A Course Material on

Computer Networks

By

Ms. P.Muruga Priya

HEAD & ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SASURIE COLLEGE OF ENGINEERING

VIJAYAMANGALAM – 638 056

QUALITY CERTIFICATE

This is to certify that the e-course material

Subject Code : CS6551

Subject : **COMPUTER NETWORKS**

Class : II Year CSE

being prepared by me and it meets the knowledge requirement of the university curriculum.

Signature of the Author

Name: P.Muruga Priya

Designation: AP/HOD,CSE

This is to certify that the course material being prepared by Ms.P.Muruga Priya is of adequate quality. She has referred more than five books among them minimum one is from abroad author.

Signature of HD

Name: P.Muruga Priya

TABLE OF CONTENTS			
S.No	**DATE**	**TOPIC**	**PAGE NO.**
UNIT I FUNDAMENTALS & LINK LAYER			
1		Building a network	**1**
2		Requirements	**1**
3		Layering and protocols	**3**
4		Internet Architecture	**23**
5		Network software	**24**
6		Performance ; Link layer Services	**28**
7		Framing	**29**
8		Error Detection	**31**
9		Flow control	**41**
UNIT II MEDIA ACCESS & INTERNETWORKING			
10		Media access control	**49**
11		Ethernet (802.3)	**52**
12		Wireless LANs	**53**
13		Ethernet (802.11)	**54**
14		Bluetooth	**58**
15		Switching and Bridging	**60**
16		Basic Internetworking	**64**
17		IP	**66**
18		CIDR	**74**
19		ARP	**76**
20		DHCP	**77**
21		ICMP	**79**
UNIT III ROUTING			
22		Routing	**80**
23		RIP	**81**
24		OSPF	**82**
25		metrics	**83**
26		Switch basics	**85**
27		Global Internet	**85**
28		Areas	**87**

CS6551 COMPUTER NETWORKS L T P C 3 0 0 3

OBJECTIVES:

The student should be made to:

- Understand the division of network functionalities into layers.
- Be familiar with the components required to build different types of networks
- Be exposed to the required functionality at each layer
- Learn the flow control and congestion control algorithms

UNIT I FUNDAMENTALS & LINK LAYER 9

Building a network – Requirements - Layering and protocols - Internet Architecture – Network software – Performance ; Link layer Services - Framing - Error Detection - Flow control

UNIT II MEDIA ACCESS & INTERNETWORKING 9

Media access control - Ethernet (802.3) - Wireless LANs – 802.11 – Bluetooth - Switching and Bridging – Basic Internetworking (IP, CIDR, ARP, DHCP, ICMP)

UNIT III ROUTING 9

Routing (RIP, OSPF, metrics) – Switch basics – Global Internet (Areas, BGP, IPv6), Multicast – addresses – multicast routing (DVMRP, PIM)

UNIT IV TRANSPORT LAYER 9

Overview of Transport layer - UDP - Reliable byte stream (TCP) - Connection management – Flow control - Retransmission – TCP Congestion control - Congestion avoidance (DECbit, RED) – QoS –Application requirements

UNIT V APPLICATION LAYER 9

Traditional applications -Electronic Mail (SMTP, POP3, IMAP, MIME) – HTTP – Web Services – DNS – SNMP **TOTAL: 45 PERIODS**

OUTCOMES:

At the end of the course, the student should be able to:

Identify the components required to build different types of networks
Choose the required functionality at each layer for given application
Identify solution for each functionality at each layer
Trace the flow of information from one node to another node in the network

TEXT BOOK:

1.Larry L. Peterson, Bruce S. Davie, "Computer Networks: A Systems Approach", Fifth Edition, Morgan Kaufmann Publishers, 2011.

REFERENCES:

1. James F. Kurose, Keith W. Ross, "Computer Networking - A Top-Down Approach Featuring the Internet", Fifth Edition, Pearson Education, 2009.
2. Nader. F. Mir, "Computer and Communication Networks", Pearson Prentice Hall Publishers, 2010.
3. Ying-Dar Lin, Ren-Hung Hwang, Fred Baker, "Computer Networks: An Open Source Approach", Mc Graw Hill Publisher, 2011.

4. Behrouz A. Forouzan, "Data communication and Networking", Fourth Edition, Tata McGraw – Hill,2011

UNIT I FUNDAMENTALS & LINK LAYER

Building a network – Requirements - Layering and protocols - Internet Architecture – Network software – Performance ; Link layer Services - Framing - Error Detection - Flow control

1. BUILDING A NETWORK

To build a computer network, that has the potential to grow to global proportions and to support applications as diverse as teleconferencing, video-on-demand, electronic commerce, distributed computing, and digital libraries.

What is network?

Network meant the set of serial lines used to attach dumb terminals to mainframe computers. To some, the term implies the voice telephone network. To others, the only interesting network is the cable network used to disseminate video signals.

The main thing these networks have in common is that they are specialized to handle one particular kind of data (keystrokes, voice, or video) and they typically connect to special-purpose devices (terminals, hand receivers, and television sets).

What distinguishes a computer network from these other types of networks?
Probably the most important characteristic of a computer network is its generality.

Computer networks are built primarily from general-purpose programmable hardware, and they are not optimized for a particular application like making phone calls or delivering television signals.

Instead, they are able to carry many different types of data, and they support a wide, and ever-growing, range of applications.

2. REQUIREMENTS

- Connectivity
- Cost-Effective Resource Sharing
- Support for Common Services
- Performance

Requirements differ according to the perspective:

1. Application programmer
List the services that his or her application needs.
Example: A guarantee that each message it sends will be delivered without error within a certain amount of time.

2. Network designer
List the properties of a cost-effective design.
Example: The network resources efficiently utilized and fairly allocated to different users.

3. Network provider

List the characteristics of a system that is easy to administer and manage.
Example: Fault can be easily isolated and it is easy to account for usage.

2.1 Connectivity

A network must provide connectivity among a set of computers

- Links and Nodes
- Types of Links or Connections
- Direction of Data Flow
- Unicast, Broadcast and Multicast

2.1.1 Links and Nodes

A network consists of two or more computers directly connected by some physical medium, such as a coaxial cable or an optical fiber. Such a physical medium is called as **links.**
The links are connected to the computers named as **nodes.**

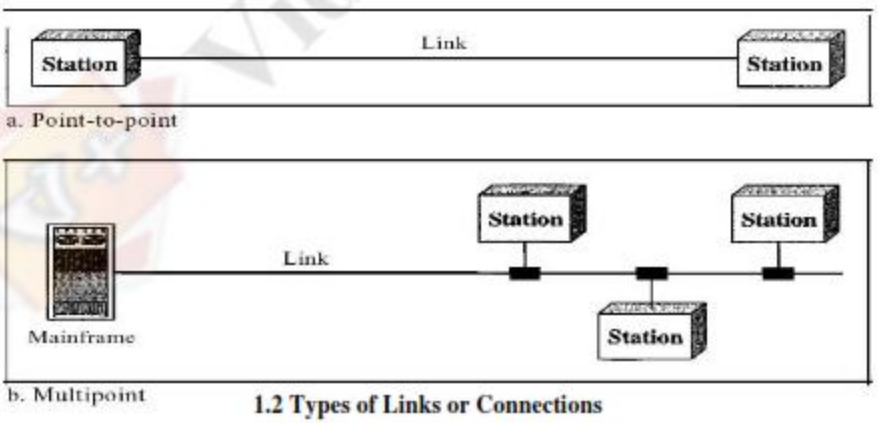
2.1.2 Types of Links or Connections

Point-to-Point

A point-to-point connection provides a dedicated link between two devices. The entire capacity of the link is reserved for transmission between those two devices.

Multipoint

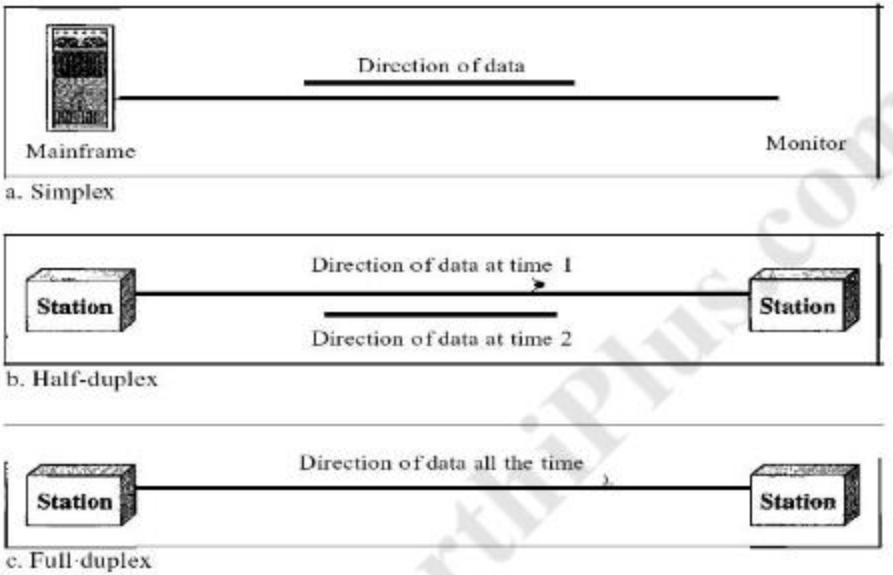
A multipoint (also called multidrop) connection is one in which more than two specific devices share a single link. In a multipoint environment, the capacity of the channel is shared, either spatially or temporally. If several devices can use the link simultaneously, it is a *spatially shared* connection. If users must take turns, it is a *timeshared* connection.



1.2 Types of Links or Connections

2.1.3 Direction of Data Flow

Communication between two devices can be simplex, half-duplex, or full-duplex as shown in Figure



Simplex:

In simplex mode, the communication is unidirectional, as on a one-way street. Only one of the two devices on a link can transmit; the other can only receive (Fig a). Keyboards and traditional monitors are examples of simplex devices.

Half-Duplex:

In half-duplex mode, each station can both transmit and receive, but not at the same time. When one device is sending, the other can only receive, and vice versa(Fig b). The half-duplex mode is like a one-lane road with traffic allowed in both directions.

Full-Duplex:

In full-duplex both stations can transmit and receive simultaneously (Fig c)

2.1.3 Unicast, Broadcast and Multicast

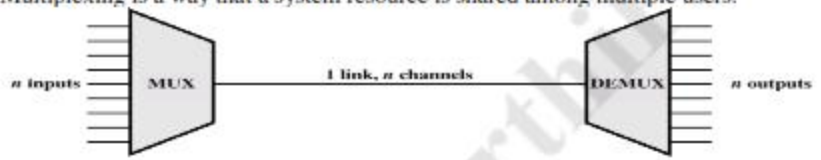
Unicast
Unicast is the term used to describe communication where a piece of information is sent from one point to another point. In this case there is just one sender, and one receiver.

Broadcast
Broadcast is the term used to describe communication where a piece of information is sent from one point to all other points. In this case there is just one sender, but the information is sent to all connected receivers.

Multicast
Multicast is the term used to describe communication where a piece of information is sent from one or more points to a set of other points. In this case there is may be one or more senders, and the information is distributed to a set of receivers (theer may be no receivers, or any other number of receivers).

2.2 Cost-Effective Resource Sharing

Multiplexing is a way that a system resource is shared among multiple users.



Two or more simultaneous transmissions on a single circuit. Transparent to end user. Multiplexing cost less.

Multiple telephone channels may share a transmission link by means of multiplexing – this sharing is static
–FDM (Frequency Division Multiplexing) is used in analogue systems (a telephone analogue channel has a nominal bandwidth of 4 kHz)
– STDM (Synchronous Time Division Multiplexing) is used in digital systems (the basic telephone digital channel has a capacity of 64 kbit/s)

2.3 Support for Common Services

A computer network provides more than packet delivery between nodes. We don't want application developers to rewrite for each application higher layer networking services.
The channel is a pipe connecting two applications. How to fill the gap between the underlying network capability and applications requirements? a set of common services– Delivery guarantees, security, delay.

2.3.1 Types of Applications

Interactive terminal and computer sessions:– Small packet length, small delay, high reliability.

- File transfer:– High packet length, high delay, high reliability
- Voice application:– Small packet length, small delay, small reliability, high arrival rate
- Video-on-demand:– Variable/high packet length, fixed delay, small reliability

- Video-conferencing– Variable/high packet length, small delay, small reliability

2.4 NETWORK CRITERIA

A network must be able to meet a certain number of criteria. The most important of these are performance, reliability, and security.

Performance:

Performance can be measured in many ways, including transit time and response time. Transit time is the amount of time required for a message to travel from one device to another. Response time is the elapsed time between an inquiry and a response. The performance of a network depends on a number of factors, including the number of users, the type of transmission medium, the capabilities of the connected hardware, and the efficiency of the software. Performance is often evaluated by two networking metrics: throughput and delay. We often need more throughputs and less delay. However, these two criteria are often contradictory. If we try to send more data to the network, we may increase throughput but we increase the delay because of traffic congestion in the network.

Reliability:

In addition to accuracy of delivery, network reliability is measured by the frequency of failure, the time it takes a link to recover from a failure, and the network's robustness in a catastrophe.

Security:

Network security issues include protecting data from unauthorized access, protecting data from damage and development, and implementing policies and procedures for recovery from breaches and data losses.

2.4.1 CATEGORIES OF NETWORK

There are three primary categories are,

1. Local area network.
2. Metropolitan area network.
3. Wide area network.

1. Local Area Network:

They are usually privately owned and link the devices in a single office, building and campus. Currently LAN size is limited to a few kilometers. It may be from two PC"s to throughout a company.

The most common LAN topologies are bus, ring and star. They have data rates from 4 to

16 Mbps. Today the speed is on increasing and can reach 100 mbps.

2. Metropolitan Area Network:

They are designed to extend over an entire city. It may be a single network or connecting a number of LANs into a large network. So the resources are shared between LANs. Example of MAN is, telephone companies provide a popular MAN service called switched multi megabit data service (SMDS).

3. Wide Area Network:

It provides a long distance transmission of data, voice, image and video information over a large geographical are like country, continent or even the whole world.

2.4.2 TOPOLOGIES:

Topology refers to the way a network is laid out either physically or logically. Two or more devices connect to a link; two or more links form a topology. It is the geographical representation of the relationship of all the links and linking devices to each other.

1. Mesh
2. Star
3. Tree
4. Bus
5. Ring

1. Mesh Topology:

Here every device has a dedicated point to point link to every other device. A fully connected mesh can have n(n-1)/2 physical channels to link n devices. It must have n-1 IO ports.

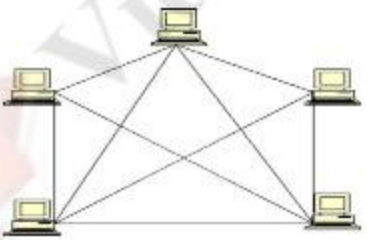


Figure: Mesh Topology

Advantages:

1. They use dedicated links so each link can only carry its own data load. So traffic problem can be avoided.

2. It is robust. If anyone link get damaged it cannot affect others
3. It gives privacy and security
4. Fault identification and fault isolation are easy.
5.

Disadvantages:

1. The amount of cabling and the number IO ports required are very large. Since every device is connected to each other devices through dedicated links.
2. The sheer bulk of wiring is larger then the available space
3. Hardware required to connect each device is highly expensive.

Example:

A mesh network has 8 devices. Calculate total number of cable links and IO ports needed.

Solution:

Number of devices = 8 Number of links = n (n-1)/2

= 8(8-1)/2

= 28

Number of port/device = n-1

= 8-1 = 7

2. STAR TOPOLOGY:

Here each device has a dedicated link to the central „hub". There is no direct traffic between devices. The transmission are occurred only through the central controller namely hub.

Figure: Star Topology

Advantages:

1. Less expensive then mesh since each device is connected only to the hub.
2. Installation and configuration are easy.

3. Less cabling is need then mesh.
4. Robustness.
5. Easy to fault identification & isolation.

Disadvantages:

1. Even it requires less cabling then mesh when compared with other topologies it still large.

TREE TOPOLOGY:

It is a variation of star. Instead of all devices connected to a central hub here most of the devices are connected to a secondary hub that in turn connected with central hub. The central hub is an active hub. An active hub contains a repeater, which regenerate the received bit pattern before sending.

Figure: Tree Topology

The secondary hub may be active or passive. A passive hub means it just precedes a physical connection only.

Advantages:

1. Can connect more than star.
2. The distance can be increased.
3. Can isolate and prioritize communication between different computers.

4. BUS TOPOLOGY:

A bus topology is multipoint. Here one long cable is act as a backbone to link all the devices are connected to the backbone by drop lines and taps. A drop line is the connection between the devices and the cable. A tap is the splice into the main cable or puncture the sheathing.

Figure: Bus Topology

Advantages:

1. Ease of installation.
2. Less cabling.

Disadvantages:

1. Difficult reconfiguration and fault isolation.
2. Difficult to add new devices.
3. Signal reflection at top can degradation in quality
4. If any fault in backbone can stops all transmission.

5. RING TOPOLOGY:

Here each device has a dedicated connection with two devices on either side of it. The signal is passed in one direction from device to device until it reaches the destination and each device have repeater.

Figure: Ring Topology

Advantages:

1. Easy to install.
2. Easy to reconfigure.
3. Fault identification is easy.

Disadvantages:

1. Unidirectional traffic.
2. Break in a single ring can break entire network.

PROTOCOLS AND STANDARDS

Protocols:

In computer networks, communication occurs between entries in different systems. An entity is anything capable of sending or receiving information. But two entities cannot communicate each other as sending or receiving. For communication occurs the entities must agree on a protocol.

A protocol is a set of rules that govern data communication. A protocol defines what is communicated how it is communicated, and when it is communicated. The key elements of a protocol are syntax, semantics and timing.

Syntax:

Syntax refers to the structure or format of the data, means to the order how it is presented.

Semantics:

Semantics refers to the meaning of each section of bits. How is a particular pattern to be interpreted, and when action is to be taken based on the interpretation.

Timing:

Timing refers to two characteristics. They are,

1. When data should be sent
2. When data to be received.

Standards:

A standard provides a model for development of a product, which is going to develop. Standards are essential to create and maintain a product.

Data communication products are fall into two categories. They are,

1. De facto
2. De jure

1. De facto:

They are further classified into

1. Proprietary
2. Non proprietary

1. Proprietary:

They are originally invented by a commercial organization as a basis for the operation of its product. They are wholly owned by the company, which invented them. They are closed standards.

2. Nonproprietary:

Groups or committees that have passed them into public domain develop them. They are open standards.

2. De jure:

They have been legislated by an officially recognized body.

STANDARDS ORGANIZATION:

Standards are developed by,

1. Standards creation committee
2. Forums
3. Regularity agencies

1. Standards creation committees:

1. International Standards Organization (ISO)
2. International Telecommunications Union – Telecommunications Standards Section (ITU-T formally CCITT)
3. The American National Standards Institute (ANSI)
4. The Institute of Electrical and Electronics Engineers (IEEE)

5. The Electronic Industries Association (EIA)
6. Telcordia

2. Forums:

1. Frame Relay Forum
2. ATM Forum & ATM consortium
3. Internet Society (ISOC) & Internet Engineering Task Force (IETF)

3. Regularity Agencies:

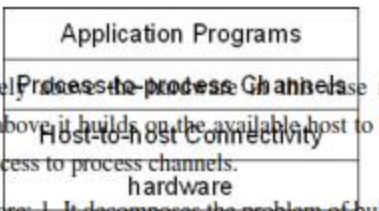
1. Federal Communication commission

NETWORK ARCHITECTURE

A computer network must provide general, cost effective, fair and robust among a large number of computers. It must evolve to accommodate changes in both the underlying technologies. To help to deal this network designers have developed general blueprints called network architecture that guide the design and implementation of networks.

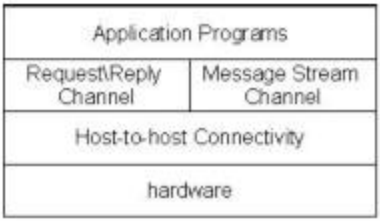
3.LAYERING AND PROTOCOL

To reduce the complexity of getting all the functions maintained by one a new technique called layering technology was introduced. In this, the architecture contains several layers and each layer is responsible for certain functions. The general idea is that the services offered by underlying hardware, and then add a sequence of layers, each providing a higher level of service. The services provided at the higher layers are implemented in terms of the services provided by the lower layers. A simple network has two layers of abstraction sandwiched between the application program and the underlying hardware.



The layer immediately above the hardware in this case might provide host to host connectivity, and the layer above it builds on the available host to host communication service and provides support for process to process channels.

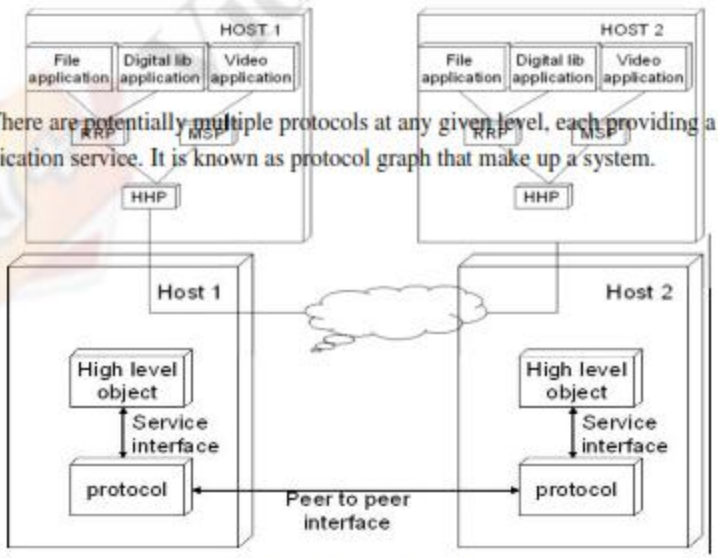
Features of layering are: 1. It decomposes the problem of building a network into more manageable components. 2. It provides a more modular design. Addition of new services and modifications are easy to implement.



In process to process channels, they have two types of channels. One for request\reply service and the other for message stream service.

A protocol provides a communication service that higher level objects use to exchange message. Each protocol defines two different interfaces. First it defines a service interface to other objects on the same system that want to use its communication services. This interface defines the operations that local objects can perform on the protocol. Second a protocol defines a peer interface to its counterpart on another machine. It defines the form and meaning of message exchanged between protocol peers to implement the communication service.

There are potentially multiple protocols at any given level, each providing a different communication service. It is known as protocol graph that make up a system.



ISO / OSI MODEL:

ISO refers International Standards Organization was established in 1947, it is a multinational body dedicated to worldwide agreement on international standards.

OSI refers to Open System Interconnection that covers all aspects of network communication. It is a standard of ISO.

Here **open system** is a model that allows any two different systems to communicate regardless of their underlying architecture. Mainly, it is not a protocol it is just a model.

OSI MODEL

The open system interconnection model is a layered framework. It has seven separate but interrelated layers. Each layer having unique responsibilities.

ARCHITECTURE

The architecture of OSI model is a layered architecture. The seven layers are,

1. Physical layer
2. Datalink layer
3. Network layer
4. Transport layer
5. Session layer
6. Presentation layer
7. Application layer

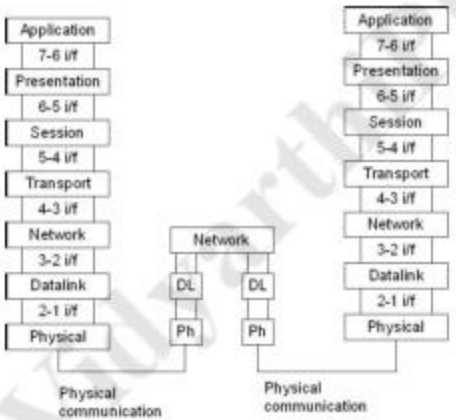
The figure shown below shows the layers involved when a message sent from A to B pass through some intermediate devices.

Both the devices A and B are formed by the framed architecture. And the intermediate nodes only having the layers are physical, Datalink and network. In every

device each layer gets the services from the layer just below to it. When the device is connected to some other device the layer of one device communicates with the corresponding layer of another device. This is known as **peer to peer process**.

Each layer in the sender adds its own information to the message. This information is known is **header** and **trailers**. When the information added at the beginning of the data is known as header. Whereas added at the end then it called as trailer. Headers added at layers 2, 3, 4, 5, 6. Trailer added at layer 2.

Each layer is connected with the next layer by using interfaces. Each interface defines what information and services a layer must provide for the layer above it.



ORGANIZATION OF LAYERS

The seven layers are arranged by three sub groups.

1. Network Support Layers
2. User Support Layers
3. Intermediate Layer

Network Support Layers:

Physical, Datalink and Network layers come under the group. They deal with the physical aspects of the data such as electrical specifications, physical connections,

physical addressing, and transport timing and reliability.

User Support Layers:

Session, Presentation and Application layers comes under the group. They deal with the interoperability between the software systems.

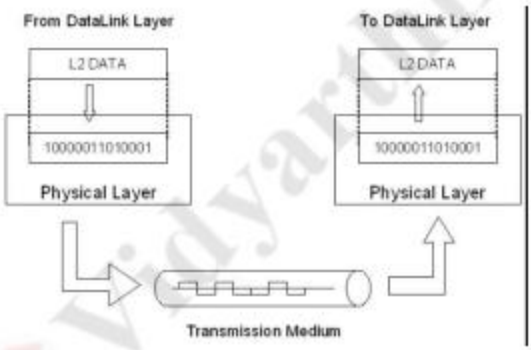
Intermediate Layer

The transport layer is the intermediate layer between the network support and the user support layers.

FUNCTIONS OF THE LAYERS

PHYSICAL LAYER

The physical layer coordinates the functions required to transmit a bit stream over a physical medium. It deals with the mechanical and electrical specifications of the interface and the transmission medium.



The functions are,

1. **Physical Characteristics Of Interfaces and Media:**
   - It defines the electrical and mechanical characteristics of the interface and the media.
   - It defines the types of transmission medium
2. **Representation of Bits**
   - To transmit the stream of bits they must be encoded into signal.
   - It defines the type of encoding weather **electrical or optical**.

3. **Data Rate**
   - It defines the transmission rate i.e. the number of bits sent per second.
4. **Synchronization of Bits**
   - The sender and receiver must be synchronized at bit level.
5. **Line Configuration**
   - It defines the type of connection between the devices.
   - Two types of connection are,
     1. point to point
     2. multipoint
6. **Physical Topology**
   - It defines how devices are connected to make a network.
   - Five topologies are,
     1. mesh
     2. star
     3. tree
     4. bus
     5. ring
7. **Transmission Mode**

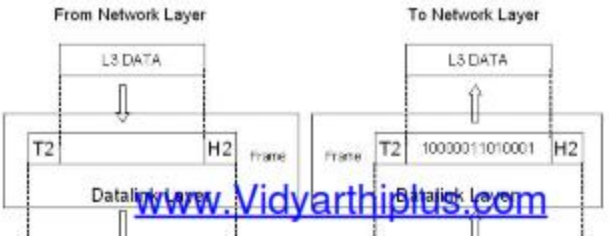
   It defines the direction of transmission between devices.

   Three types of transmission are,

   1. simplex
   2. half duplex
   3. full duplex

DATALINK LAYER

Datalink layer responsible for node-to-node delivery.



The responsibilities of Datalink layer are,

1. **Framing**

   It divides the stream of bits received from network layer into manageable data units called **frames.**

2. **Physical Addressing**

   - It adds a header that defines the physical address of the sender and the receiver.
   - If the sender and the receiver are in different networks, then the receiver address is the address of the device which connects the two networks.

3. **Flow Control**

   - It imposes a flow control mechanism used to ensure the data rate at the sender and the receiver should be same.

4. **Error Control**

   - To improve the reliability the Datalink layer adds a trailer which contains the error control mechanism like CRC, Checksum etc.

5. **Access Control**

   - When two or more devices connected at the same link, then the Datalink layer used to determine which device has control over the link at any given time.

NETWORK LAYER

When the sender is in one network and the receiver is in some other network then the network layer has the responsibility for the source to destination delivery.

The responsibilities are,

1. **Logical Addressing**

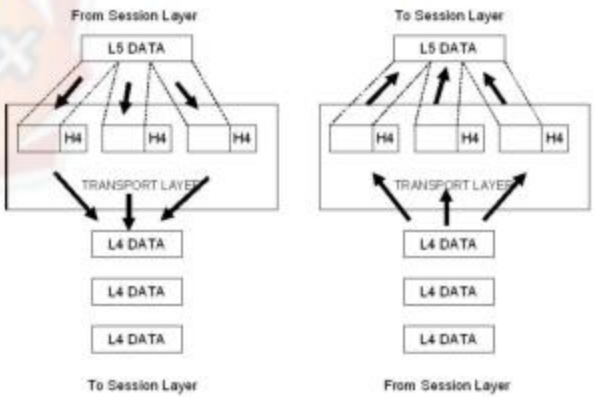
   * If a packet passes the network boundary that is when the sender and receiver are places in different network then the network layer adds a header that defines the logical address of the devices.

2. **Routing**

   * When more than one networks connected and to form an internetwork, the connecting devices route the packet to its final destination.
   * Network layer provides this mechanism.

TRANSPORT LAYER

The network layer is responsible for the end to end delivery of the entire message. It ensures that the whole message arrives in order and intact. It ensures the error control and flow control at source to destination level.

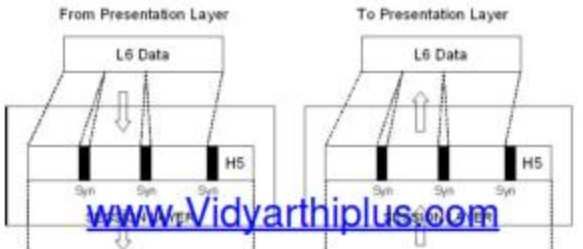


The responsibilities are,

1. **Service point Addressing**
   - A single computer can often run several programs at the same time.
   - The transport layer gets the entire message to the correct process on that computer.
   - It adds a header that defines the port address which used to identify the exact process on the receiver.
2. **Segmentation and Reassembly**
   - A message is divided into manageable units called as segments.
   - Each segment is reassembled after received that information at the receiver end.
   - To make this efficient each segment contains a sequence number.
3. **Connection Control**
   - The transport layer creates a connection between the two end ports.
   - It involves three steps. They are,
   1. connection establishment
   2. data transmission
   3. connection discard
4. **Flow Control**
   - Flow control is performed at end to end level
5. **Error Control**
   - Error control is performed at end to end level.

SESSION LAYER

It acts as a dialog controller. It establishes, maintains and synchronizes the interaction between the communication devices.



The responsibilities are,

1. **Dialog Control**
   - The session layer allows two systems to enter into a dialog.
   - It allows the communication between the devices.
2. **Synchronization**

   It adds a synchronization points into a stream of bits.

PRESENTATION LAYER

The presentation layer is responsible for the semantics and the syntax of the information exchanged.

The responsibilities are,

1. **Translation**
   - Different systems use different encoding systems.
   - The presentation layer is responsible for interoperability between different systems.
   - The presentation layer t the sender side translates the information from the sender dependent format to a common format. Likewise, at the receiver side presentation layer translate the information from common format to receiver dependent format.

2. **Encryption**
   - To ensure security encryption/decryption is used
   - Encryption means transforms the original information to another form

- Decryption means retrieve the original information from the encrypted data

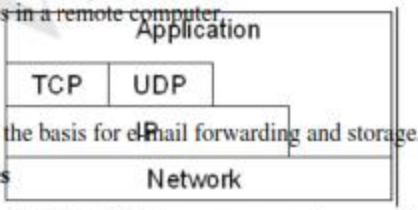
3. **Compression**
   - It used to reduce the number of bits to be transmitted.

APPLICATION LAYER

The application layer enables the user to access the network. It provides interfaces between the users to the network.

The responsibilities are,

1. **Network Virtual Terminal**
   - It is a software version of a physical terminal and allows a user to log on to a remote host.
2. **File Transfer, Access, and Management**
   - It allows a user to access files in a remote computer, retrieve files, and manage or control files in a remote computer.
   -
3. **Mail Services**
   - It provides the basis for e-mail forwarding and storage.
4. **Directory Services**
   - It provides distributed database sources and access for global information about various objects and services.



4.INTERNET ARCHITECTURE

The internet architecture evolved out of experiences with an earlier packet switched network called the ARPANET. Both the Internet and the ARPANET were funded by the

Advanced Research Projects Agency (ARPA).

The Internet and ARPANET were around before the OSI architecture, and the experience gained from building them was a major influence on the OSI reference model. Instead of having seven layers, a four layer model is often used in Internet.

At the lowest level are a wide variety of network protocols, denoted NET1, NET2 and so on. The second layer consists of a single protocol the Internet Protocol IP. It supports the interconnection of multiple networking technologies into a single, logical internetwork.

The third layer contains two main protocols the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). TCP provides a reliable byte stream channel, and UDP provides unreliable datagram delivery channel. They are called as end to end protocol they can also be referred as transport protocols.

Running above the transport layer, a range of application protocols such as FTP, TFTP, Telnet, and SMTP that enable the interoperation of popular applications.

ERROR

Networks must be able to transfer data from one device to another with complete accuracy. Some part of a message will be altered in transit than that the entire content will arrive intact. Many factors like line noise can alter or wipe out one or more bits of a given data unit. This is known as errors.

TYPES OF ERRORS

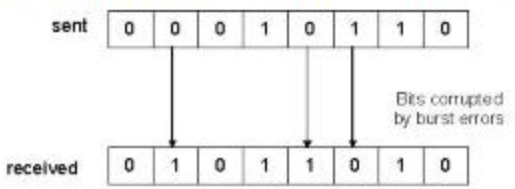
There are two types. They are, 1.

Single Bit Error

It means that only one bit of a given data unit is changed from 1 to 0 or from 0 to 1.

2. Burst Bit Error

It means that two or more bits in the data unit have changed.



- A burst bit does not necessarily means that the errors occur in consecutive bits
- The length of the bust error is measured from the first corrupted bit to the last corrupted bit. Some bits in between may not be corrupted.

5. NETWORK SOFTWARE

How to implement network software is an essential part of understanding computer networks. This section first introduces some of the issues involved in implementing an application program on top of a network, and then goes on to identify the issues involved in implementing the protocols running within the network. In many respects, network applications and network protocols are very similar—the way an application engages the services of the network is pretty much the same as the way a high-level protocol invokes the services of a low-level protocol.

5.1 Application Programming Interface (Sockets)

Most network protocols are implemented in software (especially those high in the protocol stack), and nearly all computer systems implement their network protocols as part of the operating system, when we refer to the interface "exported by the network," we are generally referring to the interface that the OS provides to its networking subsystem. This interface is often called the network *application programming interface* (API).

The advantage of industry-wide support for a single API is that applications can be easily ported from one OS to another, and that developers can easily write applications for multiple OSs. Just because two systems support the same network API does not mean that their file system, process, or graphic interfaces are the same. Still, understanding a widely adopted API like Unix sockets gives us a good place to start.

Each protocol provides a certain set of *services*, and the API provides a *syntax* by which those services can be invoked in this particular OS.

- int socket(int domain, int type, int protocol)
- int bind(int socket, struct sockaddr *address, int addr_len)
- int listen(int socket, int backlog)
- int accept(int socket, struct sockaddr *address, int *addr_len)
- int connect(int socket, struct sockaddr *address, intaddr_len)
- int send(int socket, char *message, int msg_len, int flags)
- int recv(int socket, char *buffer, int buf_len, int flags)

5.2 Example Application

The implementation of a simple client/server program that uses the socket interface to send messages over a TCP connection is discussed. The program also uses other Unix networking utilities, Our application allows a user on one machine to type in and send text to a user on another machine. It is a simplified version of the Unix talk program, which is similar to the program at the core of a web chat room.

Client program :

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
```

```
#include <netdb.h>
#define SERVER_PORT 5432
#define MAX_LINE 256
int
main(int argc, char * argv[])
{
FILE *fp;
struct hostent *hp;
struct sockaddr_in sin;
char *host;
char buf[MAX_LINE];
int s;
int len;
if (argc==2) {
host = argv[1];
}
else {
fprintf(stderr, "usage: simplex-talk host\n");
exit(1);
}
/* translate host name into peer's IP address */
hp = gethostbyname(host);
if (!hp) {
fprintf(stderr, "simplex-talk: unknown host: %s\n", host);
exit(1);
}
/* build address data structure */
bzero((char *)&sin, sizeof(sin));
sin.sin_family = AF_INET;
bcopy(hp->h_addr, (char *)&sin.sin_addr, hp->h_length);
sin.sin_port = htons(SERVER_PORT);
/* active open */
if ((s = socket(PF_INET, SOCK_STREAM, 0)) < 0) {
perror("simplex-talk: socket");
exit(1);
}
if (connect(s, (struct sockaddr *)&sin, sizeof(sin)) < 0) {
perror("simplex-talk: connect");
close(s);
exit(1);
}
/* main loop: get and send lines of text */
while (fgets(buf, sizeof(buf), stdin)) {
buf[MAX_LINE-1] = '\0';
len = strlen(buf) + 1;
send(s, buf, len, 0);
```

```
}
}
```

Server Program :

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#define SERVER_PORT 5432
#define MAX_PENDING 5
#define MAX_LINE 256
int
main()
{
struct sockaddr_in sin;
char buf[MAX_LINE];
int len;
int s, new_s;
/* build address data structure */
bzero((char *)&sin, sizeof(sin));
sin.sin_family = AF_INET;
sin.sin_addr.s_addr = INADDR_ANY;
sin.sin_port = htons(SERVER_PORT);
/* setup passive open */
if ((s = socket(PF_INET, SOCK_STREAM, 0)) < 0) {
perror("simplex-talk: socket");
exit(1);
}
if ((bind(s, (struct sockaddr *)&sin, sizeof(sin))) < 0) {
perror("simplex-talk: bind");
exit(1);
}
listen(s, MAX_PENDING);
/* wait for connection, then receive and print text */
while(1) {
if ((new_s = accept(s, (struct sockaddr *)&sin, &len)) < 0) {
perror("simplex-talk: accept");
exit(1);
}
while (len = recv(new_s, buf, sizeof(buf), 0))
fputs(buf, stdout);
close(new_s);
}
}
```

5.3 Protocol Implementation Issues

The rest of this section discusses the two primary differences between the network API and the protocol-to-protocol interface found lower in the protocol graph.

Process Model

Most operating systems provide an abstraction called a *process*, or alternatively, a *thread*. Each process runs largely independently of other processes, and the OS is responsible for making sure that resources, such as address space and CPU cycles, are allocated to all the current processes.

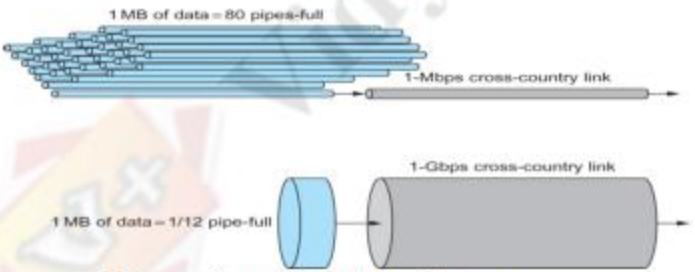
Message Buffers

A second inefficiency of the socket interface is that the application process provides the buffer that contains the outbound message when calling send, and similarly it provides the buffer into which an incoming message is copied when invoking the receive operation.This forces the topmost protocol to copy the message from the application's buffer into a network buffer, and vice versa.

6.PERFORMANCE :LINK LAYER SERVICES

6.1 Bandwidth and Latency

Network performance is measured in two fundamental ways: *bandwidth* (also called *throughput*) and *latency* (also called *delay*). The bandwidth of a network is given by the number of bits that can be transmitted over the network in a certain period of time.



- Latency = Propagation + transmit + queue
- Propagation = distance/speed of light
- Transmit = size/bandwidth
- One bit transmission => propagation is important
- Large bytes transmission => bandwidth is important

Relative importance of bandwidth and latency depends on application

- For large file transfer, bandwidth is critical

- For small messages (HTTP, NFS, etc.), latency is critical
- Variance in latency (jitter) can also affect some applications (*e.g.*, audio/video conferencing)

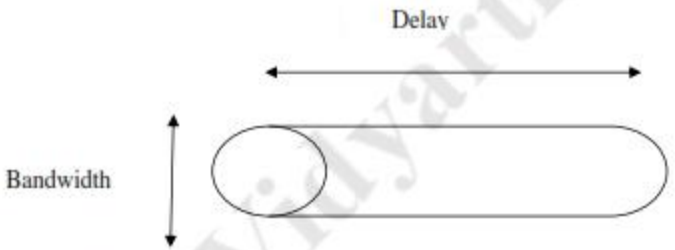
How many bits the sender must transmit before the first bit arrives at the receiver if the sender keeps the pipe full takes another one-way latency to receive a response from the receiver If the sender does not fill the pipe send a whole delay × bandwidth product's worth of data before it stops to wait for a signal the sender will not fully utilize the network

6.2 Delay × Bandwidth Product

The product of these two metrics, often called the *delay* × *bandwidth product*. A channel between a pair of processes as a hollow pipe , where the latency corresponds to the length of the pipe and the bandwidth gives the diameter of the pipe, then the delay × bandwidth product gives the volume of the pipe—the maximum number of bits that could be in transit through the pipe at any given instant.
For example, a transcontinental channel with a one-way latency of 50 ms and a bandwidth of 45 Mbps is able to hold

$$50 \times 10^{-3}\ \text{sec} \times 45 \times 10^{6}\ \text{bits/sec}$$
$$= 2.25 \times 10^{6}\ \text{bits}$$

or approximately 280 KB of data. In other words, this example channel (pipe) holds as many bytes as the memory of a personal computer from the early 1980s could hold.



The delay × bandwidth product is important to know when constructing high-performance networks because it corresponds to how many bits the sender must transmit before the first bit arrives at the receiver.

6.3 High-Speed Networks

The bandwidths available on today's networks are increasing at a dramatic rate,and there is eternal optimism that network bandwidth will continue to improve. This causes network designers to start thinking about what happens in the limit, or stated another way, what is the impact on network design of having infinite bandwidth available Although high-speed networks bring a dramatic change in the bandwidth available to applications, in many respects their impact on how we think about networking comes in what does *not* change as bandwidth increases: the speed of light.

6.4 Application Performance Needs

A network-centric view of performance; that is, we have talked in terms of what a given link or channel will support. The unstated assumption has been that application programs have simple needs—they want as much bandwidth as the network can provide. This is certainly true of the aforementioned digital library program that is retrieving a 25-MB image; the more bandwidth that is available, the faster the program will be able to return the image to the user.

If the application needs to support a frame rate of 30 frames per second, then it might request a throughput rate of 75 Mbps. The ability of the network to provide more bandwidth is of no interest to such an application because it has only so much data to transmit in a given period of time.

7.FRAMING

The sream of bits are not advisible to mqaintain in networks. When an error occurs, then the entire stream have to retransmitted. To avoid this, the framing concept is used. In this, the stream of bits are divided into manageable bit units called frames. To achive, we are using several ways. They are,

1. Byte Oriented Protocols
2. Bit Oriented Protocols
3. Clock Based Protocols

1. **BYTE ORIENTED PROTOCOLS:**

Each frame is considered as a collection of bytes rather than a collection of bits. There are two approaches. They are,

1. Sentinel approach

In this approach it uses special characters called sentinel characters to indicate where frames start and end. This approach is called character stuffing because extra characters are inserted in the data portion of the frame.

Ex: 1. Binary Synchronous Communication (BISYNC)
2. Point to Point Protocol

2. Byte Count Approach

In this approach no of bytes in frame are counted and entered in the header. The COUNT Field specifies how many bytes are contained in the frame"s body.

Ex: 1.Digital Data Communication Message Protocol

2. BIT ORIENTED PROTOCOLS:

It views the frames as a collection of bits. The Synchronous Data Link Control (SDLC) protocol developed by IBM is an example of a bit oriented protocol. It was later standardized by the ISO as the High Lever Data Link Control (HDLC)

HDLC – HIGH LEVEL DATA LINK CONTROL

It is a bit oriented data link protocol designed to support both half duplex and full duplex communication over point to point and multi point links.

FRAME FORMAT

8	16		16	8
Beginning Sequence	Header	body	CRC	Ending Sequence

HDLC denotes both the beginning and the end of a frame with the distinguished bit sequence 01111110. To guarantee that a special sequence does not appear in advertently anywhere else in the frame, HDLC uses a process called bit stuffing.

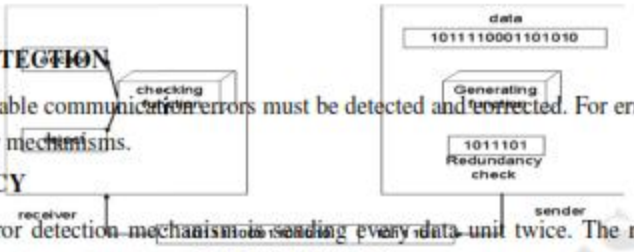
On the sending side, any time five consecutive 1s have been transmitted from the body of the message, the sender inserts a 0 before transmitting the next bit. On the receiver side, should five consecutive 1s arrive, the receiver makes its decision based on the next bit it sees. If the next bit is a 1, then one of the two things is true. Either this is the end of the frame or an error has been introduced. By looking at the next bit, it can conclude. If it sees a 0, then it is the end of frame. It else, then there must have an error and the whole frame has been discarded.

3. CLOCK BASED PROTOCOLS:

The Synchronous Optical NETwork (SONET) is one of the protocols using the clock based framing approach.

SONET:

It was developed by the ANSI for digital transmission over optical network. It addresses both the framing and encoding problems. A SONET frame has some special information to distinguish where the frame starts and ends.



8.ERROR DETECTION

For reliable communication errors must be detected and corrected. For error detection we are using many mechanisms.

REDUNDANCY

One error detection mechanism is sending every data unit twice. The receiving device then would be able to do a bit for bit comparison between the two versions of the data. Any discrepancy would indicate an error, and an appropriate correction mechanism could be used.

But instead of repeating the entire data stream, a shorter group of bits may be appended to the end of each unit. This technique is called redundancy because extra bits are redundant to the information. They are discarded as soon as the accuracy of the transmission has been determined.

TYPES

Four types of redundancy checks are used in data communications. They are,

1. vertical redundancy check (VRC)
2. longitudinal redundancy check (LRC)
3. cyclic redundancy check (CRC)
4. checksum

VERTICAL REDUNDANCY CHECK:

It is also known as parity check. In this technique a redundant bit called a parity bit is appended to every data unit so that the total number of 1s in the unit including the parity bit

becomes even for even parity or odd for odd parity.

In even parity, the data unit is passed through the even parity generator. It counts the number of 1s in the data unit. If odd number of 1s, then it sets 1 in the parity bit to make the number of 1s as even. If the data unit having even number of 1s then it sets in the parity bit to maintain the number of 1s as even. When it reaches its destination, the receiver puts all bits through an even parity checking function. If it counts even number of 1s than there is no error. Otherwise there is some error.

EXAMPLE:

The data is : 01010110

The VRC check : 010101100

In odd parity, the data unit is passed through the odd parity generator. It counts the number of 1s in the data unit. If even number of 1s, then it sets 1 in the parity bit to make the number of 1s as odd. If the data unit having odd number of 1s then it sets in the parity bit to maintain the number of 1s as odd. When it reaches its destination, the receiver puts all bits through an odd parity checking function. If it counts odd number of 1s than there is no error. Otherwise there is some error.

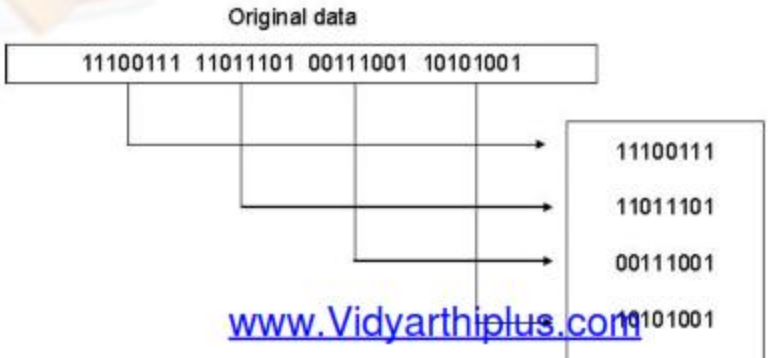
EXAMPLE

The data is: 01010110

The VRC check: 01010111

LONGITUDINAL REDUNDANCY CHECK

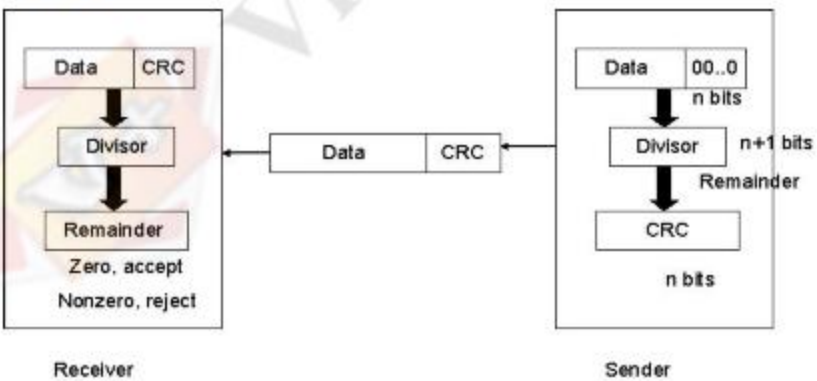
In this, a block of bits is organized in a table (rows and columns). For example, instead of sending a block of 32 bits, we organize them in a table made of four roes and eight columns. We then calculate the parity bit for each column and create a new row of eight bits which are the parity bits for the whole block



CYCLIC REDUNDANCY CHECK

CRC is based on binary division. In this a sequence of redundant bits, called CRC remainder is appended to the end of a data unit so that the resulting data unit becomes exactly divisible by a second predetermined binary number. At its destination, the incoming data unit is divided by the same number. If at this step there is no reminder, the data unit is assumed to be intact and therefore accepted. A remainder indicates that the data unit has been changed in transit and therefore must be rejected.

Here, the remainder is the CRC. It must have exactly one less bit than the divisor, and appending it to the end of the data string must make the resulting bit sequence exactly divisible by the divisor.



First, a string of n-1 0s is appended to the data unit. The number of 0s is one less than the

number of bits in the divisor which is n bits. Then the newly elongated data unit is divided by the divisor using a process called binary division. The remainder is CRC. The CRC is replaces the appended 0s at the end of the data unit.

The data unit arrives at the receiver first, followed by the CRC. The receiver treats whole string as the data unit and divides it by the same divisor that was used to find the CRC remainder. If the remainder is 0 then the data unit is error free. Otherwise it having some error and it must be discarded.

CHECKSUM

The error detection method used by the higher layer protocols is called checksum.

It consists of two arts. They are,

1. checksum generator
2. checksum checker

Checksum Generator:

In the sender, the checksum generator subdivides the data unit into equal segments of n bits. These segments are added with each other by using one"s complement arithmetic in such a way that the total is also n bits long. That total is then complemented and appended to the end of the data unit.

Checksum Checker:

The receiver subdivides the data unit as above and adds all segments together and complements the result. If the extended data unit is intact, the total value found by adding the data segments and the checksum field should be zero. Otherwise the packet contains an error and the receiver rejects it.

EXAMPLE

At the sender

Data unit: 10101001 00111001

10101001 00111001

Sum 1100010

Checksum 00011101

At the receiver

1)

Received data: 10101001 00111001 00011101

10101001 00111001 00011101

Sum	11111111
Complement	00000000

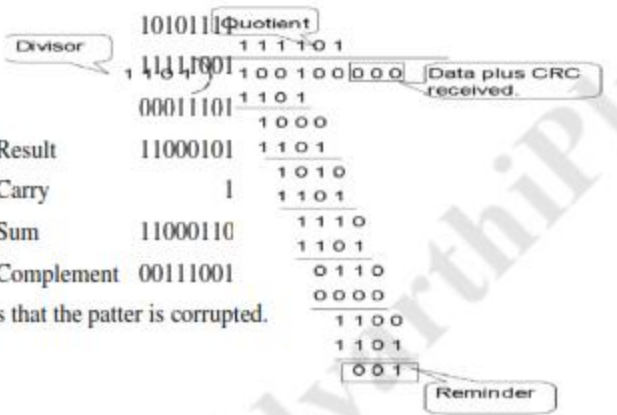
It means that the patter is ok.

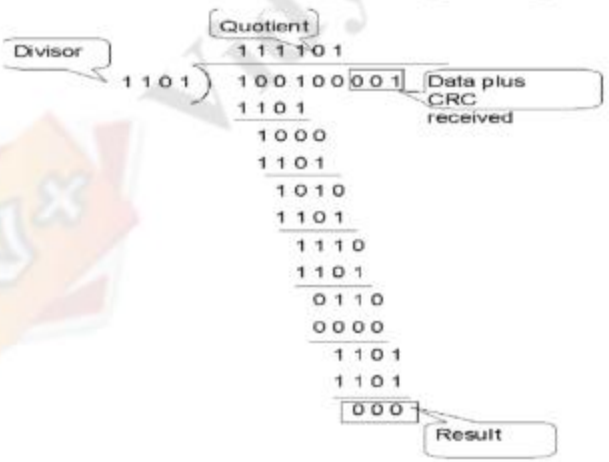
2)

Received data: 1010111 111001 00011101

Result	11000101
Carry	1
Sum	11000110
Complement	00111001

It means that the patter is corrupted.





ERROR CORRECTION

Error correction is handled in two ways. In one, when an error is discovered, the receiver can have the sender retransmit the entire data unit. In the other, a receiver can use an error correcting code, which automatically corrects certain errors.

Types of error correction:

1. Single bit error correction
2. Burst bit error correction

Single Bit Error Correction

To correct a single bit error in an ASCII character, the error correction code must determine which of the seven bits has changed. In this case we have to determine eight different states: no error, error in position 1, error in position 2, error in position 3, error in position 4, error in position 5, error in position 6, error in position 7. It looks like a three bit redundancy code should be adequate because three bits can show eight different states. But what if an error occurs in the redundancy bits? Seven bits of data and three bits of redundancy bits equal 10 bits. So three bits are not adequate.

To calculate the number of redundancy bits (r) required to correct a given number of data bits (m) we must find a relationship between m and r.

If the total number of bits in a transmittable unit is m+r then r must be able to indicate at least m+r+1 different state. Of these, one state means no error and m+r states indicate the location of an error in each of the m+r positions.

So m+r+1 state must be discoverable by r bits. And r bits can indicate 2^r different states. Therefore, 2^r must be equal to or greater than m+r+1;

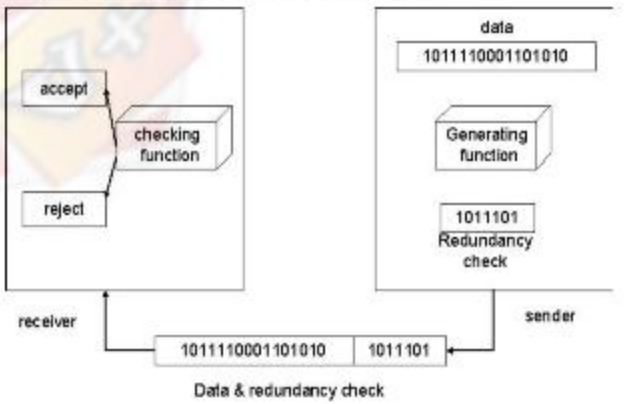
$$2^r >= m+r+1$$

NUMBER OF DATA BITS (M)	NUMBER OF REDUNDANCY BITS (R)	TOTAL BITS (M+R)
1	2	3
2	3	5
3	3	6
4	3	7
5	4	9
6	4	10
7	4	11

Hamming Code:

The hamming code can be applied to data units of any length and uses the relationship between data and redundancy bits.

Positions of redundancy bits in hamming code



The combinations used to calculate each of the four r values for a seven bit data sequence are as follows:

r1 :1,3,5,7,9,11

r2 : 2,3,6,7,10,11

r3 : 4,5,6,7

r4 : 8,9,10,11

Here, r1 bit is calculated using all bit positions whose binary representation includes a 1 in the rightmost position (0001, 0011, 0101, 0111, 1001, and 1011). The r2 bit is calculated using all bit positions with a 1 in the second position (0010, 0011, 0110, 0111, 1010 and 1011), and for r3 1 at third bit position (0100, 0101, 0110 and 0111) for r4 1 at fourth bit position (1000, 1001, 1010 and 1011).

Calculating the r Values:

In the first step, we place each bit of the original character in its appropriate positions in the 11 bit unit. Then, we calculate the even parities for the various bit combinations. The parity value of each combination is the value of the corresponding r bit. For example r1 is calculated to provide even parity for a combination of bits 3, 5, 7, 9, 11.

Error Detection and Correction:

Example:

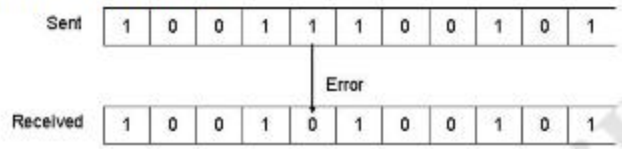
At the sender:

Data to be sent: 1001101

Redundancy bit calculation:

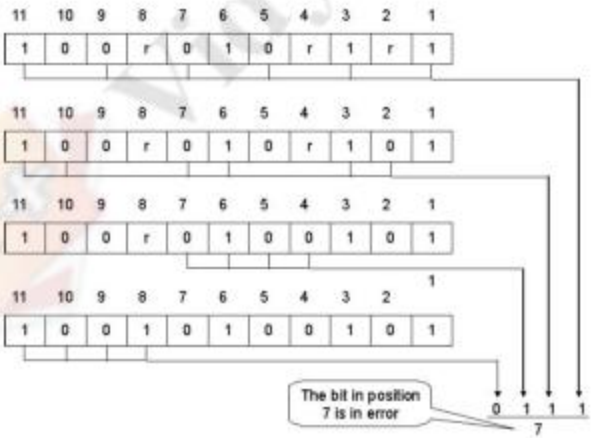
Data sent with redundancy bits: 10011100101

During transmission:



At the receiver:

The receiver takes the transmission and recalculates four new r values using the same set of bits used by the sender plus the relevant parity (r) bit for each set. Then it assembles the new parity values into a binary number in order of r position (r8, r4, r2, r1).



Once the bit is identified, the receiver can reverse its value and correct the error.

Burst Bit Error Correction:

A hamming code can be designed to correct burst errors of certain length. The number of redundancy bits required to make these corrections, however, is dramatically higher than that required for single bit errors. To correct double bit errors, for example, we must take into consideration that the two bits can be a combination of any two bits in the entire sequence. Three bit correction means any three bits in the entire sequence and so on.

9.FLOW CONTROL

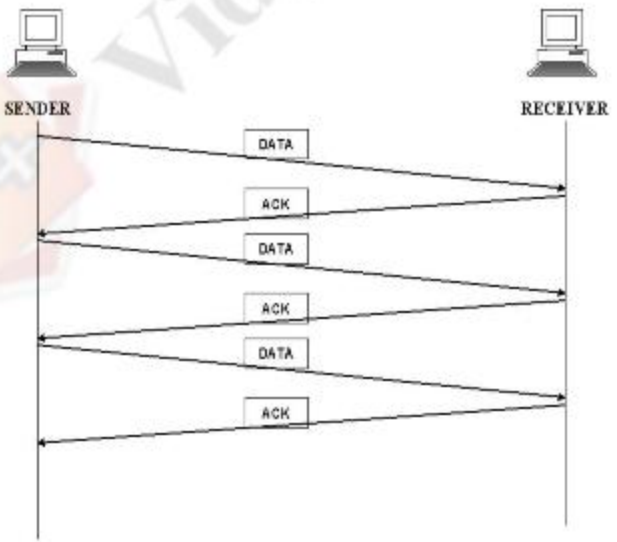
It refers to a set of procedures used to restrict the amount of data flow between sending and receiving stations. It tells the sender how much data it can transmit before it must wait for an acknowledgement from the receiver.

There are two methods are used. They are,

1. stop and wait
2. sliding window

STOP AND WAIT:

In this method the sender waits for acknowledgment after every frame it sends. Only after an acknowledgment has been received, then the sender sends the next frame.

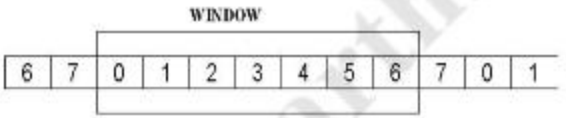


The advantage is simplicity. The disadvantage is inefficiency.

SLIDING WINDOW:

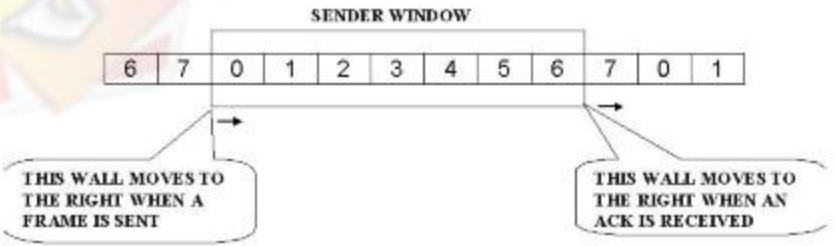
In this method, the sender can transmit several frames before needing an acknowledgment. The receiver acknowledges only some of the frames, using a single ACK to confirm the receipt of multiple data frames.

The sliding window refers to imaginary boxes at both the sender and receiver. This window provides the upper limit on the number of frames that can be transmitted before requiring an acknowledgement. To identify each frame the sliding window scheme introduces the sequence number. The frames are numbered as 0 to n-1. And the size of the window is n-1. Here the size of the window is 7 and the frames are numbered as 0,1,2,3,4,5,6,7.

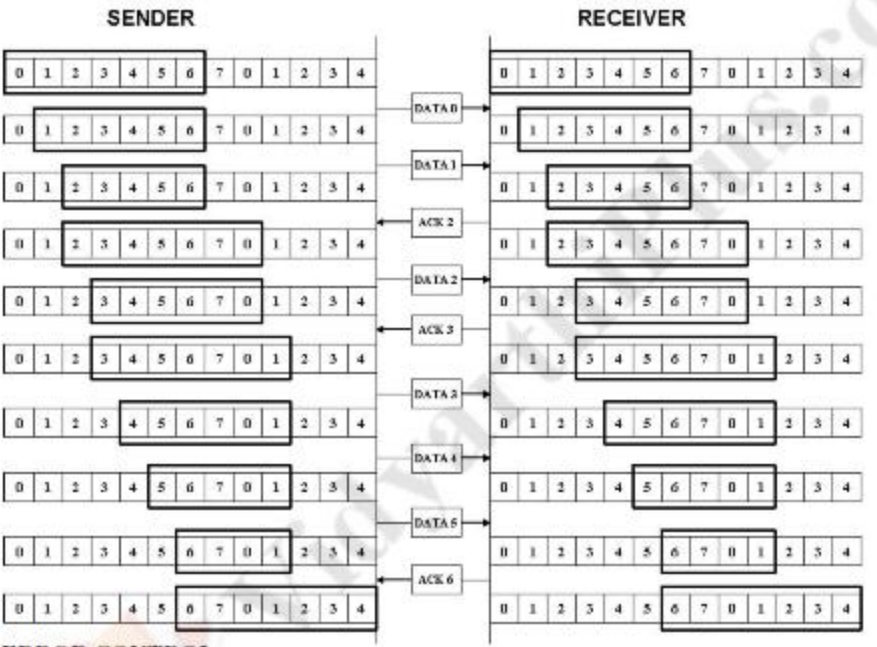


SENDER WINDOW:

At the beginning the sender"s window contains n-1 frames. As frames are sent out the left boundary of the window moves inward, shrinking the size of the window. Once an ACK receives the window expands at the right side boundary to allow in a number of new frames equal to number of frames acknowledged by that ACK.



EXAMPLE:



ERROR CONTROL

Error control is implemented in such a way that every time an error is detected, a negative acknowledgement is returned and the specified frame is retransmitted. This process is called **automatic repeat request (ARQ).**

The error control is implemented with the flow control mechanism. So there are two types in error control. They are,

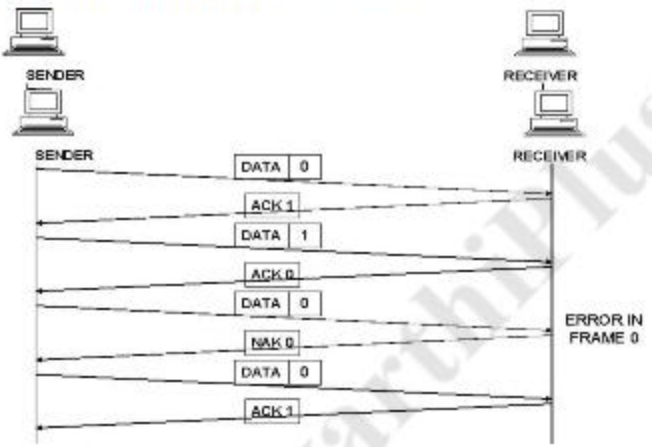
1. stop and wait ARQ
2. sliding window ARQ

STOP AND WAIT ARQ:

It is a form of stop and wait flow control, extended to include retransmission of data in case of lost or damaged frames.

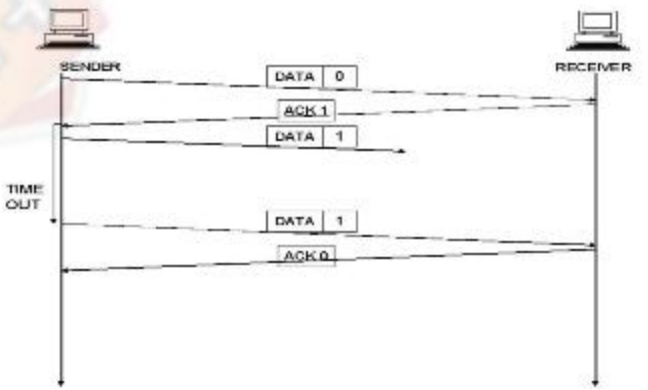
DAMAGED FRAME:

When a frame is discovered by the receiver to contain an error, it returns a NAK frame and the sender retransmits the last frame.



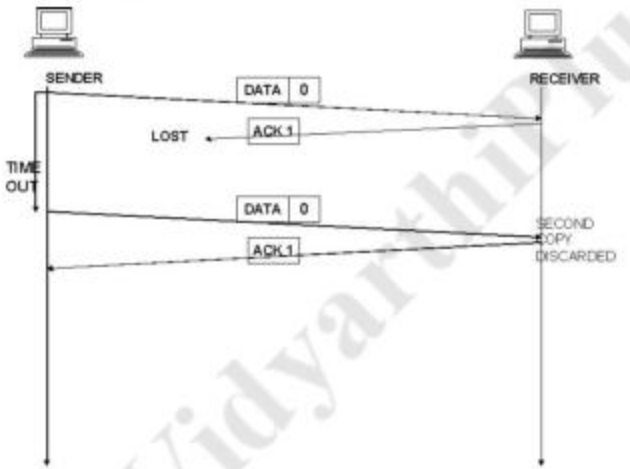
LOST DATA FRAME:

The sender is equipped with a timer that starts every time a data frame is transmitted. If the frame lost in transmission the receiver can never acknowledge it. The sending device waits for an ACK or NAK frame until its timer goes off, then it tries again. It retransmits the last data frame.



LOST ACKNOWLEDGEMENT:

The data frame was received by the receiver but the acknowledgement was lost in transmission. The sender waits until the timer goes off, then it retransmits the data frame. The receiver gets a duplicated copy of the data frame. So it knows the acknowledgement was lost so it discards the second copy.



SLIDING WINDOW ARQ

It is used to send multiple frames per time. The number of frame is according to the window size. The sliding window is an imaginary box which is reside on both sender and receiver side.

It has two types. They are,

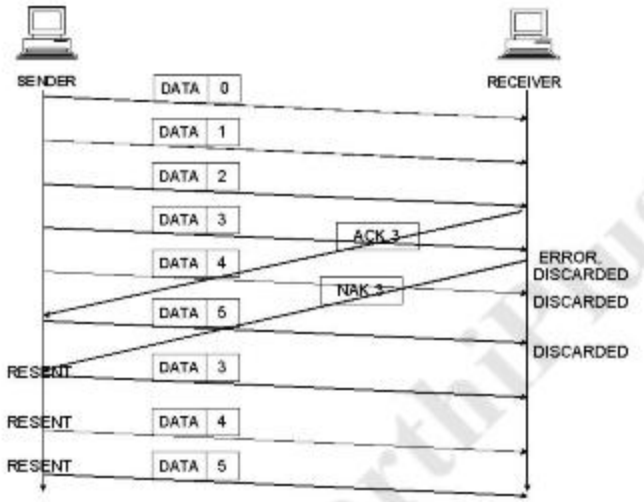
1. go-back-n ARQ
2. selective reject ARQ

GO-BACK-N ARQ:

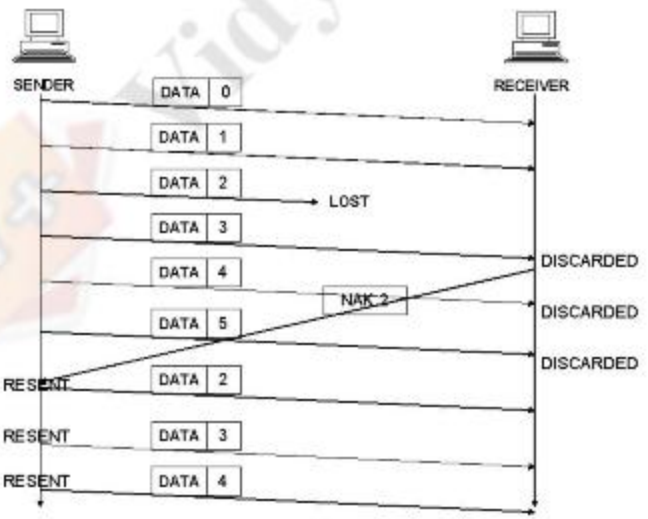
In this method, if one frame is lost or damaged, all frames sent since the last frame

acknowledged or retransmitted.

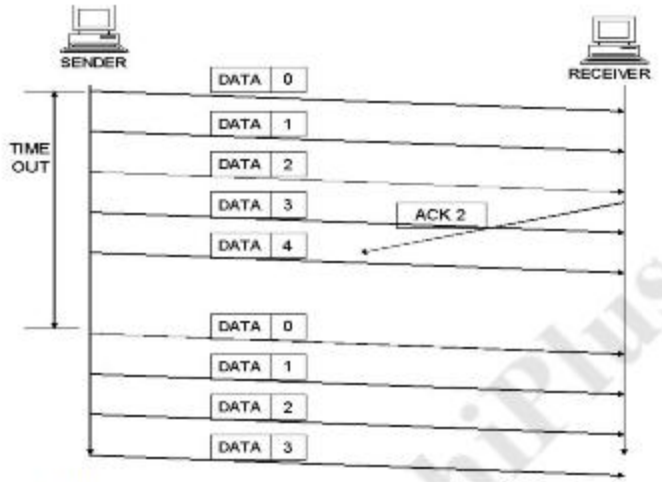
DAMAGED FRAME:



LOST FRAME:



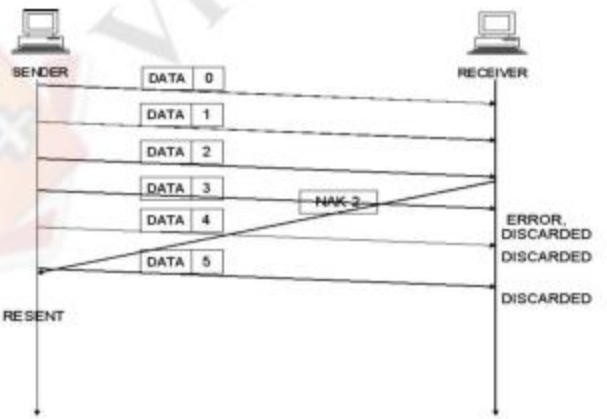
LOST ACK:



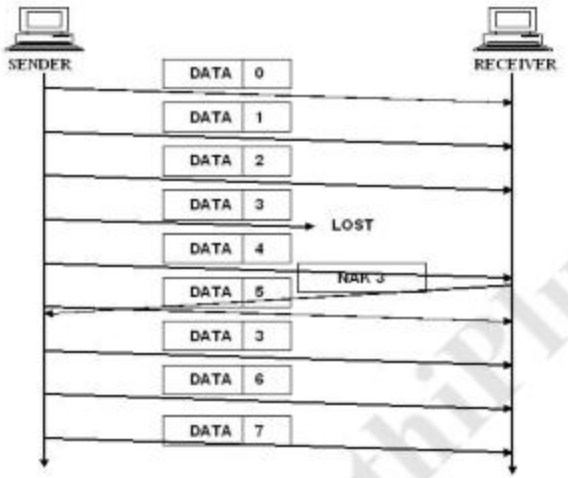
SELECTIVE REPEAT ARQ

Selective repeat ARQ re transmits only the damaged or lost frames instead of sending multiple frames. The selective transmission increases the efficiency of transmission and is more suitable for noisy link. The receiver should have sorting mechanism.

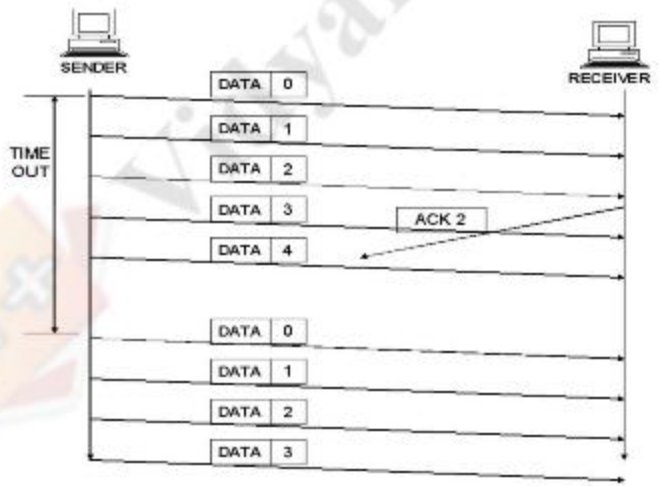
DAMAGED FRAME:



LOST FRAME



LOST ACK



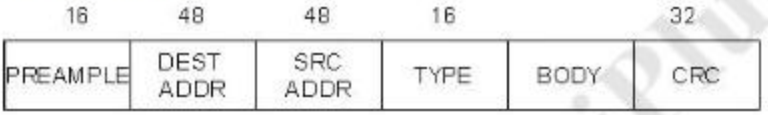
UNIT II MEDIA ACCESS & INTERNETWORKING

Media access control - Ethernet (802.3) - Wireless LANs – 802.11 – Bluetooth - Switching and Bridging – Basic Internetworking (IP, CIDR, ARP, DHCP, ICMP)

10.MEDIUM ACCESS CONTROL

This algorithm is commonly called the Ethernet"s media access control (MAC). It is typically implemented in hardware on the network adaptor.

FRAME FORMAT:



Preamble allows the receiver to synchronize with the signal. Both the source and destination hosts are identified with a 48-bit address. Each frame contains up to 1,500 bytes of data. A frame must contain at least 46 bytes of data, even if this means the host has to pad the frame before transmitting it. Each frame includes a 32-bit CRC.

ADDRESSES:

It is usually burned into ROM. Ethernet addresses are typically printed in a form humans can read as a sequence of six numbers separated by colons.

Each number corresponds to 1 byte of the 6-byte address and is given by a pair of hexadecimal digits, one for each of the 4-bit nibbles in the byte; leading 0s are dropped.

To ensure that every adaptor gets a unique address, each manufacturer of Ethernet devices is allocated a different prefix that must be prep-ended to the address on every adaptor they build.

UNICAST
MULTICAST
BROADCAST

TRANSMITTER ALGORITHM:

The receiver side of the Ethernet protocol is simple; the real smarts are implemented at the sender"s side. The transmitter algorithm is defined as follows:

When the adaptor has a frame to send and the line is busy, it waits for the line to go idle and then transmits immediately.

The Ethernet is said to be a 1-persistent protocol because an adaptor with a frame to send transmits with probability 0<=p<=1 after a line becomes idle, and defers with probability q=1-p. Because there is no centralized control it is possible for two (or more) adaptors to begin transmitting at the same time, either because both found the line to be idle or because both had been waiting for a busy line to become idle.

When this happens, the two (or more) frames are said to collide on the network. Each sender, because the Ethernet supports collision detection, is able to determine that a collision is in progress. At the moment an adaptor detects that is frame is colliding with another, it first makes sure to transmit a sure to transmit a 32-bit jamming sequence and then stops the transmission.

Thus, a transmitter will minimally send 96 bits in the case of a collision: 64-bit preamble plus 32-bit jamming sequence. One way that an adaptor will send only 96-bits which is sometimes called a runt frame is if the two hosts are close to each other. Had the two hosts been farther apart, they would have had to transmit longer, and thus send more bits, before detecting the collision.

In fact, the worst-case scenario happens when the two hosts are at opposite ends of the Ethernet. To know for sure that the frame it just sent did not collide with another frame, the transmitter may need to send as many as 512 bits.

Not coincidentally, every Ethernet frame must be at least 512 bits (64 bytes)long: 14 bytes of header plus 46 bytes of data plus 4 bytes of CRC.

Where hosts A and B are at opposite ends of the network. Suppose host A begins transmitting a frame at time t, as shown in (a). it takes it one link latency (let"s denote the latency as d) for the frame to reach host B.

Thus, the first bit of A"s frame arrives at B at time t+d, as shown in (b). Suppose an instant before host A"s frame arrives (i.e., B still sees and idle line), host B begins to transmit its own frame.

B"s frame will immediately collide with A"s frame, and this collision will be detected by

host B(c). host B will send the 32-bit jamming sequence, as described above.(B"s frame will be a runt).

Unfortunately, host A will not know that the collision occurred until B"s frame reaches it, which will happen one link latency later, at time t+2xd, as shown in (d). Host A must continue to transmit until this time in order to detect the collision. In other words, host A must transmit for 2xd should be sure that it detects all possible collisions.

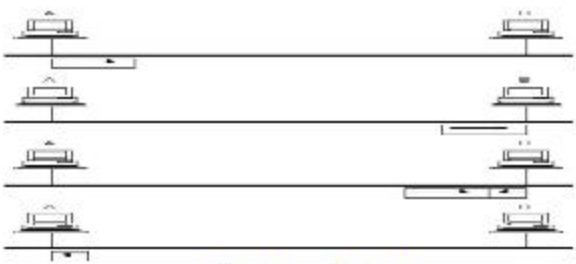
Considering that a maximally configured Ethernet is 2,500 m long, and that there may be up to four repeaters between any two hosts, the round-trip delay has been determined to be 51.2 microseconds, which on a 10-Mbps Ethernet corresponds to 512 bits.

The other way to look at this situation is that we need to limit the Ethernet"s maximum latency to a fairly small value (e.g., 512micro seconds) for the access algorithm to work; hence, an Ethernet"s maximum length must be something on the order of 2,500m.

Once an adaptor has detected a collision and stopped its transmission, it waits certain amount of time and tries again. Each time it tries to transmit but fails, the adaptor doubles the amount of time it waits before trying again.

This strategy of doubling the delay interval between each retransmission attempt is a general technique known as exponential back off. More precisely, the adaptor first delays either 0 or 51.2 microseconds, selected at random. If this effort fails, it then waits 0, 51.2, 102.4, or 153.6 microseconds (selected randomly) before trying again; this is kx51.2 for k=0...2^3-1, again selected at random.

In general, the algorithm randomly selects a k between 0 and 2^n-1 and waits kx51.2 microseconds, where n is the number of collisions experienced so far. The adaptor gives up after a given number of tries and reports a transmit error to the host. Adaptor typically retry up to 16 times, although the back off algorithm caps n in the above formula at 10.



11. ETHERNET(802.3)

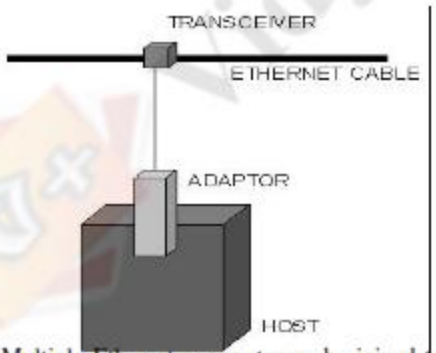
The Ethernet is developed in the mid-1970 by researches at the Xerox Palo Alto Research Center (PARC); the Ethernet is a working example of the more general carrier sense, multiple accesses with collision detect (CSMA/CD) local area network technology.

The "carrier sense" in CSMA/CD means that all the nodes can distinguish between an idle and a busy link, and "collision detect" means that all the nodes listens as it transmits and can therefore detect when a frame it is transmitting has interfered (collided) with a frame transmitted by another node.

PHYSICAL PROPERTIES:

An Ethernet segment is implemented on a coaxial cable of up to 500m. this cable is similar to the type used for cable TV, except that it typically has an impedance of 50 ohms instead of cable TV"s 75 ohms. Hosts connect to an Ethernet segment by tapping into it; taps must be at least 2.5 m apart.

A transceiver a small device directly attached to the tap detects when the line is idle and drives the signal when the host is transmitting. It also receives incoming signals. The transceiver is, in turn, connected to an Ethernet adaptor, which is plugged into the host.



Multiple Ethernet segments can be joined together by repeater. A repeater is a device that forwards digital signals, much like an amplifier forwards analog signals. However, no more than four repeaters may be positioned between any pair of hosts, meaning that an Ethernet has a total reach of only 2,500m.

An Ethernet is limited to supporting a maximum of 1,024 hosts. Terminators attached to the end of each segment absorb the signal and keep it from bouncing back and interfering with trailing signals.

STANDARDS:

There are various standards of Ethernet are,

10Base5:

The first of the physical standards defined in the IEEE 802.3 model is called 10Base5.It is also known as thick net or thick Ethernet. A segment of the original 10Base5 cable can be up to 500m long.

10Base2:

The second implementation defined by the IEEE892 series is called 10Base2.It also known as thin-net, cheapnet, cheapernet, thinwire Ethernet or thin Ethernet. In this "10" means the network operates at 10 Mbps, "Base" refers to the fact that the cable is used in a base band system and the "2" means that a given segment can be no longer than 200m

10BaseT:

The most popular standard defined in the IEEE 802.3 series is 10BaseT. It is also known as twisted pair Ethernet. The "T" stands for twisted pair. A 10BaseT segment is usually limited to less than 100m in length.

12.WIRELESS LAN'S

Wireless technologies differ in variety of dimensions, most notably in how much bandwidth they provide and how far apart communicating nodes can be. Other important differences include which part of the electromagnetic spectrum they use (including whether it requires a license) and how much power they consume. Four prominent wireless technologies:

- Blue tooth
- Wi-Fi(more formally known as 802.11)
- WiMAX(802.16)
- Third generation or 3Gcellular wireless.

The most widely used wireless links today are usually asymmetric, that is, the two endpoints are usually different kinds of nodes.

BASE STATION, usually has no mobility, but has a wired (or at least high bandwidth)

connection to the internet or other networks.

A "client node" is often mobile, and relies on its link to the base station for all its communication with other nodes. Wireless communication naturally supports point to multipoint communication, because radio waves sent by one device can be simultaneously received by many devices. However, it is often useful to create a point to point link abstraction for higher layer protocols.

This topology implies three qualitatively different levels of mobility. The first level is no mobility, such as when a receiver must be in a fixed location to receive a directional transmission from the base station, as is the case with the initial version of WiMAX. The second level is mobility within the range of a base, as is the case with Bluetooth. The third level is mobility between bases, as is the case with cell phones and Wi-Fi.

13.WI-FI (802.11)

This section takes a closer look at a specific technology centered on the emerging IEEE 802.11 standard, also known as Wi-Fi. Wi-Fi is technically a trademark, owned by a trade group called the Wi-Fi alliance that certifies product compliance with 802.11. 802.11 is designed for use in a limited geographical area (homes, office buildings, campuses) and its primarily challenge is to mediate access to a shared communication medium in this case, signals propagating through space.

PHYSICAL PROPERTIES:

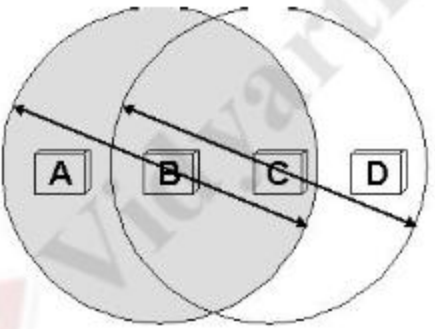
802.11 run over six different physical layer protocols. Five are based on spread spectrum radio, and one on diffused infrared (and is of historical interest only at this point). The fastest runs at a maximum of 54 Mbps.

The original 802.11 standard defined two radio based physical layers standards, one using frequency hopping and the other using direct sequence. Both provide up to 2 Mbps. Then physical layer standard 802.11 b was added. Using a variant of direct exempt 2.4GHz frequency band of the electromagnetic spectrum. Then came 802.11a, which delivers up to 54 Mbps using a variant of FDM called orthogonal frequency division multiplexing (OFDM). 802.11 a runs in the license-exempt 5GHz band. The most recent standard is 802.11g, which is backward compatible with 802.11b.

COLLISION AVOIDANCE:

A wireless protocol wait until the link becomes idle before transmitting and back off should a collision occur. Consider the situation where A and C are both within range of B but not each other. Suppose both A and C want to communicate with B and so they each send it a frame. A and C are unaware of each other since their signals do not carry that far. These two frames collide with each other at B, but unlike an Ethernet, neither A or C is aware of this collision. A and C are said to be hidden nodes with respect to each other.

A related problem called the exposed node problem where each of the four nodes is able to send and receive signals that reach just the nodes to its immediate left and right. For EX: a B can exchange frames with A and C but it cannot reach D , while C can reach B and D but not A. Suppose B is sending to A. Node C is aware of this communication because it hear B"s transmission. It would be a mistake, however, for C to conclude that it cannot transmit to anyone just because it can hear B"s transmission. For example, suppose C wants to transmit to node D. This is not a problem since C"s transmission to D will not interfere with A"s ability to receive from B.



802.11 addresses these two problems with an algorithm called multiple access with collision avoidance (MACA). The idea is for the sender and receiver ot exchange control frames with each other before the sender actually transmits any data. This exchange informs all nearby nodes that a transmission is about to begin. Specifically, the

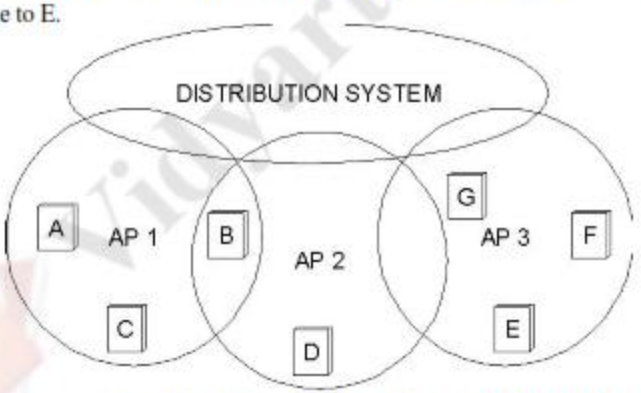
sender transmits a ***Request to send*** (RTS) frame to the receiver; the RTS frame includes a field that indicates how long the sender wants to hold the medium. The receiver then replies with a

***clear to send** (CTS)* frame. This frame echoes this length field back to the sender. Any node that sees the RTS frame will collide with each other.

802.11 does not support collision detection, but instead the senders realize the in which case they each wait a random amount of time before trying again. The amount of time a given node delay is defined by the same exponential backoff algorithm used on the Ethernet.

DISTRIBUTION SYSTEM

Instead of all nodes created equal, some nodes are allowed to roam and some are connected to a wired network infrastructure. 802.11 calls these base stations ***access points*** *(AP)*, and they are connected to each other by a so-called ***distribution system***. A distribution system that connects three access points, each of which services the nodes in some region. Although two nodes can communicate directly with each other if they are within reach of each other, the idea behind this configuration is that each node associates itself with one access point. For node A to communicate with node E, for example, A first sends a frame to its access point (AP-1), which forwards the frame across the distribution system to AP-3 , which finally transmits the frame to E.



The technique for selecting an AP is called scanning and involves the following four steps:

1. The node sends a probe frame;
2. All APs within reach reply with a probe Response frames;
3. The node selects one of the access points, and sends that AP an Association Request frames;
4. The AP replies with an Association Response frame.

Because the signal from its current AP has weakened due to the node moving away from it. Whenever a node acquires a new AP, the new AP notifies the old AP of the change via the distribution system.

Here in this fig., where node C moves from the cell serviced by AP-1 to the cell serviced by AP-2.At some point, C prefers AP-2 over AP-1,and so it associates itself with that access point.

The mechanism just described is called active scanning since the node is actively searching for an access point. APs also periodically send a BEACON frame that the capabilities of the access point; these include the transmission rates supported by the AP.

This is called passive scanning, and a node can change to this AP based on the BEACON frame simply by sending an ASSOCIATION REQUEST frame back to the access point.

FRAME FORMAT:

16	16	48	48	48	16	48	0 - 18496	32
CONTROL	DURATION	ADDR1	ADDR 2	ADDR 3	SEQCTRL	ADDR 4	PAY LOAD	CRC

The frame contains the source and destination node address, each of which is 48 bits long, up to 2,312 bytes of data, and a 32-bit CRC. The Control field contains three subfields of interest : a 6-bit Type field that indicates whether the frame carries data, is an RTS or CTS frame, or is being used by the scanning algorithm; and a pair of 1-bit fields-called ToDS and .

The 802.11 frame format is that it contains four, rather than two, address. how these address are interpreted depends on the settings of the ToDS and FromDS bits in the frame"s Control field. This is to account for the possibility that the frame had to be forwarded across the distribution systems, which would mean that the original sender is not necessarily the same as the most recent transmitting node.

Similar reasoning applies to the destination address. In the simplest case, when one node is sending directly to another, the DS bits are 0, Addr1 identifies the target node, and Addr2 identifies the source node.

In the most complex case, both DS bits are set to 1, indicating that the message went

from a wireless node onto the distribution system and then from the distribution system to another wireless node. With both bits set, Addr1 identifies the ultimate destination, Addr2 identifies the immediate sender (the one that forwarded the frame from the distribution system to the ultimate destination), Addr3 identifies the intermediate destination (the one that accepted the frame from a wireless node and forwarded it across the distribution system), and Addr4 identifies the original source. In terms of the example given in fig., Addr1 corresponds to E, Addr2 identifies AP-3, Addr3 corresponds to AP-1, and Addr4 identifies A.

14. BLUETOOTH (802.15.1)

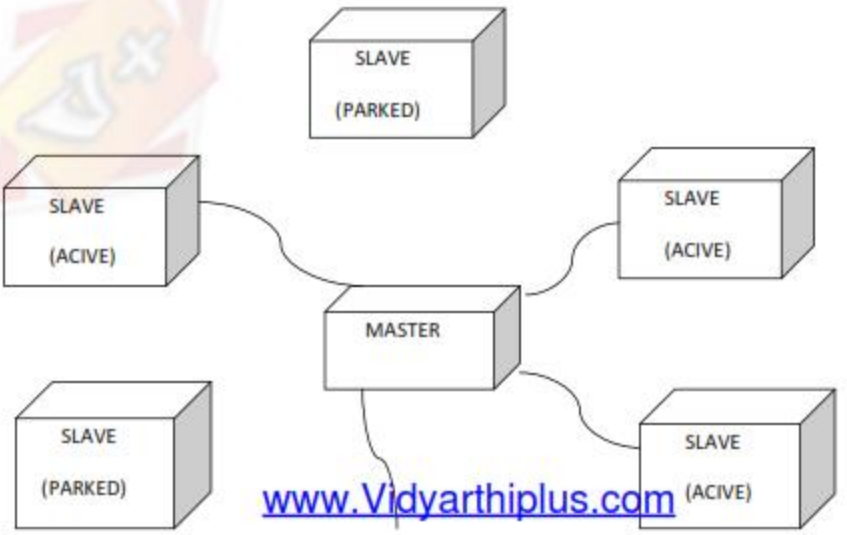
Bluetooth fills the niche of very short-range communication between mobile phones,PDAs, Notebook computers, and other personal or peripheral devices. For example, Bluetooth can be used to connect a mobile phone to a headset, or a notebook computer to a printer. Bluetooth is a more convenient alternative to connecting two devices with a wire. In such applications, it is not necessary to provide much range or bandwidth. This is fortunate for some of the target battery-powered devices, since it is important that they not consume much power.

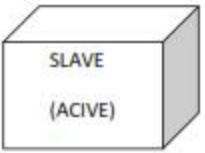
Bluetooth operates in the license-exempt band at 2.45 GHz. It has a range of only about 10 m. For this reason, and because the communicating devices typically belong to one individual or group, Bluetooth is sometimes categorized as a personal area network (PAN). Version 2.0 provides speeds up to 2.1 Mbps. Power consumption is low.

Bluetooth is specified by an industry consortium called the Bluetooth Special Interest Group. It specifies an entire suite of protocols, going beyond the link layer to define application protocols, which it calls *profiles*, for a range of applications. For example, there is a profile for synchronizing a PDA with a personal computer. Another profile gives a mobile computer access to a wired LAN in the manner of 802.11, although this was not Bluetooth's original goal. The IEEE 802.15.1 standard is based on Bluetooth but excludes the application protocols. The basic Bluetooth network configuration, called a *piconet*, consists of a master device and up to seven slave devices. Any communication is between the master and a slave; the slaves do not communicate directly with each other. Because slaves have a simpler role, their Bluetooth hardware and software can be simpler and cheaper.

Since Bluetooth operates in an license-exempt band, it is required to use spread spectrum Technique to deal with possible interference in the band. It uses frequency hopping with 79 *channels* (frequencies), using each for 625 μm at a time. This provides a natural time slot for Bluetooth to use for synchronous time division multiplexing. A frame takes up 1, 3, or 5 consecutive time slots.

A slave device can be *parked*: set to an inactive, low-power state. A parked device cannot communicate on the piconet; it can only be reactivated by the master. A piconet can have up to 255 parked devices in addition to its active slave devices. ZigBee is a newer technology that competes with Bluetooth to some extent. Devised by the ZigBee alliance and standardized as IEEE 802.15.4, it is designed for situations where the bandwidth requirements are low and power consumption must be very low to give very long battery life. It is also intended to be simpler and cheaper than Bluetooth, making it financially feasible to incorporate in cheaper devices such as a wall switch that wirelessly communicates with a ceiling-mounted fan.

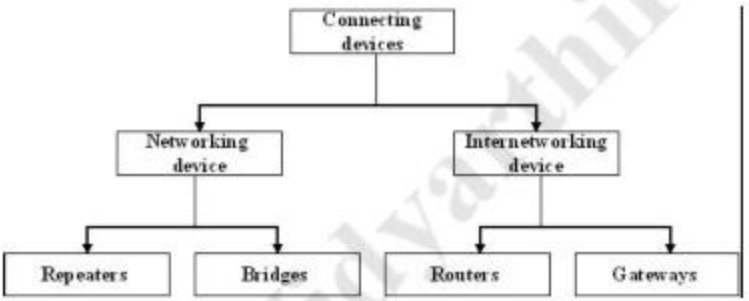


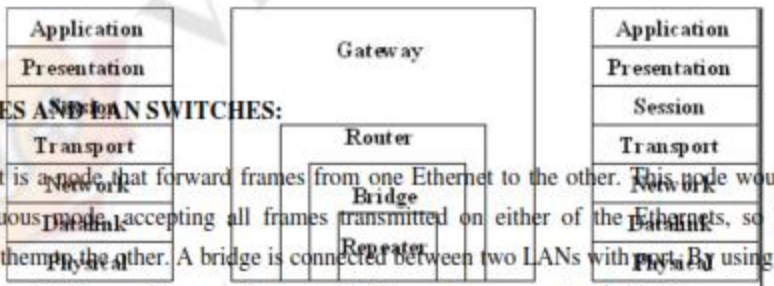


A BLUETOOTH PICONET

15.SWITCHING AND BRIDGING

Networking and internetworking devices are classified into four categories: repeaters, bridges, routers, and gateways.





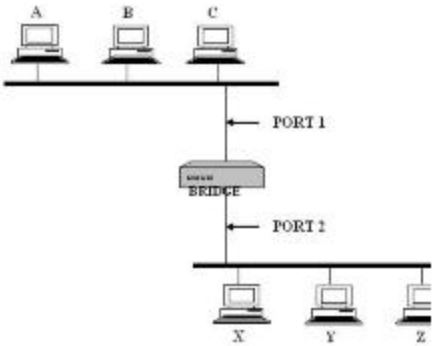
BRIDGES AND LAN SWITCHES:

It is a node that forward frames from one Ethernet to the other. This node would be in promiscuous mode, accepting all frames transmitted on either of the Ethernets, so it could forward them to the other. A bridge is connected between two LANs with port. By using the port number the LANs are addressed. Connected LANs are known as extended LAN

LEARNING BRIDGES:

Bridges maintains a forwarding table which contains each host with their port number.

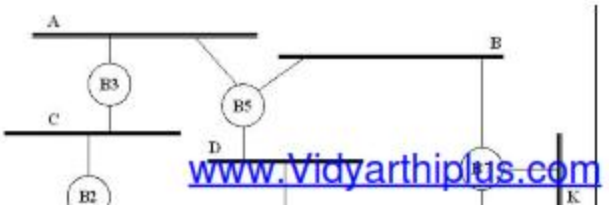
Having a human maintain this table is quite a burden, so a bridge can learn this information for itself. The idea is for each bridge to inspect the source address in all the frames it receives. When a bridge first boots, this table is empty; entries are added over time. Also a timeout is associated with each entry and the bridge is cards the entry after a specified period of time.



HOST	PORT
A	1
B	1
C	1
X	2
Y	2
Z	2

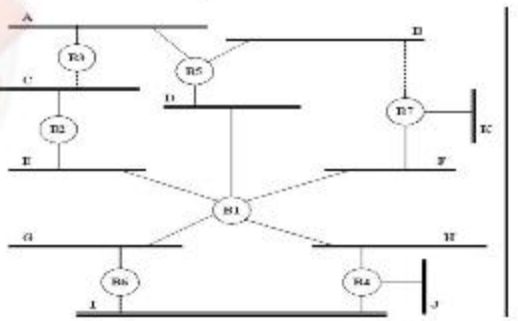
SPANNING TREE ALGORITHM

If the extended LAN is having loops then the frames potentially loop through the extended LAN forever. There are two reasons to an extended LAN to have a loop in it. One possibility is that the network is managed by more than one administrator; no single person knows the entire configuration of the network. Second, loops are built in to network on purpose to provide redundancy in case of failure. Bridges must be able to correctly handle loops. This problem is addressed by having the bridges run a distributed spanning tree algorithm.



The spanning tree algorithm wad developed by Digital Equipment Corporation. The main idea is for the bridges to select the ports over which they will forward frames. The algorithm selects as follows. Each bridge has a unique identifier. In the above example they are labeled as B1, B2, B3 … the algorithm first elects the bridge with smallest ID as the root of the spanning tree. The root bridge always forwards frames out over all of its ports. Then each bridge computes the shortest path to root and notes which of its ports is on this path. This port is also elected as the bridge"s preferred path to the root. Finally, all the bridges connected to a given LAN elect a single designated bridge that will be responsible for forwarding frames toward the root bridge. Each LANs designated bridge is the one that is closest to the root, and if two or more bridges are equally close to the root, then the bridge which having smallest ID wins.

In the above example, B1 is the root bridge since it having the smallest ID. Both B3 and B5 are connected to LAN A, but B5 is the designated bridge since it is closer to the root. Similarly B5 and B7 are connected to LAN B, but B5 is the designated bridge even they are equally closer to the root since B5 having smallest ID.



The bridges have to exchange configuration messages with each other and then decide whether or not they are the root or a designated bridge based on this message. The configuration contains three pieces of information.

1. The ID for the bridge that is sending the message
2. The ID for what the sending bridge believes to be the root bridge
3. The distance, measured in hops, from the sending bridge to the root bridge. Initially each bridge thinks it is the root bridge, so the configuration message will

contain the sending and root same ID. By receiving the configuration message from other bridges they select the root bridge. The selection will be by,

- It identifies a root with a smaller ID or
- It identifies a root with an equal ID but with a shorter distance or
- The root ID and distance are equal, but the sending bridge has a smaller ID

BROADCAST AND MULTICAST

Most LANs support both broadcast and multicast; then bridges must also support these two features.

Broadcast is simple, each bridge forward a frame with a destination broadcast address out on each active port other that the one on which the frame was received. In multicasting, each host deciding for itself whether or not to accept the message.

16. BASIC NETWORKING

The network layer is concerned with getting packets from the source all the way to the destination. The packets may require to make many hops at the intermediate routers while reaching the destination. This is the lowest layer that deals with end to end transmission. In order to achieve its goals, the network layer must know about the topology of the communication network. It must also take care to choose routes to avoid overloading of some of the communication lines while leaving others idle. The network layer-transport layer interface frequently is the interface between the carrier and the customer, that is the boundary of the subnet. The functions of this layer include :

1. Routing - The process of transferring packets received from the Data Link Layer of the source network to the Data Link Layer of the correct destination network is called routing. Involves decision making at each intermediate node on where to send the packet next so that it eventually reaches its destination. The node which makes this choice is called a router. For routing we require some mode of addressing which is recognized by the Network Layer. This addressing is different from the MAC layer addressing.

2. Inter-networking - The network layer is the same across all physical networks (such as Token-Ring and Ethernet). Thus, if two physically different networks have to communicate, the packets that arrive at the Data Link Layer of the node which connects these two physically different networks, would be stripped of their headers and passed to the Network Layer. The network layer would then pass this data to the Data Link Layer of the other physical network.
3. Congestion Control - If the incoming rate of the packets arriving at any router is more than the outgoing rate, then congestion is said to occur. Congestion may be caused by many factors. If suddenly, packets begin arriving on many input lines and all need the same output line, then a queue will build up. If there is insufficient memory to hold all of them, packets will be lost. But even if routers have an infinite amount of memory, congestion gets worse, because by the time packets reach to the front of the queue, they have already timed out (repeatedly), and duplicates have been sent. All these packets are dutifully forwarded to the next router, increasing the load all the way to the destination. Another reason for congestion are slow processors. If the router's CPUs are slow at performing the bookkeeping tasks required of them, queues can build up, even though there is excess line capacity. Similarly, low-bandwidth lines can also cause congestion.

We will now look at these function one by one.

Addressing Scheme

IP addresses are of 4 bytes and consist of :

i) The network address, followed by

ii) The host address

The first part identifies a network on which the host resides and the second part identifies the particular host on the given network. Some nodes which have more than one interface to a network must be assigned separate internet addresses for each interface. This multi-layer addressing makes it easier to find and deliver data to the destination. A fixed size for each of these would lead to wastage or under-usage that is either there will be too many network addresses and few hosts in each (which causes problems for routers who route based on the network address) or there will be very few network addresses and lots of hosts (which will be a waste for small network requirements). Thus, we do away with any notion of fixed sizes for the network and host addresses.

We classify networks as follows:

1. **Large Networks:** 8-bit network address and 24-bit host address. There are approximately 16 million hosts per network and a maximum of 126 (2^7 - 2) Class A networks can be defined. The calculation requires that 2 be subtracted because 0.0.0.0 is reserved for use as the default route and 127.0.0.0 be reserved for the loop back function. Moreover each Class A network can support a maximum of 16,777,214 (2^24 - 2) hosts

per network. The host calculation requires that 2 be subtracted because all 0's are reserved to identify the network itself and all 1s are reserved for broadcast addresses. The reserved numbers may not be assigned to individual hosts.

2. **Medium Networks:** 16-bit network address and 16-bit host address. There are approximately 65000 hosts per network and a maximum of 16,384 (2^14) Class B networks can be defined with up to (2^16-2) hosts per network.

3. **Small Networks:** 24-bit network address and 8-bit host address. There are approximately 250 hosts per network.

You might think that Large and Medium networks are sort of a waste as few corporations or organizations are large enough to have 65000 different hosts. (By the way, there are very few corporations in the world with even close to 65000 employees, and even in these corporations it is highly unlikely that each employee has his/her own computer connected to the network.) Well, if you think so, you're right. This decision seems to have been a mistake.

Address Classes

The IP specifications divide addresses into the following classes :

- Class A - For large networks

- Class B - For medium networks

- Class C - For small networks

- Class D - For multi-cast messages (multi-cast to a "group" of networks)

- Class E - Currently unused, reserved for potential uses in the future

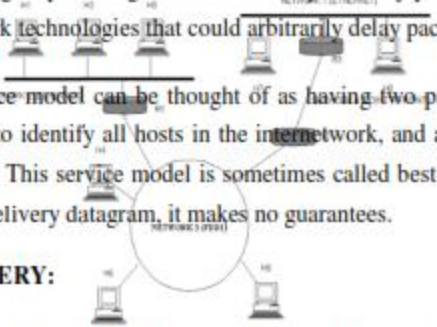
1 1 1 1 28 bits

17.IP(INTERNET PROTOCOL)

An internetwork is often referred to as a network of networks because it is made up of lots of smaller networks. The nodes that interconnect the networks are called routers. They are also sometimes called gateways, but since this term has several other connotations, we restrict our usage to router. The internet protocol is the key tool used today to build scalable, heterogeneous internetwork.

SERVICE MODEL:

The main concern in defining a service model for an internetwork is that we can provide a host-to-host service only if this service can somehow be provided over each of the underlying physical networks. For Example, it would be no good deciding that our internetwork service model was going to provide guaranteed delivery of every packet in 1 ms or less if there were underlying network technologies that could arbitrarily delay packets.



The IP service model can be thought of as having two parts: an addressing scheme, which provides a way to identify all hosts in the internetwork, and a datagram (connectionless) model of data delivery. This service model is sometimes called best effort because, although IP makes every effort to delivery datagram, it makes no guarantees.

DATAGRAM DELIVERY:

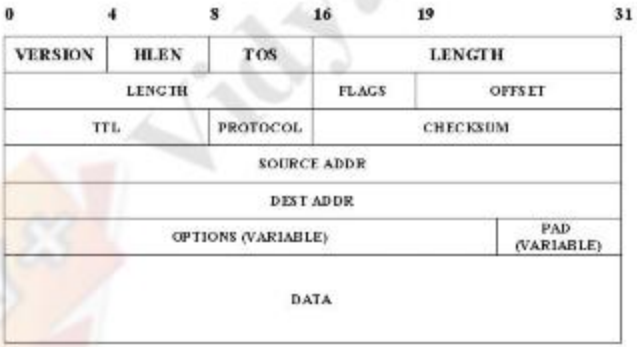
A datagram is a type of packet that happens to be sent in a connectionless manner over a

network. Every datagram carries enough information to let network forward the packet to its correct destination; there is no need for any advance setup mechanism to tell the network what to do when the packet arrives. The network makes its best effort to get it to the desired destination. The best-effort part means that if something goes wrong and the packet gets lost, corrupted, misdelivered,or in any way fails to reach its intended destination, the network does nothing-it made its best effort, and that is all it had to do. It does not make any attempt to recover from the failure. This is sometimes called an unreliable service.

PACKET FORMAT:

The IP datagram, like most packets, consists of a header followed by a number of bytes of data.

The Version field specifies the version of IP. The current version of IP is 4, and it is sometimes called IPv4^2.putting this field right at the start of the datagram makes it easy for everything else in the packet format to be redefined in subsequent versions; the header processing software starts off by looking at the version and then branches off to process the rest of the packet according to the appropriate format.



The next field, HLEN, specifies the length of the header in 32-bit words. When there are no options, which is most of the time, the header is 5 words (20 bytes) long. The 8_bit type of service (TOS) field has had a number of different definitions over the years, but its basic function is to allow packets to be treated differently based on application needs. For example, the TOS value might determine whether or not a packet should be placed in a special queue that

receives low delay.

The next 16-bit of the header contain the Length of the datagram, including the header. Unlike the HLEN field, the Length field counts bytes rather than words. Thus, the maximum size of an IP datagram is 65,535 bytes. The physical network, over which IP is running, however, may not support such long packets. For this reason, IP supports a fragmentation and reassembly process, the second word of the header contains information about fragmentation. The next byte is the time to live (TTL) field. The intent of the field is to catch packets that have been going around in routing loops and discard them, rather than let them consume resources indefinitely.

The Protocol field is simply a demultiplexing key that identifies the higher-level protocol to which this packet should be passed. These are values defined for TCP (6), UDP (17), and many other protocols that may sit above IP in the protocol graph.

The Checksum is calculated by considering the entire IP header as a sequence of 16-bit words, adding them up using ones complement arithmetic, and taking the ones complement of the result.

The last two required fields in the header are the SourceAddr and the DestinationAddr for the packet. The latter is the key to datagram delivery: every packet contains a full address for its intended destination so that forwarding decisions can be made at each router. The source address is required tom allow recipients to decide if they want to accept the packet and to enable them to reply.

Finally, there may be a number of options at the end of the header. The presence or absence of options may be determined by examining the header length (HLen) field. While options are used fairly rarely, a complete IP implementation must handle them all.

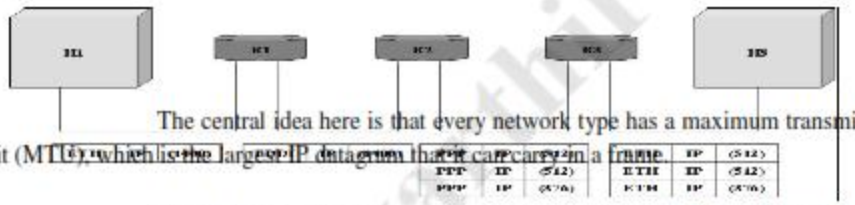
FRAGMENTATION AND REASSEMBLY:

One of the problems of providing a uniform host-to-host service model over a heterogeneous collection of network is that each network technology tends to have its own idea of how large a packet can be. For example, an Ethernet can accept packets up to 1,500 bytes long, while FDDI packets may be 4,500 bytes long.

This leaves two choices for the IP service model: make sure that all IP datagram are small enough to fit inside one packet on any network technology, or provide a means by which packets can be fragmented and reassembled when they are too big to go over a given network technology.

The latter turns out to be a good choice, especially when you consider the fact that new network technologies are always turning up, and IP needs to run over all of them; this would make it hard to pick a suitably small bound on datagram size.

This also means that a host will not send needlessly small packets, which wastes bandwidth and consumes processing resources by acquiring more headers per byte of data sent. For example, two hosts connected to FDDI networks that are interconnected by a point-to-point link would not need to send packets small enough to fit on an Ethernet.



The central idea here is that every network type has a maximum transmission unit (MTU), which is the largest IP datagram that it can carry in a frame.

PPP	IP	(512)	ETH	IP	(512)
PPP	IP	(376)	ETH	IP	(376)

The unfragmented packet has 1,400 bytes of data and a 20-byte IP header. When the packet arrives at the R2, which has an MTU of 532 bytes, it has to be fragmented. A 532-byte MTU leaves 512 bytes for data after the 20-byte IP header , so the first fragment contains 512 bytes of data. The router sets the M bit in the Flags field, meaning that there are more fragments to follow, and it sets the offset to 0,since this fragmented contains the first part of the original datagram.

The data carried in the second fragment starts with the 513th byte of the original data, so the Offset field in this header is set to 64,which is 512/8. Why the division by 8? Because the designers of IP decided that fragmentation should always happen on 8-byte boundaries, which means that the Offset field counts 8-byte chunks, not bytes. The third fragment contains the last 376 bytes of data, and the offset is now 2*512/8=128.since this is the last fragment, the M bit is not set.

START OF HEADER				
IDENT = X			1	OFFSET = 0
REST OF HEADER				
512 DATA BYTES				

START OF HEADER				
IDENT = X			0	OFFSET = 0
REST OF HEADER				

START OF HEADER

GLOBAL ADRESSES:

Global uniqueness is the first property that should be provided in an addressing scheme. Ethernet addresses are globally unique but not sufficient to address entire network. And also they are flat that is no structure in addressing.

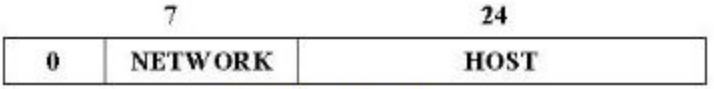
IP addresses are hierarchical. They made up of two parts, they are a network part and a host part. The network part identifies the network to which the host is connected. All hosts which are connected to the same network have same network part in their IP address. The host part then identifies each host on the particular network.

The routers are host but they are connected with two networks. So they need to have an address on each network, one for each interface.

IP addresses are divided into three different classes. They are,

1. class A
2. class B
3. class C

A)



14 16

The class of an IP address is identified in the most significant few bits. If the first bit is 0, it is a class A address. If the first bit is 1 and the second bit is 0, it is a class B address. If the first two bits are 1 and the third bit is 0, t is a class C address.

Class A addresses have 7 bits for network part and 24 bits for host part. So 126 class A networks each can accommodate 2^{24}-2 (about 16 million) hosts. The 0 and 127 are reserved.

Class B addresses have 14 bits for network part and 16 bits for host part. So 2^{14}-2 class B networks each can accommodate 2^{16}-2 (about 65,534) hosts.

Class C addresses have 21 bits for network part and 8 bits for host part. So 2^{21}-2 class C networks each can accommodate 2^{8}-2 (about 254) hosts. The 0 and 127 are reserved.

There are approximately 4 billion possible IP addresses, one half for class A, one quarter for class B and one-eighth for class C address. There are also class D and class E are there. But class D for multicast and class E are currently unused.

IP addresses are written as four decimal integers separated vy dots. Each integer represents the decimal value contained in 1 byte of the address, starting at the most significant.

DATAGRAM FORWARDING IN IP

A datagram is sent from a source to a destination, possibly passing through several routers along the way. Any node, whether it is a host or a router, first tries to establish whether it is connected to the same network as the destination. It compares the network part of the destination address with its network part. If match occurs, then it directly deliver the packet over the network. Else, then it sends to a router. Among several routers, the nearest one will be selected. If none of the entries in the table match the destination"s network number it forwards to the default router.

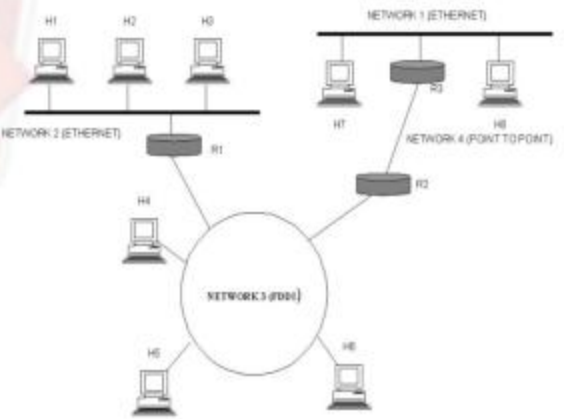
Datagram forwarding algorithm is,

```
If (networknum of destination = networknum of one of my interface) then
    Deliver packet to destination over that interface
Else
    If (networknum of destination is in my forwarding table) then
        Deliver packet to nexthop router
    Else
        Deliver packet to default router
```

For a host with only one interface and one default router in its forwarding table, this simplifies to If (networknum of destination = my networknum) then

```
    Deliver packet to destination directly
Else
    Deliver packet to default router
```

EXAMPLE



NetworkNum	NextHop
1	R3
2	R1

NetworkNum	NextHop
1	R3
2	R1
3	Interface 1
4	Interface 0

18.CLASSLESS INTERDOMAIN ROUTING (CIDR)

Way of describing IP ranges sharing a common bit prefix,we write IP/length, where IP is the first address from the range,and length is the length of the common prefix

Example

We want to describe IP addresses whose binary representation starts with

10011100.00010001.00000100.0010

First IP address from the range:10011100.00010001.00000100.00100000=156.17.4.32
prefix length = 28
Description =156.17.4.32/28

CIDR used mostly for describing single networks 156.17.4.32/28 denotes all the addresses between 156.17.4.32 and 156.17.4.47

- First address in the network is reserved (network address)
- Last address is also reserved: broadcast address.
- Remaining ones can be assigned to computers

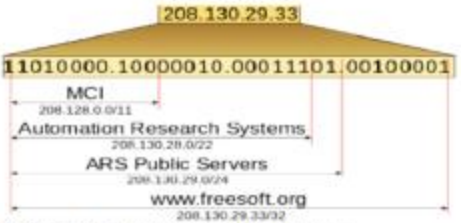
CIDR allows creating hierarchies of networks and subnetworks

Note:Top block received directly from IANA.

208.128.0.0/11

Note:

208.130.29.33/32= range consisting of one IPaddress = single computer, not really a network.



We add/xx (called subnet mask) to all IP addresses.

Example:

156.17.4.32/28: denotes whole network

156.17.4.33/28: the first computer in this network

156.17.4.46/28: the last computer in this network

156.17.4.47/28: broadcast address of this network

If you assign address 10.0.0.1 to a network card, it will be interpreted as10.0.0.1/8

Why?

Reason stems from pre-CIDR IP classes

If the first IP bit is 0, assume subnet mask /8(A class network).

Example:6.0.0.0/8

If the first IP bits are 10, assume subnet mask/16(B class network).

Example:156.17.0.0/16

If the first IP bits are 110, assume subnet mask/24(B classnetwork).

Example:200.200.200.0/24

Network 127.0.0.0/8

Interface lo (loopback)

By connecting with any computer from this network (usually with127.0.0.1), you connect with yourself. Application: it is possible to write, test and use network programs without the network.

Reserved ranges of IP addresses

Packet with such addresses should not be passed through routers.Can be used in local networks (same addresses in different networks).

Ranges:

10.0.0.0/8 (one A class network);

172.16.0.0/12 (16 B class networks);

192.168.0.0/16 (256 C class networks)

If computers with private IP addresses want to communicate with the outside world, the connecting router has to perform NetworkAddress Translation(NAT).

19.ADDRESS RESOLUTION PROTOCOL (ARP):

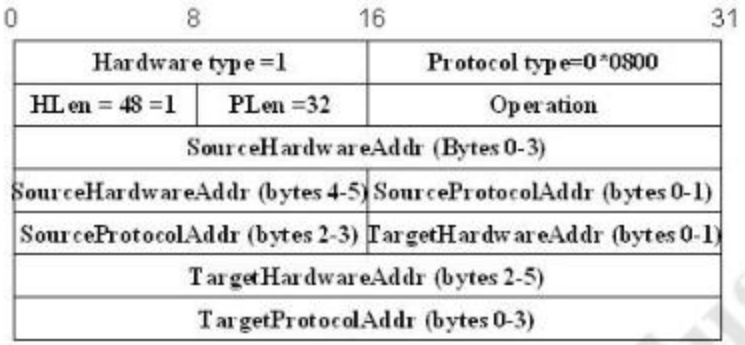
IP data grams contain IP addresses, but the physical interface hardware on the host or router can only understands the addressing scheme of that particular network. So the IP address should be translated to a link level address.

One simplest way to map an IP address in to a physical network address is to encode a host"s physical address in the host part of its IP address. For example, a host with physical address 00100001 01001001 (which has the decimal value 33 in the upper byte and 81 in the lower byte) might be given the IP address 128.96.33.81. But in class C only 8 bits for host part. It is not enough for 48 bit Ethernet address.

A more general solution would be for each host to maintain a table of address pairs, i.e, and the table would map IP addresses into physical address. While this table could be centrally managed by a system administrator and then be copied to each host ion the network, a better approach would be for each host to dynamically learn the contents of the table using the network. This can be accomplished by **Address Resolution Protocol (ARP)**. The goal of ARP is to enable each host on a network to build up a table of mappings between IP address and link level addresses.

Since these mappings may change over time, the entries are timed out periodically and

removed. This happens on the order of every 15 minutes. The set of mappings currently stored in a host is known as ARP cache or ARP table.



The above figure shows the ARP packet format for IP to Ethernet address mappings. ARP can be used for lots of other kinds of mappings the major difference is

their address size. In addition to the IP and link level addresses of both sender and target, the packet contains

- a HardwareType fiels, which specifies the type of the physical network (ex., Ethernet)
- a ProtocolType field, which specifies the higher layer protocol (ex., IP)
- HLen (hardware address length) and PLen (protocol address length) fields, which specifies the length of the link layer address and higher layer protocol address, respectively
- An Operation field, which specifies whether this is a request or a response
- The source and target hardware (Ethernet) and protocol (IP) address.

The results of the ARP process can be added as an extra column in a forwarding table.

20.DYNAMIC HOST CONFIGURATION PROTOCOL (DHCP)

Ethernet addresses are configured into the network adaptor by the manufacturer, and this process is managed in such a way that these addresses are globally unique. This is clearly a sufficient condition to ensure that any collection of hosts connected to a single Ethernet will have unique addresses. IP addresses by contrast must be not only unique on a given internetwork, but also must reflect the structure of the internetwork. They contain a network part and a host part;

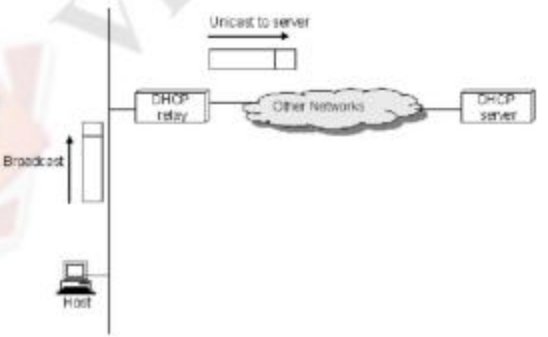
the network part must be the same for all hosts on the same network.

Thus, it is not possible for the IP addresses to be configured once into a host when it is manufactured, since that would imply that the manufacturer knew which hosts were going to end up on which networks, and it would mean that a host, once connected to one network, could never move to another. For this reason, IP addresses need to be reconfigurable.

There are some obvious drawbacks in manual configuration by system administrator. So automated configuration methods are required. The primary method uses a protocol known as **Dynamic Host Configuration Protocol (DHCP).**

DHCP relies on the existence of a DHCP server that is responsible for providing configuration information to hosts. At the simplest level, the DHCP server can function just as a centralized repository for host configuration information. The configuration information for each host could be stored in the DHCP server and automatically retrieved by each host when it is booted or connected to the network. The configuration information for each host stored in a table that is indexed by some form of unique client identifier, typically hardware address.

To contact a DHCP server the host sends a DHCPDISCOVER message to a special IP address (255.255.255.255) that is an IP broadcast address. It will received by all host and routers on the network. DHCP uses the concept of a relay agent. There is at least one relay agent on each network, and it is configured with just one piece of information, the IP address of DHCP server. When a relay agent receives a DHCPDISCOVER message, it unicasts it to the DHCP server and awaits the response, which it will send back to the requesting client



Operation	HType	HLen	Hops
Xid			
Secs		Flags	
ciaddr			
yiaddr			
siaddr			
giaddr			

The packet format is shown above. The message is sent using a protocol called the User Datagram Protocol (UDP). When trying to obtain the configuration information, the client puts its hardware address in the chaddr field. The DHCP server replies by filling in the yiaddr (your IP address) field and sending to the client.

21.ERROR REPORTING (ICMP)

While IP is perfectly willing to drop data grams when the going gets tough for example. When a router does not know how to forward the data gram or when one fragment of a datagram fails to arrive at the destination it does not necessarily fail silently. IP is always configured with a companion protocol, known as Internet Control Message Protocol (ICMP) that defines a collection of error messages that are sent back to the source host whenever a router or host is unable to process an IP data gram successfully. For example, ICMP defines error message indicating that the destination host is unreachable, that the reassembly process failed, that the TTL had reached 0, that the IP header checksum failed and so on. ICMP defines a handful of control message that a router can send back to a source host. Ex., ICMP-redirect tells the source host that there is better route to the destination

UNIT III ROUTING

Routing (RIP, OSPF, metrics) – Switch basics – Global Internet (Areas, BGP, IPv6), Multicast – addresses – multicast routing (DVMRP, PIM)

22. ROUTING

A switch or router needs to be able to look at the packet's destination address and then to determine which of the output ports is the best choice to get the packet to that address.

The forwarding table is used when a packet is being forwarded and so must contain enough information to accomplish the forwarding function. This means that a row in the forwarding table contains the mapping from a network number to an outgoing interface and some MAC information, such as the Ethernet address of the next hop.

The routing table is the table that is built up by the routing algorithms as a precursor to building the forwarding table. It generally contains mappings from network numbers to next hops. It may also contain information about how this information was learned, so that the router will be able to decide when it should discard some information.

The forwarding table needs to be structured to optimize the process of looking up a network number when forwarding a packet, while the routing table needs to be optimized for the purpose of calculating changes in topology. The forwarding table may even be implemented in specialized hardware, whereas this is rarely if ever done for the routing table.

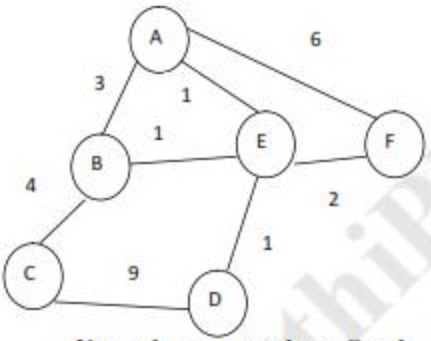
Network number	Next Hop
18	171.69.245.10

Network Number	Interface	MAC Address
18	if0	8:0:2b:e4:b:1:2

Example rows from (a) routing and (b) forwarding tables.

Network as a Graph

Routing is, in essence, a problem of graph theory. a graph representing a network. The nodes of the graph, labeled A through F, may be hosts, switches, routers, or networks. The edges of the graph correspond to the network links. Each edge has an associated *cost*, which gives some indication of the desirability of sending traffic over that link. The basic problem of routing is to find the lowest-cost path between any two nodes, where the cost of a path equals the sum of the costs of all the edges that make up the path.



Network represented as a Graph

23. ROUTING INFORMATION PROTOCOL (RIP)

Each node constructs a one-dimensional array (a vector) containing the "distances" (costs) to all other nodes and distributes that vector to its immediate neighbors. The starting assumption for distance-vector routing is that each node knows the cost of the link to each of its directly connected neighbors. A link that is down is assigned an infinite cost.

The cost of each link is set to 1, so that a least-cost path is simply the one with the fewest hops. (Since all edges have the same cost, we do not show the costs in the graph. Note that each node only knows the information in one row of the table (the one that bears its name in the left column). The global view that is presented here is not available at any single point in the network.

Information stored at node	Distance to reach node						
	A	B	C	D	E	F	G
A	0	1	1	∞	1	1	∞
B	1	0	1	∞	∞	∞	∞
C	1	1	0	1	∞	∞	∞
D	∞	∞	1	0	∞	∞	1
E	1	∞	∞	∞	∞	0	1
F	1	∞	∞	∞	∞	0	1
G	∞	∞	∞	1	∞	1	0

Initial distances stored at each node (global view).

Implementation

The code that implements this algorithm is very straightforward; we give only some of the basics here. Structure Route defines each entry in the routing table, and constant MAX_TTL specifies how long an entry is kept in the table before it is discarded. One of the most widely used routing protocols in IP networks is the Routing Information Protocol (RIP). Its widespread use is due in no small part to the fact that it was distributed along with the popular Berkeley Software Distribution (BSD) version of UNIX, from which many commercial versions of Unix were derived. It is also extremely Simple.

```
#define MAX_ROUTES 128 /* maximum size of routing table */
#define MAX_TTL 120 /* time (in seconds) until route expires */
typedef struct {
NodeAddr Destination; /* address of destination */
NodeAddr NextHop; /* address of next hop */
int Cost; /* distance metric */
u_short TTL; /* time to live */
} Route;
int numRoutes = 0;
Route routingTable[MAX_ROUTES];
```

RIP is in fact a fairly straightforward implementation of distance-vector routing. Routers running RIP send their advertisements every 30 seconds; a router also sends an update message whenever an update from another router causes it to change its routing table. One point of interest is that it supports multiple address families, not just IP. The network-address part of the advertisements is actually represented as a _family, address_ pair.

24.LINK STATE (OSPF)

Link-state routing is the second major class of intra domain routing protocol. The starting assumptions for link-state routing are rather similar to those for distance-vector routing. Each node is assumed to be capable of finding out the state of the link to its neighbors (up or down) and the cost of each link.

Reliable Flooding

Reliable flooding is the process of making sure that all the nodes participating in the routing protocol get a copy of the link-state information from all the other nodes. As the term "flooding" suggests, the basic idea is for a node to send its link-state information out on its entire directly connected links, with each node that receives this information forwarding it out on all of *its* links. This process continues until the information has reached all the nodes in the network.

- The ID of the node that created the LSP;
- A list of directly connected neighbors of that node, with the cost of the link to
- each one;
- A sequence number;
- A time to live for this packet.

One of the most widely used link-state routing protocols is OSPF. The first word, "Open," refers to the fact that it is an open, nonproprietary standard, created under the auspices of the IETF. The "SPF" part comes from an alternative name for link-state routing.

Version	Type	Message length
SourceAddr		
Area Id		
Check sum		Authentication Type
Authentication		

- Authentication of routing messages
- Additional hierarchy
- Load balancing

OSPF Header Format

There are several different types of OSPF messages, but all begin with the same header The Version field is currently set to 2, and the Type field may take the values 1 through 5. The SourceAddr identifies the sender of the message, and the AreaId is a 32-bit identifier of the area in which the node is located. The entire packet, except the authentication data, is protected by a 16-bit checksum using the same algorithm as the IP header (see Section 2.4). The Authentication type is 0 if no authentication is used; otherwise it may be 1, implying a simple password is used, or 2, which indicates that a cryptographic authentication checksum, of the sort described in Section 8.3, is used. In the latter cases the Authentication field carries the password or cryptographic checksum.Of the five OSPF message types, type 1 is the "hello" message, which a router sends to its peers to notify them that it is still alive and connected as described above.
The remaining types are used to request, send, and acknowledge the receipt of link-state messages. The basic building block of link-state messages in OSPF is known as the linkstate advertisement (LSA). One message may contain many LSAs. The LS sequence number is used exactly as described above, to detect old or duplicate LSAs.

25. METRICS

The preceding discussion assumes that link costs, or metrics, are known when we execute the routing algorithm. In this section, we look at some ways to calculate link costs that have proven effective in practice. One example that we have seen already, which is quite reasonable and very simple, is to assign a cost of 1 to all links—the least-cost route will then be the one with the fewest hops. Such an approach has several drawbacks, however. First, it does not distinguish between links on a latency basis. Thus, a satellite link with 250-ms latency looks just as attractive to the routing protocol as a terrestrial link with 1-ms latency. Second, it does not distinguish between routes on a capacity basis, making a 9.6-Kbps link look just as good as a 45-Mbps link. Finally, it does not distinguish between links based on their current load, making it impossible to route around overloaded links. It turns out that this last problem is the hardest because you are trying to capture the complex and dynamic characteristics of a link in a single scalar cost.

The ARPANET was the testing ground for a number of different approaches to link-cost calculation. (It was also the place where the superior stability of link-state over distance-vector routing was demonstrated; the original mechanism used distance vector while the later version used link state.) The following discussion traces the evolution of the ARPANET routing metric and, in so doing, explores the subtle aspects of the problem.

The original ARPANET routing metric measured the number of packets that were queued waiting to be transmitted on each link, meaning that a link with 10 packets queued waiting to be transmitted was assigned a larger cost weight than a link with 5 packets queued for transmission. Using queue length as a routing metric did not work well, however, since queue length is an artificial measure of load—it moves packets toward the shortest queue rather than toward the destination, a situation all too familiar to those of us who hop from line to line at the grocery store. Stated more precisely, the original ARPANET routing mechanism suffered from the fact that it did not take either the bandwidth or the latency of the link into consideration.

A second version of the ARPANET routing algorithm, sometimes called the "new routing mechanism," took both link bandwidth and latency into consideration and used delay, rather than just queue length, as a measure of load. This was done as follows. First, each incoming packet was timestamped with its time of arrival at the router (ArrivalTime); its departure time from the router (DepartTime) was also recorded. Second, when the link-level ACK was received from the other side, the node computed the delay for that packet as

Delay = (DepartTime– ArrivalTime) +TransmissionTime +Latency

where TransmissionTime and Latency were statically defined for the link and captured the link's bandwidth and latency, respectively. Notice that in this case, DepartTime – ArrivalTime represents the amount of time the packet was delayed (queued) in the node due to load. If the ACK did not arrive, but instead the packet timed out, then DepartTime was reset to the time the packet was *retransmitted*. In this case, DepartTime – ArrivalTime captures the reliability of the link—the more frequent the retransmission of packets, the less reliable the link, and the more we want to avoid it. Finally, the weight assigned to each link was derived from the average delay experienced by the packets recently sent over that link.

- A highly loaded link never shows a cost of more than three times its cost when idle;

- The most expensive link is only seven times the cost of the least expensive;
- A high-speed satellite link is more attractive than a low-speed terrestrial link;
- Cost is a function of link utilization only at moderate to high loads.

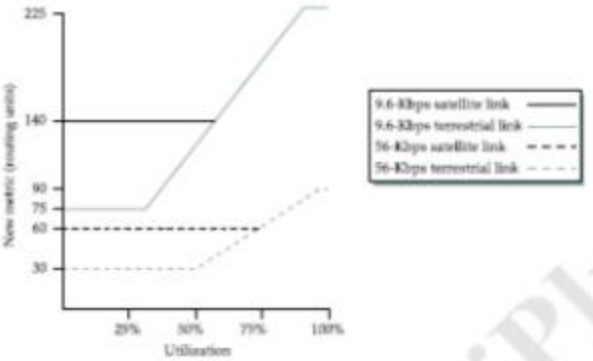


Figure 4.21 Revised ARPANET routing metric versus link utilization.

26.SWITCH BASICS

We saw a variety of ways to build a switch, ranging from a general-purpose workstation with a suitable number of network interfaces to some sophisticated hardware designs. The control processor is responsible for running the routing protocols discussed above, among other things, and generally acts as the central point of control of the router. The switching fabric transfers packets from one port to another, just as in a switch; and the ports provide a range of functionality to allow the router to interface to links of various types (e.g., Ethernet or SONET).

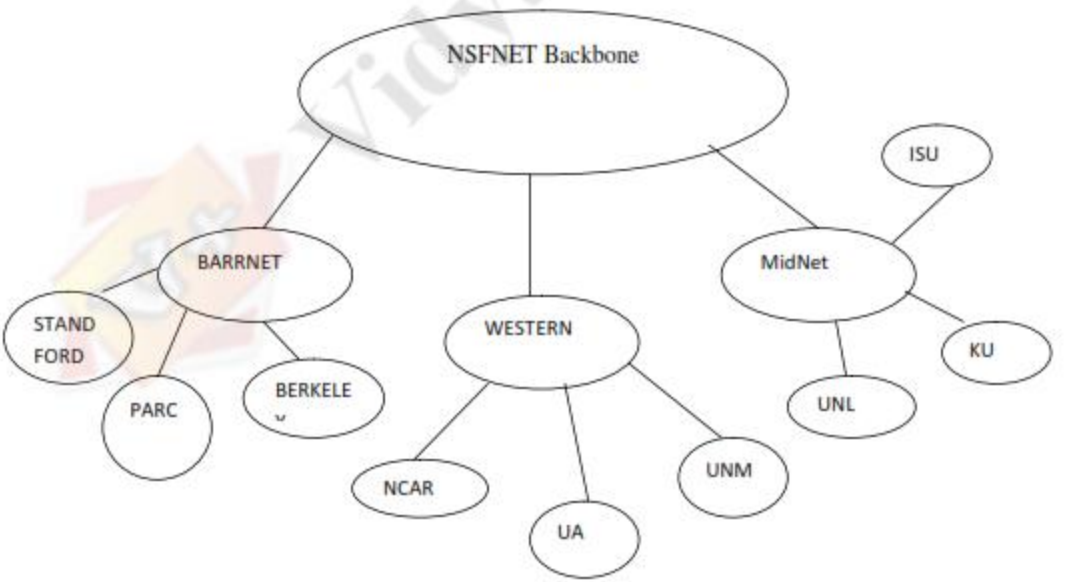
Another consequence of the variable length of IP datagrams is that it can be harder to characterize the performance of a router than a switch that forwards only cells. Routers can usually forward a certain number of packets per second, and this implies that the total throughput in *bits* per second depends on packet size. Router designers generally have to make a choice as to what packet length they will support at *line rate*. That is, if (pps) packets per second is the rate at which packets arriving on a particular port can be forwarded, and linerate is the physical speed of the port in bits per second, then there will be some packetsize in bits such that:

$$\text{packetsize} \times \text{pps} = \text{linerate}$$

This is the packet size at which the router can forward at line rate; it is likely to be able to sustain line rate for longer packets but not for shorter packets. Sometimes a designer might decide that the right packet size to support is 40 bytes, since that is the minimum size of an IP packet that has a TCP header attached. Another choice might be the expected *average* packet size, which can be determined by studying traces of network traffic.

27.GLOBAL INTERNET

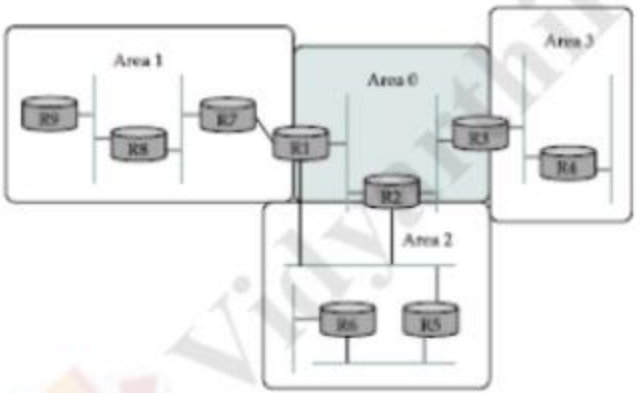
We have seen how to connect a heterogeneous collection of networks to create an internetwork and how to use the simple hierarchy of the IP address to make routing in an internet somewhat scalable. We say "somewhat" scalable because even though each router does not need to know about all the hosts connected to the internet, it does, in the model described so far, need to know about all the networks connected to the internet. Today's Internet has tens of thousands of networks connected to it. Routing protocols such as those we have just discussed do not scale to those kinds of numbers. This section looks at a variety of techniques that greatly improve scalability and that have enabled the Internet to grow as far as it has. Before getting to these techniques, we need to have a general picture in our heads of what the global Internet looks like. It is not just a random interconnection of Ethernets, but instead it takes on a shape that reflects the fact that it interconnects many different organizations. The regional networks were, in turn, connected by a nationwide backbone. In 1990 this backbone was funded by the National Science Foundation (NSF) and was therefore called the NSFNET backbone. Although the detail is not shown in this figure, the provider networks are typically built from a large number of point-to-point links (e.g., DS-3 or OC-3 links) that connect to routers; similarly, each end user site is typically not a single network, but instead consists of multiple physical networks connected by routers and bridges. For example, it is quite likely that different providers will have different ideas about the best routing protocol to use within their network, and on how metrics should be assigned to links in their network. Because of this independence, each provider's network is usually a single *autonomous system (AS)*. The fact that the Internet has a discernible structure can be used to our advantage as we tackle the problem of scalability. In fact, we need to deal with two related scaling issues. The first is the scalability of routing. We need to find ways to minimize the number of network numbers that get carried around in routing protocols and stored in the routing tables of routers. The second is address utilization that is, making sure that The IP address space does not get consumed too quickly.



The tree structure of the Internet in 1990.

28. AREAS

An area is a set of routers that are administratively configured to exchange link-state information with each other. There is one special area—the backbone area, also known as area 0. An example of a routing domain divided into areas is shown in Figure . Routers R1, R2, and R3 are members of the backbone area. They are also members of at least one no backbone area; R1 is actually a member of both area 1 and area 2. A router that is a member of both the backbone area and a nonbackbone area is an area border router (ABR). Note that these are distinct from the routers that are at the edge of an AS, which are referred to as AS border routers for clarity. All the routers in the area send link-state advertisements to each other, and thus develop a complete, consistent map of the area. However, the link-state advertisements of routers that are not area border routers do not leave the area in which they originated. This has the effect of making the flooding and route calculation processes considerably more scalable. For
example, router R4 in area 3 will never see a link-state advertisement from router R8 in area 1. As a consequence, it will know nothing about the detailed topology of areas other than its own.



A domain divided into areas.

The route from sending node to mobile node can be significantly suboptimal. One of the most extreme examples is when a mobile node and the sending node are on the same network, but the home network for the mobile node is on the far side of the Internet.

The sending node addresses all packets to the home network; they traverse the Internet to reach the home agent, which then tunnels them back across the Internet to reach the foreign agent. Clearly it would be nice if the sending node could find out that the mobile node is actually on the same network and deliver the packet directly.

In the more general case, the goal is to deliver packets as directly as possible from sending node to mobile node without passing through a home agent.

This is sometimes referred to as the triangle routing problem since the path from sender to mobile node via home agent takes two sides of a triangle, rather than the third side that is the direct path. The basic idea behind the solution to triangle routing is to let the sending node know the care-of address of the mobile node.

The sending node can then create its own tunnel to the foreign agent. This is treated as an optimization of the process just described. If the sender has been equipped with the necessary software to learn the care of address and create its own tunnel, then the route can be optimized; if not, packets just follow the suboptimal route.

Mobile routing provides some interesting security challenges. For example, an attacker wishing to intercept the packets destined to some other node in an internetwork could contact the home agent for that node and announce itself as the new foreign agent for the node. Thus, it is clear that some authentication mechanisms are required.

When a home agent sees a packet destined for one of the mobile nodes that it supports, it can deduce that the sender is not using the optimal route. Therefore, it sends a binding update message back to the source, in addition to forwarding the data packet to the foreign agent.

The source, if capable, uses this binding update to create an entry in a binding cache, which consists of a list of mappings from mobile node addresses to care-of addresses. The next time this source has a data packet to send to that mobile node, it will find the binding in the cache and can tunnel the packet directly to the foreign agent.

29.BGB (BROADER GATEWAY PROTOCOL) INTERDOMAIN ROUTING

The Internet is organized as autonomous systems, each of which is under the control of a single administrative entity. A corporation's complex internal network might be a single AS, as may the network

of a single Internet service provider. A key design goal of interdomain routing is that policies like the example above, and much more complex ones, should be supported by the interdomain routing system.

To make the problem harder, I need to be able to implement such a policy without any help from other ASs, and in the face of possible misconfiguration or malicious behavior by other ASs.

There have been two major interdomain routing protocols in the recent history of the Internet. The first was the Exterior Gateway Protocol (EGP). EGP had a number of limitations, perhaps the most severe of which was that it constrained the topology of the Internet rather significantly. EGP basically forced a treelike topology onto the Internet, or to be more precise, it was designed when the Internet had a treelike topology, such as that illustrated in Figure 4.24. EGP did not allow for the topology to become more general. Note that in this simple treelike structure, there is a single backbone, and autonomous systems are connected only as parents and children and not as peers.

The replacement for EGP is the Border Gateway Protocol (BGP), which is in its fourth version at the time of this writing (BGP-4). BGP is also known for being rather complex. This section presents the highlights of BGP-4.

As a starting position, BGP assumes that the Internet is an arbitrarily interconnected set of ASs. Given this rough sketch of the Internet, if we define *local traffic* as traffic that originates at or terminates on nodes within an AS, and *transit traffic* as traffic that passes through an AS, we can classify ASs into three types:

■ **Stub AS:** an AS that has only a single connection to one other AS; such an AS will only carry local traffic. The small corporation in Figure 4.29 is an example of a stub AS.
■ **Multihomed AS**: an AS that has connections to more than one other AS but that refuses to carry transit traffic;
■ **Transit AS:** an AS that has connections to more than one other AS and that is designed to carry both transit and local traffic, such as the backbone providers. The first is simply a matter of scale. An Internet backbone router must be able to forward any packet second challenge in inter domain routing arises from the autonomous nature of
the domains. Note that each domain may run its own interior routing protocols, and use any scheme they choose to assign metrics to paths. This means that it is impossible to calculate meaningful path costs for a path that crosses multiple ASs. A cost of 1,000 across one provider might imply a great path, but it might mean an unacceptably bad one from another provider. As a result, interdomain routing advertises only reach ability. The concept of reach ability is basically a statement that "you can reach this network through this AS." This means that for interdomain routing to pick an optimal path is essentially impossible.
The third challenge involves the issue of trust. Provider A might be unwilling to believe certain advertisements from provider B for fear that provider B will advertise erroneous routing information. For example, trusting provider B when he advertises a great route to anywhere in the Internet can be a disastrous choice if provider B turns out to have made a mistake configuring his routers or to have insufficient capacity to carry the traffic. the task of forwarding packets between ASs. BGP does not belong to either of the two main classes of routing protocols (distance-vector and link-state protocols)

Figure 4.30 Example of a network running BGP

Integrating Interdomain and Intradomain Routing

prefix. The final level of complexity comes in backbone networks, which learn so much routing information from BGP that it becomes too costly to inject it into the intradomain protocol. For example, if a border router wants to inject 10,000 prefixes that it learned about from another AS, it will have to send very big link-state packets to the other routers in that AS, and their shortest-path calculations are going to become very complex.
For this reason, the routers in a backbone network use a variant of BGP called interior BGP (iBGP) to effectively redistribute the information that is learned by the BGP speakers at the edges of the AS to all the other routers in the AS. (The other variant of BGP, discussed above, runs between ASs and is called exterior BGP or eBGP.) Ibgp enables any router in the AS to learn the best border router to use when sending a packet to any address.