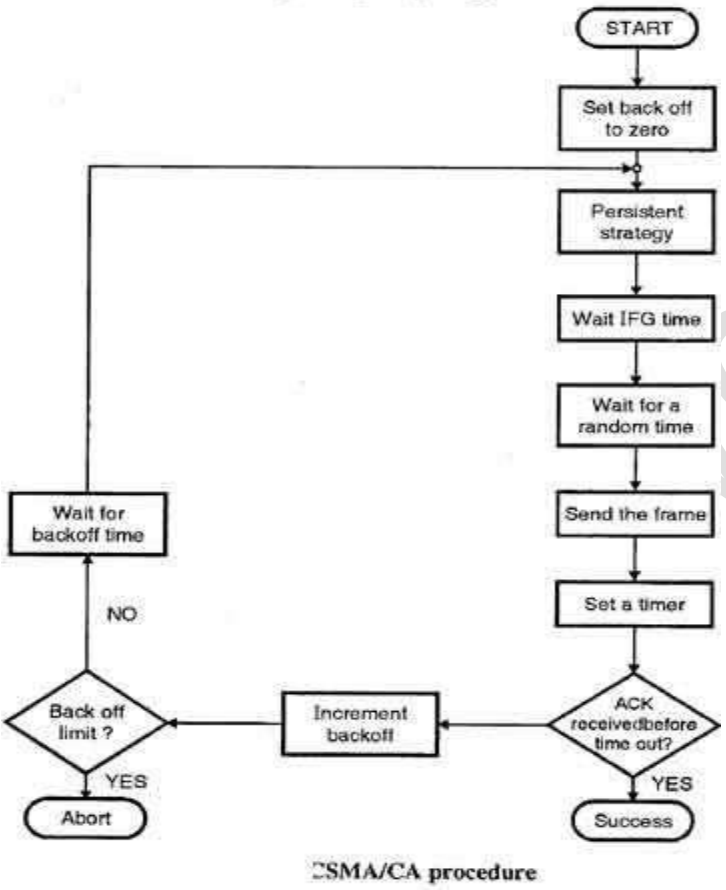


Fig 3.11 Flow Chart of CSMA/CA

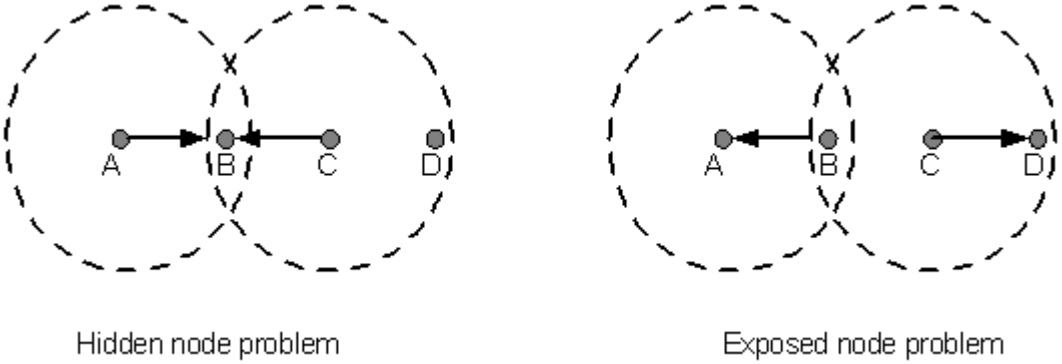
LINK: https://www.youtube.com/watch?v=74zIRH-bj2c

Hidden Node Problem

In the case of wireless network it is possible that A is sending a message to B, but C is out of its range and hence while "listening" on the network it will find the network to be free and might try to send packets to B at the same time as A. So, there will be a collision at B. The problem can be looked upon as if A and C are hidden from each other. Hence it is called the "hidden node problem".

Exposed Node Problem

If C is transmitting a message to D and B wants to transmit a message to A, B will find the network to be busy as B hears C trnasmitting. Even if B would have transmitted to A, it would not have been a problem at A or D. CSMA/CD would not allow it to transmit message to A, while the two transmissions could have gone in parallel.



Hidden node problem Exposed node problem

Fig 3.12 Hidden and Exposed Node Problem

Collision Free Protocols: Basic Bit Map, BRAP, Binary Count Down

Collision Free Protocols

A collision-free protocol for transmitting frames between stations connected over a shared transmission medium such as an IEEE 802.3 Ethernet LAN. A logical ring is formed and a token is circulated among the connected stations part of the logical ring (not all connected stations are required to be part of the logical ring). Transmitting from any one station, part of the logical ring, is permitted only while holding the token, therefore preventing collisions. A collision-free protocol, over a standard Ethernet infrastructure, becomes feasible, yet remains compatible with the standard collision protocol, thus improving performances.

Basic Bit Map

1. Assume N stations are numbered from 1 to N.
2. There is a contention period of N slots (bits).
3. Each station has one slot time during the contention period, numbered 1 to N.
4. Station J sends a 1-bit reservation during Jth slot time if it wants to transmit a frame.
5. Every station sees all the 1-bit reservation transmitted during the contention period, so each station knows which stations want to transmit.
6. After the contention period, each station that asserted its desire to transmit sends its frame in the order of station number.

BRAP

Backup Route Aware Routing Program (BRAP) is a protocol that provides interdomain routing. BRAP uses reverse paths and backup paths to ensure fast failure recovery in networking systems.

Binary Countdown

In this protocol, a node which wants to signal that it has a frame to send does so by writing its address into the header as a binary number. The arbitration is such that as soon as a node sees that a higher bit position that is 0 in its address has been overwritten with a 1, it gives up. The final result is the address of the node which is allowed to send. After the node has transmitted the whole process is repeated all over again. Given below is an example situation.

Nodes	Addresses
A	0010
B	0101
C	1010
D	1001

	1010

Node C having higher priority gets to transmit. The problem with this protocol is that the nodes with higher address always wins. Hence this creates a priority which is highly unfair and hence undesirable.

MLMA protocol

Multi-Level Multi-Access (MLMA): The problem with BRAP is the delay when the channel is lightly loaded. When there is no frame to be transmitted, the N-bit headers just go on and on until a station inserts a 1 into its mini slot. On average, the waiting time would be N=2. MLAM scheme 41 is nearly as efficient under high channel load, but has shorter delay under low channel load. In MLAM, a station wants to transmit a frame sends its identification in a particular format. A group of 10 bits (called decade) is used to represent a digit of the station number 48.

Limited Contention Protocols: Adaptive Tree Walk

Contention based and Contention - free has their own problems. Under conditions of light load, contention is preferable due to its low delay. As the load increases, contention becomes increasingly less attractive, because the overload associated with channel arbitration becomes greater. Just the reverse is true for contention - free protocols. At low load, they have high delay, but as the load increases, the channel efficiency improves rather than getting worse as it does for contention protocols.

Obviously, it would be better if one could combine the best properties of the contention and contention - free protocols, that is, protocol which used contention at low loads to provide low delay, but used a contention-free technique at high load to provide good channel efficiency. Such protocols do exist and are called Limited contention protocols.

It is obvious that the probability of some station acquiring the channel could only be increased by decreasing the amount of competition. The limited contention protocols do exactly that. They first divide the stations up into (not necessarily disjoint) groups. Only the members of group 0 are permitted to compete for slot 0. The competition for acquiring the slot within a group is contention based. If one of the members of that group succeeds, it acquires the channel and transmits a frame. If there is collision or no node of a particular group wants to send then the members of the next group compete for the next slot. The probability of a particular node is set to a particular value (optimum).

Adaptive Tree Walk Protocol

Initially all the nodes are allowed to try to acquire the channel. If it is able to acquire the channel, it sends its frame. If there is collision then the nodes are divided into two equal groups and only one of these groups compete for slot 1. If one of its member acquires the channel then the next slot is reserved for the other group. On the other hand, if there is a collision then that group is again subdivided and the same process is followed. This can be better understood if the nodes are thought of as being organised in a binary tree.

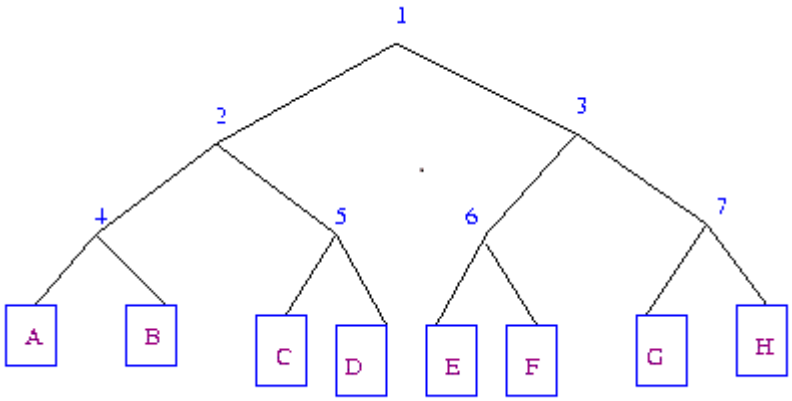


Fig 3.13 Adaptive Tree Walk

Many improvements could be made to the algorithm. For example, consider the case of nodes G and H being the only ones wanting to transmit. At slot 1 a collision will be detected and so 2 will be tried and it will be found to be idle. Hence it is pointless to probe 3 and one should directly go to 6,7.

URN Protocol

In computing, a uniform resource name (URN) is the historical name for a uniform resource identifier (URI) that uses the scheme. A URI is a string of characters used to identify a name of a web resource. Such identification enables interaction with representations of the web resource over a network, typically the World Wide Web, using specific protocols.

URNs were intended to serve as persistent, location-independent identifiers, allowing the simple mapping of namespaces into a single URN namespace. The existence of such a URI does not imply availability of the identified resource, but such URIs are required to remain globally unique and persistent, even when the resource ceases to exist or becomes unavailable.

(Uniform Resource Name) A name that identifies a resource on the Internet. Unlike URLs, which use network addresses (domain, directory path, file name), URNs use regular words that are protocol and location independent. Providing a higher level of abstraction, URNs are persistent (never change) and require a resolution service similar to the DNS system in order to convert names into real addresses. For the most part, URNs have evolved into XRI identifiers (see XDI). See URI and URL.

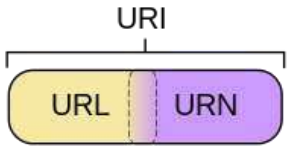


Fig 3.14 URN Protocol

High Speed LAN: Fast Ethernet, Gigabit Ethernet

Name	IEEE Standards	Data Rate	Media Type	Maximum Distance
Ethernet	802.3	10 Mbps	10Base-T	100 meters
Fast Ethernet/ 100Base-T	802.3u	100 Mbps	100Base-TX 100Base-FX	100 meters 2000 meters
Gigabit Ethernet/ GigE	802.3z	1000 Mbps	1000Base-T 1000Base-SX 1000Base-LX	100 meters 275/550 meters 550/5000 meters
10 Gigabit Ethernet	IEEE 802.3ae	10 Gbps	10GBase-SR 10GBase-LX4 10GBase-LR/ER 10GBase-SW/LW/EW	300 meters 300m MMF/ 10km SMF 10km/40km 300m/10km/40km

Key Differences between Fast Ethernet and Gigabit Ethernet

- Gigabit Ethernet is more advanced technology than Fast Ethernet having speed of 1000 Mbit/s, 10 times more than speed of Fast Ethernet, which is 100 Mbit/s.
- Due to more bit transfer speed and higher bandwidth, Gigabit Ethernet results in better performance than Fast Ethernet.
- Gigabit Ethernet is more expensive than Fast Ethernet. Upgrading of Fast Ethernet from Standard Ethernet is easy and cost effective while upgrading of Gigabit Ethernet from Fast Ethernet is complex and expensive.
- Configuration problems in Gigabit Ethernet are more complex than Fast Ethernet. Devices used in Gigabit Ethernet must have same configuration to function fully. While in Fast Ethernet, connected devices configure automatically with the system.
- Every network can support 100 Mbit/s but cannot support 1000 Mbit/s. So, specific network is required that can support the Gigabit Ethernet.
- Maximum length of 10 km network can be achieved in Fast Ethernet, if 100BASE-LX10 version is being used. While 70 km network length can be achieved in Gigabit Ethernet, if Single Mode Fiber (1,310 nm wavelength) is being used as a medium.

- Faster Ethernet runs on both optical fiber cable and unshielded twisted pair cable. Gigabit Ethernet runs on either 1000BASE-T twisted pair cable, 1000BASE-X optical fiber or 1000BASE-CX shielded balanced copper cable.
- Fast Ethernet is economical but provides the slow transfer speed as compared to the Gigabit Ethernet that provides the faster transfer rate but is very expensive. The ports of Gigabit Ethernet cost four times the price per port of Fast Ethernet.
- IEEE Standard for Gigabit Ethernet is IEEE 802.3-2008 and the IEEE Standards for Fast Ethernet are 802.3u-1995, 802.3u-1995 and 802.3u-1995.
- Upgrade from simple Ethernet to Fast Ethernet is relatively simple and economical as compared to the upgrade from Fast Ethernet to Gigabit Ethernet.
- Gigabit Ethernet requires specifically designed network devices that can support the standard 1000Mbps data rate. Fast Ethernet requires no specific network devices.
- Manual configuration is the must-have element in the setup of Gigabit Ethernet where most of the devices required prior configuration in order to be compatible with Gigabit Ethernet. While in Fast Ethernet there is no scene of configuration as connected devices automatically configured according to the requirement of Fast Ethernet.
- If you need the more bandwidth then Gigabit Ethernet will provide you the more bandwidth at the best possible frequency as compared to the Fast Ethernet.

FDDI

The Fiber Distributed Data Interface (FDDI) specifies a 100-Mbps token-passing, dual-ring LAN using fiber-optic cable. FDDI is frequently used as high-speed backbone technology because of its support for high bandwidth and greater distances than copper. It should be noted that relatively recently, a related copper specification, called Copper Distributed Data Interface (CDDI), has emerged to provide 100-Mbps service over copper. CDDI is the implementation of FDDI protocols over twisted-pair copper wire. This article focuses mainly on FDDI specifications and operations, but it also provides a high-level overview of CDDI.

FDDI uses dual-ring architecture with traffic on each ring flowing in opposite directions (called counter-rotating). The dual rings consist of a primary and a secondary ring. During normal operation, the primary ring is used for data transmission, and the secondary ring remains idle. As will be discussed in detail later in this article, the primary purpose of the dual rings is to provide superior reliability and robustness.

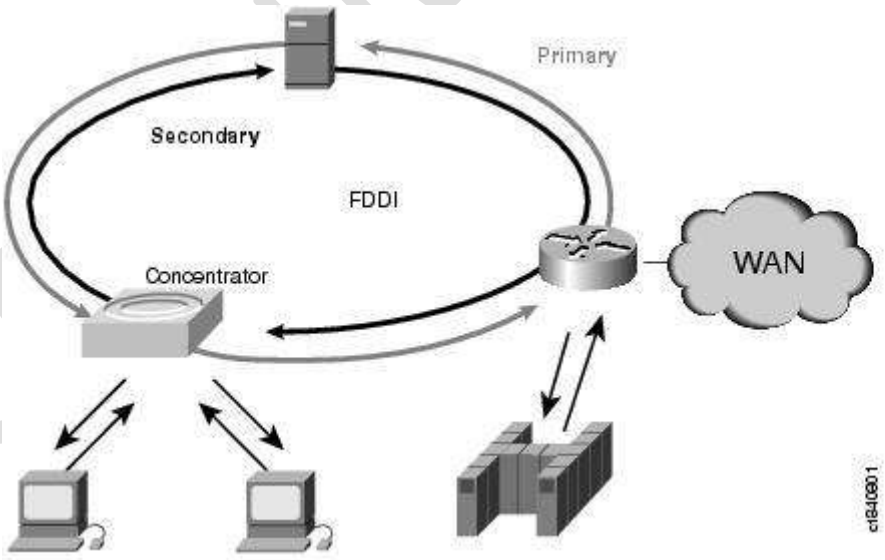


Fig 3.14 FDDI

FDDI Transmission Media

FDDI uses optical fiber as the primary transmission medium, but it also can run over copper cabling. As mentioned earlier, FDDI over copper is referred to as Copper-Distributed Data Interface (CDDI). Optical

fiber has several advantages over copper media. In particular, security, reliability, and performance all are enhanced with optical fiber media because fiber does not emit electrical signals. A physical medium that does emit electrical signals (copper) can be tapped and therefore would permit unauthorized access to the data that is transiting the medium. In addition, fiber is immune to electrical interference from radio frequency interference (RFI) and electromagnetic interference (EMI). Fiber historically has supported much higher bandwidth (throughput potential) than copper, although recent technological advances have made copper capable of transmitting at 100 Mbps. Finally, FDDI allows 2 km between stations using multimode fiber, and even longer distances using a single mode.

FDDI defines two types of optical fiber: single-mode and multimode. A mode is a ray of light that enters the fiber at a particular angle. Multimode fiber uses LED as the light-generating device, while single-mode fiber generally uses lasers.

Multimode fiber allows multiple modes of light to propagate through the fiber. Because these modes of light enter the fiber at different angles, they will arrive at the end of the fiber at different times. This characteristic is known as modal dispersion. Modal dispersion limits the bandwidth and distances that can be accomplished using multimode fibers. For this reason, multimode fiber is generally used for connectivity within a building or a relatively geographically contained environment.

Single-mode fiber allows only one mode of light to propagate through the fiber. Because only a single mode of light is used, modal dispersion is not present with single-mode fiber. Therefore, single-mode fiber is capable of delivering considerably higher performance connectivity over much larger distances, which is why it generally is used for connectivity between buildings and within environments that are more geographically dispersed.

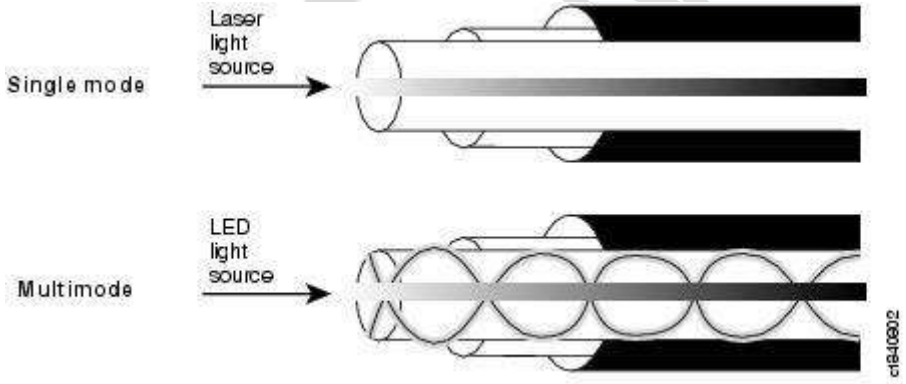


Fig 3.14 FDDI transmission medium

Performance Measuring Metrics

- **Latency:** It can take a long time for a packet to be delivered across intervening networks. In reliable protocols where a receiver acknowledges delivery of each chunk of data, it is possible to measure this as round-trip time.
- **Packet loss:** In some cases, intermediate devices in a network will lose packets. This may be due to errors, to overloading of the intermediate network, or to intentional discarding of traffic in order to enforce a particular service level.
- **Retransmission**: When packets are lost in a reliable network, they are retransmitted. This incurs two delays: First, the delay from re-sending the data; and second, the delay resulting from waiting until the data is received in the correct order before forwarding it up the protocol stack.
- **Throughput:** The amount of traffic a network can carry is measured as throughput, usually in terms such as kilobits per second. Throughput is analogous to the number of lanes on a highway, whereas latency is analogous to its speed limit.

IEEE Standards 802 series & their variant

802.2 Logical Link Control

The technical definition for 802.2 is "the standard for the upper Data Link Layer sublayer also known as the Logical Link Control layer. It is used with the 802.3, 802.4, and 802.5 standards (lower DL sublayers)."

802.2 "specifies the general interface between the network layer (IP, IPX, etc) and the data link layer (Ethernet, Token Ring, etc).

Basically, think of the 802.2 as the "translator" for the Data Link Layer. 802.2 is concerned with managing traffic over the physical network. It is responsible for flow and error control. The Data Link Layer wants to send some data over the network, 802.2 Logical Link Control helps make this possible. It also helps by identifying the line protocol, like NetBIOS, or Netware.

The LLC acts like a software bus allowing multiple higher layer protocols to access one or more lower layer networks. For example, if you have a server with multiple network interface cards, the LLC will forward packers from those upper layer protocols to the appropriate network interface. This allows the upper layer protocols to not need specific knowledge of the lower layer networks in use.

802.3 Ethernet

802.3 is the standard which Ethernet operates by. It is the standard for CSMA/CD (Carrier Sense Multiple Access with Collision Detection). This standard encompasses both the MAC and Physical Layer standards.

CSMA/CD is what Ethernet uses to control access to the network medium (network cable). If there is no data, any node may attempt to transmit, if the nodes detect a collision, both stop transmitting and wait a random amount of time before retransmitting the data.

The original 802.3 standard is 10 Mbps (Megabits per second). 802.3u defined the 100 Mbps (Fast Ethernet) standard, 802.3z/802.3ab defined 1000 Mbps Gigabit Ethernet, and 802.3ae define 10 Gigabit Ethernet.

Commonly, Ethernet networks transmit data in packets, or small bits of information. A packet can be a minimum size of 72 bytes or a maximum of 1518 bytes.

The most common topology for Ethernet is the star topology.

802.5 Token Ring

Token ring is designed to use the ring topology and utilizes a token to control the transmission of data on the network.

The token is a special frame which is designed to travel from node to node around the ring. When it does not have any data attached to it, a node on the network can modify the frame, attach its data and transmit. Each node on the network checks the token as it passes to see if the data is intended for that node, if it is; it accepts the data and transmits a new token. If it is not intended for that node, it retransmits the token on to the next node.

The token ring network is designed in such a way that each node on the network is guaranteed access to the token at some point. This equalizes the data transfer on the network. This is different from an Ethernet network where each workstation has equal access to grab the available bandwidth, with the possible of a node using more bandwidth than other nodes.

Originally, token ring operated at a speed of about 4 Mbps and 16 Mbps. 802.5t allows for 100 Mbps speeds and 802.5v provides for 1 Gbps over fibber.

Token ring can be run over a star topology as well as the ring topology.

There are three major cable types for token ring: Unshielded twisted pair (UTP), Shielded twisted pair (STP), and fibber.

Token ring utilizes a Multi-station Access Unit (MAU) as a central wiring hub. This is also sometimes called a MSAU when referring to token ring networks.

802.11 Wireless Network Standards

802.11 is the collection of standards setup for wireless networking. You are probably familiar with the three popular standards: 802.11a, 802.11b, 802.11g and latest one is 802.11n. Each standard uses a frequency to connect to the network and has a defined upper limit for data transfer speeds.

802.11a was one of the first wireless standards. 802.11a operates in the 5Ghz radio band and can achieve a maximum of 54Mbps. Wasn't as popular as the 802.11b standard due to higher prices and lower range.

802.11b operates in the 2.4Ghz band and supports up to 11 Mbps. Range of up to several hundred feet in theory. The first real consumer option for wireless and very popular.

802.11g is a standard in the 2.4Ghz band operating at 54Mbps. Since it operates in the same band as 802.11b, 802.11g is compatible with 802.11b equipment. 802.11a is not directly compatible with 802.11b or 802.11g since it operates in a different band.

Wireless LANs primarily use CSMA/CA - Carrier Sense Multiple Access/Collision Avoidance. It has a "listen before talk" method of minimizing collisions on the wireless network. This results in less need for retransmitting data.

Wireless standards operate within a wireless topology.

RGPVNOTES.IN

We hope you find these notes useful.

You can get previous year question papers at https://qp.rgpvnotes.in .

If you have any queries or you want to submit your study notes please write us at rgpvnotes.in@gmail.com

LIKE & FOLLOW US ON FACEBOOK
facebook.com/rgpvnotes.in

RGPVNOTES.IN

Program : **B.Tech**

Subject Name: **Computer Networks**

Subject Code: **CS-602**

Semester: **6th**

Department of Computer Science and Engineering
CS602 Computer Networks
Subject Notes: UNIT-IV

Syllabus: **Network Layer: Need, Services Provided , Design issues, Routing algorithms: Least CostRouting algorithm, Dijkstra's algorithm, Bellman-ford algorithm, Hierarchical Routing,Broadcast Routing, Multicast Routing. IP Addresses, Header format, Packet forwarding,Fragmentation and reassembly, ICMP, Comparative study of IPv4 & IPv6.**

Network Layer: Need

The network layer is considered the backbone of the OSI Model. It selects and manages the best logical path for data transfer between nodes. This layer contains hardware devices such as routers, bridges, firewalls and switches, but it actually creates a logical image of the most efficient communication route and implements it with a physical medium.

Network layer protocols exist in every host or router. The router examines the header fields of all the IP packets that pass through it. Internet Protocol and Netware IPX/SPX are the most common protocols associated with the network layer.

In the OSI model, the network layer responds to requests from the layer above it (transport layer) and issues requests to the layer below it (data link layer).

Network Layer: Services

It translates logical network address into physical address.

1. Routers and gateways operate in the network layer. Mechanism is provided by Network Layer for routing the packets to final destination.
2. Connection services are provided including flow control, error control and packet sequence control.
3. Breaks larger packets into small packets.

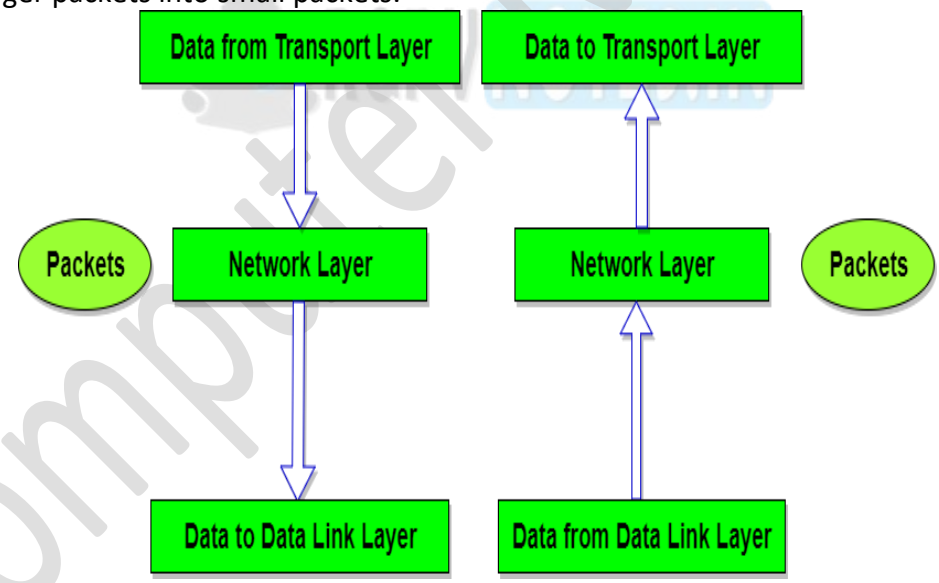


Fig 4.1 Network Layer

There are two types of service that can be provided by the network layer:

1. An unreliable connectionless service.
2. A connection-oriented, reliable or unreliable, service.

Network Layer: **Design issues**

a) Store-and-Forward Packet Switching
b) Services Provided to the Transport Layer
c) Implementation of Connectionless Service
d) Implementation of Connection-Oriented Service

a) Store-and-Forward Packet Switching

A host with a packet to send transmits it to the nearest router, either on its own LAN or over a point-to-pointlink to the carrier. The packet is stored there until it has fully arrived so the checksum can be verified. Then it is forwarded to the next router along the path until it reaches the destination host, where it is delivered. This mechanism is store-and-forward packet switching.

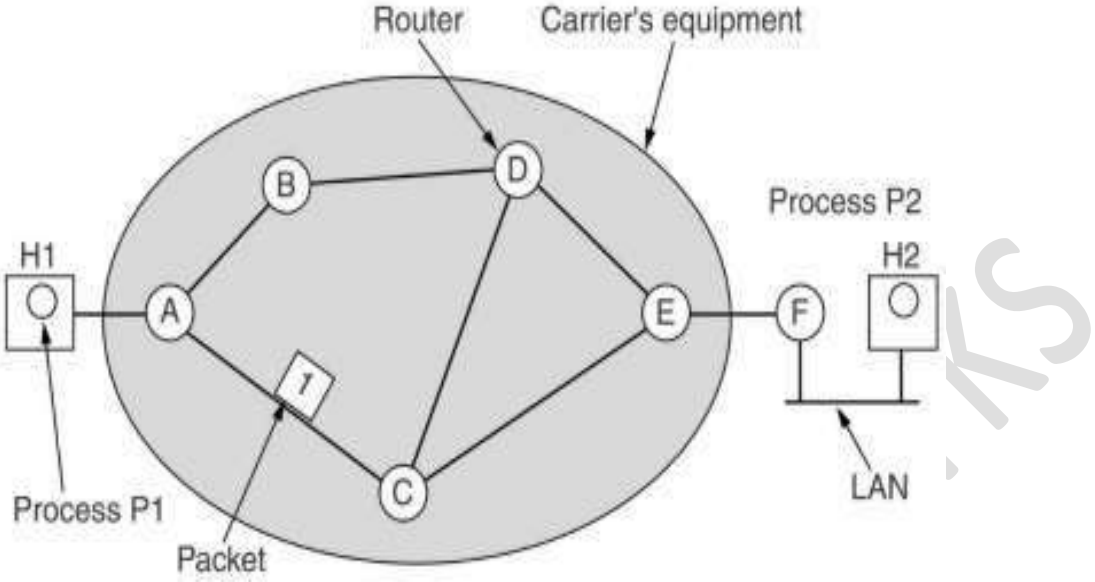


Fig. 4.2Store and Forward Packet Switching

b) Services Provided to the Transport Layer

The network layer services have been designed with the following goals:

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

c) Implementation of Connectionless Service

If connectionless service is offered, packets are injected into the subnet individually and routed independently of each other. No advance setup is needed. In this context, the packets are frequently called **datagrams** and the subnet is called a datagram subnet.

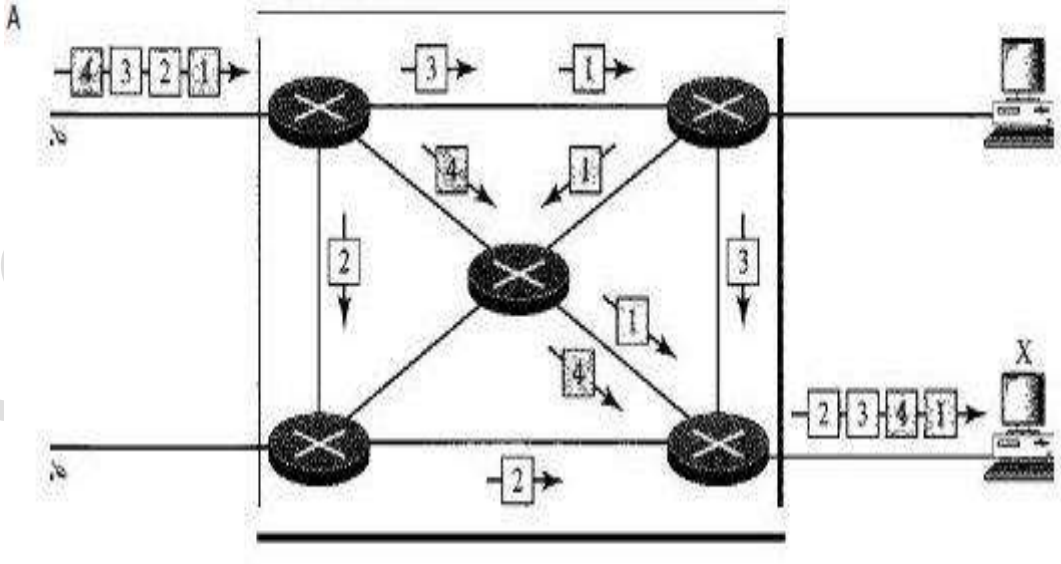


Fig. 4.3Connectionless Service

d) Implementation of Connection-Oriented Service

If connection-oriented service is used, a path from the source router to the destination router must be established before any data packets can be sent. This connection is called a **VC (virtual circuit)** and the subnet is called a virtual-circuit subnet.

The Process is completed in three phase

1. Establishment Phase.
2. Data transfer Phase.
3. Connection release Phase.

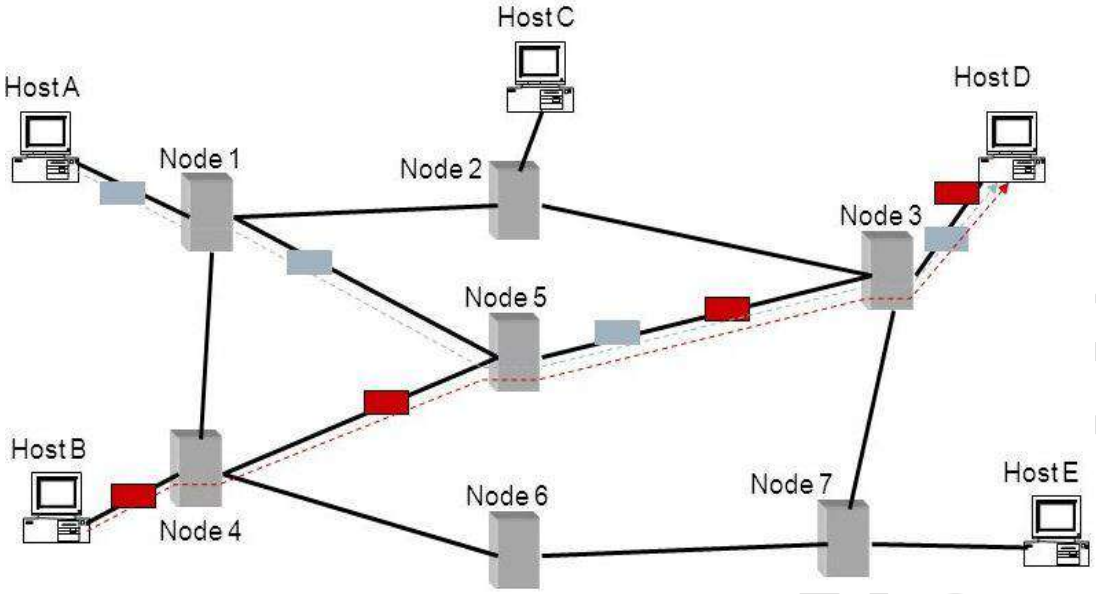


Fig. 4.4Connection-Oriented Service

Comparison of datagram and virtual-circuit subnets

Issue	Datagram subnet	Virtual-circuit subnet
Circuit setup	Not Needed.	Required.
Addressing	Each packet contains the full source and destination address.	Each packet contains a short VC number.
State information	Routers do not hold state information about connections.	Each VC requires router table space per connection.
Routing	Each packet is routed independently.	Route chosen when VC is set up: all packets follow it.
Effect of router failures	None, except for packets lost during the crash.	All VCs that passed through the failed router are terminated.
Quality of services and Congestion Control	Difficult.	Easy if enough resources can be allocated in advance for each VC.

Table 4.1 Comparison of datagram and virtual-circuit subnets

Routing algorithms:

A routing algorithm is a set of step-by-step operations used to direct Internet traffic efficiently. When a packet of data leaves its source, there are many different paths it can take to its destination. The routing algorithm is used to determine mathematically the best path to take.

Properties of routing algorithm:

Correctness: The routing should be done properly and correctly so that the packets may reach their proper destination.

Simplicity: The routing should be done in a simple manner so that the overhead is as low as possible. With increasing complexity of the routing algorithms the overhead also increases.

Robustness: Once a major network becomes operative, it may be expected to run continuously for years without any failures. The algorithms designed for routing should be robust enough to handle hardware and software failures and should be able to cope with changes in the topology and traffic without requiring all jobs in all hosts to be aborted and the network rebooted every time some router goes down.

Stability: The routing algorithms should be stable under all possible circumstances.

Fairness: Every node connected to the network should get a fair chance of transmitting their packets. This is generally done on a first come first serve basis.

Optimality: The routing algorithms should be optimal in terms of throughput and minimizing mean packet delays. Here there is a trade-off and one has to choose depending on his suitability.

Routing can be grouped into two categories

1. Adaptive Routing Algorithm: These algorithms change their routing decisions to reflect changes in the topology and in traffic as well. These get their routing information from adjacent routers or from all routers. The optimization parameters are the distance, number of hops and estimated transit time. This can be further classified as follows:

1. Centralized: In this type some central node in the network gets entire information about the network topology, about the traffic and about other nodes. This then transmits this information to the respective routers. The advantage of this is that only one node is required to keep the information. The disadvantage is that if the central node goes down the entire network is down, i.e. single point of failure.

2. Isolated: In this method the node decides the routing without seeking information from other nodes. The sending node does not know about the status of a particular link. The disadvantage is that the packet may be send through a congested route resulting in a delay. Some examples of this type of algorithm for routing are:

a. Hot Potato: When a packet comes to a node, it tries to get rid of it as fast as it can, by putting it on the shortest output queue without regard to where that link leads. A variation of this algorithm is to combine static routing with the hot potato algorithm. When a packet arrives, the routing algorithm takes into account both the static weights of the links and the queue lengths.

b. Backward Learning: In this method the routing tables at each node gets modified by information from the incoming packets. One way to implement backward learning is to include the identity of the source node in each packet, together with a hop counter that is incremented on each hop. When a node receives a packet in a particular line, it notes down the number of hops it has taken to reach it from the source node. If the previous value of hop count stored in the node is better than the current one then nothing is done but if the current value is better than the value is updated for future use. The problem with this is that when the best route goes down then it cannot recall the second best route to a particular node. Hence all the nodes have to forget the stored information periodically and start all over again.

3. Distributed: In this the node receives information from its neighbouring nodes and then takes the decision about which way to send the packet. The disadvantage is that if in between the interval it receives information and sends the packet something changes then the packet may be delayed.

2. Non-Adaptive Routing Algorithm: These algorithms do not base their routing decisions on measurements and estimates of the current traffic and topology. Instead the route to be taken in going from one node to the other is computed in advance, off-line, and downloaded to the routers when the network is booted. This is also known as static routing. This can be further classified as:

1. Flooding: Flooding adapts the technique in which every incoming packet is sent on every outgoing line except the one on which it arrived. One problem with this method is that packets may go in a loop. As a result of this a node may receive several copies of a particular packet which is undesirable. Some techniques adapted to overcome these problems are as follows:

a. Sequence Numbers: Every packet is given a sequence number. When a node receives the packet it sees its source address and sequence number. If the node finds that it has sent the same packet earlier then it will not transmit the packet and will just discard it.

b. Hop Count: Every packet has a hop count associated with it. This is decremented (or incremented) by one by each node which sees it. When the hop count becomes zero (or a maximum possible value) the packet is dropped.

c. Spanning Tree: The packet is sent only on those links that lead to the destination by constructing a spanning tree routed at the source. This avoids loops in transmission but is possible only when all the intermediate nodes have knowledge of the network topology.

Flooding is not practical for general kinds of applications. But in cases where high degree of robustness is desired such as in military applications, flooding is of great help.

2. Random Walk: In this method a packet is sent by the node to one of its neighbours randomly. This algorithm is highly robust. When the network is highly interconnected, this algorithm has the property of making excellent use of alternative routes. It is usually implemented by sending the packet onto the least queued link.

The Optimality Principle

The optimality principle states that if router J is on the optimal path from router I to router K, then the optimal path from J to K also falls along the same route. As a consequence of that principle, we can see that the set of optimal routes from all sources to a given destination form a tree rooted at the destination. Such tree is called a **sink tree.**

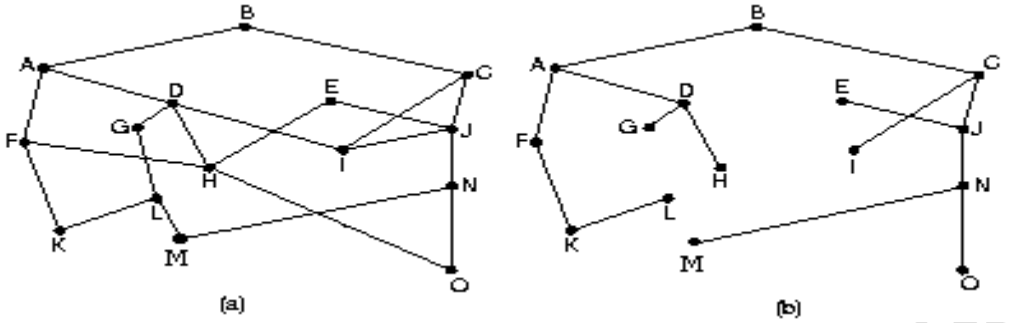


Fig. 4.5 (a) Subnet (b) Sink tree for router B

Shortest Path Algorithm (Least Cost Routing algorithm)

- In this the path length between each node is measured as a function of distance, Bandwidth, average traffic, communication cost, mean queue length, measured delay etc.
- By changing the weighing function, the algorithm then computes the shortest path measured according to any one of a number of criteria or a combination of criteria.
- For this a graph of subnet is drawn. With each node of graph representing a router and each arc of the graph representing a communication link. Each link has a cost associated with it.

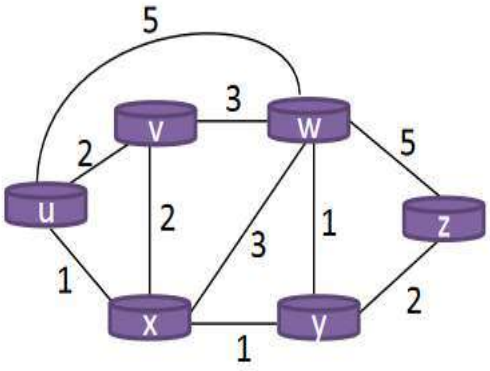
Two algorithms for computing the shortest path between two nodes of a graph are:-

1. Dijkstra's Algorithm
2. Bellnam-Ford Algorithm

1. Dijkstra's algorithm:

1. Compute the least cost path from one node to all other nodes in the network.
2. Iterative algorithm - After the kth iteration, the least cost paths for k destination nodes are found.
3. D(v): cost of the least cost path from source node to destination v
4. p(v): previous node of v along the least-cost path from source.
5. N': set of nodes to which the least-cost path is found.

- Source is node u.



Step	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					

Bellman-ford algorithm:

Following are the detailed steps.

Input: Graph and a source vertex *src*

Output: Shortest distance to all vertices from *src*. If there is a negative weight cycle, then shortest distances are not calculated, negative weight cycle is reported.

1) This step initializes distances from source to all vertices as infinite and distance to source itself as 0. Create an array dist[] of size |V| with all values as infinite except dist[src] where src is source vertex.

2) This step calculates shortest distances. Do following |V|-1 times where |V| is the number of vertices in given graph.

a) Do following for each edge u-v

If dist[v] >dist[u] + weight of edge uv, then update dist[v]

dist[v] = dist[u] + weight of edge uv

3) This step reports if there is a negative weight cycle in graph. Do following for each edge u-v

If dist[v] >dist[u] + weight of edge uv, then "Graph contains negative weight cycle"

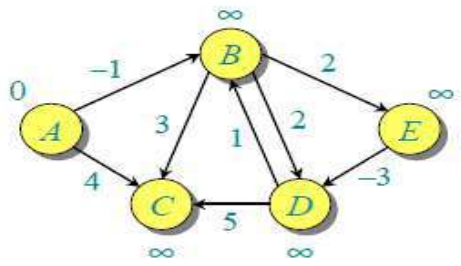
The idea of step 3 is, step 2 guarantees shortest distances if graph doesn't contain negative weight cycle. If we iterate through all edges one more time and get a shorter path for any vertex, then there is a negative weight cycle

How does this work? Like other Dynamic Programming Problems, the algorithm calculates shortest paths in bottom-up manner. It first calculates the shortest distances which have at-most one edge in the path. Then, it calculates shortest paths with at-most 2 edges, and so on. After the i-th iteration of outer loop, the shortest paths with at most i edges are calculated. There can be maximum |V| – 1 edge in any simple path that is why the outer loop runs |v| – 1 times. The idea is, assuming that there is no negative weight cycle, if we have calculated shortest paths with at most i edges, then an iteration over all edges guarantees to give shortest path with at-most (i+1) edges

Example

let us understand the algorithm with following example graph. The images are taken from this source.

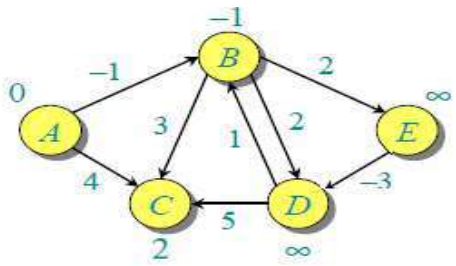
Let the given source vertex be 0. Initialize all distances as infinite, except the distance to source itself. Total number of vertices in the graph is 5, so *all edges must be processed 4 times.*



A	B	C	D	E
0	∞	∞	∞	∞

Fig 4.7Bellman-ford algorithm

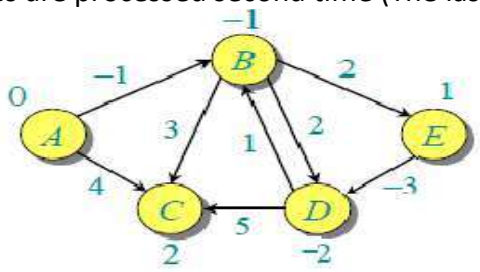
Let all edges are processed in following order: (B,E), (D,B), (B,D), (A,B), (A,C), (D,C), (B,C), (E,D). We get following distances when all edges are processed first time. The first row in shows initial distances. The second row shows distances when edges (B, E), (D,B), (B,D) and (A,B) are processed. The third row shows distances when (A,C) is processed. The fourth row shows when (D,C), (B,C) and (E,D) are processed.



A	B	C	D	E
0	∞	∞	∞	∞
0	−1	∞	∞	∞
0	−1	4	∞	∞
0	−1	2	∞	∞

Fig 4.8Bellman-ford algorithm (Example Step-1)

The first iteration guarantees to give all shortest paths which are at most 1 edge long. We get following distances when all edges are processed second time (The last row shows final values).



A	B	C	D	E
0	∞	∞	∞	∞
0	−1	∞	∞	∞
0	−1	4	∞	∞
0	−1	2	∞	∞
0	−1	2	∞	1
0	−1	2	1	1
0	−1	2	−2	1

Fig 4.9Bellman-ford algorithm (Example Step-2)

The second iteration guarantees to give all shortest paths which are at most 2 edges long. The algorithm processes all edges 2 more times. The distances are minimized after the second iteration, so third and fourth iterations don't update the distances.

Hierarchical Routing:

1. As the number of routers becomes large, the overhead involved in maintaining routing information becomes prohibitive.
2. Internet providers want to manage their network as they wish, while still being able to connect to other networks.
3. Organizing routers into autonomous systems (ASs) solve these problems.
4. Routers within the same AS all run the same routing algorithm (e.g., Dijkstrar DV).Intra-AS routing protocol
5. One or more routers in an AS are responsible to forward packets to destinations outside AS.
6. How to route packets outside an AS?
7. Inter-AS routing protocol: – Obtain reachability information from neighbouring ASs, and Propagate the reachability information to all routers in AS.
8. In the Internet, all ASs run the same inter-AS routing protocol: BGP (Border Gateway Protocol)–Uses a DV-like algorithm.

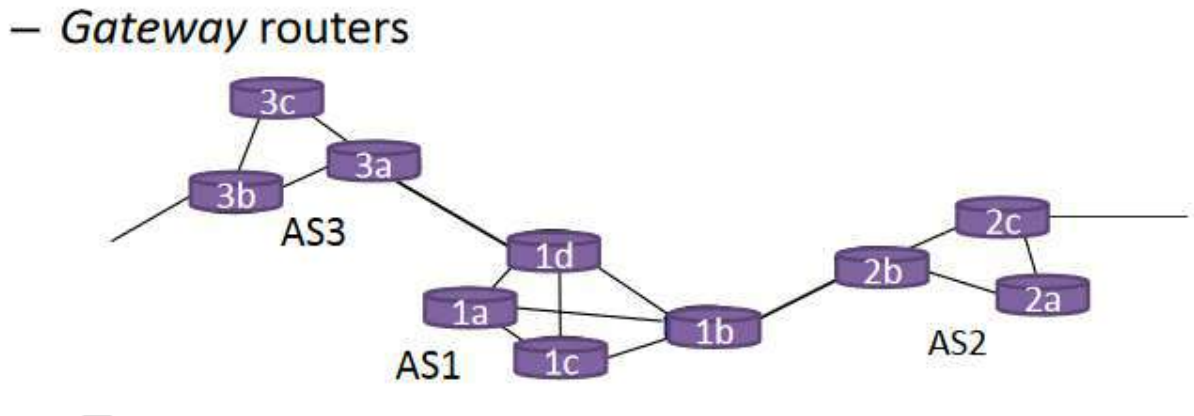


Fig 4.10Hierarchical Routing

Broadcast Routing:

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

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

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

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

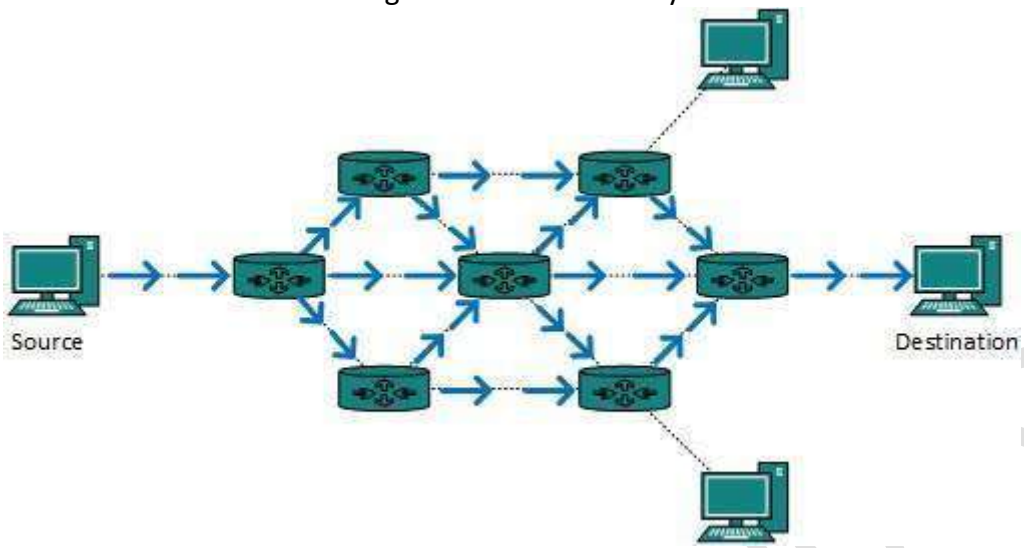


Fig 4.11Broadcast routing

This method is easy on router's CPU but may cause the problem of duplicate packets received from peer routers.

Reverse path forwarding is a technique, in which router knows in advance about its predecessor from where it should receive broadcast. This technique is used to detect and discard duplicates.

Multicast Routing:

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

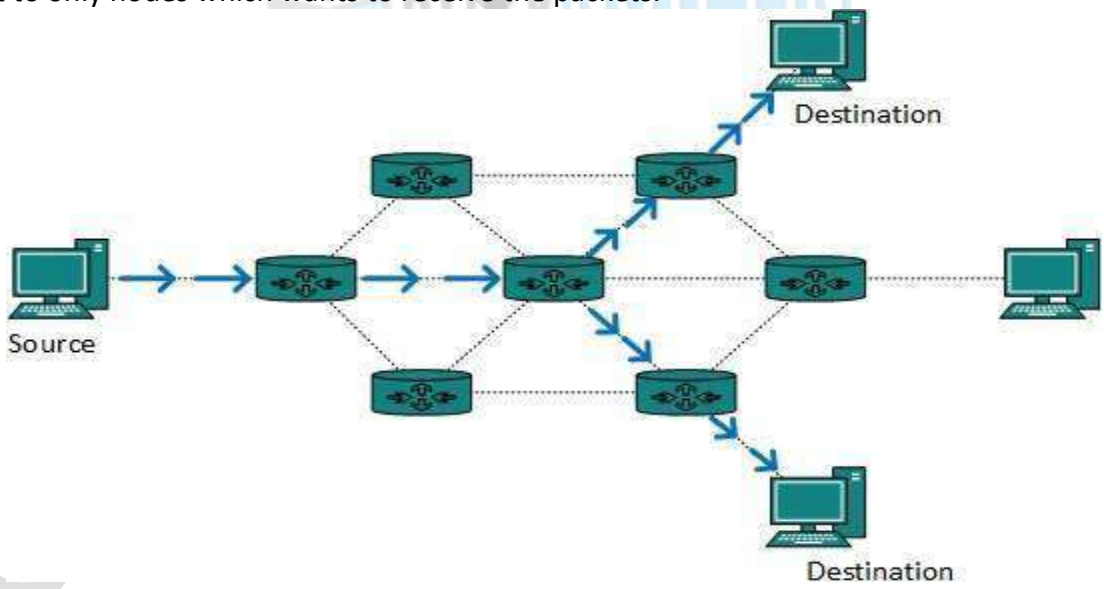


Fig 4.12Multicast routing

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

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

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

Protocol Independent Multicast is commonly used now. It has two flavour's:

- **PIM Dense Mode**
  This mode uses source-based trees. It is used in dense environment such as LAN.
- **PIM Sparse Mode**
  This mode uses shared trees. It is used in sparse environment such as WAN.

Congestion Control Algorithms:

Congestion

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

Effects of Congestion

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

General Principles of Congestion Control Principles

1. Many problems in complex systems, such as computer networks, can be viewed from a control theory point of view. This approach leads to dividing all solutions into two groups: open loop and closed loop.
2. Open loop solutions attempt to solve the problem by good design, in essence, to make sure it does not occur in the first place. Once the system is up and running, midcourse corrections are not made.
3. Tools for doing open-loop control include deciding when to accept new traffic, deciding when to discard packets and which ones, and making scheduling decisions at various points in the network. All of these have in common the fact that they make decisions without regard to the current state of the network.
4. In contrast, closed loop solutions are based on the concept of a feedback loop. This approach has three parts when applied to congestion control:
   1. Monitor the system to detect when and where congestion occurs.
   2. Pass this information to places where action can be taken.
   3. Adjust system operation to correct the problem.

Congestion control algorithms

- **Leaky Bucket Algorithm**

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

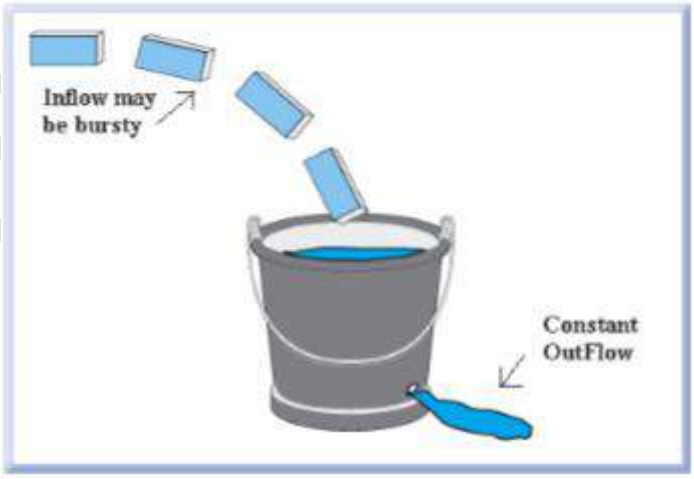


Fig 4.13Leaky bucket algorithm

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

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

- **Token bucket Algorithm**

Need of token bucket Algorithm:-

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

Steps of this algorithm can be described as follows:

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

Let's understand with an example,

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

Let's understand with an example,

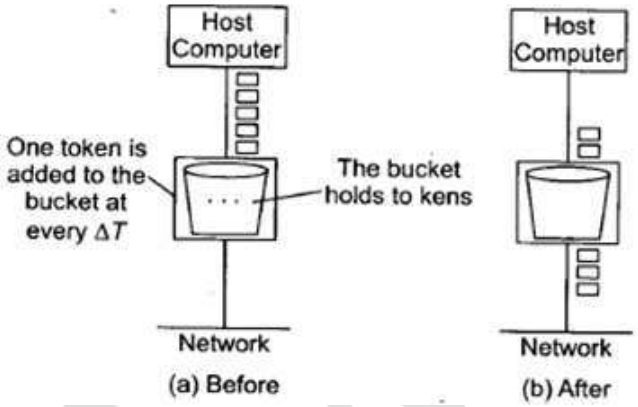


Fig 4.14Token bucket algorithm

Prevention Policies:

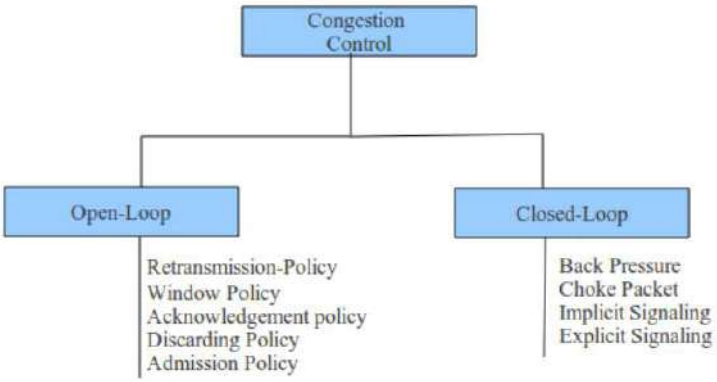
These systems are designed to minimize congestion in the first place, rather than letting it happen and reacting after the fact. They try to achieve their goal by using appropriate policies at various levels.

Layer	Policies
Transport	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy • Timeout determination
Network	• Virtual circuits versus datagram inside the subnet • Packet queueing and service policy • Packet discard policy • Routing algorithm • Packet lifetime management
Data link	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy

Table 4.2 Prevention Policies

- A jumpy sender that times out quickly and retransmits all outstanding packets using go back n will put a heavier load on the system than will a leisurely sender that uses selective repeat. Closely related to this is the buffering policy.
- If receivers routinely discard all out-of-order packets, these packets will have to be transmitted again later, creating extra load. With respect to congestion control, selective repeat is clearly better than go back n.
- In the transport layer, the same issues occur as in the data link layer, but in addition, determining the timeout interval is harder because the transit time across the network is less predictable than the transit time over a wire between two routers.

- If the timeout interval is too short, extra packets will be sent unnecessarily. If it is too long, congestion will be reduced but the response time will suffer whenever a packet is lost.



Open Loop Congestion Control:

In open-loop congestion control, policies are applied to prevent congestion before it happens. In these mechanisms, congestion control is handled by either the source or the destination

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 in general may increase congestion in the network. However, a good retransmission policy can prevent congestion. The retransmission policy and the retransmission timers must be designed to optimize efficiency and at the same time prevent congestion. For example, the retransmission policy used by TCP is designed to prevent or alleviate congestion

Window Policy

The type of window at the sender may also affect congestion. The Selective Repeat window is better than the Go-Back-N window for congestion control. In the Go-Back-N window, when the timer for a packet times out, several packets may be resent, although some may have arrived safe and sound at the receiver. This duplication may make the congestion worse. The Selective Repeat window, on the other hand, tries to send the specific packets that have been lost or corrupted.

Acknowledgment Policy:

The acknowledgment policy imposed by the receiver may also affect congestion. If the receiver does not acknowledge every packet it receives, it may slow down the sender and help prevent congestion. Several approaches are used in this case. 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. We need to know that the acknowledgments are also part of the load in a network. Sending fewer acknowledgments means imposing fewer loads on the network.

Discarding Policy:

A good discarding policy by the routers may prevent congestion and at the same time may not harm the integrity of the transmission. For example, in audio transmission, if the policy is to discard less sensitive packets when congestion is likely to happen, the quality of sound is still preserved and congestion is prevented or alleviated.

Admission Policy:

An admission policy, which is a quality-of-service mechanism, can also prevent congestion in virtual-circuit networks. Switches in a flow first check the resource requirement of a flow before admitting it to the network. A router can deny establishing a virtual- circuit connection if there is congestion in the network or if there is a possibility of future congestion.

Closed-Loop Congestion Control

Closed-loop congestion control mechanisms try to alleviate congestion after it happens. Several mechanisms have been used by different protocols.

Back-pressure:

The technique of backpressure refers to a congestion control mechanism in which a congested node stops receiving data from the immediate upstream node or nodes. This may cause the upstream node or nodes

to become congested, and they, in turn, reject data from their upstream nodes or nodes. And so on. 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. The backpressure technique can be applied only to virtual circuit networks, in which each node knows the upstream node from which a flow of data is corning.

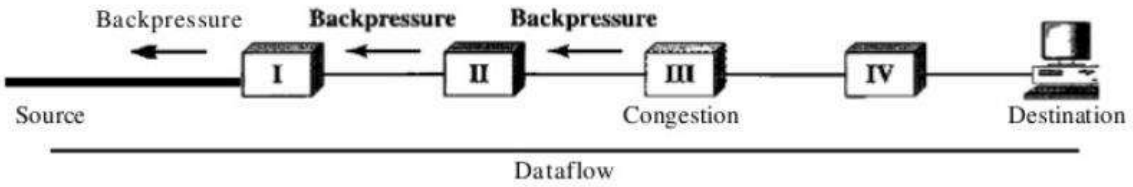


Fig 4.15Back pressure

Node III in the figure has more input data than it can handle. It drops some packets in its input buffer and informs node II to slow down. Node II, in turn, may be congested because it is slowing down the output flow of data. If node II is congested, it informs node I to slow down, which in turn may create congestion. If so, node I inform the source of data to slow down. This, in time, alleviates the congestion. Note that the pressure on node III is moved backward to the source to remove the congestion. None of the virtual-circuit networks we studied in this book use backpressure. It was, however, implemented in the first virtual-circuit network, X.25. The technique cannot be implemented in a datagram network because in this type of network, a node (router) does not have the slightest knowledge of the upstream router.

Choke Packet

A choke packet is a packet sent by a node to the source to inform it of congestion. Note the difference between the backpressure and choke packet methods. In backpressure, the warning is from one node to its upstream node, although the warning may eventually reach the source station. In the choke packet method, the warning is from the router, which has encountered congestion, to the source station directly. The intermediate nodes through which the packet has travelled are not warned. We have seen an example of this type of control in ICMP. When a router in the Internet is overwhelmed datagrams, it may discard some of them; but it informs the source host, using a source quench ICMP message. The warning message goes directly to the source station; the intermediate routers, and does not take any action. Figure shows the idea of a choke packet.

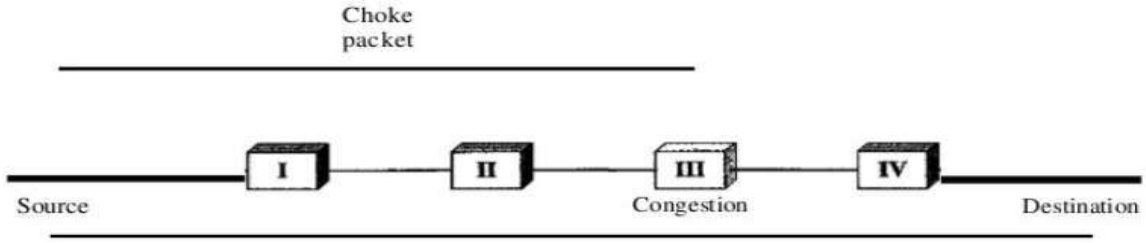


Fig 4.16Choke packet

Implicit Signalling

In implicit signalling, there is no communication between the congested node or nodes and the source. The source guesses that there is congestion somewhere in the network from other symptoms. For example, when a source sends several packets and there is no acknowledgment for a while, one assumption is that the network is congested. The delay in receiving an acknowledgment is interpreted as congestion in the network; the source should slow down. We will see this type of signalling when we discuss TCP congestion control later in the chapter.

Explicit Signalling

The node that experiences congestion can explicitly send a signal to the source or destination. The explicit signalling method, however, is different from the choke packet method. In the choke packet method, a separate packet is used for this purpose; in the explicit signalling method, the signal is included in the packets that carry data. Explicit signalling, as we will see in Frame Relay congestion control, can occur in either the forward or the backward direction.

Backward Signalling

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.

Forward Signalling

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.

Congestion Control in Virtual-Circuit Subnets

1. Admission control: In this approach, once the congestion is signalled, no new connections are set up until the problem is solved. This type of approach is often used in normal telephone networks. When the exchange is overloaded, then no new calls are established.
2. Allow new virtual connections other than the congested area.
3. Negotiate an agreement between the host and the network when the connection is setup. This agreement specifies the volume and shape of traffic, quality of service, maximum delay and other parameters. The network will reserve resources (Buffer space, Bandwidth and CPU cycle) along the path when the connection is set up. Now congestion is unlikely to occur on the new connections because all the necessary resources are guaranteed to be available. The disadvantage of this approach is that it may leads to wasted bandwidth because of some idle connection.

Congestion Control in Datagram subnets

Congestion control in Datagram Subnets is achieved by sending warning to sender in advance. Each router can easily monitor the utilization of its output lines. If utilization is greater than threshold value then output line may be congested in future so mark it as warning state. Each newly arriving packet is checked to see if its output line is in warning state. If it is, some action is taken. The actions are:

1. The warning bit
2. Choke packets
3. Hop-by-hop choke packet

1. The warning bit

When a new packet is to be transmitted on the output line marked as warning state, a special bit is added in header to signal this state. At the destination, this information is sent back with ACK to the sender so that it could cut the traffic. When warning bit is absent, sender increases its transmitting rate.

Note: It uses a whole trip (source -> destination -> source) to tell the source to slow down.

2. Choke packets

In this approach, the router sends a choke packet back to the source host. The original packet is marked so that it would not generate any more choke packets further along the path and is then forwarded in the usual way. When the source gets the choke packet, it is required to reduce the traffic by X packets.

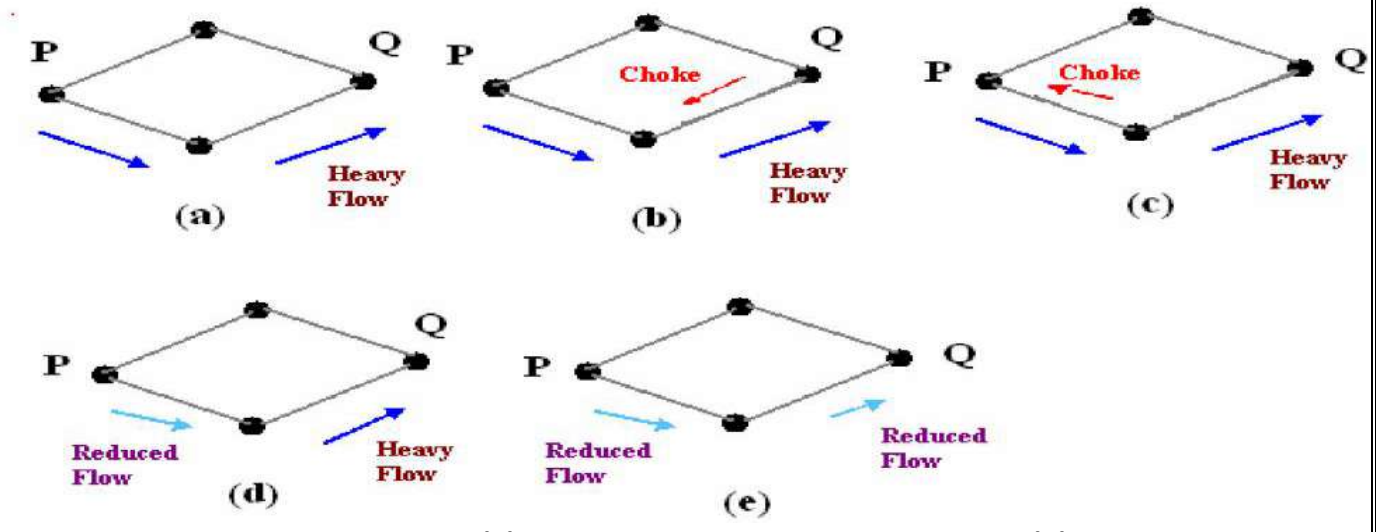


Fig 4.17Functioning of choke packets, (a) Heavy traffic between nodes P and Q, (b) Node Q sends the Choke packet to P, (c) Choke packet reaches P, (d) P reduces the flow and sends a reduced flow out, (e) Reduced flow reaches node Q.

Problem: It does not work well if the choke packet travels a long distance to reach the source because reduction of flow starts from source node rather than intermediate node. This problem can be solved by hop-by-hop approach.

3. Hop-by-hop choke packet

In this approach, unlike choke packet, reduction of flow starts from intermediate node rather than source node. To understand this, let us refer the figure 2. When the choke packet reaches the nearest router (say R) from router Q, it reduces the flow. However, router R now requires devoting more buffers to the flow since the source is still sending at full blast but it gives router Q immediate relief. In the next step, the choke packet reaches P and flow genuinely slow down. The net effect of hop-by-hop scheme is to provide quick relief at the point of congestion at the price of using up more buffers upstream.

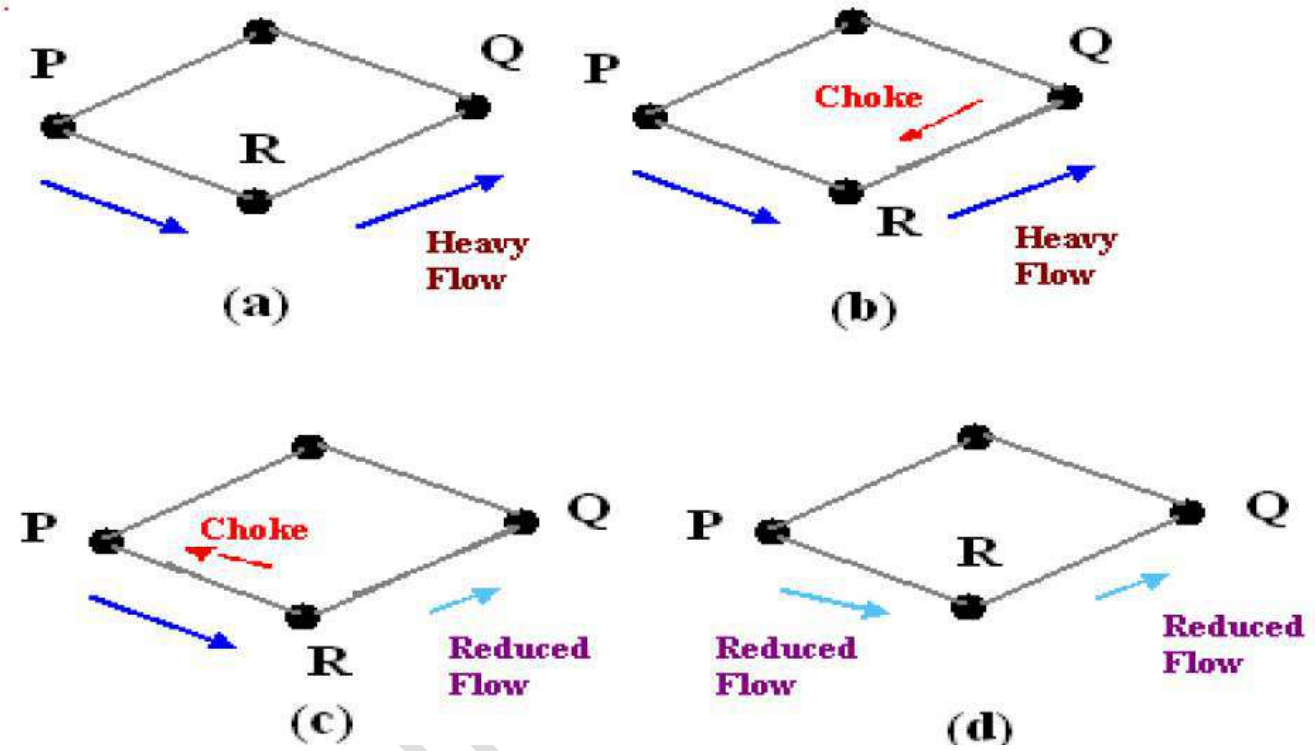


Fig 4.18Functioning of Hop-by-Hop choke packets, (a) Heavy traffic between nodes P and Q, (b) Node Q sends the Choke packet to P, (c) Choke packet reaches R, and the flow between R and Q is decreased, (d) Choke packer reaches P, and P reduces the flow out.

IP protocol:

Internet Protocol (IP) is the principal set (or communications protocol) of digital message formats and rules for exchanging messages between computers across a single network or a series of interconnected networks, using the Internet Protocol Suite (often referred to as TCP/IP). Messages are exchanged as datagrams, also known as data packets or just packets.

IP is the primary protocol in the Internet Layer of the Internet Protocol Suite, which is a set of communications protocols consisting of four abstraction layers: link layer (lowest), Internet layer, transport layer and application layer (highest).

The main purpose and task of IP is the delivery of datagrams from the source host (source computer) to the destination host (receiving computer) based on their addresses. To achieve this, IP includes methods and structures for putting tags (address information, which is part of metadata) within datagrams. The process of putting these tags on datagrams is called encapsulation.

Think of an analogy with the postal system. IP is similar to the U.S. Postal System in that it allows a package (a datagram) to be addressed (encapsulation) and put into the system (the Internet) by the sender (source host). However, there is no direct link between sender and receiver.

The package (datagram) is almost always divided into pieces, but each piece contains the address of the receiver (destination host). Eventually, each piece arrives at the receiver, often by different routes and at different times. These routes and times are also determined by the Postal System, which is the IP. However, the Postal System (in the transport and application layers) puts all the pieces back together before delivery to the receiver (destination host).

Note: IP is actually a connectionless protocol, meaning that the circuit to the receiver (destination host) does not need be set up before transmission (by the source host). Continuing the analogy, there does not need to be a direct connection between the physical return address on the letter/package and the recipient address before the letter/package is sent.

When format and rules were applied to allow connections, the connection-oriented Transmission Control Protocol was created. The two together forms the Internet Protocol Suite, often referred to as TCP/IP.

Internet Protocol version 4 (IPv4) was the first major version of IP. This is the dominant protocol of the Internet. However, iPv6 is active and in use, and its deployment is increasing all over the world.

Addressing and routing are the most complex aspects of IP. However, intelligence in the network is located at nodes (network interconnection points) in the form of routers which forward datagrams to the next known gateway on the route to the final destination. The routers use interior gateway protocols (IGPs) or external gateway protocols (EGPs) to help with making forwarding route decisions. Routes are determined by the routing prefix within the datagrams. The routing process can therefore become complex. But at the speed of light (or nearly so) the routing intelligence determines the best route, and the datagram pieces and datagram all eventually arrive at their destination.

IP Address

An Internet Protocol address (IP address) is a logical numeric address that is assigned to every single computer, printer, switch, router or any other device that is part of a TCP/IP-based network.

The IP address is the core component on which the networking architecture is built; no network exists without it. An IP address is a logical address that is used to uniquely identify every node in the network. Because IP addresses are logical, they can change. They are similar to addresses in a town or city because the IP address gives the network node an address so that it can communicate with other nodes or networks, just like mail is sent to friends and relatives.

The numerals in an IP address are divided into 2 parts:

- The network part specifies which networks this address belongs to and
- The host part further pinpoints the exact location.

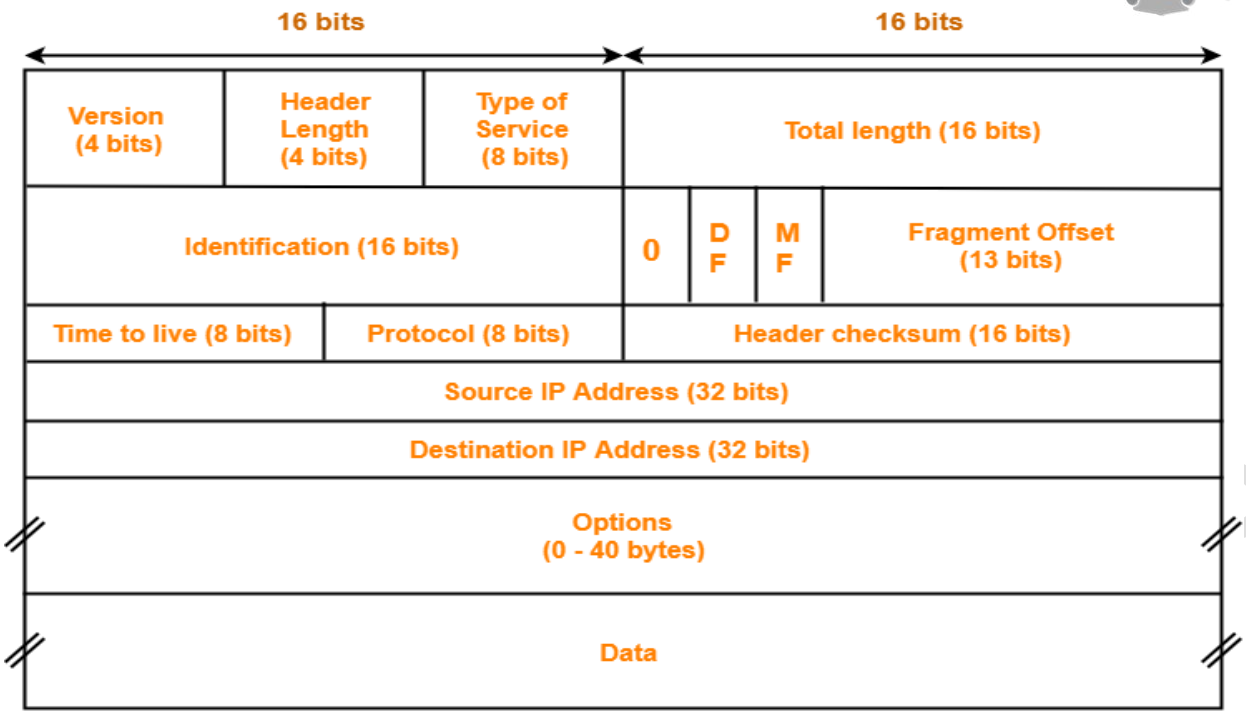
An IP address is the most significant and important component in the networking phenomena that binds the World Wide Web together. The IP address is a numeric address assigned to every unique instance that is connected to any computer communication network using the TCP/IP communication protocols.

Network nodes are assigned IP addresses by the Dynamic Host Configuration Protocol server as soon as the nodes connect to a network. DHCP assigns IP addresses using a pool of available addresses which are part of the whole addressing scheme. Though DHCP only provides addresses that are not static, many machines reserve static IP addresses that are assigned to that entity forever and cannot be used again.

IP addresses falls into two types:

- Classfull IP addressing is a legacy scheme which divides the whole IP address pools into 5 distinct classes—A, B, C, D and E.
- Classless IP addressing has an arbitrary length of the prefixes.

Header format of IPv4:



IPv4 Header

Fig:4.19 IPV4 Header

VERSION: Version of the IP protocol (4 bits), which is 4 for IPv4

HLEN: IP header length (4 bits), which is the number of 32 bit words in the header. The minimum value for this field is 5 and the maximum is 15.

Type of service: Low Delay, High Throughput, Reliability (8 bits)

Total Length: Length of header + Data (16 bits), which has a minimum value 20 bytes and the maximum is 65,535 bytes.

Identification: Unique Packet Id for identifying the group of fragments of a single IP datagram (16 bits)

Flags: 3 flags of 1 bit each : reserved bit (must be zero), do not fragment flag, more fragments flag (same order)

Fragment Offset: Represents the number of Data Bytes ahead of the particular fragment in the particular Datagram. Specified in terms of number of 8 bytes, which has the maximum value of 65,528 bytes.

Time to live: Datagram's lifetime (8 bits), It prevents the datagram to loop through the network by restricting the number of Hops taken by a Packet before delivering to the Destination.

Protocol: Name of the protocol to which the data is to be passed (8 bits)

Header Checksum: 16 bits header checksum for checking errors in the datagram header.

Source IP address: 32 bits IP address of the sender

Destination IP address: 32 bits IP address of the receiver

Option: Optional information such as source route, record route. Used by the Network administrator to check whether a path is working or not.

Due to the presence of options, the size of the datagram header can be of variable length (20 bytes to 60 bytes).

Header format of IPV6:

Version (4-bits) : Indicates version of Internet Protocol which contains bit sequence 0110.

Traffic Class (8-bits) : The Traffic Class field indicates class or priority of IPv6 packet which is similar to *Service Field* in IPv4 packet. It helps routers to handle the traffic based on priority of the packet. If congestion occurs on router then packets with least priority will be discarded.

As of now only 4-bits are being used (and remaining bits are under research), in which 0 to 7 are assigned to Congestion controlled traffic and 8 to 15 are assigned to Uncontrolled traffic.

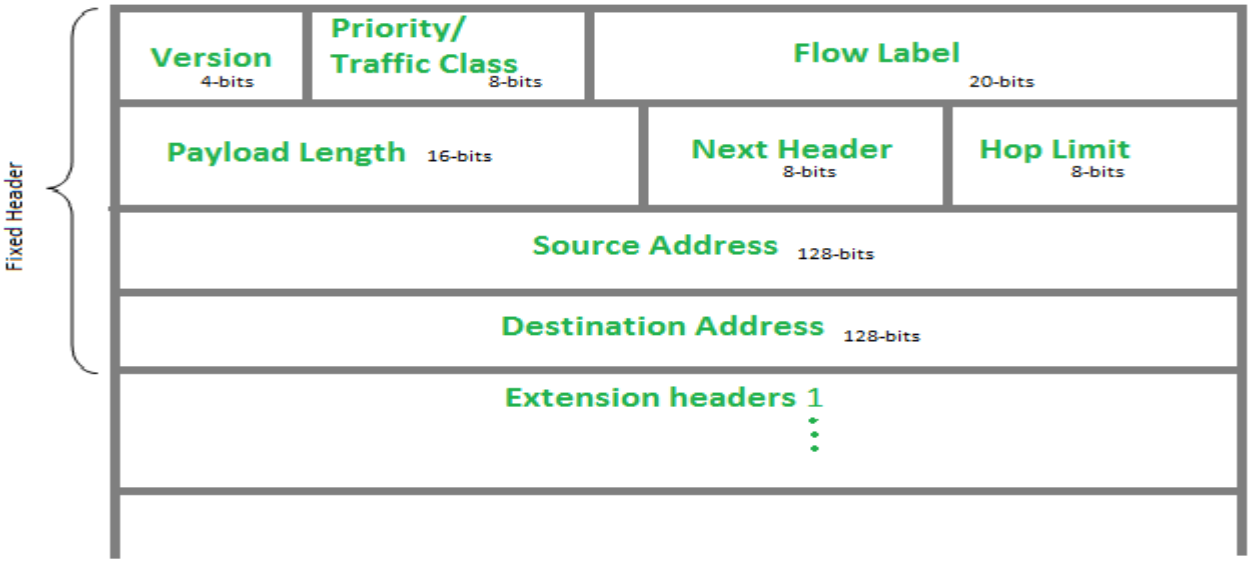


Fig:4.20 IPV6 Header

Flow Label (20-bits) : Flow Label field is used by source to label the packets belonging to the same flow in order to request special handling by intermediate IPv6 routers, such as non-default quality of service or real time service. In order to distinguish the flow, intermediate router can use source address, destination address and flow label of the packets. Between a source and destination multiple flows may exist because many processes might be running at the same time. Routers or Host that do not support the functionality of flow label field and for default router handling, flow label field is set to 0. While setting up the flow label, source is also supposed to specify the lifetime of flow.

Payload Length (16-bits) : It is a 16-bit (unsigned integer) field, indicates total size of the payload which tells routers about amount of information a particular packet contains in its payload. Payload Length field includes extension headers(if any) and upper layer packet. In case length of payload is greater than 65,535 bytes (payload up to 65,535 bytes can be indicated with 16-bits), then the payload length field will be set to 0 and jumbo payload option is used in the Hop-by-Hop options extension header.

Next Header (8-bits) : Next Header indicates type of extension header(if present) immediately following the IPv6 header. Whereas In some cases it indicates the protocols contained within upper-layer packet, such as TCP, UDP.

Hop Limit (8-bits) : Hop Limit field is same as TTL in IPv4 packets. It indicates the maximum number of intermediate nodes IPv6 packet is allowed to travel. Its value gets decremented by one, by each node that forwards the packet and packet is discarded if value decrements to 0. This is used to discard the packets that are stuck in infinite loop because of some routing error.

Source Address (128-bits) : Source Address is 128-bit IPv6 address of the original source of the packet.

Destination Address (128-bits) : Destination Address field indicates the IPv6 address of the final destination(in most cases). All the intermediate nodes can use this information in order to correctly route the packet.

Extension Headers : In order to rectify the limitations of *IPv4 Option Field*, Extension Headers are introduced in IPversion 6. The extension header mechanism is very important part of the IPv6 architecture. Next Header field of IPv6 fixed header points to the first Extension Header and this first extension header points to the second extension header and so on.

Comparative study of IPv4 & IPv6:

Sl. No.	IPv4	IPv6
1	Addresses are 32 bits (4 bytes) in length.	Addresses are 128 bits (16 bytes) in length.
2	Address (A) resource records in DNS to map host names to IPv4 addresses.	Address (AAAA) resource records in DNS to map host names to IPv6 addresses.
3	Pointer (PTR) resource records in the IN-ADDR.ARPA DNS domain to map IPv4 addresses to host names.	Pointer (PTR) resource records in the IP6.ARPA DNS domain to map IPv6 addresses to host names.
4	IPSec is optional and should be supported externally	IPSec support is not optional
5	Header does not identify packet flow for QoS handling by routers	Header contains Flow Label field, which Identifies packet flow for QoS handling by router.
6	Both routers and the sending host fragment packets.	Routers do not support packet fragmentation. Sending host fragments packets
7	Header includes a checksum.	Header does not include a checksum.
8	ARP uses broadcast ARP request to resolve IP to MAC/Hardware address.	Multicast Neighbour Solicitation messages resolve IP addresses to MAC addresses.
9	Internet Group Management Protocol (IGMP) manages membership in local subnet groups	Multicast Listener Discovery (MLD) messages manage membership in local subnet groups.
10	Broadcast addresses are used to send traffic to all nodes on a subnet.	IPv6 uses a link-local scope all-nodes multicast address.
11	Configured either manually or through DHCP.	Does not require manual configuration or DHCP.
12	Must support a 576-byte packet size (possibly fragmented).	Must support a 1280-byte packet size (without fragmentation).

Table 4.3 Comparison between IPv4 and IPv6

Packet Forwarding:

Packet forwarding is done when uIP receives a packet that has a destination IP address that does not match any of the IP addresses of the node. A node typically has multiple addresses: one or more unicast addresses and at least one broadcast or multicast address. Packets that do not match the addresses should be forwarded to a neighboring node, either because the address matches that of the neighbor or because the neighbor has a route to the destination address.

Packet forwarding occurs only when uIP has been configured to be a router. The packet forwarding mechanism is then invoked as part of the output processing.

The packet forwarding mechanism is modular and does not specify any particular routing mechanism to be used. Rather, a routing mechanism will register itself with the forwarding module upon startup. For every packet, the forwarding mechanism asks the routing module to look up the destination IP address and return the address to the next-hop neighbor. The routing module may implement this any way it wants by using a table of destination addresses, a table of network prefixes, a hash table of addresses, a cache of the recently used routes, or any other way it finds suitable. The routing protocol may perform a route discovery for each address not found in its cache.

By separating packet forwarding and packet routing, uIP can adapt a wide range of requirements such as routing performance and memory requirements, as well as take advantage of future development in routing protocols. A system with strict memory requirements and low routing performance requirements may use a cache configuration that prompts frequent network route discoveries, whereas a system with

strict requirements on routing performance but lax memory requirements may choose a larger cache setting.

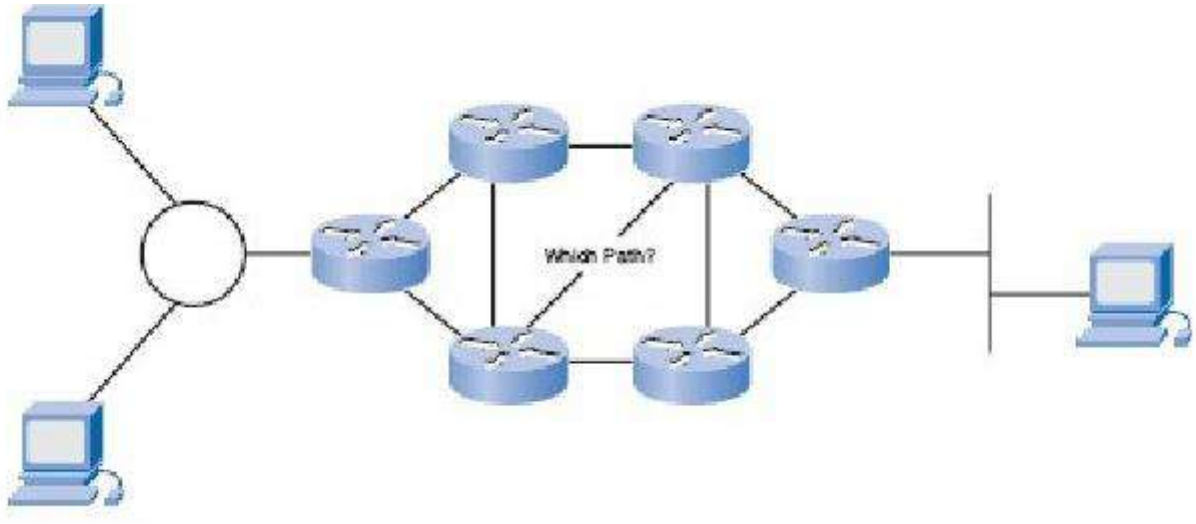


Fig. 4.21 Packet forwarding

Fragmentation:

Fragmentation is the process of breaking a packet into smaller pieces so that they will fit into the frames of the underlying network. The receiving system reassembles the pieces into the original packets. The term MTU (maximum transmission unit) refers to the maximum amount of data that can travel in a frame. Different networks have different MTU sizes, so packets may need to be fragmented in order to fit within the frames of the network that they transit.

Internetworking protocols such as IP use fragmentation because each of the networks that a packet may travel over could have a different frame size. Fragmentation occurs at routers that connect two networks with different MTUs. While it is possible to design an internal network with the same MTU size, this is not an option on the Internet, which includes thousands of independently managed interconnected networks. Fragmentation is always undesirable because it reduces performance. In fact, fragmentation is not allowed in IPv6. Large packets are always preferable.

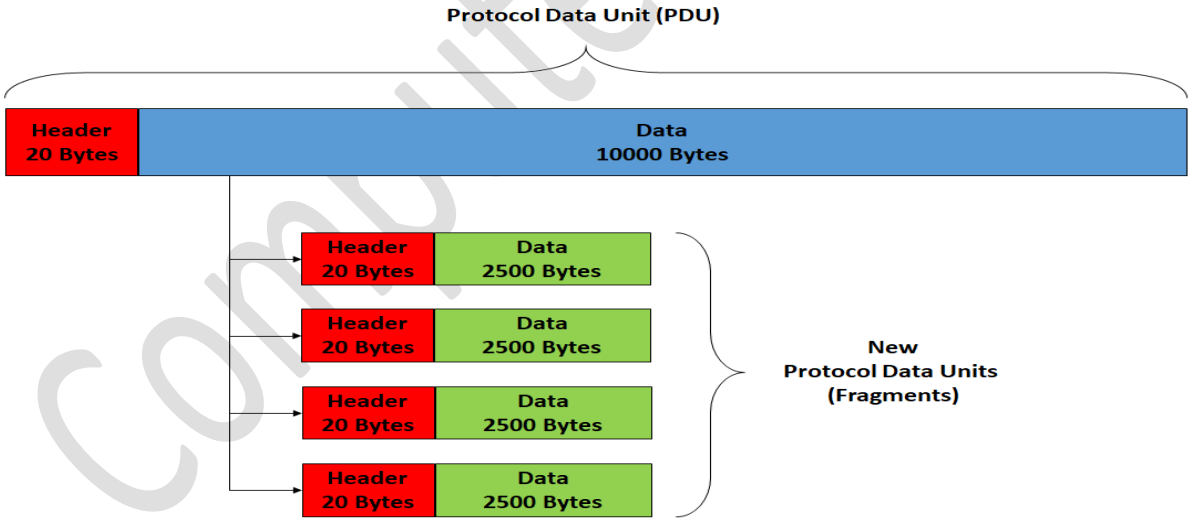


Fig:4. 22 Fragmentations

Reassembly is the reverse of segmentation. Protocol Data Units are put back together in the correct order to reassemble a stream of data in its original form.

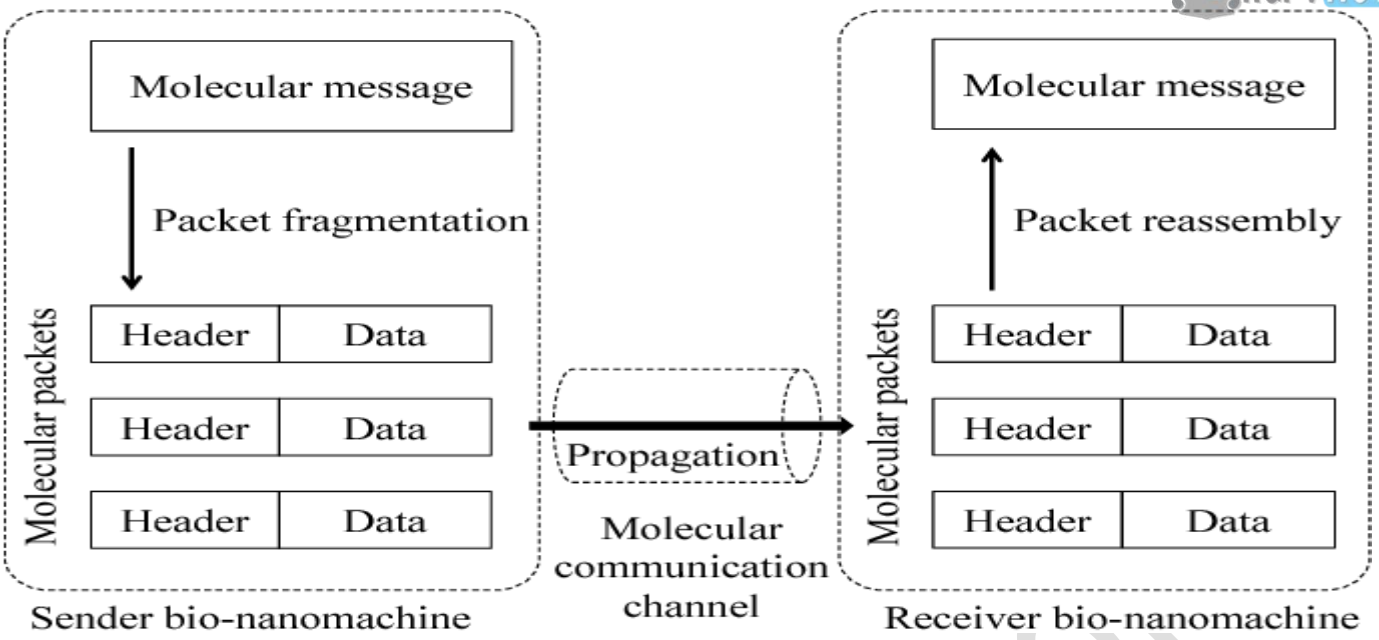


Fig:4.23 Packet fragmentation and Packet reassembly

ICMP:

ICMP (Internet Control Message Protocol) is an error-reporting protocol network devices like routers use to generate error messages to the source IP address when network problems prevent delivery of IP packets. ICMP creates and sends messages to the source IP address indicating that a gateway to the Internet that a router, service or host cannot be reached for packet delivery. Any IP network device has the capability to send, receive or process ICMP messages.

ICMP is *not* a transport protocol that sends data between systems.

While ICMP is not used regularly in end-user applications, it is used by network administrators to troubleshoot Internet connections in diagnostic utilities including ping and tracer route.

One of the main protocols of the IP suite, ICMP is used by routers, intermediary devices or hosts to communicate error information or updates to other routers, intermediary devices or hosts. The widely used IPv4 (Internet Protocol version 4) and the newerIPv6 use similar versions of the ICMP protocol (ICMPv4 and ICMPv6, respectively).

ICMP messages are transmitted as datagram's and consist of an IP header that encapsulates the ICMP data. ICMP packets are IP packets with ICMP in the IP data portion. ICMP messages also contain the entire IP header from the original message, so the end system knows which packet failed

The ICMP header appears after the IPv4 or IPv6 packet header and is identified as IP protocol number 1. The complex protocol contains three fields:

- The major type that identifies the ICMP message;
- The minor code that contains more information about the type field; and
- The checksum that helps detect errors introduced during transmission.

Following the three fields is the ICMP data and the original IP header to identify which packets actually failed.

ICMP has been used to execute denial-of-services attacks (also called the ping of death) by sending an IP packet larger than the number of bytes allowed by the IP protocol.

RGPVNOTES.IN

We hope you find these notes useful.

You can get previous year question papers at https://qp.rgpvnotes.in .

If you have any queries or you want to submit your study notes please write us at rgpvnotes.in@gmail.com

LIKE & FOLLOW US ON FACEBOOK
facebook.com/rgpvnotes.in

RGPVNOTES.IN

Program : **B.Tech**

Subject Name: **Computer Networks**

Subject Code: **CS-602**

Semester: **6th**

Department of Computer Science and Engineering
CS602 Computer Networks
Subject Notes: UNIT-V

Syllabus: **Transport Layer: Design Issues, UDP: Header Format, Per-Segment Checksum, Carrying Unicast/Multicast Real-Time Traffic, TCP: Connection Management, Reliability of Data Transfers, TCP Flow Control, TCP Congestion Control, TCP Header Format, TCP Timer Management. Application Layer: WWW and HTTP, FTP, SSH, Email (SMTP, MIME, IMAP), DNS, Network Management (SNMP).**

Transport Layer: Design Issues

- Accepting data from Session layer split it into segments and send to the network layer.
- Ensure correct delivery of data with efficiency.
- Isolate upper layers from the technological changes.
- Error control and flow control.

UDP: Header Format

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

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

Features

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

0	15 16	31
16-Bit Source Port	16-Bit Destination Port	
16- Bit UDP Length	16- Bit UDP Checksum	
Data (If Any)		

Fig 5.1 UDP Header Format

The UDP header consists of four fields each of 2 bytes in length:

Source Port (UDP packets from a client use this as a service access point (SAP) to indicate the session on the local client that originated the packet. UDP packets from a server carry the server SAP in this field)

Destination Port (UDP packets from a client use this as a service access point (SAP) to indicate the service required from the remote server. UDP packets from a server carry the client SAP in this field)

UDP length (The number of bytes comprising the combined UDP header information and payload data)

UDP Checksum (A checksum to verify that the end to end data has not been corrupted by routers or bridges in the network or by the processing in an end system. The algorithm to compute the checksum is the Standard Internet Checksum algorithm. This allows the receiver to verify that it was the intended destination of the packet, because it covers the IP addresses, port numbers and protocol number, and it verifies that the packet is not truncated or padded, because it covers the size field. Therefore, this protects an application against receiving corrupted payload data in place of, or in addition to, the data that was sent. In the cases where this check is not required, the value of 0x0000 is placed in this field, in which case the data is not checked by the receiver.

Pre-Segment Checksum

Checksum is a simple error detection mechanism to determine the integrity of the data transmitted over a network. Communication protocols like TCP/IP/UDP implement this scheme in order to determine whether the received data is corrupted along the network. The sender of an IPv4 datagram would compute the checksum value based on the data and embed it in the frame. The receiver would also compute the checksum locally and based on the result ascertain the data integrity. Similarly the TCP/UDP data which forms the payload for the IP datagram would have its checksum computed and embedded as a part of the TCP/UDP frame.

In an attempt to improve performance and to assist drivers in ensuring data integrity, checksum computation is increasingly being done in hardware. The checksum offload feature can be implemented as a combination of hardware and software functions - the hardware assists the driver in completing the checksum computation.

This functionality can be enabled in Asics and disabled in the existing drivers (TCP/IP protocol stack) easily.

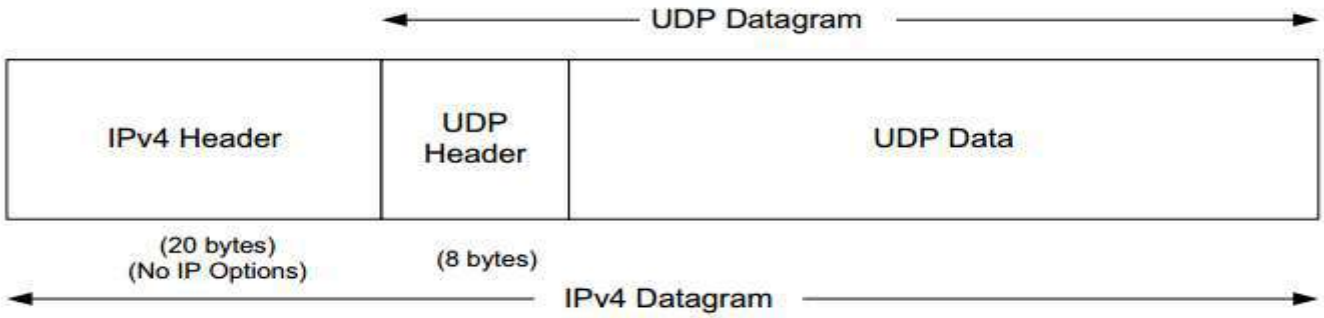


Fig 5.2 UDP Per-Segment Checksum

Unicast/Multicast Real-Time Traffic

Data is transported over a network by three simple methods i.e. Unicast, Broadcast, and Multicast.

- **Unicast**: from one source to one destination i.e. One-to-One
- **Broadcast**: from one source to all possible destinations i.e. One-to-All
- **Multicast**: from one source to multiple destinations stating an interest in receiving the traffic i.e. One-to-Many

Note: There is no separate classification for Many-to-Many applications, for example, video conferencing or online gaming, where multiple sources for the same receiver and where receivers often are double as sources. This service model works on the basis of one-to-many multicast and for that reason requires no unique protocol. The original multicast design i.e. RFC 1112, supports both the ASM (any-source-multicast) based on many-to-many service model and the SSM (source specific multicast) based on a one-to-many model.

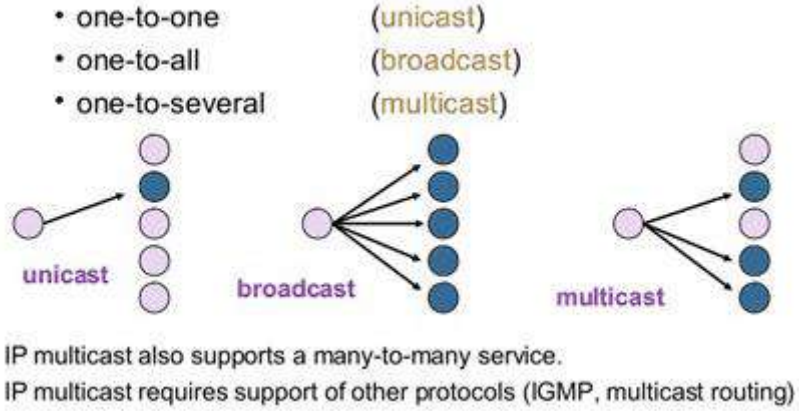


Fig 5.3 IP Services

Unicast

1. Traffic is sent from one host to another. A replica of each packet in the data stream goes to every host that requests it.

2. The implementation of unicast applications is a bit easy as they use well-established IP protocols; however, they are particularly incompetent when there is a need for many-to-many communications. In the meantime, all packets in the data stream must be sent to every host requesting access to the data stream. However, this type of transmission is ineffective in terms of both network and server resource as it equally presents obvious scalability issues.
3. This is a one-to-one connection between the client and the server. Unicast uses IP provision techniques such as TCP (transmission control protocol) and UDP (user datagram protocol), which are session-based protocols. Once a Windows media player client connects via unicast to a Windows media server that client gets a straight connection to the server. Every unicast client that connects to the server takes up extra bandwidth. For instance, if you have 10 clients all performing 100 Kbps (kilobits per second) streams, it means those clients taking up 1,000 Kbps. But you a have single client using the 100 Kbps stream, only 100 Kbps is being used.

Multicast

Multicast lets server's direct single copies of data streams that are then simulated and routed to hosts that request it.

Hence, rather than sending thousands of copies of a streaming event, the server instead streams a single flow that is then directed by routers on the network to the hosts that have specified that they need to get the stream. This removes the requirement to send redundant traffic over the network and also be likely to reduce CPU load on systems, which are not using the multicast system, yielding important enhancement to efficiency for both server and network.

Multicast is true broadcast?

The multicast source depends on multicast-enabled routers to forward the packets to all clients' subnets that have clients listening. However, there is no direct affiliation between clients and Windows media server. The Windows media server creates an ".nsc" (NetShow channel) file when the multicast station is first formed. Usually, the .nsc file is sent to the client from a web server. This file holds data that the Windows media player requires to listen for the multicast. This is quite same to fine-tuning a station on a radio. Every client which eavesdrops to the multicast includes no extra overhead on the server. In fact, the server sends out only single stream per multicast station. The equal load is experienced on the server whether only a single client or multiple clients are listening.

Important note

Multicast on the Internet is usually not a concrete solution because only small sections of the Internet are enabled with Multicast. On the other hand, in corporate environments where all routers are multicast-enabled can save quite a bit of bandwidth.

So what is the difference between Multicast and Unicast?

There are two central methods that Windows Media servers use to send data to Windows Media Player clients i.e. Unicast and Multicast…

Multicast or Unicast both can be used for broadcasting live video or audio.

TCP

TCP is a unicast connection-oriented protocol. Before either end can send data to the other, a connection must be established between them. TCP detects and repairs essentially all the data transfer problems that may be introduced by packet loss, duplication, or errors at the IP layer (or below).

Because of its management of connection state (information about the connection kept by both endpoints), TCP is a considerably more complicated protocol than UDP.

A TCP connection is defined to be a 4-tuple consisting of two IP addresses and two port numbers. It is a pair of endpoints or sockets where each endpoint is identified by an (IP address, port number) pair.

A connection typically goes through three phases:

- Setup
- Data transfer (called established)
- Teardown (closing).

Some of the difficulty in creating a robust TCP implementation is handling all of the transitions between and among these phases correctly.

Connection Management:

A connection typically goes through three phases:

- Setup
- Data transfer (called established)
- Teardown (closing).

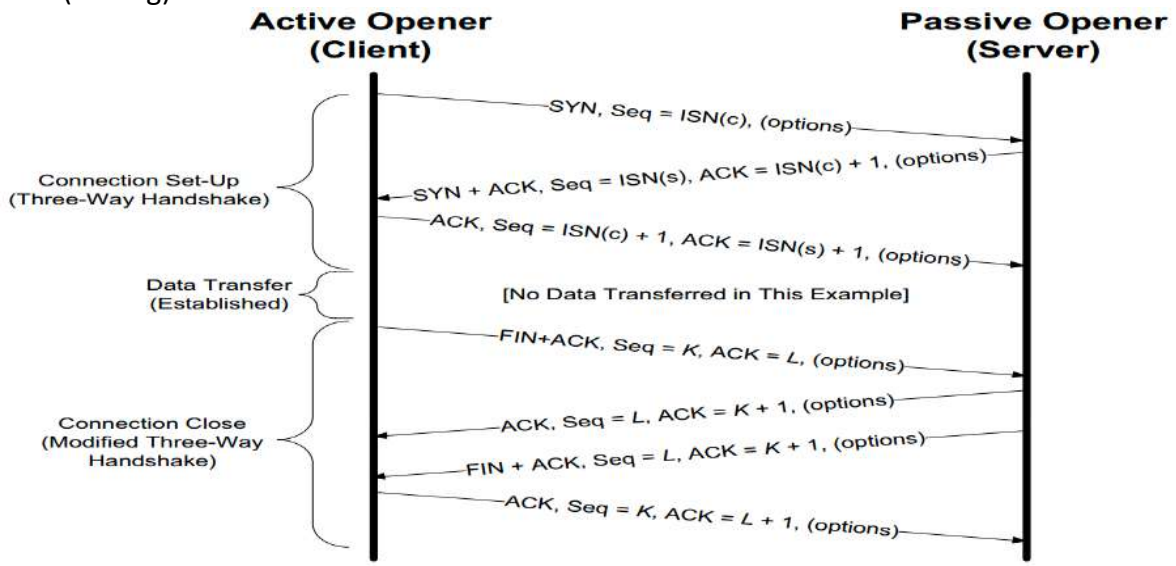


Fig 5.4 TCP- Connection Management

Reliability of Data Transfers

Let us consider the problem of reliable data transfer in a general context. This is appropriate since the problem of implementing reliable data transfer occurs not only at the transport layer, but also at the link layer and the application layer as well. The general problem is thus of central importance to networking. Indeed, if one had to identify a ``top-10'' list of fundamentally important problems in all of networking, this would be a top candidate to lead that list.

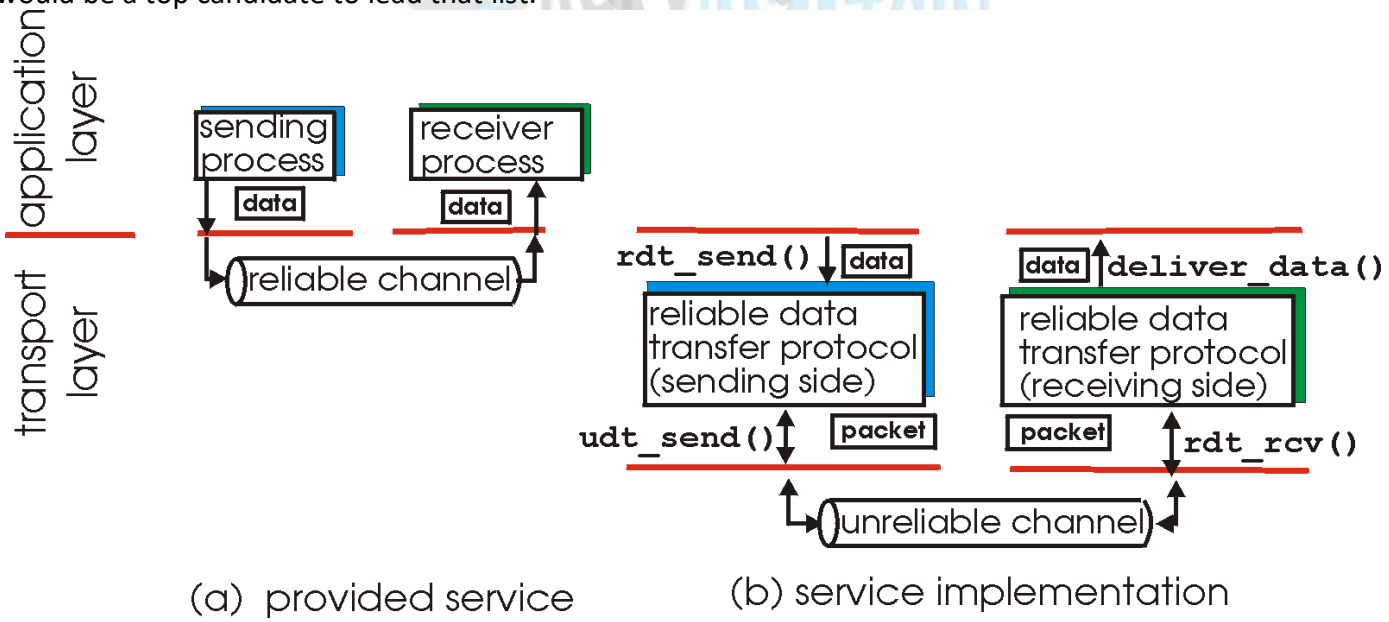


Fig 5.5 Reliable data transfer: service model and service implementation.

It is the responsibility of a **reliable data transfer protocol** to implement this service abstraction. This task is made difficult by the fact that layer below the reliable data transfer protocol may be unreliable. For example, TCP is a reliable data transfer protocol that is implemented on top of an unreliable (IP) end-end network layer. More generally, the layer beneath the two reliably-communicating endpoints might consist of a single physical link (e.g., as in the case of a link-level data transfer protocol) or a global internetwork (e.g., as in the case of a transport-level protocol). For our purposes, however, we can view this lower layer simply as an unreliable point-to-point channel.

The internet network layer provides only best effort service with no guarantee that packets arrive at their destination. Also, since each packet is routed individually it is possible that packets are received out of order. For connection-oriented service provided by TCP, it is necessary to have a reliable data transfer (RDT) protocol to ensure delivery of all packets and to enable the receiver to deliver the packets in order to its application layer.

A simple alternating bit RDT protocol can be designed using some basic tools. This protocol is also known as a stop-and-wait protocol: after sending each packet the sender stops and waits for feedback from the receiver indicating that the packet has been received.

Stop-and-wait RDT protocols have poor performance in a long-distance connection. At best, the sender can only transmit one packet per round-trip time. For a 1000 mile connection this amounts to approximately 1 packet (about 1500 bytes) every 20 ms that results in a pathetic 75 KB per second rate.

To improve transmission rates, a realistic RDT protocol must use pipelining. This allows the sender to have a large number of packets "in the pipeline". This phrase refers to packets that have been sent but whose receipt has not yet verified by the receiver.

TCP Flow Control

TCP is the protocol that guarantees we can have a reliable communication channel over an unreliable network. When we send data from a node to another, packets can be lost, they can arrive out of order, the network can be congested or the receiver node can be overloaded. When we are writing an application, though, we usually don't need to deal with this complexity, we just write some data to a socket and TCP makes sure the packets are delivered correctly to the receiver node. Another important service that TCP provides is what is called *Flow Control*. Let's talk about what that means and how TCP does its magic.

What is Flow Control (and what it's not)

Flow Control basically means that TCP will ensure that a sender is not overwhelming a receiver by sending packets faster than it can consume. It's pretty similar to what's normally called *Back pressure* in the Distributed Systems literature. The idea is that a node receiving data will send some kind of feedback to the node sending the data to let it know about its current condition.

It's important to understand that this is **not** the same as *Congestion Control*. Although there's some overlap between the mechanisms TCP uses to provide both services, they are distinct features. Congestion control is about preventing a node from overwhelming the network (i.e. the links between two nodes), while Flow Control is about the end-node.

How it works

When we need to send data over a network, this is normally what happens.

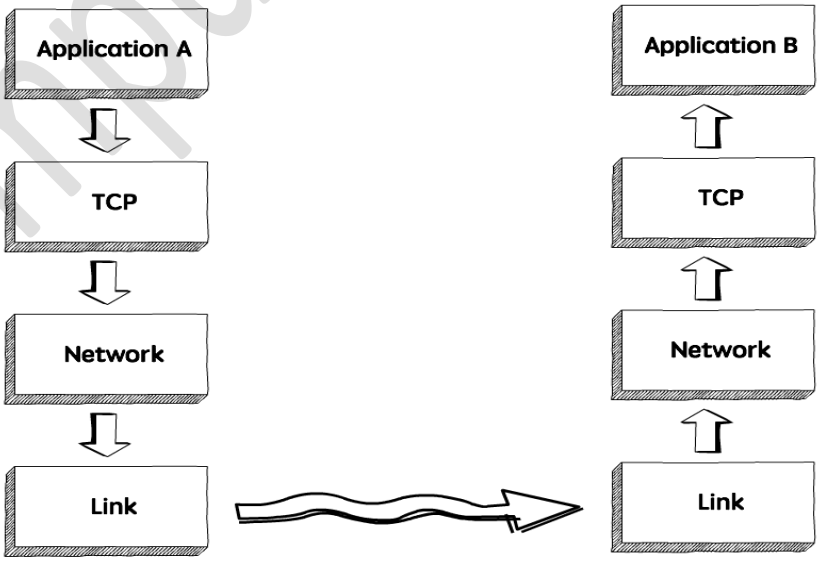


Fig 5.6 TCP Flow Control- works.

The sender application writes data to a socket, the transport layer (in our case, TCP) will wrap this data in a segment and hand it to the network layer (e.g. IP), that will somehow route this packet to the receiving node. On the other side of this communication, the network layer will deliver this piece of data to TCP, that will make it available to the receiver application as an exact copy of the data sent, meaning it will not

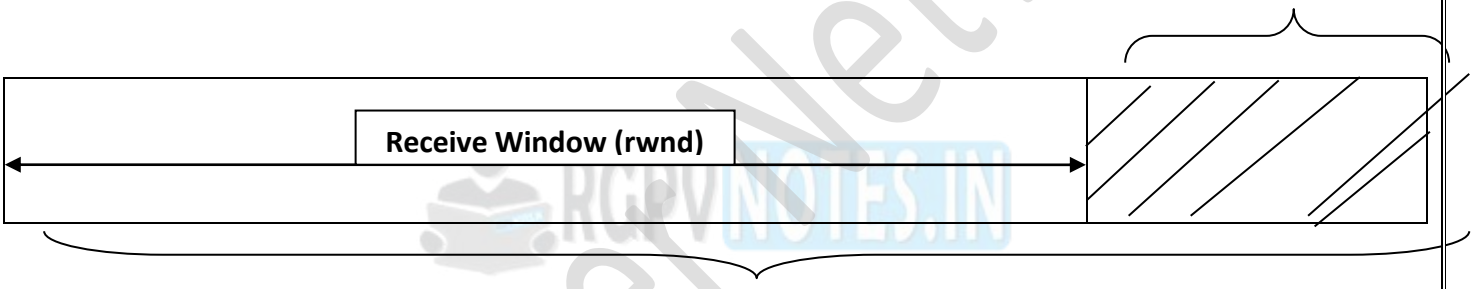
deliver packets out of order, and will wait for a retransmission in case it notices a gap in the byte stream. If we zoom in, we will see something like this.

Fig 5.7 TCP Flow Control.

TCP stores the data it needs to send in the send buffer, and the data it receives in the receive buffer. When the application is ready, it will then read data from the receive buffer.

Flow Control is all about making sure we don't send more packets when the receive buffer is already full, as the receiver wouldn't be able to handle them and would need to drop these packets.

To control the amount of data that TCP can send, the receiver will advertise its Receive Window (rwnd), that is, the spare room in the receive buffer. **Data in the Buffer**



TCP Receive Buffer

Fig 5.8 TCP Flow Control - Receive Window

Every time TCP receives a packet, it needs to send an ack message to the sender, acknowledging it received that packet correctly, and with this ack message it sends the value of the current receive window, so the sender knows if it can keep sending data.

TCP Congestion Control

TCP uses a congestion window and a congestion policy that avoid congestion. Previously, we assumed that only receiver can dictate the sender's window size. We ignored another entity here, the network. If the network cannot deliver the data as fast as it is created by the sender, it must tell the sender to slow down. In other words, in addition to the receiver, the network is a second entity that determines the size of the sender's window.

Congestion policy in TCP –

1. Slow Start Phase: starts slowly increment is exponential to threshold
2. Congestion Avoidance Phase: After reaching the threshold increment is by 1
3. Congestion Detection Phase: Sender goes back to Slow start phase or Congestion avoidance phase.

Slow Start Phase: exponential increment – In this phase after every RTT the congestion window size increments exponentially.

Initially cwnd = 1

After 1 RTT, cwnd = 2^ (1) = 2

2 RTT, cwnd = 2^ (2) = 4

3 RTT, cwnd = 2^ (3) = 8

Congestion Avoidance Phase: additive increment – This phase starts after the threshold value also denoted as ssthresh. The size of cwnd(congestion window) increases additive. After each RTT cwnd = cwnd + 1.

Initially cwnd = i
After 1 RTT, cwnd = i+1
2 RTT, cwnd = i+2
3 RTT, cwnd = i+3

Congestion Detection Phase: multiplicative decrement – If congestion occurs, the congestion window size is decreased. The only way a sender can guess that congestion has occurred is the need to retransmit a segment. Retransmission is needed to recover a missing packet which is assumed to have been dropped by a router due to congestion. Retransmission can occur in one of two cases: when the RTO timer times out or when three duplicate ACKs are received.

- **Case 1: Retransmission due to Timeout –** In this case congestion possibility is high.
  (a) ssthresh is reduced to half of the current window size.
  (b) Set cwnd = 1
  (c) Start with slow start phase again.
- **Case 2: Retransmission due to 3 Acknowledgement duplicates –** In this case congestion possibility is less.
  (a) ssthresh value reduces to half of the current window size.
  (b) Set cwnd= ssthresh.
  (c) Start with congestion avoidance phase.

  Example – Assume a TCP protocol experiencing the behaviour of slow start. At 5th transmission round with a threshold (ssthresh) value of 32 goes into congestion avoidance phase and continues till 10th transmission. At 10th transmission round, 3 duplicate ACKs are received by the receiver and enter into additive increase mode. Timeout occurs at 16th transmission round. Plot the transmission round (time) vs congestion window size of TCP segments.

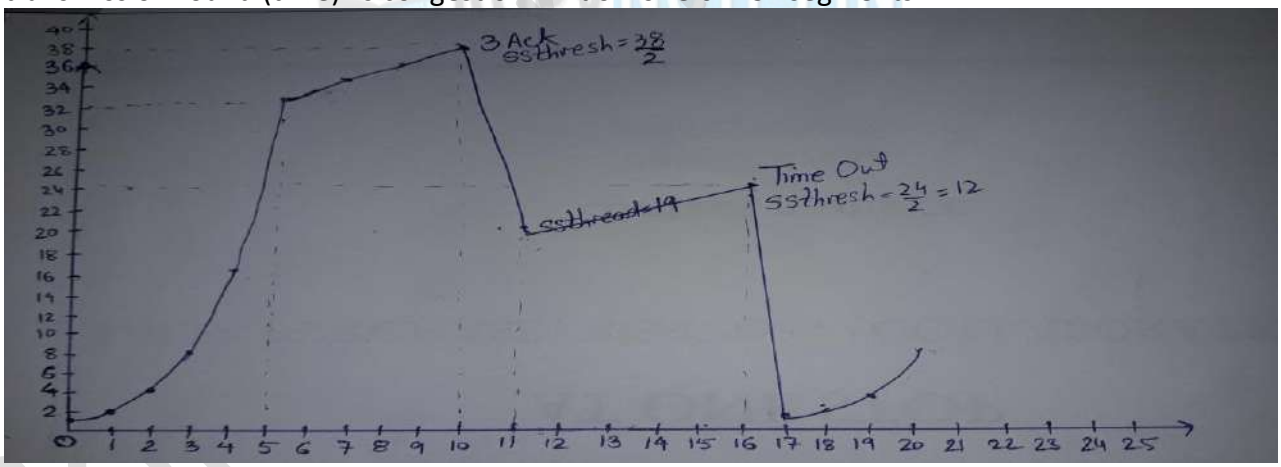


Fig 5.9 Example

TCP Header Format

Each TCP header has ten required fields totalling 20 bytes (160 bits) in size.
They can also optionally include an additional data section up to 40 bytes in size.
This is the layout of TCP headers:

1. Source TCP port number (2 bytes)
2. Destination TCP port number (2 bytes)
3. Sequence number (4 bytes)
4. Acknowledgment number (4 bytes)
5. TCP data offset (4 bytes)
6. Reserved data (3 bits)
7. Control flags (up to 9 bits)
8. Window size (2 bytes)
9. TCP checksum (2 bytes)

10. Urgent pointer (2 bytes)
11. TCP optional data (0-40 bytes)

TCP inserts header fields into the message stream in the order listed above.

- *Source and destination TCP port numbers* are the communication endpoints for sending and receiving devices.
- Message senders use *sequence numbers* to mark the ordering of a group of messages. Both senders and receivers use the acknowledgment *numbers* field to communicate the sequence numbers of messages that are either recently received or expected to be sent.
- The *data offset field* stores the total size of a TCP header in multiples of four bytes. A header not using the optional TCP field has a data offset of 5 (representing 20 bytes), while a header using the maximum-sized optional field has a data offset of 15 (representing 60 bytes).
- *Reserved data* in TCP headers always has a value of zero. This field serves the purpose of aligning the total header size as a multiple of four bytes (important for the efficiency of computer data processing).
- TCP uses a set of six standard and three extended *control flags* (each an individual bit representing *on* or *off*) to manage data flow in specific situations. One bit flag, for example, initiates TCP connection reset logic. The detailed operation of these fields goes beyond the scope of this article.
- TCP senders use a number called *window size* to regulate how much data they send to a receiver before requiring an acknowledgment in return. If the window size becomes too small, network data transfer will be unnecessarily slow, while if the window size becomes too large, the network link can become saturated (unusable for any other applications) or the receiver may not be able to process incoming data quickly enough (also resulting in slow performance). Windowing algorithms built into the protocol dynamically calculate size values and use this field of TCP headers to coordinate changes between senders and receivers.
- The *checksum* value inside a TCP header is generated by the protocol sender as a mathematical technique to help the receiver detect messages that are corrupted or tampered with.
- The urgent pointer field is often set to zero and ignored, but in conjunction with one of the control flags, it can be used as a data offset to mark a subset of a message as requiring priority processing.
- Usages of optional TCP data go beyond the scope of this article but include support for special acknowledgment and window scaling algorithms.

<table>
<tr><td colspan="3">Source Port Number
(2 Bytes)</td><td>Destination Port Number
(2 Bytes)</td></tr>
<tr><td colspan="4">Sequence Number
(4 Bytes)</td></tr>
<tr><td colspan="4">Acknowledgement Number
(4 Bytes)</td></tr>
<tr><td>Data Offset
(4 Bits)</td><td>Reserved
(3 Bits)</td><td>Control Flags
(9 Bits)</td><td>Window Size
(2 Bytes)</td></tr>
<tr><td colspan="3">Checksum
(2 Bytes)</td><td>Urgent Pointer
(2 Bytes)</td></tr>
<tr><td colspan="4">Optional Data</td></tr>
</table>

Fig 5.10 TCP Header Format (20-60 Bytes)

YOUTUBE LINK: https://www.youtube.com/watch?v=wlTtOqeklJI

TCP Timer Management

- TCP uses multiple timers (at least conceptually) to do its work. The most important of these is the RTO (Retransmission Time Out). When a segment is sent, a retransmission timer is started. If the segment is acknowledged before the timer expires, the timer is stopped. If, on the other hand, the timer goes off before the acknowledgement comes in, the segment is retransmitted (and the timer OS started again). It is difficult to predict how long should the timeout be.
- This problem is much more difficult in the transport layer than in data link protocols such as 802.11. In the latter case, the expected delay is measured in microseconds and is highly predictable (i.e.,

has a low variance), so the timer can be set to go off just slightly after the acknowledgement is expected, as shown in Fig below. Since acknowledgements are rarely delayed in the data link layer (due to lack of congestion), the absence of an acknowledgement at the expected time generally means either the frame or the acknowledgement has been lost.

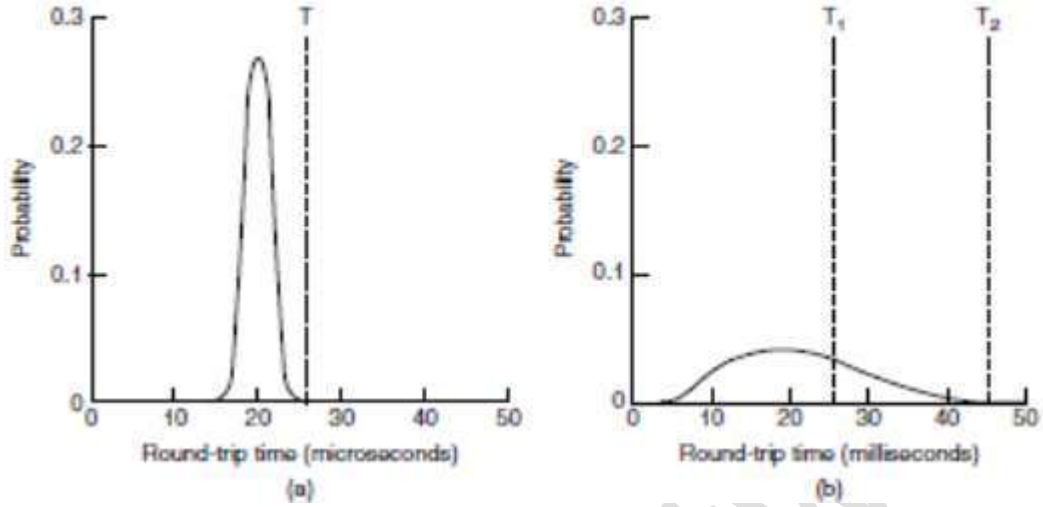


Fig. 5.11 (a) Probability density of acknowledgement arrival times in the data link layer. (b) Probability density of acknowledgement arrival times for TCP.

- TCP is faced with a radically different environment. The probability density function for the time it takes for a TCP acknowledgement to come back looks more like Fig. (b) than Fig. (a) as shown above. It is larger and more variable. Determining the round-trip time to the destination is tricky. Even when it is known, deciding on the timeout interval is also difficult. If the timeout is set too short, say, T1 in Fig. (b), unnecessary retransmissions will occur, clogging the Internet with useless packets. If it is set too long (e.g., T2), performance will suffer due to the long retransmission delay whenever a packet is lost. Furthermore, the mean and variance of the acknowledgement arrival distribution can change rapidly within a few seconds as congestion builds up or is resolved.
- The solution is to use a dynamic algorithm that constantly adapts the timeout interval, based on continuous measurements of network performance. The algorithm generally used by TCP is due to Jacobson (1988) and works as follows. For each connection, TCP maintains a variable, SRTT (Smoothed Round-Trip Time) that is the best current estimate of the round-trip time to the destination in question.
- When a segment is sent, a timer is started, both to see how long the acknowledgement takes and also to trigger a retransmission if it takes too long. If the acknowledgement gets back before the timer expires, TCP measures how long the acknowledgement took, say, R. It then updates SRTT according to the formula.

$SRTT = \alpha SRTT + (1-\alpha) R$

Where α is a smoothing factor that determines how quickly the old values are forgotten. Typically, α = 7/8. This kind of formula is a EWMA (Exponentially Weighted Moving Average) or low-pass filter that discards noise in the samples.

Session layer:

This layer is primarily concerned with coordinating

- Applications as they interact on different hosts.
- Support the dialog between cooperating application programs
- The session layer offers provisions for efficient data transfer.
- The session layer decides when to turn communication on and off between two computer.
- Provides duplex, half-duplex or simplex communications between devices.

Authentication: Before establishing a session with some network peer, it is important for one of the computers to know that another peer it is communicating to is a legitimate one.
Authorization: Authorization is more like "Are you authorized to do so?"
Example:
If someone knows my email address and password, he can easily authenticate himself as 'Amar Shekhar' and he can log in as well. However, since he is not the right person to access my personal email account, so he is not an authorized person to do so.
So the basic difference between the two is: authentication is the process of verifying that "you are who you say you are", authorization is the process of verifying that "you are permitted to do what you are trying to do". Authorization thus presupposes authentication.

Session layer protocol
PAP: Password Authentication Protocol (PAP) provides a simple method for the peer to establish its identity. This is done only upon initial link establishment.
SCP: The Session Control Protocol (SCP) manages logical links for DECnet (DECnet is a suite of network protocols created by Digital Equipment Corporation) connections.
H.245: Call Control Protocol for Multimedia Communication
It is a protocol for the transmission of call management and control signals in packet-based networks using H.323 equipment. The H.245 specification is used in audio, video, and data transmissions, as well as in voice over IP (VoIP). H.245 messages are sent over special channels called H.245 control channels.

Presentation Layer
It responds to service requests from the application layer and issues service requests to the session layer concerned with syntax and semantics of the information exchanged between two systems.

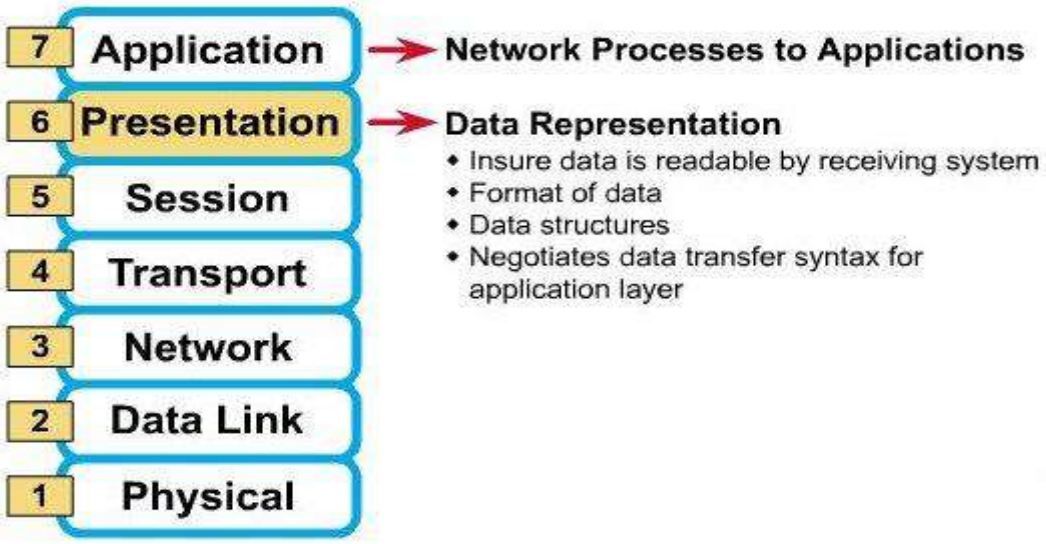


Fig. 5.12 Presentation Layer

Data Conversion- it is responsible for the conversion of computer data from one format to another. Different computers encode data in different ways on the basis of certain standards. On top of that, each computer program handles data in a different manner. Data conversion comes in handy in those situations when the representation of data is needed on different platforms.
Character Code Translation- Before being transmitted, the data remains in the form of characters and numbers. This data has to be changed to bit streams before transmission. The presentation layer is responsible for interoperability between encoding methods as different computers use different encoding methods. It translates data between the formats that a network requires and the format a computer needs.
Data Compression- Data compression is related closely to data representation. One way to transmit a 32-bit integer is to simply encode it as four bytes and send it on its way. However, if it is known that 95% of the numbers sent are between 0 and 250 then it may be better to transmit these integers as a single

unsigned byte, and to use the code 255 to indicate that the following data is a true 32-bit integer. In this case while it is true that 5 bytes will be needed instead of 4, the gain from being able to use one byte most of the time certainly offsets any losses.

Data compression can be approached in three general ways.

1. **The finiteness of the set of symbols: -** A typical book has about 20 characters in its title. Expressed in ASCII such a book title requires 140 bits. By simply giving each book a sequence number, it is possible to reduce the numbers of bits needed from 140 to 26 or fewer. However the receiver must have the numbered book list.
2. **The relative frequencies with which the symbols are used: -** If we are transmitting a DNA code and note that the relative frequencies of the four constituents of DNA are 0.7, 0.2, 0.07, and 0.03, then we can encode the transmission as 0, 10, 110, and 111 respectively. This leads to a tree encoding with 0 for left and 1 for right. (Related to Huffman coding)
3. **The context in which a symbol appears: -** If we are transmitting a DNA code and note that the signalling. Say we wish to transmit the following:
   0001000001000000100000000000000100000010001
   Run lengths of zeros. ie:
   3,5,6,14,6,3
   or
   011 101 110 111 111 110 011

Encryption and Decryption

Encryption is the process of translating plain text data (plaintext) into something that appears to be random and meaningless (ciphertext). Decryption is the process of converting ciphertext back to plaintext.

To encrypt more than a small amount of data, symmetric encryption is used. A symmetric key is used during both the encryption and decryption processes. To decrypt a particular piece of ciphertext, the key that was used to encrypt the data must be used.

The goal of every encryption algorithm is to make it as difficult as possible to decrypt the generated ciphertext without using the key. If a really good encryption algorithm is used, there is no technique significantly better than methodically trying every possible key. For such an algorithm, the longer the key, the more difficult it is to decrypt a piece of ciphertext without possessing the key. It is difficult to determine the quality of an encryption algorithm. Algorithms that look promising sometimes turn out to be very easy to break, given the proper attack. When selecting an encryption algorithm, it is a good idea to choose one that has been in use for several years and has successfully resisted all attacks.

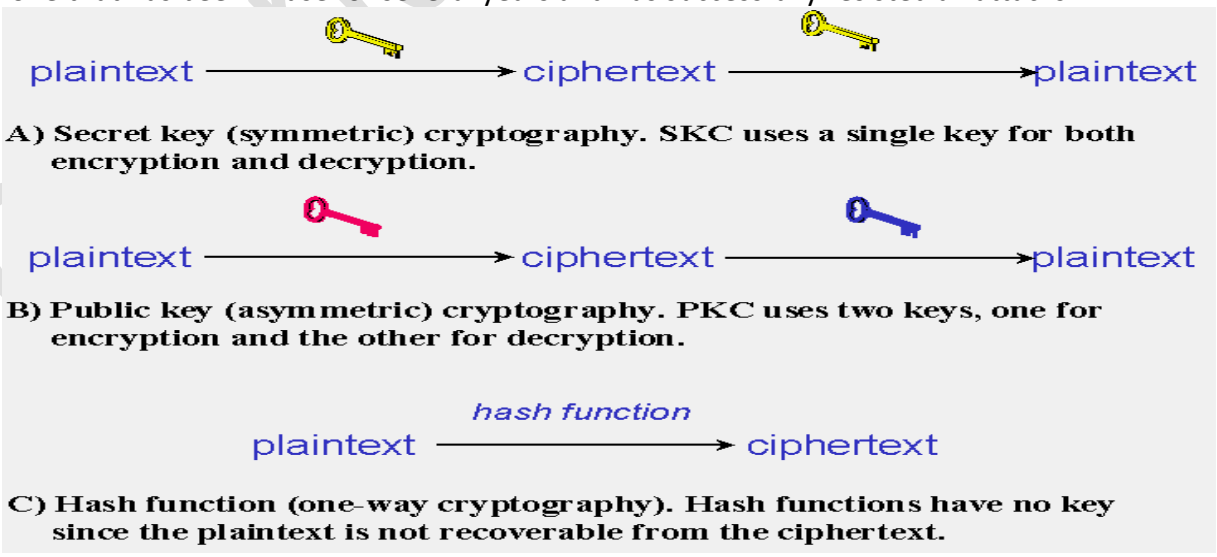


Fig. 5.13 Encryption and Decryption

Presentation layer protocol

LLP

The full form of LPP is Lightweight Presentation Protocol. It is presentation layer protocol Used to provide ISO presentation services on top of TCP/IP based protocol stacks. It is defined in RFC 1085. This protocol refers to an approach used for providing stream-lined support of open source interface (OSI) application services on top of Transmission Control Protocol/Internet Protocol-based (TCP/IP) network for some constrained environments . It is designed for particular class of OSI applications, namely those entities whose application context contains only an ACSE (Association Control Service Element) and ROSE (Remote Operations Service Element). Their is one more thing, Lightweight Presentation Protocol is not applicable to entities whose application context is more extensive. Used to provide ISO presentation services on top of TCP/IP based protocol stacks. It is defined in RFC 1085.

TELNET

A terminal emulation that enables a user to connect to a remote host or device using a telnet client, usually over port 23. For example, typing telnet hostname would connect a user to a host named hostname. Telnet enables a user to manage an account or device remotely. For example, a user may telnet into a computer that hosts their website to manage his or her files remotely.

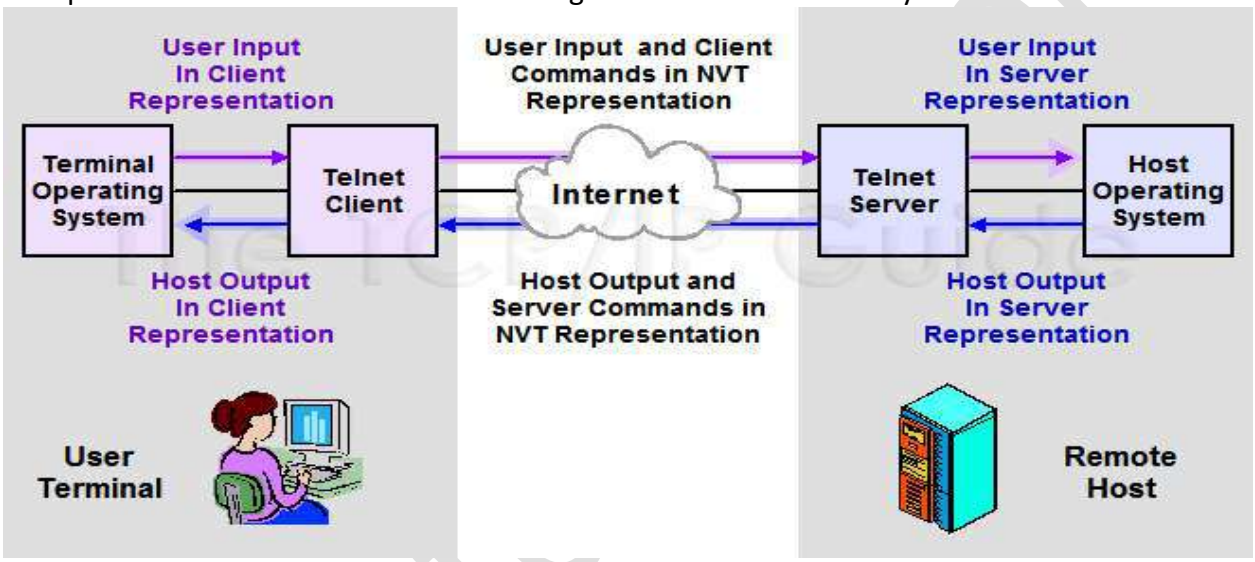


Fig. 5.14 TELNET

X.25 Packet Assembler Disassembler (PAD)

X.25 is a standard suite of protocols used for packet-switched communications over a wide area network—a WAN. A protocol is an agreed-upon set of procedures and rules. Two devices that follow the same protocols can understand each other and exchange data.

X-25 offered three basic layers of protocols:

- Physical layer
- Data link layer
- Packet layer

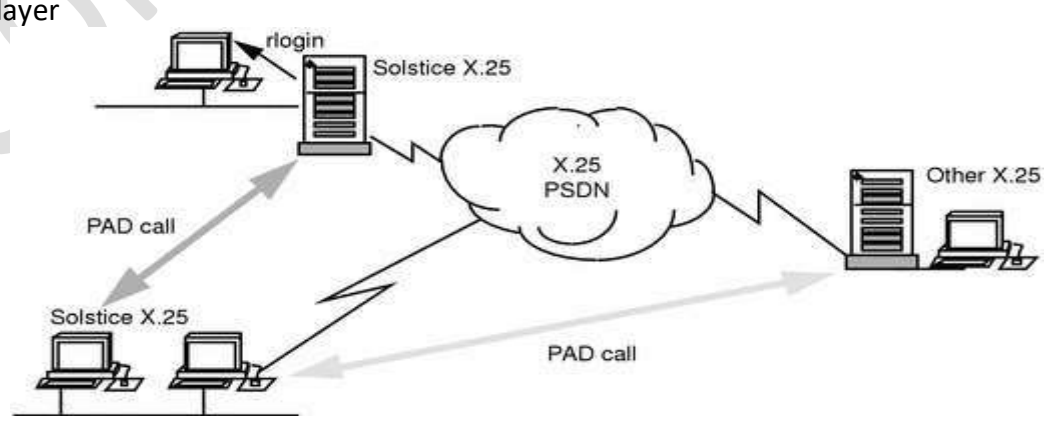


Fig. 5.15 X.25 Packet Assembler Disassembler (PAD)

Application Layer

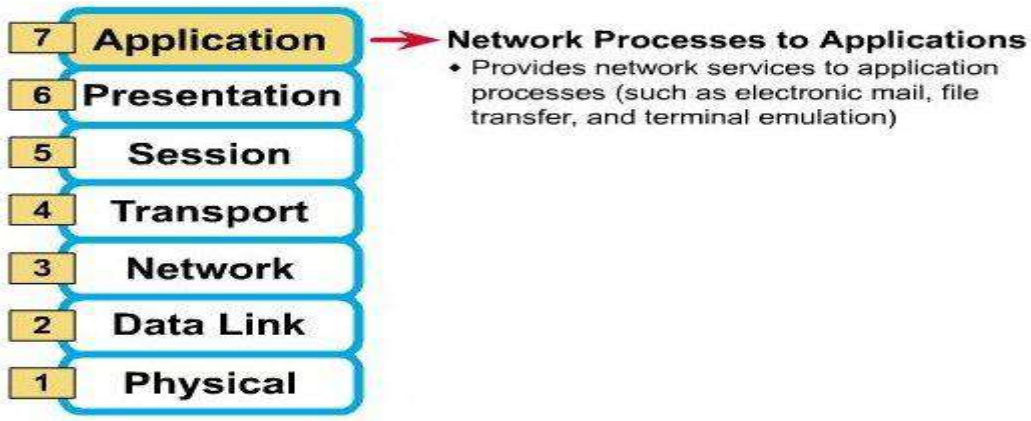


Fig. 5.16 Application Layer

HTTP : Short for HyperText Transfer Protocol, HTTP is a set of standards that allow users of the World Wide Web to exchange information found on web pages. When accessing any web page entering http:// in front of the address tells the browser to communicate over HTTP. For example, the URL for Computer Hope is https://www.computerhope.com. Today's browsers no longer require HTTP in front of the URL since it is the default method of communication. However, it is kept in browsers because of the need to separate protocols such as FTP.

FTP- Stands for "File Transfer Protocol." FTP is a protocol designed for transferring files over the Internet. Files stored on an FTP server can be accessed using an FTP client, such as a web browser, FTP software program, or a command line interface.

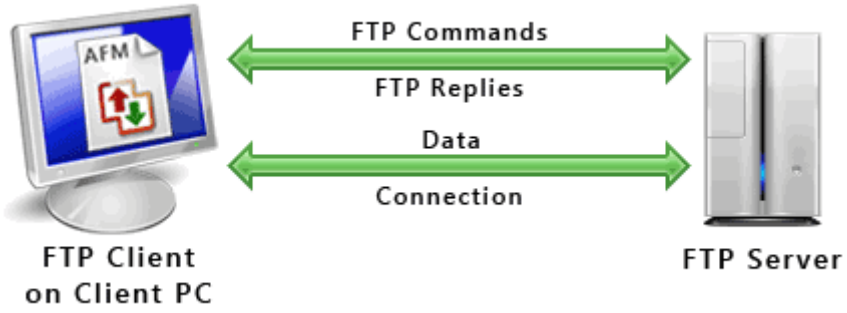


Fig. 5.17 FTP

SSH- The Secure Shell (SSH) protocol performs public-key encryption using a host key and a server key. SSH improves security by providing a means for the storage system to authenticate the client and by generating a session key that encrypts data sent between the client and storage system.

The SSH server version running on Data ONTAP is Data ONTAP SSH version 1.0. For information about the Common Vulnerabilities and Exposures (CVE) fixes implemented in Data ONTAP, see the Suspected Security Vulnerabilities page on the NetApp Support Site.

Data ONTAP supports the SSH 1.x protocol and the SSH 2.0 protocol.

Data ONTAP supports the following SSH clients:

- OpenSSH client version 4.4p1 on UNIX platforms
- SSH Communications Security client (SSH Tectia client) version 6.0.0 on Windows platforms
- Vandyke SecureCRT version 6.0.1 on Windows platforms
- PuTTY version 0.6.0 on Windows platforms
- F-Secure SSH client version 7.0.0 on UNIX platforms

SSH uses three keys to improve security:

- Host key

SSH uses the host key to encrypt and decrypt the session key. You determine the size of the host key, and Data ONTAP generates the host key when you configure SecureAdmin.

- Server key

SSH uses the server key to encrypt and decrypt the session key. You determine the size of the server key when you configure SecureAdmin. If SSH is enabled, Data ONTAP generates the server key when any of the following events occur:

- o You start SecureAdmin
- o An hour elapses
- o The storage system reboots

- Session key

SSH uses the session key to encrypt data sent between the client and storage system. The session key is created by the client. To use the session key, the client encrypts the session key using the host and server keys and sends the encrypted session key to the storage system, where it is decrypted using the host and server keys. After the session key is decrypted, the client and storage system can exchange encrypted data.

EMAIL: E-mail Protocols are set of rules that help the client to properly transmit the information to or from the mail server. Here in this tutorial, we will discuss various protocols such as SMTP, POP, and IMAP.

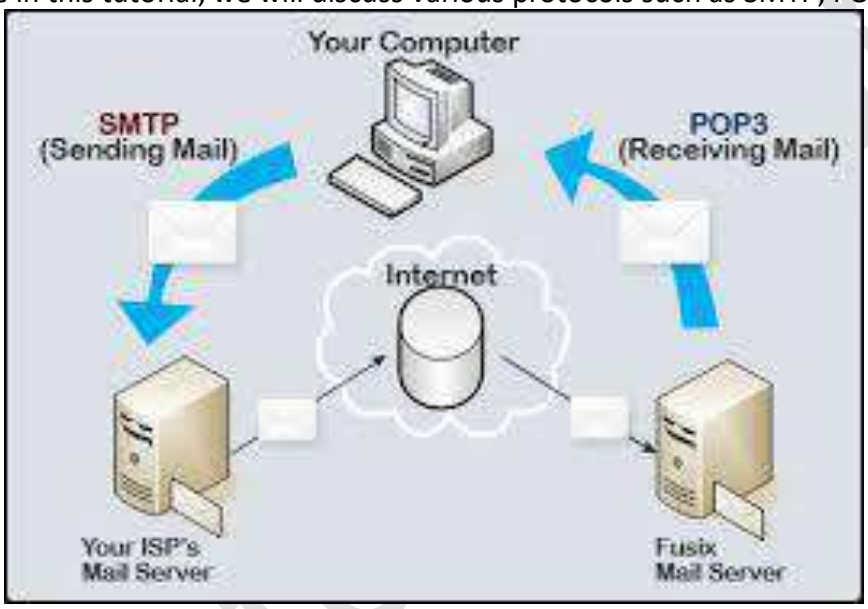


Fig. 5.18 e-mail Protocol

SMTP : SMTP stands for Simple Mail Transfer Protocol. It was first proposed in 1982. It is a standard protocol used for sending e-mail efficiently and reliably over the internet.

- SMTP is application level protocol.
- SMTP is connection oriented protocol.
- SMTP is text based protocol.
- It handles exchange of messages between e-mail servers over TCP/IP network.
- Apart from transferring e-mail, SMPT also provides notification regarding incoming mail.
- When you send e-mail, your e-mail client sends it to your e-mail server which further contacts the recipient mail server using SMTP client.
- These SMTP commands specify the sender's and receiver's e-mail address, along with the message to be send.
- The exchange of commands between servers is carried out without intervention of any user.
- In case, message cannot be delivered, an error report is sent to the sender which makes SMTP a reliable protocol.

IMAP : IMAP stands for Internet Mail Access Protocol. It was first proposed in 1986. There exist five versions of IMAP as follows:

1. Original IMAP
2. IMAP2
3. IMAP3
4. IMAP2bis
5. IMAP4

- IMAP allows the client program to manipulate the e-mail message on the server without downloading them on the local computer.
- The e-mail is hold and maintained by the remote server.
- It enables us to take any action such as downloading, delete the mail without reading the mail.It enables us to create, manipulate and delete remote message folders called mail boxes.
- IMAP enables the users to search the e-mails.
- It allows concurrent access to multiple mailboxes on multiple mail servers.

MIME:

Multipurpose Internet Mail Extension (MIME) is a standard which was proposed by Bell Communications in 1991 in order to expand limited capabilities of email.
MIME is a kind of add on or a supplementary protocol which allows non-ASCII data to be sent through SMTP. It allows the users to exchange different kinds of data files on the Internet: audio, video, images, application programs as well.
Limitations of Simple Mail Transfer Protocol (SMTP):

1. SMTP has a very simple structure
2. It's simplicity however comes with a price as it only send messages in NVT 7-bit ASCII format.
3. It cannot be used for languages that do not support 7-bit ASCII format such as- French, German, Russian, Chinese and Japanese, etc. so it cannot be transmitted using SMTP. So, in order to make SMTP more broad we use MIME.
4. It cannot be used to send binary files or video or audio data.

Features of MIME –

1. It is able to send multiple attachments with a single message.
2. Unlimited message length.
3. Binary attachments (executables, images, audio, or video files) which may be divided if needed.
4. MIME provided support for varying content types and multi-part messages.

MIME with SMTP and POP –SMTP transfers the mail being a message transfer agent from senders side to the mailbox of receiver side and stores it and MIME header is added to the original header and provides additional information. while POP being the message access agent organizes the mails from the mail server to the receivers computer. POP allows user agent to connect with the message transfer agent.

SNMP: Simple Network Management Protocol (SNMP) is an application-layer protocol used to manage and monitor network devices and their functions. SNMP provides a common language for network devices to relay management information within single- and multivendor environments in a local area network (LAN) or wide area network (WAN). The most recent iteration of SNMP, version 3, includes security enhancements that authenticate and encrypt SNMP messages as well as protect packets during transit.
One of the most widely used protocols; SNMP is supported on an extensive range of hardware -- from conventional network equipment like routers, switches and wireless access points to endpoints like

printers, scanners and internet of things (IoT) devices. In addition to hardware, SNMP can be used to monitor services such as Dynamic Host Configuration Protocol (DHCP). Software agents on these devices and services communicate with a network management system (NMS) also referred to as an SNMP manager, via SNMP to relay status information and configuration changes.

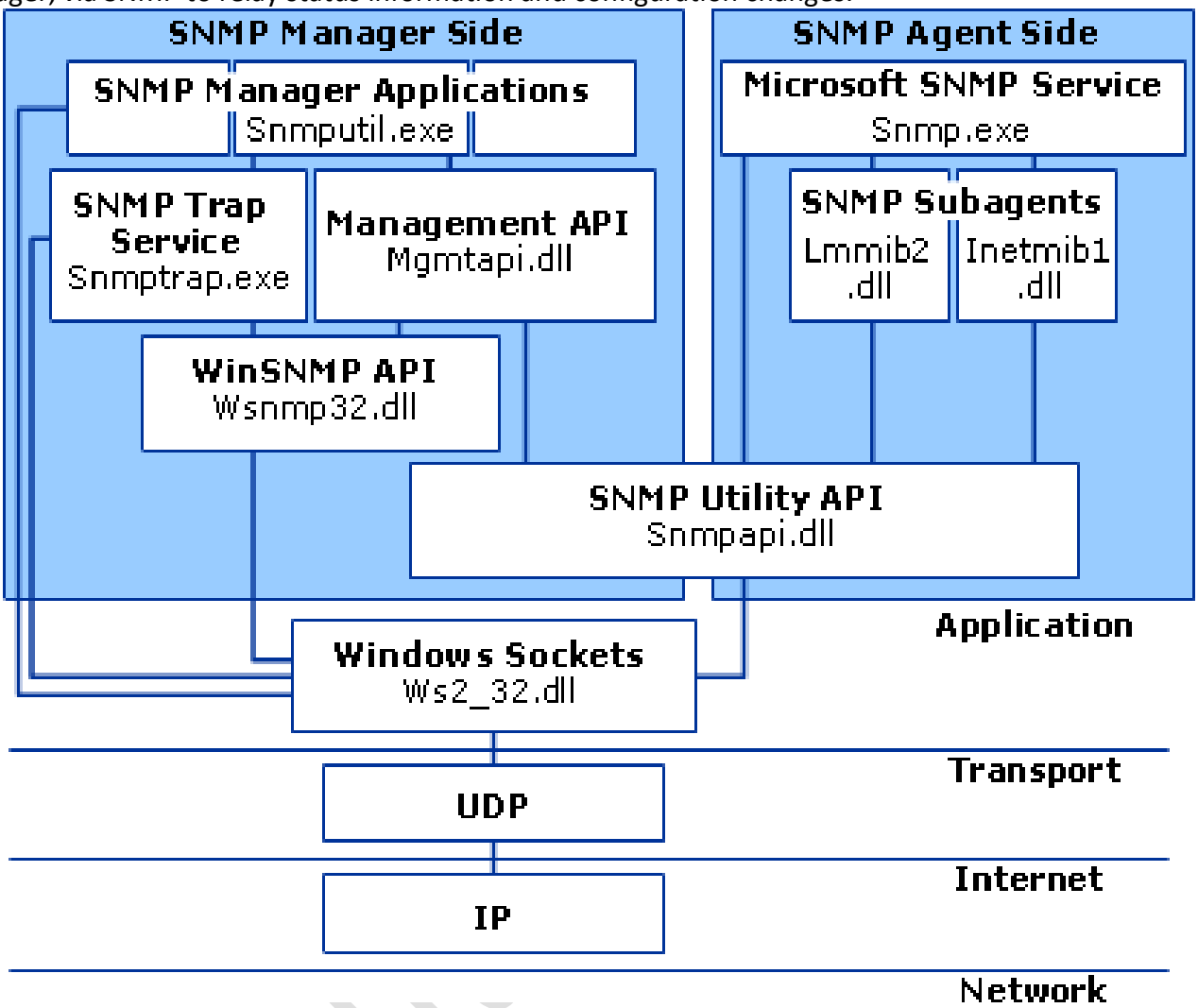


Fig. 5.19 Simple Network Management Protocol

DNS : The domain name system (DNS) is the way that internet domain names are located and translated into internet protocol (IP) addresses. The domain name system maps the name people use to locate a website to the IP addresses that a computer uses to locate a website. For example, if someone types TechTarget.com into a web browser, a server behind the scenes will map that name to the IP address 206.19.49.149.

Web browsing and most other internet activity rely on DNS to quickly provide the information necessary to connect users to remote hosts. DNS mapping is distributed throughout the internet in a hierarchy of authority. Access providers and enterprises, as well as governments, universities and other organizations, typically have their own assigned ranges of IP addresses and an assigned domain name; they also typically run DNS servers to manage the mapping of those names to those addresses. Most URLs are built around the domain name of the web server that takes client requests.

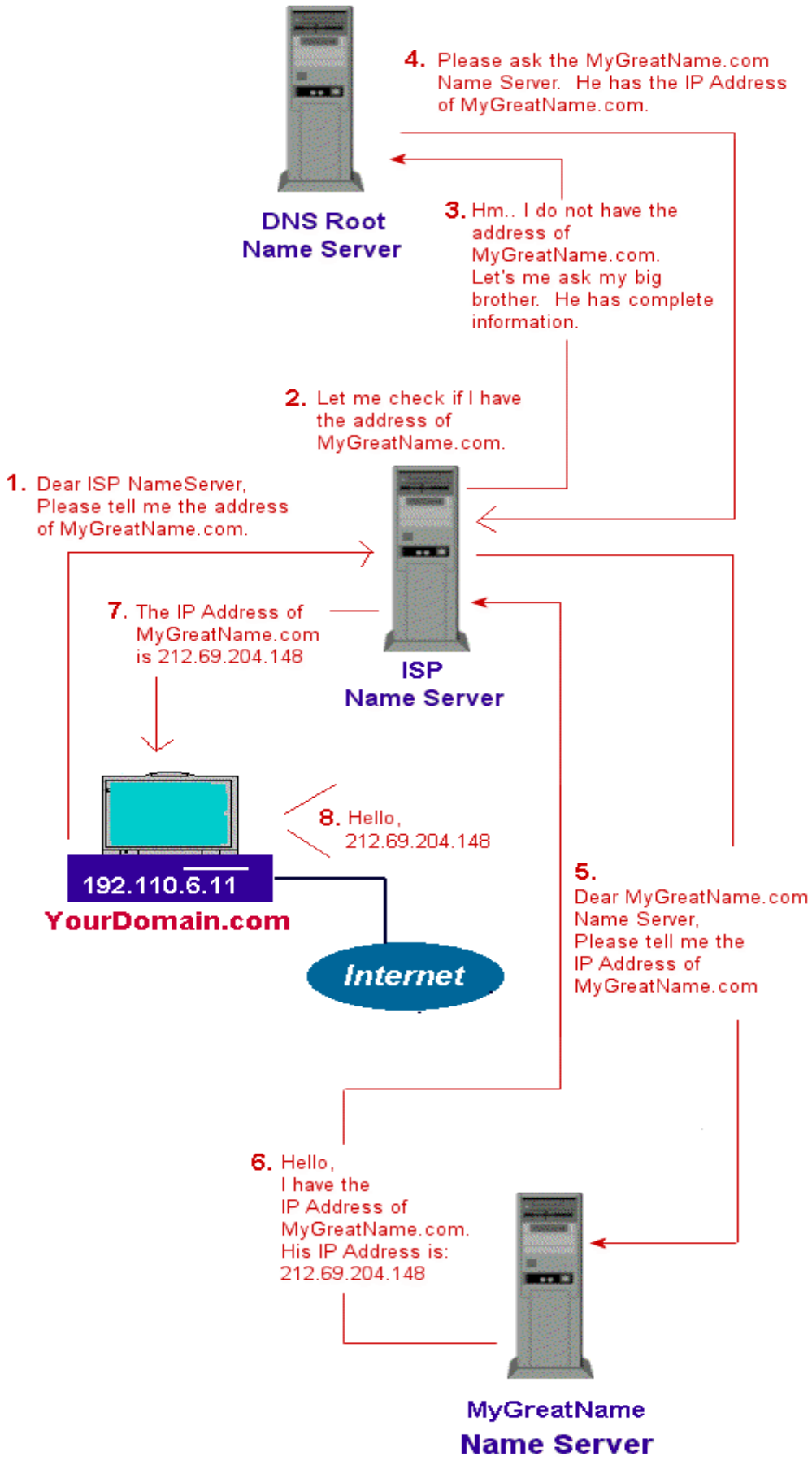


Fig. 5.20 Domain Name Server

RGPVNOTES.IN

We hope you find these notes useful.

You can get previous year question papers at https://qp.rgpvnotes.in .

If you have any queries or you want to submit your study notes please write us at rgpvnotes.in@gmail.com

LIKE & FOLLOW US ON FACEBOOK
facebook.com/rgpvnotes.in