	**M.Sc. (IT) Sem-3** **PAPER- MITM2104T** **Computer Networks** **UNIT No.1**
Center for Distance and Online Education, PunjabiUniversity,Patiala	**Lesson No:** **1.1 Computer Networks** **1.2 Network Topologies** **1.3 Reference Models OSI and TCP/IP** **1.4 Introduction To NOVEL NETWARE and ARPANET** **1.5 Data Link Layer** **1.6 Medium Access Sublayer And LAN Protocols** **1.7 IEEE 802 Standards**

(Syllabus)

MITM2104T: Computer Networks

Maximum Marks: 70 **Maximum Time: 3 Hrs.**

Minimum Pass Marks: 35%

A) INSTRUCTIONS FOR THE PAPER SETTER

The question paper will consist of three Sections: A, B and C. Sections A and B will have four questions each from the respective section of the syllabus and will carry 10.5 marks for each question. Section C will consist of 7-15 short answer type questions covering the entire syllabus uniformly and will carry a total of 28 marks.

B) INSTRUCTIONS FOR THE CANDIDATES

1. Candidates are required to attempt five questions in all, selecting two questions each from Section A and Section B and compulsory question of Section C.

2. Use of non-programmable scientific calculator is allowed.

SECTION A

Computer networks: uses of computer networks, Goals and applications of networks, computer network structure and architecture, reference models: OSI model, TCP/IP model, Comparison of TCP/IP and OSI models, Introduction to Novell Netware, and ARPANET.

Medium Access Sublayer : Static and dynamic channel allocation for LAN and MAN ALOHA Protocols, **LAN Protocols** : CSMA, CSMA/CD, Collision Free protocol, BRAP, MLMA, Binary countdown, Limited contention protocol, Urn Protocol, Adaptive tree walk protocol.

Networking and Internetworking devices: Repeater, bridges, routers, gateways, switches.

SECTION B

High speed LAN: FDDI, Fast Ethernet, HIPPI, Fiber channel.

LAN IEEE 802.x standards.

Routing: Static vs. Dynamic Routing, various Routing Algorithms.

Congestion Control: Causes of Congestion, Various Congestion Control Strategies and Algorithms

Mobile telephone, mobile telephone switching office.

Internet protocols: Principles of Internetworking, connectionless internetworking, Internet protocols, IPv6.

Network Security: Security requirements and attacks, encryption Public key encryption and digital Signatures. distributed applications: SNMP, SMTP, HTTP.

Reference Books :

1. A.S. Tannenbaum, “Computer Networks", 3rd Edition, Prentice Hall, 1999.
2. Data Communications & Networking by Forouzan, Tata McGraw Hills.
3. D.E. Cormer," Computer Networks and Internet”, 2nd Edition, Addison Wesley Publication, 2000.
4. D.E. Cormer and D.L. Stevens," Inter-networking with TCP-IP: Design, Implementation and Internals", Vol. II, Prentice Hall, 1990.
5. D. Bertsekas and R.Gallagar, “Data Networks”, 2nd Edition, Prentice-Hall, 1992.
6. Stevens W.R.," UNIX Network Programming," Prentice Hall, 1990.

Lesson No. 1.1 **Author: Dr. Amandeep Kaur**
Converted into SLM by: Dr. Vishal Singh

COMPUTER NETWORKS

Objectives:

The objective of this lesson is to:

- Introduce to the students the concept of computer networks
- Help the students understand the goals and applications of computer networks
- Make the students aware of how computer networks are useful for business, homes etc
- Help them to understand the concept of network structure and architecture

Introduction

On the most basic level, a computer network is a collection of devices that can store and manipulate electronic data, interconnected in such a way that network users can store, retrieve, and share information.

We know that the computers have made a lot of progress in a very short time. Initially the computer systems were highly centralized, usually within a single large room. The old model of a single computer serving all of the organization's computational needs has been replaced by one in which a large number of separate but interconnected computers do the job. These systems are called computer networks. The term computer network can be related to as a collection of autonomous computers interconnected by a single technology. Two computers are said to be interconnected if they are able to exchange information. Networks come in many sizes, shapes and forms. Before we start to examine the technical issues in detail, it is worth devoting some time to pointing out what are the basic goals and applications of computer networks and what they can be used for.

1.1 Goals and Applications of Networks

Some of the basics goals of computer networks are:

1) Instantaneous, coordinated information storage and retrieval
2) Combining the skills of different people and the power of different equipment, regardless of the physical locations of the people or the equipment.
3) Enables people to easily share information, allowing them to work more securely, efficiently, and productively.
4) To Control who will have access to sensitive data, equipment, and other resources through their security feature.
5) Help users to share expensive equipment.

Traditional application areas of computer networks are:

1) Business Applications
   a) To extract and correlate information about the entire company
   b) To make all programs, equipment, and especially data available to anyone on the network without regard to the physical location of the resource and the user
   c) Most companies have customer records, inventories, accounts receivable, financial statements, tax information, and much more online.
   d) Doing business electronically
   e) To provide a powerful communication medium among employees.
2) Home Applications
   a) Access to remote information.
   b) Person-to-person communication.
   c) Interactive entertainment.
   d) Electronic commerce.

Emerging applications areas are

1) Mobile users

1.2 Use of Computer Networks

Use of computer networks in business

Many companies have a substantial number of computers. The computers are connected in order to extract and correlate information about the entire company and to make all programs, equipment, and especially data available to anyone on the network without regard to the physical location of the resource and the user.

Every large and medium-sized company and many small companies are vitally dependent on computerized information. Most companies have customer records, inventories, accounts receivable, financial statements, tax information, and much more online.

A computer network can provide a powerful communication medium among employees. Virtually every company that has two or more computers now has **e-mail** (**electronic mail**), which employees generally use for a great deal of daily communication. But e-mail is not the only form of improved communication made possible by computer networks. With a network, it is easy for two or more people who work far apart to write a

report together. Yet another form of computer-assisted communication is videoconferencing. Using this technology, employees at distant locations can hold a meeting, seeing and hearing each other and even writing on a shared virtual blackboard.

Many companies use computer networks for doing business electronically with other companies, especially suppliers and customers. Computer network can also be used for doing business with consumers over the Internet.

Use of computer networks in home applications

In recent years a large number of people use computer for Internet access. Some of the more popular uses of the Internet for home users are as follows:

Access to remote information: Access to remote information comes in many forms. It can be surfing the World Wide Web for information or just for fun. Information available includes the arts, business, cooking, government, health, history, hobbies, recreation, science, sports, travel, and many others. Many newspapers have gone on-line and can be personalized. People can also access the on-line digital library. Many professional organizations, such as the ACM (*www.acm.org*) and the IEEE Computer Society (*www.computer.org*), already have many journals and conference proceedings on-line.

Person-to-person communication: The second broad category of network use is person-to-person communication, basically the 21st century's answer to the 19th century's telephone. E-mail is already used on a daily basis by millions of people all over the world and its use is growing rapidly. It already routinely contains audio and video as well as text and pictures. Smell may take a while. Worldwide newsgroups, with discussions on every conceivable topic, are already commonplace among a select group of people, and this phenomenon will grow to include the population at large. Another type of person-to-person communication often goes by the name of **peer-to-peer** communication, to distinguish it from the client-server model (Parameswaran et al., 2001). In this form, individuals who form a loose group can communicate with others in the group, as shown in Fig. 1-3. Every person can, in principle, communicate with one or more other people; there is no fixed division into clients and servers. Peer-to-peer communication really hit the big time around 2000 with a service called Napster

The next generation of peer-to-peer systems eliminates the central database by having each user maintain his own database locally, as well as providing a list of other nearby people who are members of the system. This form of communication is expected to grow considerably in the future. Other communication-oriented applications include using the Internet to carry telephone calls, video phone, and Internet radio, three rapidly growing areas.

Interactive entertainment: It is a huge and growing industry. Video on demand is one such example of interactive entertainment. In some years it may become possible to select any movie or television program ever made, in any country, and have it displayed on our screen instantly. New films may become interactive, where the user is occasionally

prompted for the story direction with alternative scenarios provided for all cases. Live television may also become interactive, with the audience participating in quiz shows, choosing among contestants

Electronic commerce: Home shopping is already popular and enables users to inspect the on-line catalogs of thousands of companies. Some of these catalogs will soon provide the ability to get an instant video on any product by just clicking on the product's name. After the customer buys a product electronically but cannot figure out how to use it, on-line technical support may be consulted. Another area in which e-commerce is already happening is access to financial institutions. Many people already pay their bills, manage their bank accounts, and handle their investments electronically. This will surely grow as networks become more secure.

Some other uses of computer networks are:

Computer networks may become hugely important to people who are geographically challenged, giving them the same access to services as people living in the middle of a big city.

Telelearning may radically affect education; universities may go national or international. Telemedicine is only now starting to catch on (e.g., remote patient monitoring) but may become much more important.

Mobile users

Mobile computers, such as notebook computers and personal digital assistants (PDAs), are one of the fastest growing segments of the computer industry. Many owners of these computers have desktop machines back at the office and want to be connected to their home base even when away from home or en route. Since having a wired connection is impossible in cars and airplanes, there is a lot of interest in wireless networks.

Wireless networks are of great value to fleets of trucks, taxis, delivery vehicles, and repairpersons for keeping in contact with home. Wireless networks are also important to the military. If you have to be able to fight a war anywhere on earth on short notice, counting on using the local networking infrastructure is probably not a good idea. It is better to bring your own. Although wireless networking and mobile computing are often related, they are not identical. For example, if a traveler plugs a notebook computer into the telephone jack in a hotel room, he has mobility without a wireless network.

There are also the true mobile, wireless applications, ranging from the portable office to people walking around a store with a PDA doing inventory. At many busy airports, car rental return clerks work in the parking lot with wireless portable computers. They type in the license plate number of returning cars, and their portable, which has a built-in printer, calls the main computer, gets the rental information, and prints out the bill on the spot. As wireless technology becomes more widespread, numerous other applications are likely to emerge like:

- Wireless network can be used for utility meter reading. If electricity, gas, water, and other meters in people's homes were to report usage over a wireless network, there would be no need to send out meter readers.
- A whole different application area for wireless networks is the expected merger of cell phones and PDAs into tiny wireless computers.
- Wireless networks can be used for mobile maps etc

1.3 Network Structure and Architecture

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

To help deal with this complexity, network designers have developed general structure of network which is called **network architecture**. The network architecture guides the design and implementation of networks.

When the system gets complex, abstraction is added to it. The idea of an abstraction is to define a unifying model that can capture some important aspect of the system, encapsulate this model in an object that provides an interface that can be manipulated by other components of the system, and hide the details of how the object is implemented from the users of the object. The challenge is to identify abstractions that simultaneously provide a service that proves useful in a large number of situations and that can be efficiently implemented in the underlying system.

Abstractions naturally lead to layering, especially in network systems. The general idea is that you start with the services offered by the underlying hardware, and then add a sequence of layers, each providing a higher (more abstract) level of service. The services provided at the high layers are implemented in terms of the services provided by the low layers

In computer programming, layering means organizing the program into separate functional components that interact in sequentially and hierarchically order with layer above it and the layer below it. Computer networking programs are often layered.

To reduce their design complexity, most networks are organized as a stack of layers or levels, each one built upon the one below it. The number of layers, the name of each layer, the contents of each layer, and the function of each layer differ from network to network. The layered network architecture is shown in Fig. 1 below.

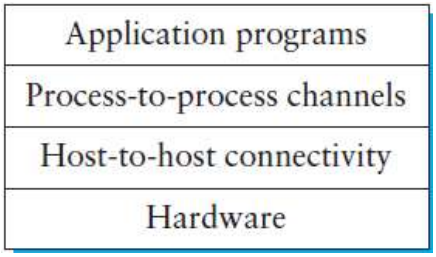

Fig. 1. Example layered network architecture

The purpose of each layer is to offer certain services to the higher layers, shielding those layers from the details of how the offered services are actually implemented. In a sense, each layer is a kind of virtual machine, offering certain services to the layer above it. This concept is actually a familiar one and used throughout computer science, where it is variously known as information hiding, abstract data types, data encapsulation, and object-oriented programming.

Layering provides two features:

- First, it decomposes the problem of building a network into more manageable components.
- Second, it provides a more modular design. If you decide that you want to add some new service, you may only need to modify the functionality at one layer, reusing the functions provided at all the other layers.

The abstract objects that make up the layers of a network system are called **protocols**. That is, a protocol provides a communication service that higher-level objects (such as application processes, or perhaps higher-level protocols) use to exchange messages. The network protocols and interfaces are shown in Fig. 2.

Each protocol defines two different interfaces. First, it defines a service interface to the other objects on the same computer that want to use its communication services. This service interface defines the operations that local objects can perform on the protocol. For example, a request/reply protocol would support operations by which an application can send and receive messages. Second, a protocol defines a peer interface to its counterpart (peer) on another machine. This second interface defines the form and meaning of messages exchanged between protocol peers to implement the communication service.

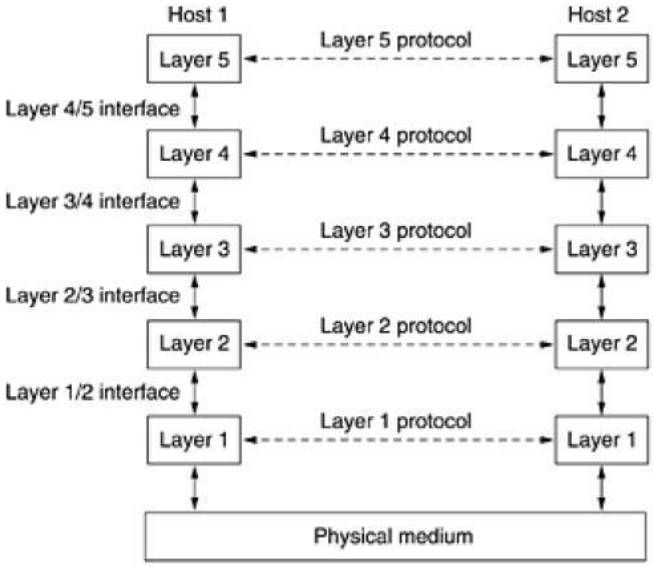

Fig. 2. A five-layer network is shown with service interface and peer interface

The entities comprising the corresponding layers on different machines are called **peers**. The peers may be processes, hardware devices, or even human beings. In other words, it is the peers that communicate by using the protocol.

In reality, no data are directly transferred from layer **n** on one machine to layer **n** on another machine. Instead, each layer passes data and control information to the layer immediately below it, until the lowest layer is reached. Below layer 1 is the physical medium through which actual communication occurs.

Between each pair of adjacent layers is an interface. The interface defines which primitive operations and services the lower layer makes available to the upper one. When network designers decide how many layers to include in a network and what each one should do, one of the most important considerations is defining clean interfaces between the layers. Doing so, in turn, requires that each layer perform a specific collection of well understood functions. In addition to minimizing the amount of information that must be passed between layers, clear-cut interfaces also make it simpler to replace the implementation of one layer with a completely different implementation (e.g., all the telephone lines are replaced by satellite channels) because all that is required of the new implementation is that it offer exactly the same set of services to its upstairs neighbor as the old implementation did.

Thus **network architecture** can also be defined as a set of layers and protocols. The specification of architecture must contain enough information to allow an implementer to write the program or build the hardware for each layer so that it will correctly obey the appropriate protocol. Neither the details of the implementation nor the specification of the interfaces is part of the architecture because these are hidden away inside the machines and not visible from the outside. It is not even necessary that the interfaces on all machines in a network be the same, provided that each machine can correctly use all the protocols. A list of protocols used by a certain system, one protocol per layer, is called a **protocol stack**.

Design issues for the layers

Every layer needs a mechanism for identifying senders and receivers. Since a network normally has many computers, some of which have multiple processes, a means is needed for a process on one machine to specify with whom it wants to talk. As a consequence of having multiple destinations, some form of addressing is needed in order to specify a specific destination.

Another set of design decisions concerns the rules for data transfer. In some systems, data only travel in one direction; in others, data can go both ways. The protocol must also determine how many logical channels the connection corresponds to and what their priorities are. Many networks provide at least two logical channels per connection, one for normal data and one for urgent data.

Error control is an important issue because physical communication circuits are not perfect. Many error-detecting and error-correcting codes are known, but both ends of the connection must agree on which one is being used.

In addition, the receiver must have some way of telling the sender which messages have been correctly received and which has not.

Not all communication channels preserve the order of messages sent on them. To deal with a possible loss of sequencing, the protocol must make explicit provision for the receiver to allow the pieces to be reassembled properly. An obvious solution is to number the pieces, but this solution still leaves open the question of what should be done with pieces that arrive out of order.

An issue that occurs at every level is how to keep a fast sender from swamping a slow receiver with data.

Another problem that must be solved at several levels is the inability of all processes to accept arbitrarily long messages. This property leads to mechanisms for disassembling, transmitting, and then reassembling messages. A related issue is the problem of what to do when processes insist on transmitting data in units that are so small that sending each one separately is inefficient. Here the solution is to gather several small messages heading toward a common destination into a single large message and dismember the large message at the other side.

When it is inconvenient or expensive to set up a separate connection for each pair of communicating processes, the underlying layer may decide to use the same connection for multiple, unrelated conversations. As long as this multiplexing and demultiplexing is done transparently, it can be used by any layer.

When there are multiple paths between source and destination, a route must be chosen. Sometimes this decision must be split over two or more layers. This topic is called routing.

1.4 Summary

In this lesson we have discussed Computer networks, which play a pivotal role in modern technology by enabling the seamless communication and sharing of information between devices and users. It facilitates efficient data transmission, allowing individuals and organizations to access resources, services, and applications remotely. Networks provide a foundation for internet connectivity, supporting global communication, online collaboration, and e-commerce. Additionally, computer networks underpin cloud computing, enabling scalable and on-demand access to computing resources. Their role extends to data storage, backup, and disaster recovery, enhancing data availability and security. In essence, computer networks are the backbone of today's interconnected world, driving innovation, productivity, and digital transformation. We have also discussed the concept of network structure and architecture in detail.

1.5 Self Check Exercise

Q1. Give the main goals of computer networks.

Q2. List any five business applications of the computer networks.

Q3. Why computer networks have layered structure? Explain clearly.

Q4. What is meant by protocols and why are they needed?

Q5. Write a note on design issues for layers

1.6 Suggested Readings

Andrew S. Tanenbaum, Computer Networks, Prentice Hall India, Third Edition.
Forouzan, Data Communication and Networking, Tata McGraw Hill.

Lesson No. 1.2 **Author: Dr. Dilraj Singh**
Converted into SLM by: Dr. Vishal Singh

NETWORK TOPOLOGIES

Objectives

Objectives:

The objective of this lesson is to:

- Introduce the concept of Network topology and its types
- Discuss Network and its various types

1. Introduction: Topology

The term *topology* refers to the way in which a network is laid out physically. Two or more devices connect to a link; two or more links form a topology. The topology of a network is the geometric representation of the relationship of all the links and linking devices (usually called nodes) to one another. Network topologies describe the ways in which the elements of a network are mapped. They describe the physical and logical arrangement of the network nodes. The physical topology of a network refers to the Configuration of cables, computers, and other peripherals.

2. Types of Topologies

• Bus Topology

• Star Topology

- Ring Topology
- Mesh Topology
- Tree Topology
- Hybrid Topology

2.1 Bus Topology

A **bus topology** is multipoint. One long cable acts as a **backbone** to link all the devices in a network as shown in Figure 1. All the nodes (file server, workstations, and peripherals) on a bus topology are connected by one single cable. A bus topology consists of a main run of cable with a terminator at each end. All nodes (file server, workstations, and peripherals) are connected to the linear cable.

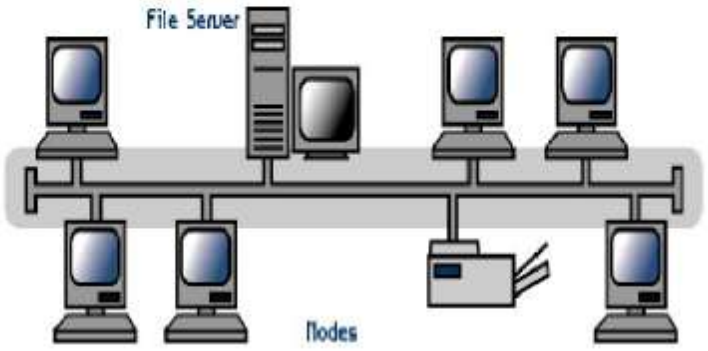

Figure: 1 Bus Topology

Advantages

- The data being transmitted between two nodes passes through all the intermediate nodes. A central server is not required for the management of this topology.
- The traffic is unidirectional and the data transmission is high-speed.
- The adding or removing of network nodes is easy, as the process requires changing only two connections.
- The configuration makes it easy to identify faults in network nodes.
- In this topology, each node has the opportunity to transmit data. Thus, it is a very organized network topology.
- It is less costly than a star topology.

Disadvantages

- The cable length is limited. This limits the number of network nodes that can be connected.
- This network topology can perform well only for a limited number of nodes. When the number of devices connected to the bus increases, the efficiency decreases.
- It is suitable for networks with low traffic. High traffic increases load on the bus, and the network efficiency drops.
- It is heavily dependent on the central bus. A fault in the bus leads to network failure.
- It is not easy to isolate faults in the network nodes.
- Each device on the network "sees" all the data being transmitted, thus posing a security risk.

2.2 Ring Topology

In a ring topology, each device has a dedicated point-to-point connection with only the two devices on either side of it. A signal is passed along the ring in one direction, from device to device, until it reaches its destination. Each device in the ring incorporates a repeater. In a ring network as shown in Figure 2, every device has exactly two neighbors for communication purposes. All messages travel through a ring in the same direction. A failure in any cable or device breaks the loop and can take down the entire network. To implement a ring network we use the Token Ring technology. A token, or small data packet, is continuously passed around the network. When a device needs to transmit, it reserves the token for the next trip around, then attaches its data packet to it.

Advantage of Ring Topology

- Very orderly network where every device has access to the token and the opportunity to transmit.
- Easier to Mange than a Bus Network
- Good Communication over long distances
- Handles high volume of traffic

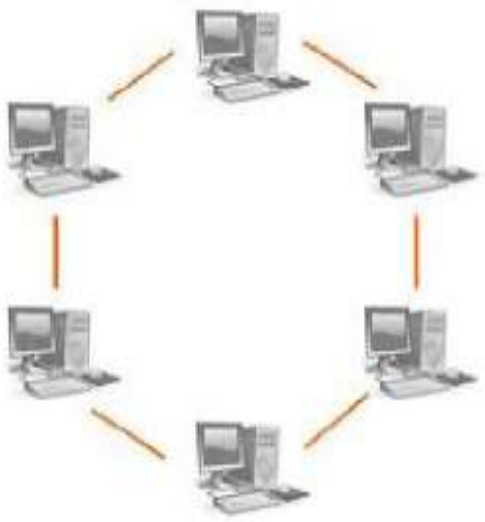

Figure 2. Ring Topology

Disadvantages of Ring Topology

- The failure of a single node of the network can cause the entire network to fail.
- The movement or changes made to network nodes affects the performance of the entire network

2.3 Star Topology

In a star topology, each device has a dedicated point-to-point link only to a central controller, usually called a hub. The devices are not directly linked to one another. In a star network, each node (file server, workstations, and peripherals) is connected to a central device called a hub. The hub takes a signal that comes from any node and passes it along to all the other nodes in the network. Data on a star network passes through the hub, switch, or concentrator before continuing to its destination. The hub, switch, or concentrator manages and controls all functions of the network. The star topology reduces the chance of network failure by connecting all of the systems to a central node.

Advantages of Star Topology

- Easy to manage
- Easy to locate problems (cable/workstations)
- Easier to expand than a bus or ring topology.
- Easy to install and wire.
- Easy to detect faults and to remove parts.

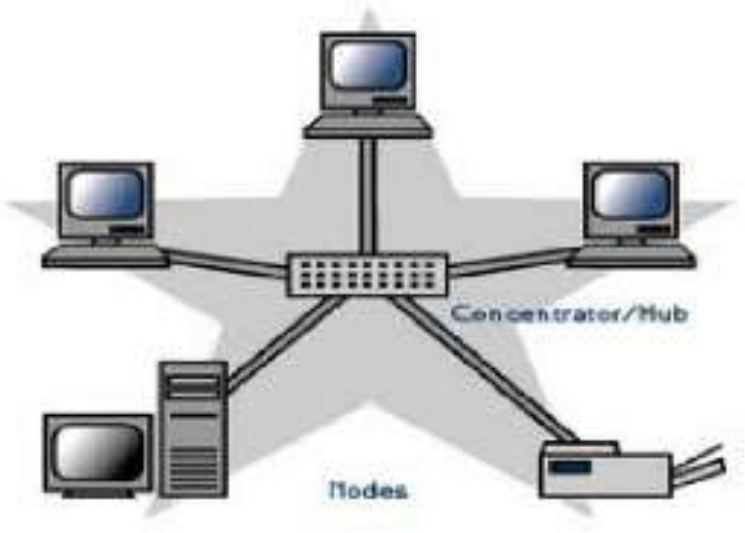

Figure 3. Star Topology

Disadvantages of Star Topology

• Requires more cable length than a linear topology.

• If the hub or concentrator fails, nodes attached are disabled.

• More expensive because of the cost of the concentrators.

2.4 Tree Topology

A tree topology (hierarchical topology) can be viewed as a collection of star networks arranged in a hierarchy. This tree has individual peripheral nodes which are required to transmit to and receive from one other only and are not required to act as repeaters or regenerators. The tree topology arranges links and nodes into distinct hierarchies in order to allow greater control and easier troubleshooting. This is particularly helpful for colleges, universities and schools so that each of the connection to the big network in some way.

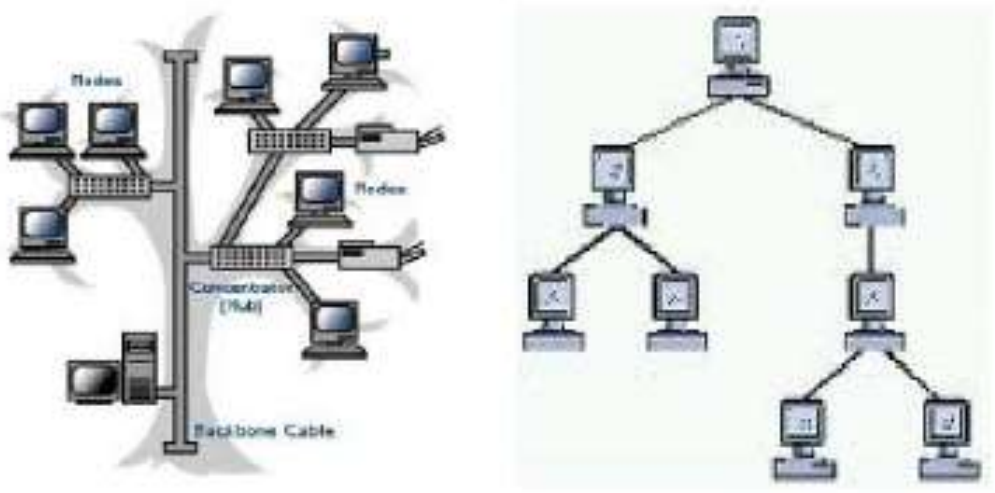

Figure 4. Tree Topology

Advantages of a Tree Topology

• Point-to-point wiring for individual segments.

• Supported by several hardware and software vendors.

• All the computers have access to the larger and their immediate networks.

Disadvantages of a Tree Topology

• Overall length of each segment is limited by the type of cabling used.

• If the backbone line breaks, the entire segment goes down.

• More difficult to configure and wire than other topologies.

2.5 Mesh Topology

In a mesh topology, every device has a dedicated point-to-point link to every other device. The term *dedicated* means that the link carries traffic only between the two devices it connects. In this topology, each node is connected to every other node in the network. Implementing the mesh topology is expensive and difficult. In this type of network, each node may send message to destination through multiple paths. While the data is travelling on the Mesh Network it is automatically configured to reach the destination by taking the shortest route which means the least number of hops.

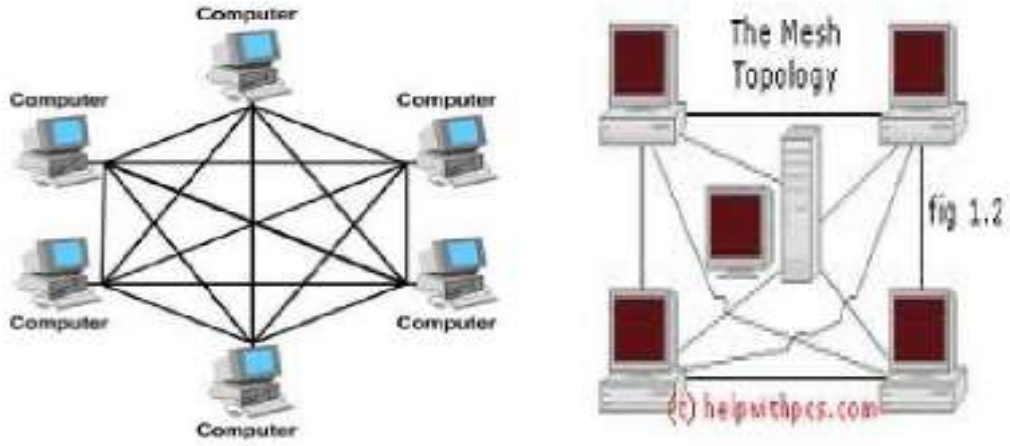

Figure 5. Mesh Topology

Advantages of Mesh Topology

• No traffic problem as there are dedicated links.

• It has multiple links, so if one route is blocked then other routes can be used for data communication.

• Points to point links make fault identification easy.

Disadvantage of Mesh Topology

• There is mesh of wiring which can be difficult to manage.

• Installation is complex as each node is connected to every node.

• Cabling cost is high.

2.6 Hybrid Topology

A network can be hybrid. For example, we can have a main star topology with each branch connecting several stations in a bus topology. A combination of any two or more network topologies. A hybrid topology always accrues when two different basic network topologies are connected. It is a mixture of above mentioned topologies. Usually, a central computer is attached with sub-controllers which in turn participate in a variety of topologies

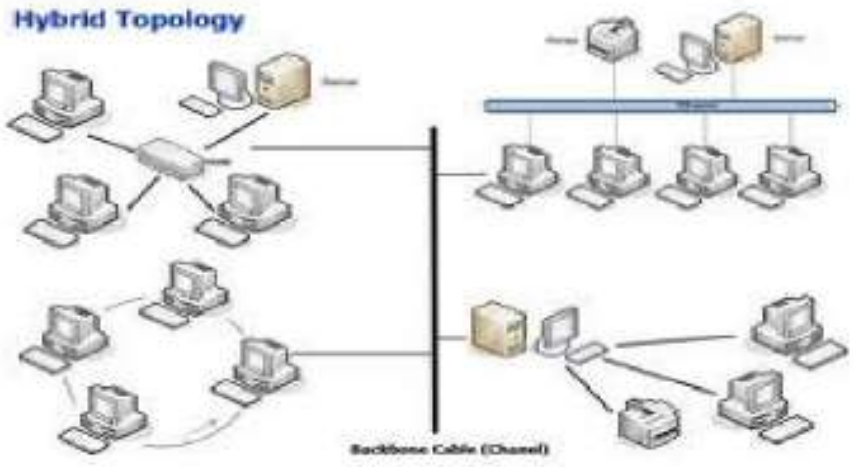

Figure 6. Hybrid Topology

Advantages of a Hybrid Topology

• It is extremely flexible.

• It is very reliable.

Disadvantages of a Hybrid Topology

• Expensive

3. Categories of Networks

Today when we speak of networks, we are generally referring to two primary categories: Local-area networks (LAN) and wide-area networks (WAN). The category into which a network falls is determined by its size. A LAN normally covers a less area and a WAN can be worldwide. Networks of a size in between are normally referred to as metropolitan area Networks (MAN) and span tens of miles.

3.1 Local Area Network

A local area network (LAN) is usually privately owned and links the devices in a single office, building, or campus. Depending on the needs of an organization and the type of technology used, a LAN can be as simple as two PCs and a printer in someone's home office; or it can extend throughout a company and include audio and video peripherals. Currently, LAN size is limited to a few kilometers. LANs are designed to allow resources to be shared between personal computers or workstations. The resources to be shared can include hardware (e.g., a printer), software (e.g., an application program), or data. A common example of a LAN, found in many business environments, links a workgroup of task-related computers, for example, engineering workstations or accounting PCs. One of the computers may be given a large capacity disk drive and may become a server to

clients. Software can be stored on this central server and used as needed by the whole group. In this example, the size of the LAN may be determined by licensing restrictions on the number of users per copy of software, or by restrictions on the number of users licensed to access the operating system. In addition to size, LANs are distinguished from other types of networks by their transmission media and topology. In general, a given LAN will use only one type of transmission medium. The most common LAN topologies are bus, ring, and star. Early LANs had data rates in the 4 to 16 megabits per second (Mbps) range. Today, however, speeds are normally 100 or 1000 Mbps.

3.2 Wide Area Network

A wide area network (WAN) provides long-distance transmission of data, image, audio, and video information over large geographic areas that may comprise a country, a continent, or even the whole world. A WAN can be as complex as the backbones that connect the Internet or as simple as a dial-up line that connects a home computer to the Internet. The point-to-point WAN is normally a line leased from a telephone or cable TV provider that connects a home computer or a small LAN to an Internet service provider (lSP). This type of WAN is often used to provide Internet access. A good example of a switched WAN is the asynchronous transfer mode (ATM) network, which is a network with fixed-size data unit packets called cells.

3.3 Metropolitan Area Networks

A metropolitan area network (MAN) is a network with a size between a LAN and a WAN. It normally covers the area inside a town or a city. It is designed for customers who need a high-speed connectivity, normally to the Internet, and have endpoints spread over a city or part of city. A good example of a MAN is the part of the telephone company network that can provide a high-speed DSL line to the customer. Another example is the cable TV network that originally was designed for cable TV, but today can also be used for high-speed data connection to the Internet.

4. Concept of Protocol services

In computer networks, communication occurs between entities in different systems. An entity is anything capable of sending or receiving information. However, two entities cannot simply send bit streams to each other and expect to be understood. For communication to occur, the entities must agree on a protocol. A protocol is a set of rules that govern data communications. A protocol defines what is communicated, how it is communicated, and when it is communicated.

In a **Layered Network Architecture**, the services are grouped in a hierarchy of layers

– Layer N uses services of layer N-1

– Layer N provides services to layer N+1

Layered Communications

A communication layer is completely defined by

(a) A **peer protocol** which specifies how entities at layer-N communicate

(b) The **service interface** which specifies how adjacent layers at the same system communicate

The following protocol architectures are used in Networks

- **OSI Reference Model**

An ISO standard that covers all aspects of network communications is the Open Systems Interconnection model. It was first introduced in the late 1970s. An open system is a set of protocols that allows any two different systems to communicate regardless of their underlying architecture. The purpose of the OSI model is to show how to facilitate communication between different systems without requiring changes to the logic of the underlying hardware and software. The OSI model is a layered framework for the design of network systems that allows communication between all types of computer systems. It consists of seven separate but related layers, each of which defines a part of the process of moving information across a network

- **TCP/IP Protocols Suite**

The TCPIIP protocol suite was developed prior to the OSI model. Therefore, the layers in the TCP/IP protocol suite do not exactly match those in the OSI model. The original TCP/IP protocol suite was defined as having four layers: host-to-network, internet, transport, and application. *TCP/IP* is a hierarchical protocol made up of interactive modules, each of which provides a specific functionality; however, the modules are not necessarily interdependent. Whereas the OSI model specifies which functions belong to each of its layers, the layers of the *TCP/IP* protocol suite contain relatively independent protocols that can be mixed and matched depending on the needs of the system.

5. Summary

In this lesson we have discussed Network topology, which refers to the physical or logical arrangement of devices, nodes, and connections in a computer network. It determines how data flows and is transmitted between devices. Different network topologies, such as bus, star, ring, mesh, and hybrid, offer varying degrees of efficiency, fault tolerance, and scalability. Network topology influences factors like communication speed, ease of maintenance, and system reliability. For instance, a star topology simplifies troubleshooting as each device connects directly to a central hub, while a mesh topology ensures redundancy by connecting every node to multiple others. The choice of topology is crucial in designing networks that suit specific organizational needs, optimizing performance and minimizing potential points of failure.

6. Self Check Exercise

1. What is Network topology? What are the benefits of network topology?
2. Discuss the various types of network topologies in detail.
3. Explain the different types of Computer networks.

7. Suggested Readings

1. Andrew S. Tannenbaum, “Computer Networks", 3rd Edition, Prentice Hall.
2. Behrouz A. Forouzan, “Data Communications & Networking”, Fourth edition, Tata McGraw Hills.
3. D.E. Cormer and D.L Stevens, "Internetworking with TCP/IP: Design implementations and Internals, “Vol II , Prentice Hall, 1990.
4. D.E. Cormer," Computer Networks and Internet”, 2nd Edition, Addison Wesley Publication, 2000.
5. D. Bertsekas and R.Gallagar, “Data Networks”, 2nd Edition, Prentice-Hall, 1992.
6. Internetworking Technologies Handbook, Fourth Edition, By Cisco Systems, Inc. Publisher: Cisco Press, Pub Date: September 11, 2003, ISBN: 1-58705-119-2 Pages:1128.

Lesson No. 1.3 **Author: Dr. Amandeep Kaur**
Converted into SLM by: Dr. Vishal Singh

REFERENCE MODELS OSI AND TCP/IP

Objectives

After reading this lesson you should be able to:

- Understand the concept of network reference models
- Understand two main reference models (OSI and TCP/IP)
- Compare OSI and TCP/IP reference model

Introduction

The concept of layered network architecture was built into many reference models. Each of these models has different number of layers. In the next sections we will discuss two important network architectures, the OSI reference model and the TCP/IP reference model. Although the protocols associated with the OSI model are rarely used any more, the model itself is actually quite general and still valid, and the features discussed at each layer are still very important. The TCP/IP model has the opposite properties: the model itself is not of much use but the protocols are widely used.

1.1 Open Standards Interconnect (OSI) Model

The reference model for communication programs, Open System Interconnection (OSI), is a layered set of protocols in which programming at both ends of a communications exchange uses an identical set of layers. The International Standards Organization (ISO) developed a theoretical model of how networks should behave and how they are put together. This model is called the **Open Standards Interconnect (OSI) Model**. In the OSI model, there are seven layers, each reflecting a different function that

has to be performed in order for program-to-program communication to take place between computers.

This model is based on a proposal developed by the International Standards Organization (ISO) as a first step toward international standardization of the protocols used in the various layers (Day and Zimmermann, 1983).

The model is called the ISO OSI (Open Systems Interconnection) Reference Model because it deals with connecting open systems—that is, systems that are open for communication with other systems. We will just call it the OSI model for short.

OSI model for networking specifies:

- How information should be handled when being transported over a network.
- How software should interact with the network.
- Layers at which specific networking functions are performed.
- Layer specific functions should be invisible to the layer above it and below it.
- The method of communication at the boundaries between layers.

The Fig. 1.1 shows the OSI reference models seven layers. The principles that were applied to arrive at the seven layers can be briefly summarized as follows:

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

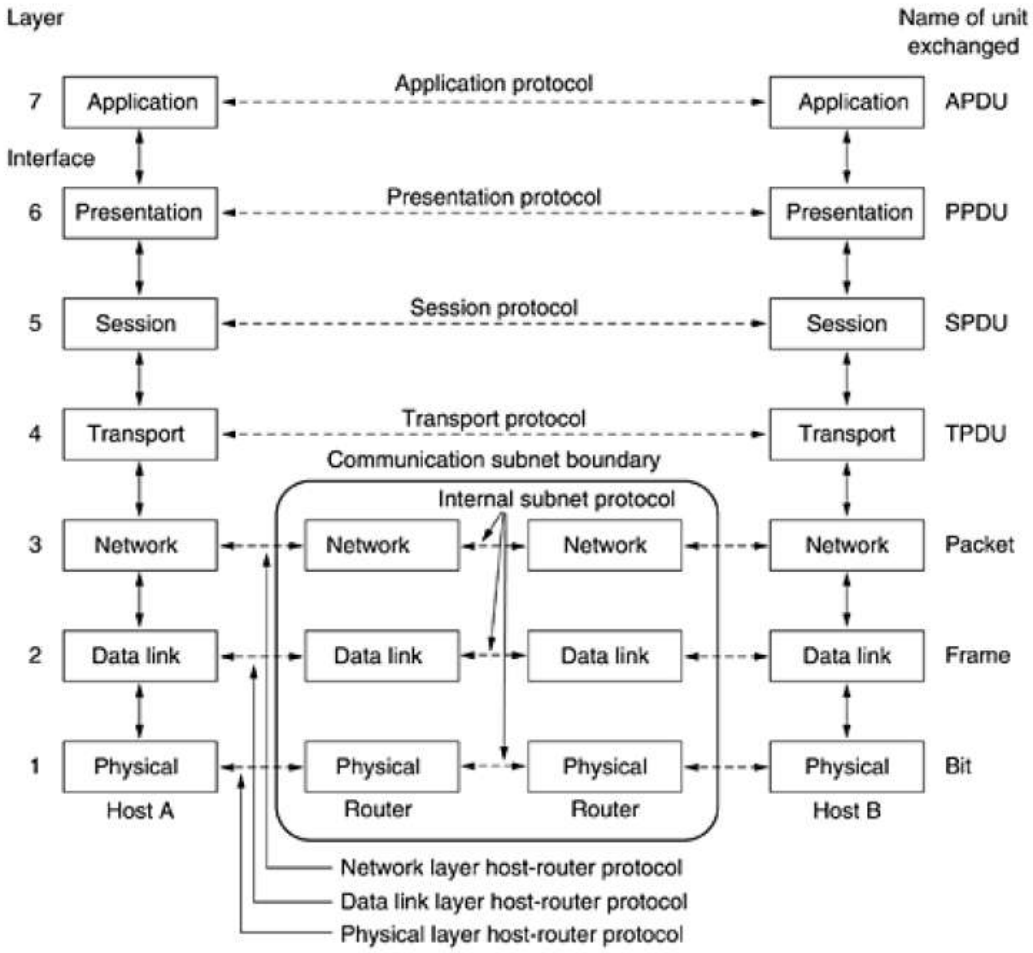

Fig. 1 The OSI reference model

Detailed description of each layer of OSI reference model:

Physical Layer:

The physical layer is responsible for the following jobs:

- Communication with the data link layer above it.
- Defines physical means of sending data over network devices
- Interfaces between network medium and devices
- Fragmentation of data into frames
- Reassembly of frames into data link Protocol Data Units.
- Defines optical, electrical and mechanical characteristics

The physical layer provides for physical connectivity between networked devices. Transmission and receipt of data from the physical medium (copper wire, fiber, radio frequencies, barbed wire, string etc.). The physical layer receives data from the data link Layer, and transmits it to the wire. The physical layer controls the electrical and mechanical functions related to the transmission and receipt of a communications signal. It also manages the encoding and decoding of data contained within the modulated signal. Note that for two devices to communicate, they must be connected to the same type of physical medium (wiring). 802.3 Ethernet to 802.3 Ethernet, **Fiber Distributed Data**

Interface (FDDI) to FDDI, serial to serial etc. Two end stations using different protocols can only communicate through a multi-protocol bridge or a router.

The design issues here largely deal with mechanical, electrical, and timing interfaces, and the physical transmission medium, which lies below the physical layer. It should be noted that in most modern network interface adaptors, the physical and data link functions are performed by the adaptor.

Example Physical Protocols: PCM, Ethernet

Data Link Layer:

The data link Layer is the second layer of the OSI model. The main task of the data link layer is to transform a raw transmission facility into a line that appears free of undetected transmission errors to the network layer. The primary functions of data link layers are:

- Communication with the network layer above.
- Segmentation of upper layer datagrams (also called packets) into frames in sizes that can be handled by the communications hardware.
- Bit ordering. (The data link layer organizes the pattern of data bits into frames before transmission. The frame formatting issues such as stop and start bits, bit order, parity and other functions are handled here. Management of big-endian / little-endian issues is also managed at this layer).
- Communication with the physical layer below.
- Flow regulation: (To keep a fast transmitter from drowning a slow receiver in data).
- Error handling.
- Control access to the shared channel.

Thus the layer provides reliable transmission of data across a physical link. The data link layer is concerned with physical addressing, network topology, physical link management, error notification, ordered delivery of frames, and flow control. It should be noted that in most modern network interface adaptors, the Physical and Datalink functions are performed by the network interface adaptor.

Example Data Link Protocols: IEEE 802.2, IEEE 802.3, 802.5 - Token Ring, FDDI etc

Network Layer:

The network layer is concerned with the following primary functions:

- Communication with the Transport layer above.
- Encapsulation of Transport data into *Network* Layer Protocol Data Units.
- Management of connectivity and routing between hosts or networks.
- Communication with the data link layer below.
- Determines how data are transferred between network devices.
- Routes packets according to unique network device addresses.
- Provides flow and congestion control to prevent network resource depletion

A key design issue is determining how packets are routed from source to destination. Routes can be based on static tables that are "wired into" the network and rarely changed. They can also be determined at the start of each conversation, for example, a terminal

session (e.g., a login to a remote machine). Finally, they can be highly dynamic, being determined anew for each packet, to reflect the current network load.

If too many packets are present in the subnet at the same time, they will get in one another's way, forming bottlenecks. The control of such congestion also belongs to the network layer. More generally, the quality of service provided (delay, transit time, jitter, etc.) is also a network layer issue. When a packet has to travel from one network to another to get to its destination, many problems can arise. The addressing used by the second network may be different from the first one. The second one may not accept the packet at all because it is too large. The protocols may differ, and so on. It is up to the network layer to overcome all these problems to allow heterogeneous networks to be interconnected. In broadcast networks, the routing problem is simple, so the network layer is often thin or even nonexistent.

Examples of network layer protocols include: Internet Protocol (IP), Internet Control Message Protocol (ICMP), Internet Gateway Management Protocol (IGMP) etc

Transport Layer:

- Manages end-to-end message delivery in network.
- Provides reliable and sequential packet delivery through error recovery and flow control mechanisms.
- Provides connectionless and connection oriented packet delivery.
- Communicate with the Session layer above.
- Reassemble *transport* Protocol Data Units into data streams .
- Reliable protocols operating at this layer will detect errors and lost data, recover lost data and manage retransmission of data.
- Segmentation of data streams into *transport* Protocol Data Units.
- Communicate with the Network layer below.

The basic function of the transport layer is to accept data from above, split it up into smaller units if need be, pass these to the network layer, and ensure that the pieces all arrive correctly at the other end. Furthermore, all this must be done efficiently and in a way that isolates the upper layers from the inevitable changes in the hardware technology.

The transport layer also determines what type of service to provide to the session layer, and, ultimately, to the users of the network. The most popular type of transport connection is an error-free point-to-point channel that delivers messages or bytes in the order in which they were sent. However, other possible kinds of transport service are the transporting of isolated messages, with no guarantee about the order of delivery, and the broadcasting of messages to multiple destinations. The transport layer is a true end-to-end layer, all the way from the source to the destination. In other words, a program on the source machine carries on a conversation with a similar program on the destination machine, using the message headers and control messages. In the lower layers, the protocols are between each machine and its immediate neighbors, and not between the ultimate source and destination machines, which may be separated by many routers.

If networking software performs reliable data transfer functions, then the detection of errors and retransmission of data to recover those errors or lost data will occur in software managing this layer. The transport layer may use a variety of techniques such as a Cyclic Redundancy Check, windowing and acknowledgements. If data is lost or damaged it is the *transport* layer's responsibility to recover from that error.

Examples of transport layer protocols include: Transmission Control Protocol (Reliable), User Datagram Protocol (Unreliable), Sequenced Packet Exchange

Session Layer:

The session layer performs the following functions:

- Communication with the presentation layer above.
- Organize and manage one or more connections per application, between hosts.
- Communication with the transport layer below.
- Controls establishment and termination of logic links between users.
- Reports upper layer errors.

The session layer allows users on different machines to establish sessions between them. Sessions offer various services, including dialog control (keeping track of whose turn it is to transmit), token management (preventing two parties from attempting the same critical operation at the same time), and synchronization (checkpointing long transmissions to allow them to continue from where they were after a crash).

The session layer tracks connections, also called sessions. The *session* layer should keep track of multiple file downloads requested by a particular FTP application, or multiple telnet connections from a single terminal client, or web page retrievals from a web server.

With TCP/IP this functionality is handled by application software addressing a connection to a remote machine and using a different local port number for each connection.

Examples: *Sessions* are used to keep track of individual connections to remote servers. Your web browser is an excellent example of the use of *sessions*.

Your web browser (an application layer object) opens a web page. That page contains text, graphics, Macromedia Flash objects and perhaps a Java applet. The graphics, the Flash object and the Java applet are all stored as separate files on the web server. To access them, a separate download must be started. Your web browser opens a separate *session* to the web server to download each of the individual files. The *session* layer keeps track of which packets and data belong to which file and keeps track of where they go (in this case, to your web browser).For e.g.: RPC, NetBIOS etc

Presentation Layer:

The presentation layer performs the following functions :

- Communication with the application layer above.
- Translation of data conforming to cross-platform standards into formats understood by the local machine.
- Masks the differences of data formats between dissimilar systems.
- Specifies architecture-independent data transfer format.
- Encodes and decodes data; Encrypts and decrypts data; Compresses and decompresses data.

- Communication with the session layer below.

The presentation layer handles the conversion of data between a Standards-based or platform independent formats to a format understood by the local machine. This allows for data to be transported between devices and still be understood. Unlike lower layers, which are mostly concerned with moving bits around, the presentation layer is concerned with the syntax and semantics of the information transmitted. In order to make it possible for computers with different data representations to communicate, the data structures to be exchanged can be defined in an abstract way, along with a standard encoding to be used "on the wire."
Examples of Presentation Layer Functions: Conversion of a Sun .RAS raster graphic to JPG., Conversion of ASCII to IBM EBCDIC, Conversion of .wav to .mp3

Application Layer:

- Defines interface to user processes for communication and data transfer in network.
- Provides standardized services such as virtual terminal, file and job transfer and operations.

The application layer contains a variety of protocols that are commonly needed by users. One widely-used application protocol is HTTP (HyperText Transfer Protocol), which is the basis for the World Wide Web. When a browser wants a Web page, it sends the name of the page it wants to the server using HTTP. The server then sends the page back. Other application protocols are used for file transfer, electronic mail, and network news.

The OSI model defines the *application* layer as being the user interface. The OSI application layer is responsible for displaying data and images to the user in a human-recognizable format and to interface with the presentation layer below it.
Examples of applications that utilize the network are: Telnet, FTP etc

1.2 TCP/IP Model

Let us now turn from the OSI reference model to the reference model used in the grandparent of all wide area computer networks, the ARPANET, and its successor, the worldwide Internet.

The ARPANET was a research network sponsored by the DoD (U.S. Department of Defense). It eventually connected hundreds of universities and government installations, using leased telephone lines. When satellite and radio networks were added later, the existing protocols had trouble interworking with them, so new reference architecture was needed. Thus, the ability to connect multiple networks in a seamless way was one of the major design goals from the very beginning. This architecture later became known as the **TCP/IP Reference Model.**

DoD wanted a network model with the following main features

- Ability to connect multiple networks.

- To remain intact as long as the source and destination machines were functioning, even if some of the machines or transmission lines in between were suddenly put out of operation.

These requirements led to the choice of a packet-switching network based on a connectionless internetwork layer. This layer, called the internet layer, is the linchpin that holds the whole architecture together. Its job is to permit hosts to inject packets into any network and have them travel independently to the destination (potentially on a different network). They may even arrive in a different order than they were sent, in which case it is the job of higher layers to rearrange them, if in-order delivery is desired. Note that "internet" is used here in a generic sense, even though this layer is present in the Internet.

The TCP/IP model uses four layers that logically span the equivalent of the top six layers of the OSI reference model. The TCP/IP architectural model has four layers that approximately match six of the seven layers in the OSI Reference Model. The TCP/IP model does not address the physical layer, which is where hardware devices reside. The physical layer is not covered by the TCP/IP model because the data link layer is considered the point at which the interface occurs between the TCP/IP stack and the underlying networking hardware. The next three layers—*network interface*, *internet* and *(host-to-host) transport*—correspond to layers 2, 3 and 4 of the OSI model. The TCP/IP *application* layer conceptually "blurs" the top three OSI layers. It's also worth noting that some people consider certain aspects of the OSI session layer to be arguably part of the TCP/IP host-to-host transport layer. The layers in OSI and TCP/IP are shown in Fig. 2

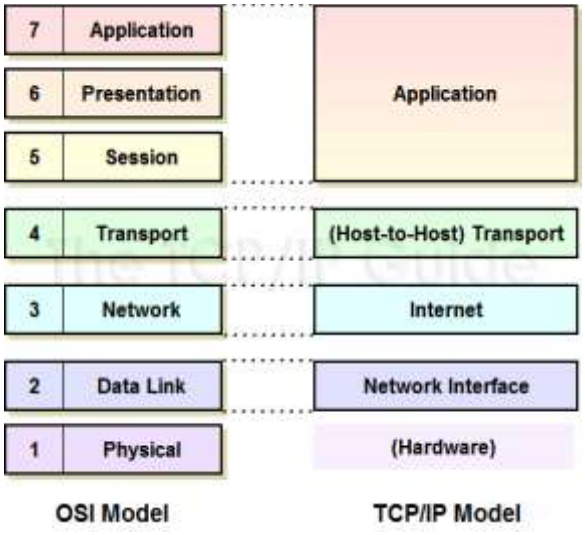

Fig. 2 Network Interface Layer

The TCP/IP model does not exactly match the OSI model. There is no universal agreement regarding how to describe TCP/IP with a layered model but it is generally agreed that there are fewer levels than the seven layers of the OSI model. Most descriptions present from three to five layers. In this lesson the layers of the TCP/IP(Fig. 3) model are defined as follows:

Network Access Layer

Below the internet layer is a great void. The TCP/IP reference model does not really say much about what happens here, except to point out that the host has to connect to the network using some protocol so it can send IP packets to it. This protocol is not defined and varies from host to host and network to network.

Or we can say in TCP/IP the Data Link Layer and Physical Layer are normally grouped together. TCP/IP makes use of existing Data Link and Physical Layer standards rather than defining its own. Most RFCs that refer to the Data Link Layer describe how IP utilizes existing data link protocols such as Ethernet, Token Ring, FDDI, HSSI, and ATM. The characteristics of the hardware that carries the communication signal are typically defined by the Physical Layer. This describes attributes such as pin configurations, voltage levels, and cable requirements. Examples of Physical Layer standards are RS-232C, V.35, and IEEE 802.3.

Internet Layer

In the OSI Reference Model the Network Layer isolates the upper layer protocols from the details of the underlying network and manages the connections across the network. The Internet Protocol (IP) is normally described as the TCP/IP Network Layer. Because of the Inter-Networking emphasis of TCP/IP this is commonly referred to as the Internet Layer. All upper and lower layer communications travel through IP as they are passed through the TCP/IP protocol stack.

The internet layer defines an official packet format and protocol called IP (Internet Protocol). The job of the internet layer is to deliver IP packets where they are supposed to go. Packet routing is clearly the major issue here, as is avoiding congestion. For these reasons, it is reasonable to say that the TCP/IP internet layer is similar in functionality to the OSI network layer.

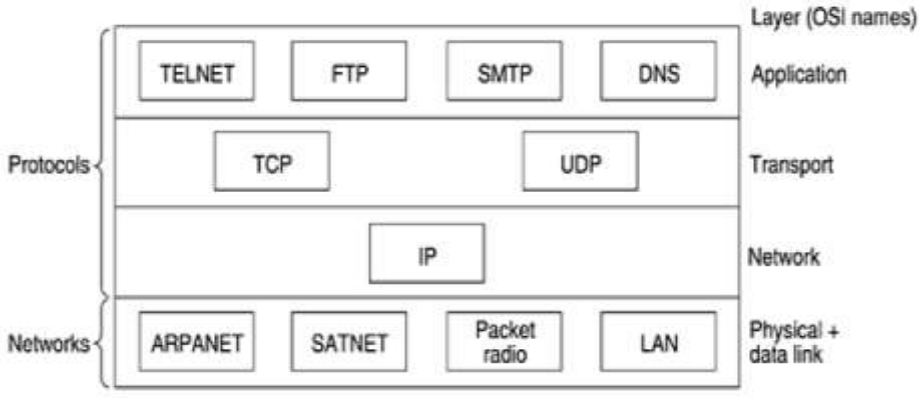

Fig. 3. TCP/IP Protocol Stack

Transport Layer

The layer above the internet layer in the TCP/IP model is now usually called the transport layer. It is designed to allow peer entities on the source and destination hosts to carry on a conversation, just as in the OSI transport layer. Two end-to-end transport

protocols have been defined here. The first one, TCP (Transmission Control Protocol), is a reliable connection-oriented protocol that allows a byte stream originating on one machine to be delivered without error on any other machine in the internet. It fragments the incoming byte stream into discrete messages and passes each one on to the internet layer. At the destination, the receiving TCP process reassembles the received messages into the output stream. TCP also handles flow control to make sure a fast sender cannot swamp a slow receiver with more messages than it can handle. The second protocol in this layer, UDP (User Datagram Protocol), is an unreliable, connectionless protocol for applications that do not want TCP's sequencing or flow control and wish to provide their own. It is also widely used for one-shot, client-server-type request-reply queries and applications in which prompt delivery is more important than accurate delivery, such as transmitting speech or video. Since the model was developed, IP has been implemented on many other networks.

Application Layer

In TCP/IP the Application Layer also includes the OSI Presentation Layer and Session Layer. An application is any process that occurs above the Transport Layer. This includes all of the processes that involve user interaction. The application determines the presentation of the data and controls the session. In TCP/IP the terms **socket** and **port** are used to describe the path over which applications communicate. There are numerous application level protocols in TCP/IP, including Simple Mail Transfer Protocol (SMTP) and Post Office Protocol (POP) used for e-mail, Hyper Text Transfer Protocol (HTTP) used for the World-Wide-Web, and File Transfer Protocol (FTP). Most application level protocols are associated with one or more port number.

The virtual terminal protocol (Telnet) allows a user on one machine to log onto a distant machine and work there. The file transfer protocol provides a way to move data efficiently from one machine to another. Electronic mail was originally just a kind of file transfer, but later a specialized protocol (SMTP) was developed for it. Many other protocols have been added to these over the years: the Domain Name System (DNS) for mapping host names onto their network addresses, NNTP, the protocol for moving USENET news articles around, and HTTP, the protocol for fetching pages on the World Wide Web, and many others.

The four layer structure of TCP/IP is built as information is passed down from applications to the physical network layer. When data is sent, each layer treats all of the information it receives from the layer above as data and adds control information to the front of that data. This control information is called a **header**, and the addition of a header is called **encapsulation (Fig 4)**. When data is received, the opposite procedure takes place as each layer removes its header before passing the data to the layer above.

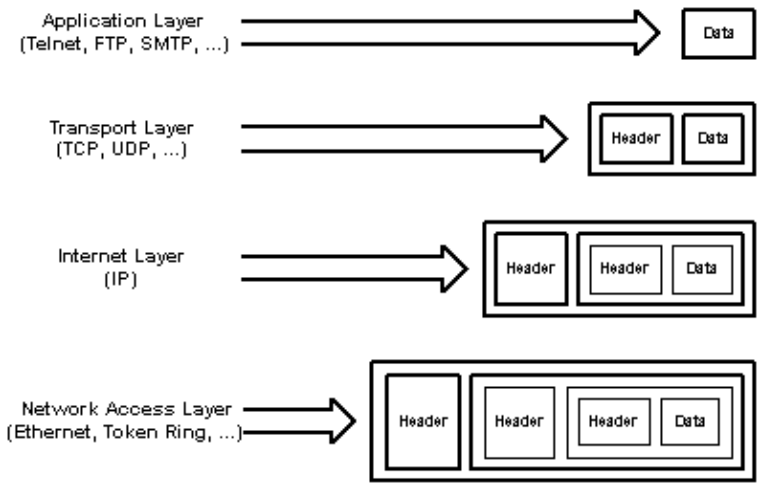

Fig 4 Shows the control information added as the data passes from application to network access layer.

1.3 Comparison of the OSI and TCP/IP Reference Models

The OSI and TCP/IP reference models have much in common. Both are based on the concept of a stack of independent protocols. Also, the functionality of the layers is roughly similar. For example, in both models the layers up through and including the transport layer are there to provide an end-to-end, network-independent transport service to processes wishing to communicate. These layers form the transport provider. Again in both models, the layers above transport are application-oriented users of the transport service.

Despite these fundamental similarities, the two models also have many differences. In this section we will focus on the key differences between the two reference models.

Three concepts are central to the OSI model:

1. Services. 2. Interfaces. 3. Protocols.

Probably the biggest contribution of the OSI model is to make the distinction between these three concepts explicit. Each layer performs some services for the layer above it. The **service** definition tells what the layer does, not how entities above it access it or how the layer works. It defines the layer's semantics.

A layer's **interface** tells the processes above it how to access it. It specifies what the parameters are and what results to expect. It, too, says nothing about how the layer works inside. Finally, the peer **protocols** used in a layer are the layer's own business. It can use any protocols it wants to, as long as it gets the job done (i.e., provides the offered services). It can also change them at will without affecting software in higher layers. These ideas fit very nicely with modern ideas about object-oriented programming. An object, like a layer, has a set of methods (operations) that processes outside the object can invoke. The semantics of these methods define the set of services that the object offers. The methods' parameters and results form the object's interface.

The code internal to the object is its protocol and is not visible or of any concern outside the object. The **TCP/IP model did not originally clearly distinguish between service, interface, and protocol**, although people have tried to retrofit it after the fact to make it more OSI-like. For example, the only real services offered by the internet layer are SEND IP PACKET and RECEIVE IP PACKET.

As a consequence, **the protocols in the OSI model are better hidden than in the TCP/IP model and can be replaced relatively easily as the technology changes**. Being able to make such changes is one of the main purposes of having layered protocols in the first place.

The OSI reference model was devised before the corresponding protocols were invented. This ordering means that **the model was not biased toward one particular set of protocols**, a fact that made it quite general. The downside of this ordering is that the designers did not have much experience with the subject and did not have a good idea of which functionality to put in which layer. For example, the data link layer originally dealt only with point-to-point networks. When broadcast networks came around, a new sublayer had to be hacked into the model. When people started to build real networks using the OSI model and existing protocols, it was discovered that these networks did not match the required service specifications (wonder of wonders), so convergence sublayers had to be grafted onto the model to provide a place for papering over the differences. Finally, the committee originally expected that each country would have one network, run by the government and using the OSI protocols, so no thought was given to internetworking.

With **TCP/IP** the reverse was true: **the protocols came first**, and the model was really just a description of the existing protocols. There was no problem with the protocols fitting the model. They fit perfectly. The only trouble was that **the model did not fit any other protocol stacks.** Consequently, it was not especially useful for describing other, non-TCP/IP networks.

An obvious difference between the two models is the number of layers: the OSI model has seven layers and the TCP/IP has four layers. Both have (inter)network, transport, and application layers, but the other layers are different. Another difference is in the area of connectionless versus connection-oriented communication. **The OSI model supports both connectionless and connection-oriented communication in the network layer, but only connection oriented communication in the transport layer**, where it counts (because the transport service is visible to the users). **The TCP/IP model has only one mode in the network layer (connectionless) but supports both modes in the transport layer, giving the users a choice**. This choice is especially important for simple request-response protocols.

Table 1.1 Main differences between OSI and TCI/IP Reference Models

	OSI Model Reference	**TCP/IP Model Reference**

Service, interface and protocol	Protocols in the OSI model are better hidden and can be replaced relatively easily as the technology changes, which is one of the main objective of layered protocols.	Service, interface and protocol are not clearly defined. For example, the only real services offered by the Internet layer are - Send IP Packet - Receive IP Packet
Functionalities	Because models were invented before protocols, functionalities put in each layer are not very optimized.	In this case, the protocols have been invented before models, so the functionalities are perfectly described.
Numbers of layers	Seven layers, Network (Internet), Transport and Application layers being similar to TCP/IP.	Only four layers.
Connectionless/ Connection-oriented communication	Both connectionless and connection-oriented communications are supported in the network layer, but only connection-oriented communication in the transport layer.	Only one mode in the network layer (connectionless) but both modes in the transport layer are supported, giving the users a choice.

1.4 Summary

In this lesson we have discussed network reference models, which provide a conceptual framework to standardize and guide the design, implementation, and understanding of computer networks. The most prominent reference model is the OSI (Open Systems Interconnection) model, which consists of seven layers representing different network functions, from physical transmission to application services. Another widely used model is the TCP/IP (Transmission Control Protocol/Internet Protocol) suite, comprising four layers and serving as the foundation of the modern internet.

These models facilitate interoperability between different network components and vendors, promoting a modular and layered approach to network development. They aid in troubleshooting, protocol development, and communication standardization. By breaking down complex networking tasks into distinct layers, reference models enhance the ability to manage and comprehend intricate network systems, ensuring seamless communication and compatibility across diverse hardware and software environments.

1.5 Self Check Exercise

Q1. What is meant by network reference models? Explain any one reference model.
Q2. What are the various goals of OSI reference model?
Q3. What is FTP and what is its main function?
Q4. Differentiate between OSI and TCP/IP reference models.
Q5. What are the main functions of transport layer?

1.6 Suggested Readings

Andrew S. Tanenbaum, Computer Networks, Prentice Hall India, Third Edition.
Forouzan, Data Communication and Networking, Tata McGraw Hill

Lesson No. 1.4 **Author: Dr. Amandeep Kaur**
Converted into SLM by: Dr. Vishal Singh

INTRODUCTION TO NOVEL NETWARE AND ARPANET

Objectives

After reading this lesson you should be able to:

- Understand novell netware
- Understand ARPANET

Introduction

1.1 Novell Netware

Novell netware is the most popular network system. It was designed to be used by companies downsizing from mainframe to a network of PCs. It is based on the client server model. Netware uses a proprietary protocol stack. It is based on Xerox Network System (XNS) but with certain modifications.

Novell Netware came into being before OSI and is thus not based on it. Its structure is more like TCP/IP. The Fig 1 gives the reference model for Novell Netware

Layers			
Application	SAP	File Server	…..
Transport	NCP	SPX	
Network	IPX		

Data Link	Ethernet	Token ring	ARCnet
Physical	Ethernet	Token ring	ARCnet

Fig. 1 The Novell Netware reference model

The physical and data link layers can be chosen from among various industry standards, including Ethernet, IBM token ring etc.

The network layer runs an unreliable connectionless internetwork protocol called IPX. The IPX is functionally similar to IP, except that it uses 10-bytes addresses instead of 4 – bytes addresses. Above IPX comes a connection oriented transport protocol called NCP (Network core protocol). It provides various services besides user data transport and is the main protocol of Netware. A second protocol SPX is also available but provides only transport. Other protocols like TCP can also be used. In this the session and presentation layers are not present. Various application protocols are present in the application layer.

The format of IPX packet is shown in Fig 2. The checksum field is rarely used, since the underlying data link layer also provides checksum. The packet length field tells how long the entire packet is (header + data). The transport control field counts how many networks the packet has traversed. When this exceeds the maximum the packet is discarded. The packet type field is used to mark various control packets. The two addresses each contain a 32-bit network number, a 48-bit machine number (the 802 LAN address) and 16-bit local address (socket) on the machine. Finally we have the data; maximum size is determined by the underlying network.

Bytes	2	2	1	1	12	12	Variable length
	Checksum	Packet length	Transport control	Packet type	Destination address	Source address	data

Fig. 2 : A Novel Netware IPX packet

The working of Netware

About once a minute, each server broadcasts a packet giving its address and telling what services it offers. These broadcasts use the SAP (service advertising protocol) protocol. The packets are seen and collected by special agent processes running on the router machines. The agents use the information contained in them to construct databases of which servers are running where.

When a client machine is booted, it broadcasts a request asking where the nearest server is. The agent on the local router machine sees this request, looks in its database of

servers and matches up the request with the best server. The choice of server to use is then sent back to the client. The client can now establish an NCP connection with the server. Using this connection the client and server negotiate the maximum packet size. From this point on the client can access the file system and other services using this connection.

1.2 ARPANET

History of ARPANET

The ARPANET was designed in the 1960s for the US Defense Department, so as to develop new bombproof, distributed packet-switching network technology. It got its name from the provider of the funds, the Advanced Research Projects Agency (ARPA).

In the late 1950s during the Cold War, the DoD wanted a command-and-control network that could survive a nuclear war. At that time, all military communications used the public telephone network, which was considered vulnerable. The black dots in Fig. 3 represent telephone switching offices, each of which was connected to thousands of telephones. These switching offices were, in turn, connected to higher-level switching offices (toll offices), to form a national hierarchy with only a small amount of redundancy. The vulnerability of the system was that the destruction of a few key toll offices could fragment the system into many isolated islands.

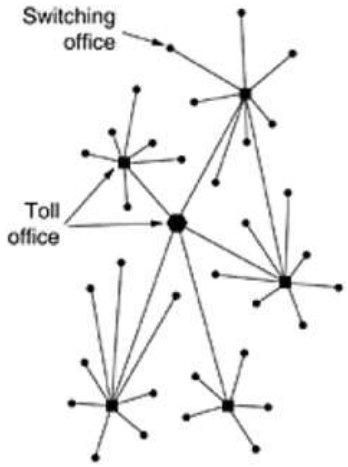

Fig. 3 Structure of the telephone system

Around 1960, the DoD awarded a contract to the RAND Corporation to find a solution. One of its employees, Paul Baran, came up with the highly distributed and fault-tolerant design of Fig. 4. AT&T dismissed Baran's ideas.

Fig. 4 Baran's proposed distributed switching system

In 1957 DoD set up ARPA, the Advanced Research Projects Agency. ARPA had no scientists or laboratories; in fact, it had nothing more than an office and a small (by Pentagon standards) budget. It did its work by issuing grants and contracts to universities and companies whose ideas looked promising to it. In 1967, the attention of ARPA's then director, Larry Roberts, turned to networking. He contacted various experts to decide what to do. One of them, Wesley Clark, suggested building a packet-switched subnet, giving each host its own router, as illustrated in Fig 5 and thus ARPANET came into being.

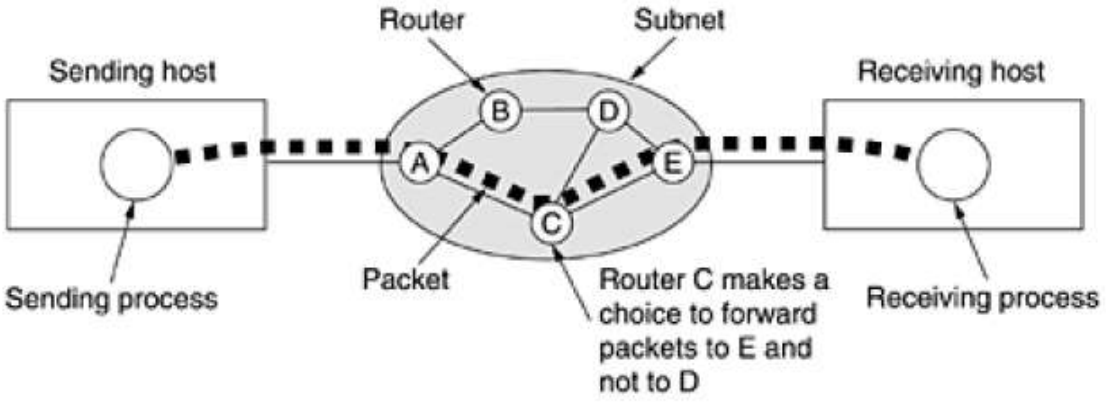

Fig. 5 A stream of packets from sender to receiver

The ARPANET was a research network sponsored by the DoD (U.S. Department of Defense). It eventually connected hundreds of universities and government installations, using leased telephone lines. When satellite and radio networks were added later, the existing protocols had trouble interworking with them, so new reference architecture was needed. Thus, the ability to connect multiple networks in a seamless way was one of the major design goals from the very beginning. Given the DoD's worry that some of its

precious hosts, routers, and internetwork gateways might get blown to pieces at a moment's notice, another major goal was that the network be able to survive loss of subnet hardware, with existing conversations not being broken off. In other words, DoD wanted connections to remain intact as long as the source and destination machines were functioning, even if some of the machines or transmission lines in between were suddenly put out of operation. Furthermore, a flexible architecture was needed since applications with divergent requirements were envisioned, ranging from transferring files to realtime speech transmission. This led to the development of TCP/IP Reference Model, after its two primary protocols.

Original Structure of ARPANET

The original design of ARPANET is shown in Fig. 6, The backbone of the ARPANET consisted of packet-switching computers, called IMPs (Interface Message Processors), and connected by, for the time, superfast 56 Kbit/s lines. Conventional computers with appropriate communications software were then connected to these IMP nodes.

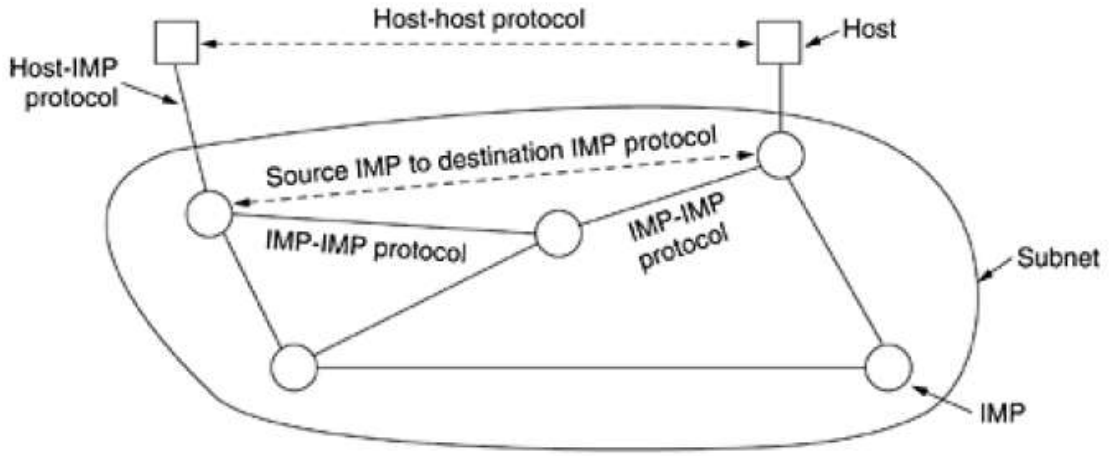

Fig 6 The original ARPANET design

In the autumn of 1969 the first ARPANET computer was connected to the ARPANET's IMP node at the University of California at Los Angeles (UCLA). Doug Engelbart's hypertext-project computer at Stanford Research Institute (SRI) was the next. By the end of the year, the network also included the computers at the University of California, Santa Barbara (UCSB) and the University of Utah, i.e. four in all. All the computers used different operating systems and they were able to talk to each other across the network with equal status.

During the 1970s, the ARPANET grew to connect research institutes and laboratories supported by the Department of Defense in various parts of the USA. Many of these were university laboratories studying data processing and computer networks, which developed the TCP/IP network and its applications for the Internet.

Public demonstration of the ARPANET by Bob Kahn of BBN. The demonstration consisted of a "packet switch", and a TIP (Terminal Interface Processor) in the basement of the Washington Hilton Hotel. The public could use the TIP to run distributed applications across the US. According to Vinton Cerf, the demonstration was a "roaring success".

In 1980 IP became the official standard of the US Department of Defense, and the original ARPANET adopted IP on 1.1.1983, when it became a major part of the Internet. At this time, Defense Department computers were separated from the ARPANET to form their own MILNET network. In 1986 the NSFNET constructs its own backbone network to run in parallel to the ARPANET. And, finally in 1990, with everyone having gone over to using the newer, faster Internet backbone network, the original ARPANET with its network address 10.0.0.0 was shut down.

1.3 Summary

In this lesson we have discussed Novell NetWare, which was a widely used network operating system (NOS) in the 1980s and 1990s. It provided robust file and print services, user authentication, and network management for local area networks (LANs). NetWare introduced the concept of directory services with Novell Directory Services (NDS), later known as eDirectory, offering centralized user and resource management. Despite its popularity, NetWare faced competition from Microsoft's Windows NT and other NOS solutions, leading to a decline in market share over time. We have also discussed ARPANET, which was a pioneering network that laid the foundation for the modern internet. Developed by the United States Department of Defense in the late 1960s, ARPANET connected computers at various research institutions, enabling them to exchange data and communicate. It introduced key concepts like packet switching and TCP/IP protocols. ARPANET's success led to the creation of a global network of networks, eventually evolving into the internet we use today.

1.4 Self Check Exercise

Q1. Write a note on Novell Netware?
Q2. Who developed ARPANET and why?
Q3. Which were the main goals set forth before the development of ARPANET?
Q4. Explain the working of Novell Netware?

1.5 Suggested Readings

Andrew S. Tanenbaum, Computer Networks, Prentice Hall India, Third Edition.
Forouzan, Data Communication and Networking, Tata McGraw Hill

Lesson No. 1.5 **Author: Dr. Dilraj Singh**
Converted into SLM by: Dr. Vishal Singh

DATA LINK LAYER

Objectives

Objectives:

The objective of this lesson is to:

- Introduce to the students the concept of Data link layer
- Study various protocols of data link layer

1. Introduction: Data Link Layer

The data link layer transforms the physical layer, a raw transmission facility, to a link responsible for node-to-node (hop-to-hop) communication. Specific responsibilities of the data link layer include framing, addressing, flow control, error control, and media access control. The data link layer divides the stream of bits received from the network layer into manageable data units called frames. The data link layer adds a header to the frame to define the addresses of the sender and receiver of the frame.

1.1 Data Link Layer Design Issues

The data link layer uses the services of the physical layer to send and receive bits over communication channels. It has a number of functions, including:

1. Providing a well-defined service interface to the network layer.

2. Dealing with transmission errors.
3. Regulating the flow of data so that slow receivers are not swamped by fast senders.
To accomplish these goals, the data link layer takes the packets it gets from the **network layer** and encapsulates them into **frames** for transmission. Each frame contains a frame header, a payload field for holding the packet, and a frame trailer, as illustrated in Figure 1 Frame management forms the heart of what the data link layer does.

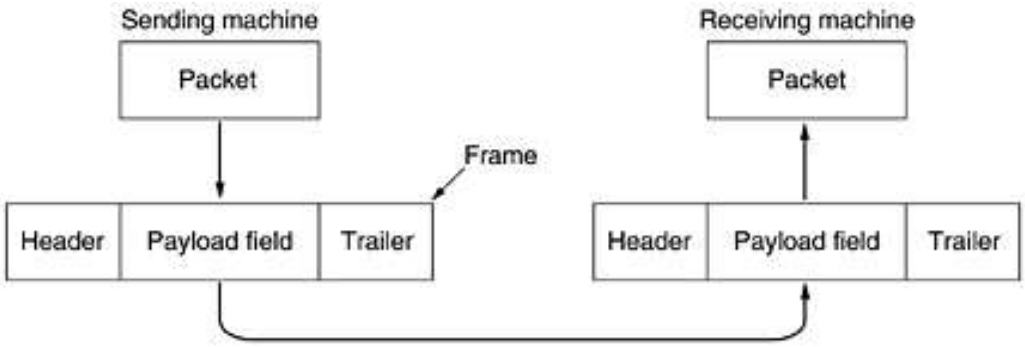

Figure 1: Relationship between packets and frames.

1.1.1 Services Provided to the Network Layer

Data link describes how a shared communication channel can be accessed, and how a data frame can be reliably transmitted. Data link layer is responsible for transmitting data from source network layer to destination network layer

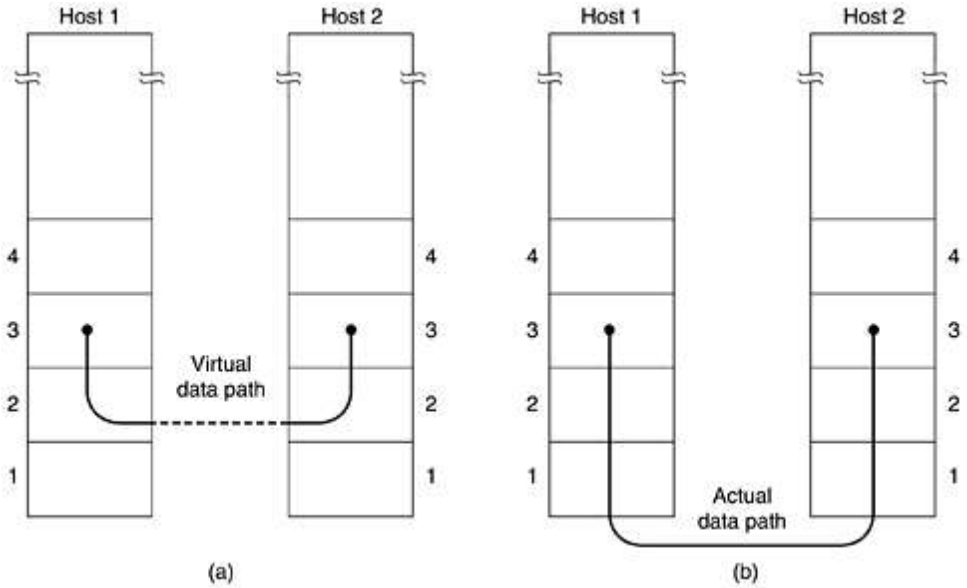

Figure 2: (a) Virtual communication. (b) Actual communication.

Source network layer passes a number of bits to data link layer, who then packs them into frames and relies on physical layer to do actual communication.

The basic services commonly provided are:

1. Unacknowledged connectionless: no attempt to recover wrong or lost frame in layer 2, having least overhead, appropriate when error rate is very low (LANs) so recovery is left to higher layers

2. Acknowledged connectionless: sender knows a frame has arrived safely or not, useful over unreliable channels (wireless systems)
3. Acknowledged connection-oriented: highly reliable but overhead is very high, usually for WANs.

1.1.2 Framing

Data link layer is responsible for making physical link reliable and, to do so, it breaks up network layer data stream into small blocks, a process called **segmentation**, and adds header and frame flag to each block to form a frame, a process called **encapsulation.**

Header generally contains three parts or fields

1. Address: address of sender and/or receiver

2. Error detecting code: a checksum of the frame for error detection

3. Control: additional information to implement protocol functions

The receiving data link layer must know the start and end of a frame according to the frame flag. A good design must make it easy for a receiver to find the start of new frames while using little of the channel bandwidth. We will look at four methods:

1. **Byte count**: The first framing method uses a field in the header to specify the number of bytes in the frame

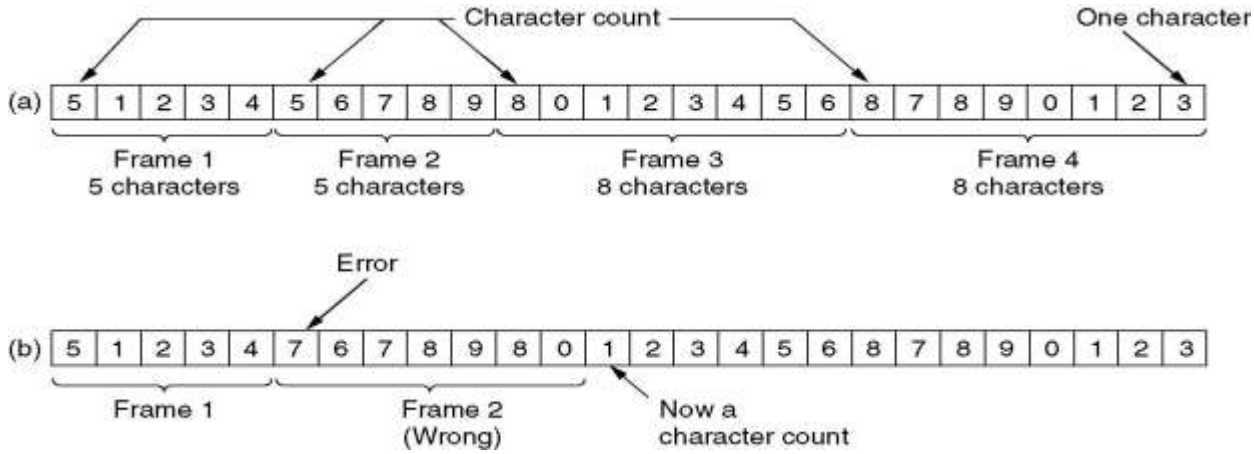

Figure 3: A character stream. (a) Without errors. (b) With one error

2. **Flag bytes with byte stuffing: Each frame start and end with special bytes**. Often the same byte, called a flag byte, is used as both the starting and ending delimiter. This byte is shown in Figure 3 (a) as FLAG.

It may happen that the flag byte occurs in the data. One way to solve this problem is to have the sender's data link layer insert a special escape byte (ESC) just before each "accidental" flag byte in the data. This technique is called **byte stuffing**, Figure 3 (b).

3. **Flag bits with bit stuffing**. Frame flag must be a special bit pattern that never appears any other place inside a frame. Bit stuffing: Use 01111110 as frame flag and this bit pattern must be made special. So sender adds a 0 bit whenever it encounters five consecutive 1 bits in data, and receiver deletes the 0 bit that follows five consecutive 1 bits in the received data.

4. **Physical layer coding violations.** Recall line coding defines how 0 and 1 bits are transmitted in voltage pulses, and deliberately violating the rule can be used to signify something special

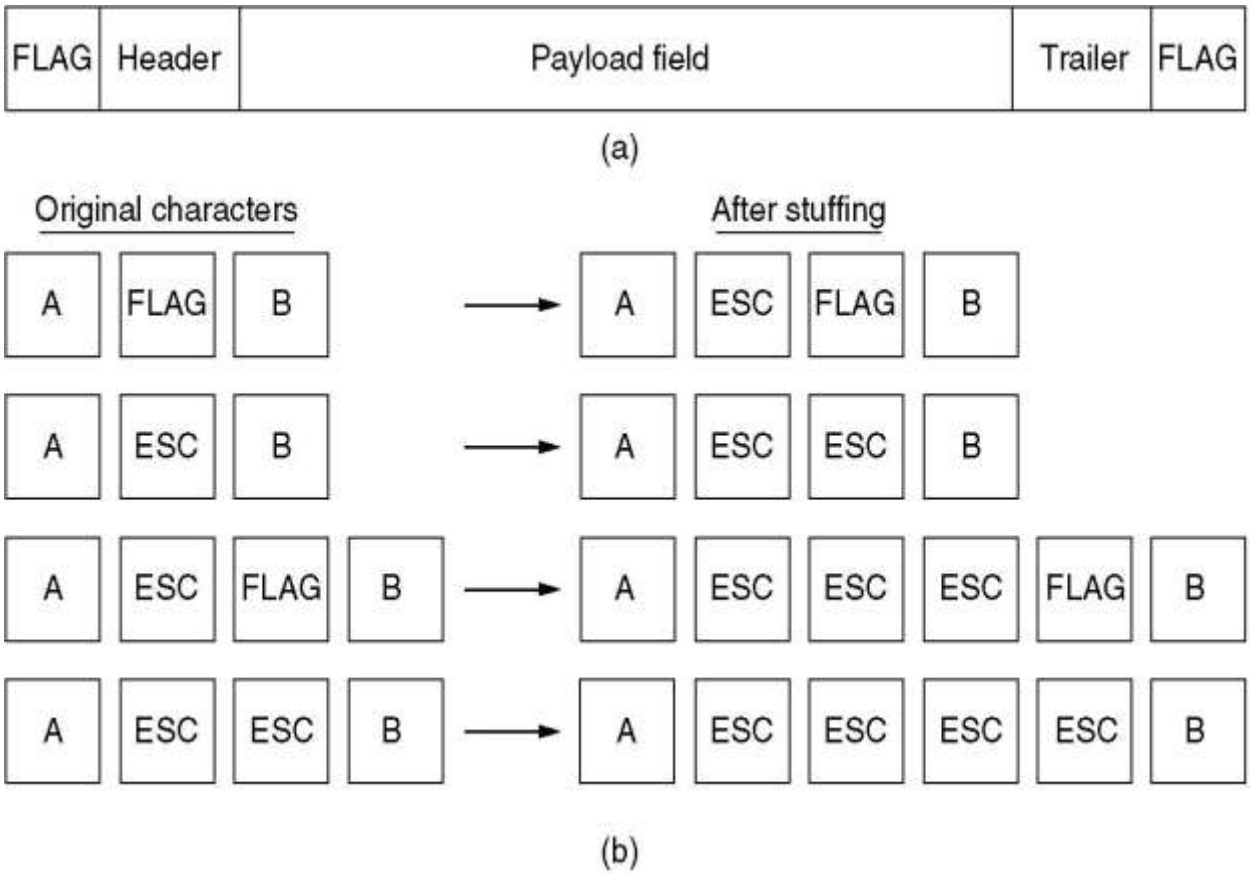

Figure 4: (a) A frame delimited by flag bytes. (b) Four examples of byte sequences before and after stuffing.

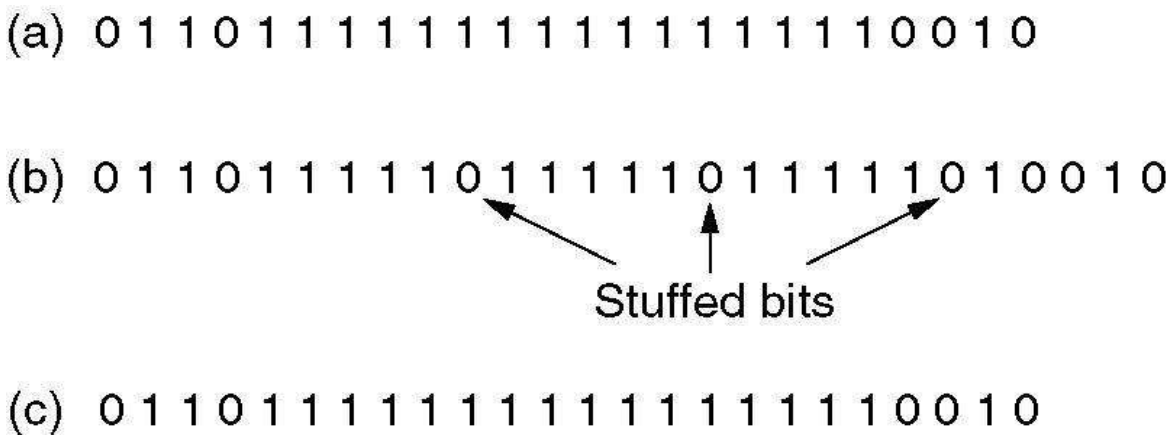

Figure 5: Bit stuffing

(a) The original data.
(b) The data as they appear on the line without the frame flag (01111110).
(c) The data as they are stored in receiver's memory after de-stuffing.

1.1.3 Error Control

Having solved the problem of marking the start and end of each frame, we come to the next problem: how to make sure all frames are eventually delivered to the network layer at the destination and in the proper order. Suppose that the sender just kept outputting frames without regard to whether they were arriving properly. This might be fine for unacknowledged connectionless service, but would most certainly not be fine for reliable, connection-oriented service.

The usual way to ensure reliable delivery is to provide the sender with some feedback about what is happening at the other end of the line. Typically, the protocol calls for the receiver to send back special control frames bearing positive or negative acknowledgements about the incoming frames. If the sender receives a positive acknowledgement about a frame, it knows the frame has arrived safely. On the other hand, a negative acknowledgement means that something has gone wrong, and the frame must be transmitted again.

An additional complication comes from the possibility that hardware troubles may cause a frame to vanish completely (e.g., in a noise burst). In this case, the receiver will not react at all, since it has no reason to react. It should be clear that a protocol in which the sender transmits a frame and then waits for an acknowledgement, positive or negative, will hang forever if a frame is ever lost due to, for example, malfunctioning hardware.

This possibility is dealt with by introducing timers into the data link layer. When the sender transmits a frame, it generally also starts a timer. The timer is set to expire after an interval long enough for the frame to reach the destination, be processed there, and have the acknowledgement propagate back to the sender. Normally, the frame will be correctly received and the acknowledgement will get back before the timer runs out, in which case the timer will be canceled.

However, if either the frame or the acknowledgement is lost, the timer will go off, alerting the sender to a potential problem. The obvious solution is to just transmit the frame again. However, when frames may be transmitted multiple times there is a danger that the receiver will accept the same frame two or more times and pass it to the network layer more than once. To prevent this from happening, it is generally necessary to assign sequence numbers to outgoing frames, so that the receiver can distinguish retransmissions from originals.

The whole issue of managing the timers and sequence numbers so as to ensure that each frame is ultimately passed to the network layer at the destination exactly once, no more and no less, is an important part of the data link layer's duties. Later in this chapter, we will look at a series of increasingly sophisticated examples to see how this management is done.

1.1.4 Flow Control

Another important design issue that occurs in the data link layer (and higher layers as well) is what to do with a sender that systematically wants to transmit frames faster than the receiver can accept them. This situation can easily occur when the sender is running on a fast (or lightly loaded) computer and the receiver is running on a slow (or heavily loaded) machine. The sender keeps pumping the frames out at a high rate until the receiver is completely swamped. Even if the transmission is error free, at a certain point the receiver will simply be unable to handle the frames as they arrive and will start to lose some. Clearly, something has to be done to prevent this situation.

Two approaches are commonly used. In the first one, feedback-based flow control, the receiver sends back information to the sender giving it permission to send more data or at least telling the sender how the receiver is doing. In the second one, rate-based flow control, the protocol has a built-in mechanism that limits the rate at which senders may transmit data, without using feedback from the receiver. In this chapter we will study feedback-based flow control schemes because rate-based schemes are never used in the data link layer.

Various feedback-based flow control schemes are known, but most of them use the same basic principle. The protocol contains well-defined rules about when a sender may transmit the next frame. These rules often prohibit frames from being sent until the receiver has granted permission, either implicitly or explicitly.

2. Elementary Data Link Protocols

Data link layer can combine framing, flow control, and error control to achieve the delivery of data from one node to another. The protocols are normally implemented in software by using one of the common programming languages.

We divide the discussion of protocols into those that can be used for noiseless (error-free) channels and those that can be used for noisy (error-creating) channels. The

protocols in the first category cannot be used in real life, but they serve as a basis for understanding the protocols of noisy channels. Figure

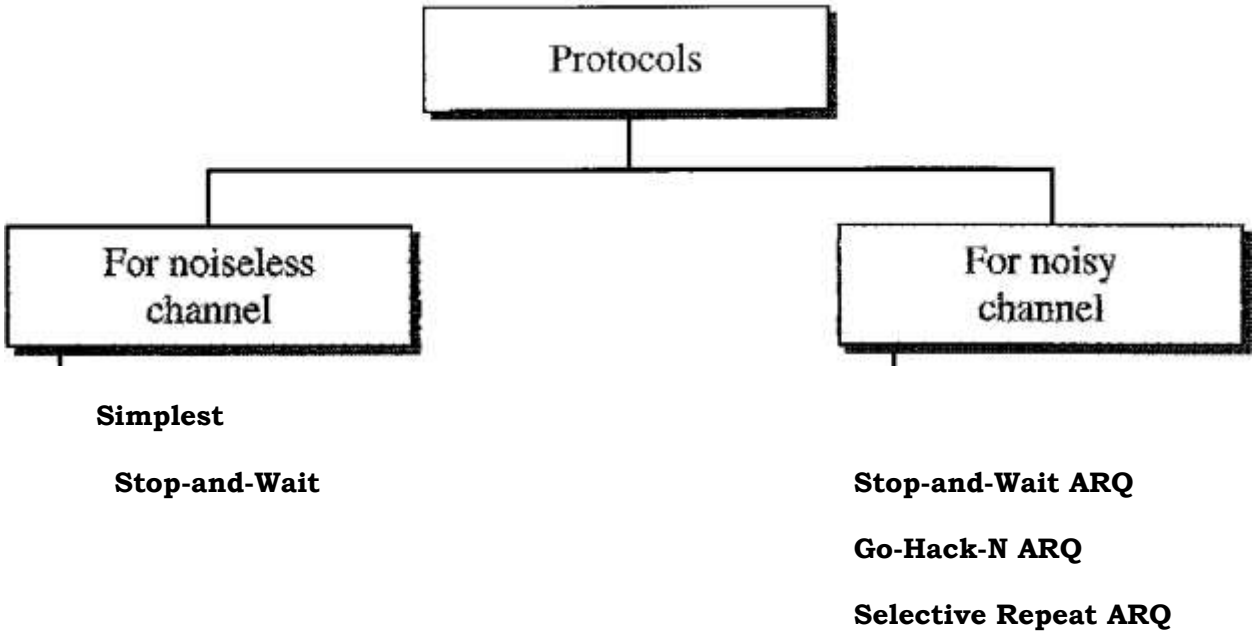

Figure 6. Protocol Classification

2.1 NOISELESS CHANNELS

Let us first assume we have an ideal channel in which no frames are lost, duplicated, or corrupted. We introduce two protocols for this type of channel. The first is a protocol that does not use flow control; the second is the one that does. Of course, neither has error control because we have assumed that the channel is a perfect noiseless channel.

2.1.1 Simplest Protocol

Our first protocol, which we call the Simplest Protocol for lack of any other name, is one that has no flow or error control. Like other protocols we will discuss in this chapter, it is Unidirectional protocol in which data frames are traveling in only one direction-from the sender to receiver. We assume that the receiver can immediately handle any frame it receives with a processing time that is small enough to be negligible. The data link layer of the receiver immediately removes the header from the frame and hands the data packet to its network layer, which can also accept the packet immediately. In other words, the receiver can never be overwhelmed with incoming frames.

Design

There is no need for flow control in this scheme. The data link layer at the sender site gets data from its network layer, makes a frame out of the data, and sends it. The data link layer at the receiver site receives a frame from its physical layer, extracts data from the frame, and delivers the data to its network layer. The data link layers of the sender and receiver provide transmission services for their network layers. The data link layers use the services provided by their physical layers (such as signaling, multiplexing, and so on) for the physical transmission of bits. Figure 7 shows a design.

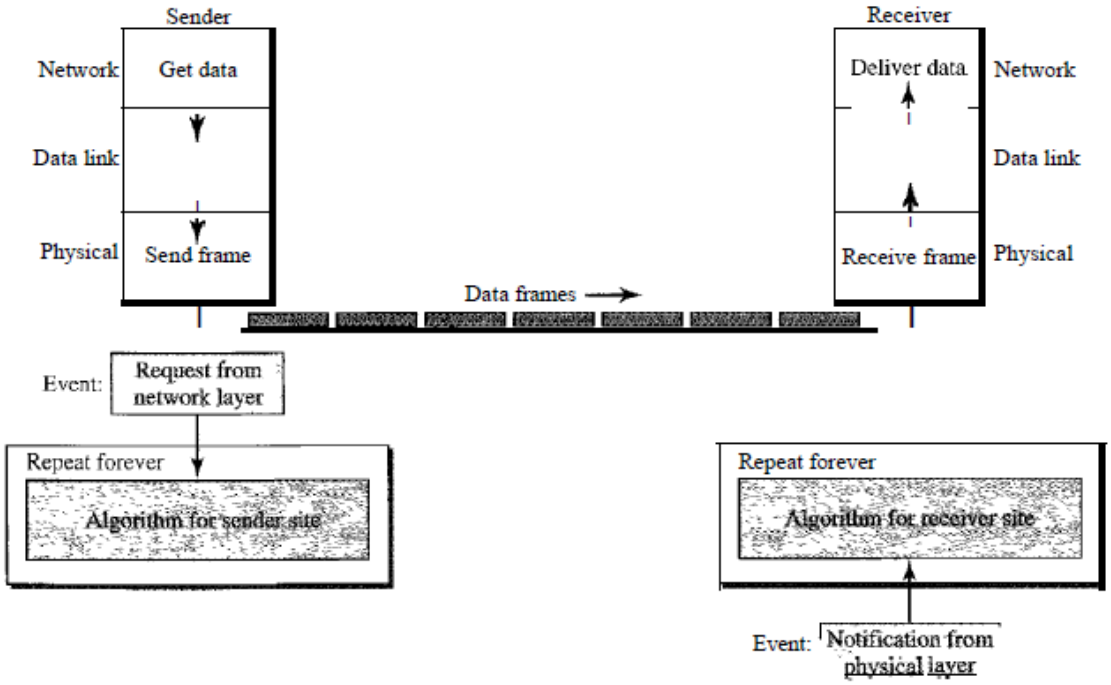

Figure 7. The design of the simplest protocol with no flow or error control

We need to elaborate on the procedure used by both data link layers. The sender site cannot send a frame until its network layer has a data packet to send. The receiver site cannot deliver a data packet to its network layer until a frame arrives. If the protocol is implemented as a procedure, we need to introduce the idea of events in the protocol. The procedure at the sender site is constantly running; there is no action until there is a request from the network layer. The procedure at the receiver site is also constantly running, but there is no action until notification from the physical layer arrives. Both procedures are constantly running because they do not know when the corresponding events will occur.

2.1.2 Stop-and-Wait Protocol

If data frames arrive at the receiver site faster than they can be processed, the frames must be stored until their use. Normally, the receiver does not have enough storage space, especially if it is receiving data from many sources. This may result in either the

discarding of frames or denial of service. To prevent the receiver from becoming overwhelmed with frames, we somehow need to tell the sender to slow down. There must be feedback from the receiver to the sender. The protocol we discuss now is called the Stop-and-Wait Protocol because the sender sends one frame, stops until it receives confirmation from the receiver (okay to go ahead), and then sends the next frame. We still have unidirectional communication for data frames, but auxiliary ACK frames (simple tokens of acknowledgment) travel from the other direction. We add flow control to our previous protocol.

The Figure 8 illustrates the mechanism. Comparing this figure with Figure 6, we can see the traffic on the forward channel (from sender to receiver) and the reverse channel. At any time, there is either one data frame on the forward channel or one ACK frame on the reverse channel. We therefore need a half-duplex link.

2.2 NOISY CHANNELS

Although the Stop-and-Wait Protocol gives us an idea of how to add flow control to its predecessor, noiseless channels are nonexistent. We can ignore the error (as we sometimes do), or we need to add error control to our protocols. We discuss three protocols in this section that use error control.

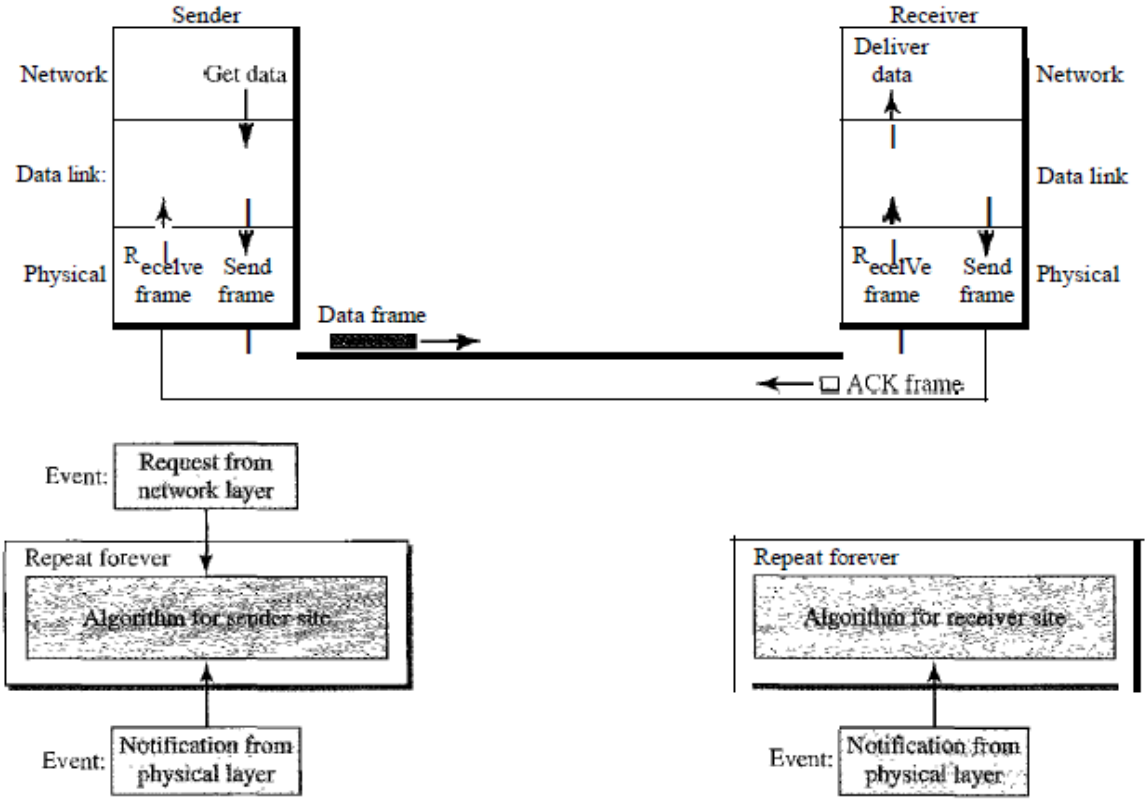

Figure 8. Design of Stop-and- Wait Protocol

2.2.1 Stop-and-Wait Automatic Repeat Request

Our first protocol, called the Stop-and-Wait Automatic Repeat Request (Stop-and Wait ARQ), adds a simple error control mechanism to the Stop-and-Wait Protocol. Let us see how this protocol detects and corrects errors.

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

Since an ACK frame can also be corrupted and lost, it too needs redundancy bits and a sequence number. The ACK frame for this protocol has a sequence number field. In this protocol, the sender simply discards a corrupted ACK frame or ignores an out-of-order one.

Sequence Numbers

As we discussed, the protocol specifies that frames need to be numbered. This is done by using sequence numbers. A field is added to the data frame to hold the sequence number of that frame. One important consideration is the range of the sequence numbers. Since we want to minimize the frame size, we look for the smallest range that provides unambiguous communication. The sequence numbers of course can wrap around. For example, if we decide that the field is m bits long, the sequence numbers start from 0, go to 2m - 1, and then are repeated.

Let us reason out the range of sequence numbers we need. Assume we have used x as a sequence number; we only need to use x + 1 after that. There is no need for x + 2. To show this, assume that the sender has sent the frame numbered x. Three things can happen.

1. The frame arrives safe and sound at the receiver site; the receiver sends an acknowledgment. The acknowledgment arrives at the sender site, causing the sender to send the next frame numbered x + 1.

2. The frame arrives safe and sound at the receiver site; the receiver sends an acknowledgment, but the acknowledgment is corrupted or lost. The sender resends the frame (numbered x) after the time-out. Note that the frame here is a duplicate. The receiver can recognize this fact because it expects frame x + I but frame x was received.
3. The frame is corrupted or never arrives at the receiver site; the sender resends the frame (numbered x) after the time-out. We can see that there is a need for sequence numbers x and x + I because the receiver needs to distinguish between case 1 and case 2. But there is no need for a frame to be numbered x + 2. In case 1, the frame can be numbered x again because frames x and x + 1 are acknowledged and there is no ambiguity at either site. In cases 2 and 3, the new frame is x + I, not x + 2. If only x and x + 1 are needed, we can let x = 0 and x + I == 1. This means that the sequence is 0, I, 0, I, 0, and so on.

Acknowledgment Numbers

Since the sequence numbers must be suitable for both data frames and ACK frames, we use this convention: The acknowledgment numbers always announce the sequence number of the next frame expected by the receiver. For example, if frame 0 has arrived safe and sound, the receiver sends an ACK frame with acknowledgment 1 (meaning frame 1 is expected next). If frame 1 has arrived safe and sound, the receiver sends an ACK frame with acknowledgment 0 (meaning frame 0 is expected).

Design

Figure 9 shows the design of the Stop-and-Wait ARQ Protocol. The sending device keeps a copy of the last frame transmitted until it receives an acknowledgment for that frame. A data frames uses a seqNo (sequence number); an ACK frame uses an ackNo (acknowledgment number). The sender has a control variable, which we call Sn (sender, next frame to send), that holds the sequence number for the next frame to be sent (0 or 1).

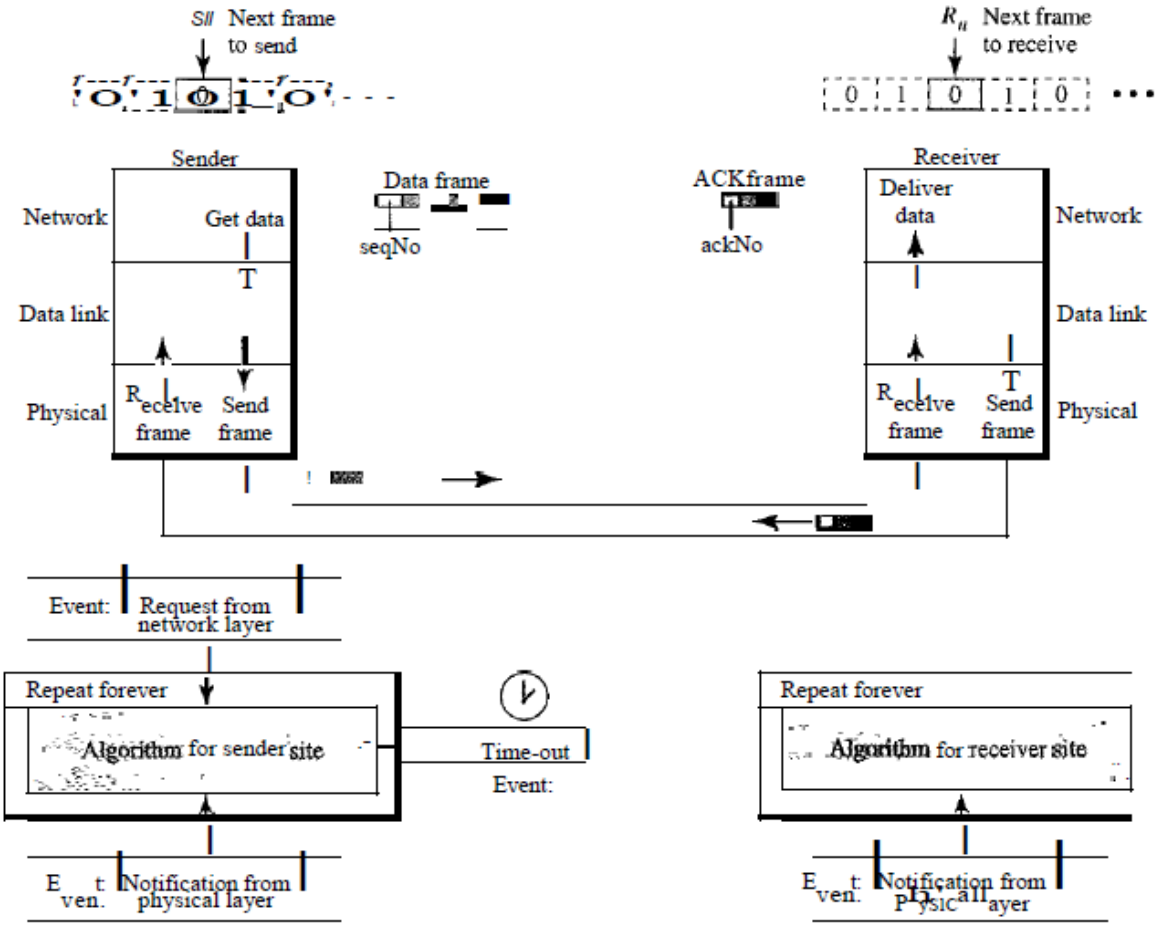

Figure 9. Design of the Stop-and-Wait ARQ Protocol

The receiver has a control variable, which we call Rn (receiver, next frame expected), that holds the number of the next frame expected. When a frame is sent, the value of Sn is incremented (modulo-2), which means if it is 0, it becomes 1 and vice versa. When a frame is received, the value of Rn is incremented (modulo-2), which means if it is 0, it becomes 1 and vice versa. Three events can happen at the sender site; one event can happen at the receiver site. Variable Sn points to the slot that matches the sequence number of the frame that has been sent, but not acknowledged; Rn points to the slot that matches the sequence number of the expected frame.

2.2.2 Go-Back-N Automatic Repeat Request

To improve the efficiency of transmission (filling the pipe), multiple frames must be in transition while waiting for acknowledgment. In other words, we need to let more than one frame be outstanding to keep the channel busy while the sender is waiting for acknowledgment. In this section, we discuss one protocol that can achieve this goal; The first is called Go-Back-N Automatic Repeat Request (the rationale for the name will become

clear later). In this protocol we can send several frames before receiving acknowledgments; we keep a copy of these frames until the acknowledgments arrive.

Sequence Numbers

Frames from a sending station are numbered sequentially. However, because we need to include the sequence number of each frame in the header, we need to set a limit. If the header of the frame allows m bits for the sequence number, the sequence numbers range from 0 to 2m - 1. For example, if m is 4, the only sequence numbers are 0 through 15 inclusive. However, we can repeat the sequence. So the sequence numbers are :

0, 1,2,3,4,5,6, 7,8,9, 10, 11, 12, 13, 14, 15,0, 1,2,3,4,5,6,7,8,9,10, 11, ...

In other words, the sequence numbers are modulo-2m. In the Go-Back-N Protocol, the sequence numbers are modulo 1!" where m is the size of the sequence number field in bits.

Sliding Window

In this protocol (and the next), the sliding window is an abstract concept that defines the range of sequence numbers that is the concern of the sender and receiver. In other words, the sender and receiver need to deal with only part of the possible sequence numbers. The range which is the concern of the sender is called the send sliding window; the range that is the concern of the receiver is called the receive sliding window. We discuss both here.

The send window is an imaginary box covering the sequence numbers of the data frames which can be in transit. In each window position, some of these sequence numbers define the frames that have been sent; others define those that can be sent. The maximum size of the window is 2m - 1 for reasons that we discuss later. In this chapter, we let the size be fixed and set to the maximum value, but we will see in future chapters that some protocols may have a variable window size. Figure 10 shows a sliding window of size 15 (m =4). The window at any time divides the possible sequence numbers into four regions. The first region, from the far left to the left wall of the window, defines the sequence

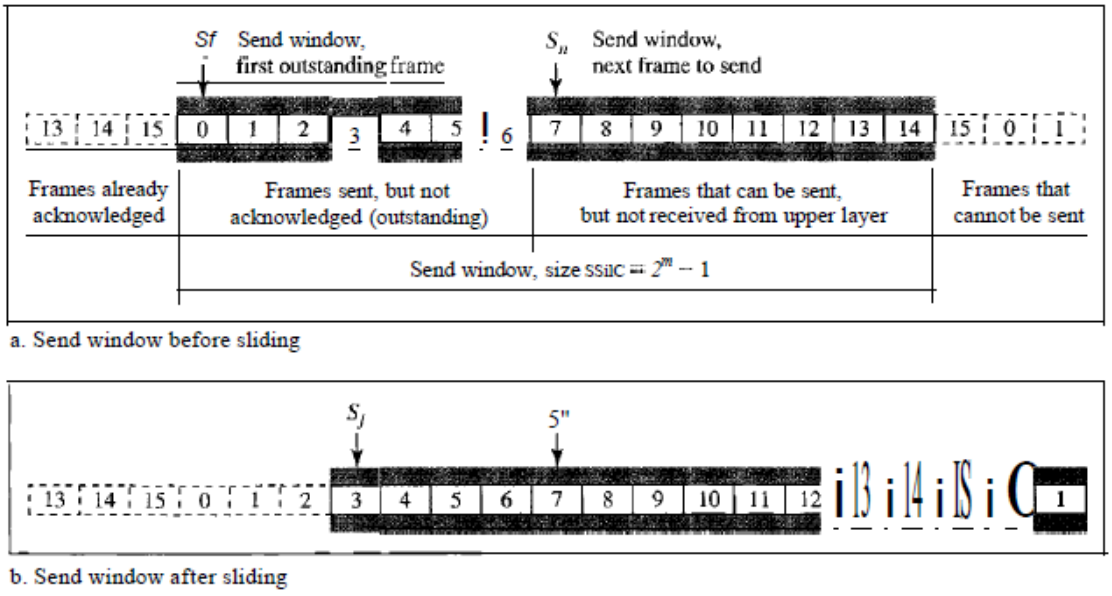

Figure 10. Send window for Go-Back-NARQ

numbers belonging to frames that are already acknowledged. The sender does not worry about these frames and keeps no copies of them. The second region, colored in Figure 9 a, defines the range of sequence numbers belonging to the frames that are sent and have an unknown status. The sender needs to wait to find out if these frames have been received or were lost. We call these outstanding frames. The third range, white in the figure, defines the range of sequence numbers for frames that can be sent; however, the corresponding data packets have not yet been received from the network layer. Finally, the fourth region defines sequence numbers that cannot be used until the window slides, as we see next.

The window itself is an abstraction; three variables define its size and location at any time. We call these variables Sf (send window, the first outstanding frame), Sn (send window, the next frame to be sent), and Ssize (send window, size). The variable Sf defines the sequence number of the first (oldest) outstanding frame. The variable Sn holds the sequence number that will be assigned to the next frame to be sent. Finally, the variable size defines the size of the window, which is fixed in our protocol.

In Figure 10.b shows how a send window can slide one or more slots to the right when an acknowledgment arrives from the other end. As we will see shortly, the

acknowledgments in this protocol are cumulative, meaning that more than one frame can be acknowledged by an ACK frame. In Figure 9b, frames 0, I, and 2 are acknowledged, so the window has slid to the right three slots. Note that the value of Sf is 3 because frame 3 is now the first outstanding frame.

The receive window makes sure that the correct data frames are received and that the correct acknowledgments are sent. The size of the receive window is always I. The

receiver is always looking for the arrival of a specific frame. Any frame arriving out of order is discarded and needs to be resent. Figure 11 shows the receive window.

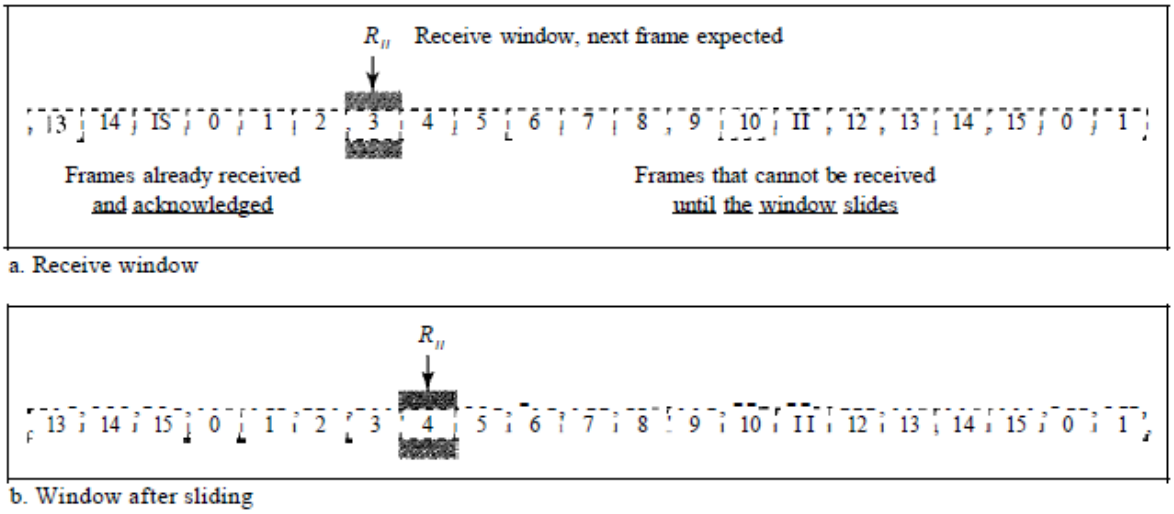

Figure 11. Receive window for Go-Back-NARQ

Note that we need only one variable Rn (receive window, next frame expected) to define this abstraction. The sequence numbers to the left of the window belong to the frames already received and acknowledged; the sequence numbers to the right of this window define the frames that cannot be received. Any received frame with a sequence number in these two regions is discarded. Only a frame with a sequence number matching the value of Rn is accepted and acknowledged. The receive window also slides, but only one slot at a time. When a correct frame is received (and a frame is received only one at a time), the window slides.

Timers

Although there can be a timer for each frame that is sent, in our protocol we use only one. The reason is that the timer for the first outstanding frame always expires first; we send all outstanding frames when this timer expires.

Acknowledgment

The receiver sends a positive acknowledgment if a frame has arrived safe and sound and in order. If a frame is damaged or is received out of order, the receiver is silent and will discard all subsequent frames until it receives the one it is expecting. The silence of the receiver causes the timer of the unacknowledged frame at the sender site to expire. This, in turn, causes the sender to go back and resend all frames, beginning with the one with the expired timer. The receiver does not have to acknowledge each frame received. It can send one cumulative acknowledgment for several frames.

Resending a Frame

When the timer expires, the sender resends all outstanding frames. For example, suppose the sender has already sent frame 6, but the timer for frame 3 expires. This means that frame 3 has not been acknowledged; the sender goes back and sends frames 3, 4,5, and 6 again. That is why the protocol is called Go-Back-N ARQ.

Design

Figure 12 shows the design for this protocol. As we can see, multiple frames can be in transit in the forward direction, and multiple acknowledgments in the reverse direction.

The idea is similar to Stop-and-Wait ARQ; the difference is that the send window allows us to have as many frames in transition as there are slots in the send window. Send Window Size We can now show why the size of the send window must be less than 2m. As an example, we choose m =2, which means the size of the window can be 2m - 1, or 3.

Figure 12 compares a window size of 3 against a window size of 4. If the size of the window is 3 (less than 22) and all three acknowledgments are lost, the frame timer expires and all three frames are resent. The receiver is now expecting frame 3, not frame 0, so the duplicate frame is correctly discarded. On the other hand, if the size of the window is 4 (equal to 22) and all acknowledgments are lost, the sender will send a duplicate of frame 0. However, this time the window of the receiver expects to receive frame 0, so it accepts frame 0, not as a duplicate, but as the first frame in the next cycle. This is an error.

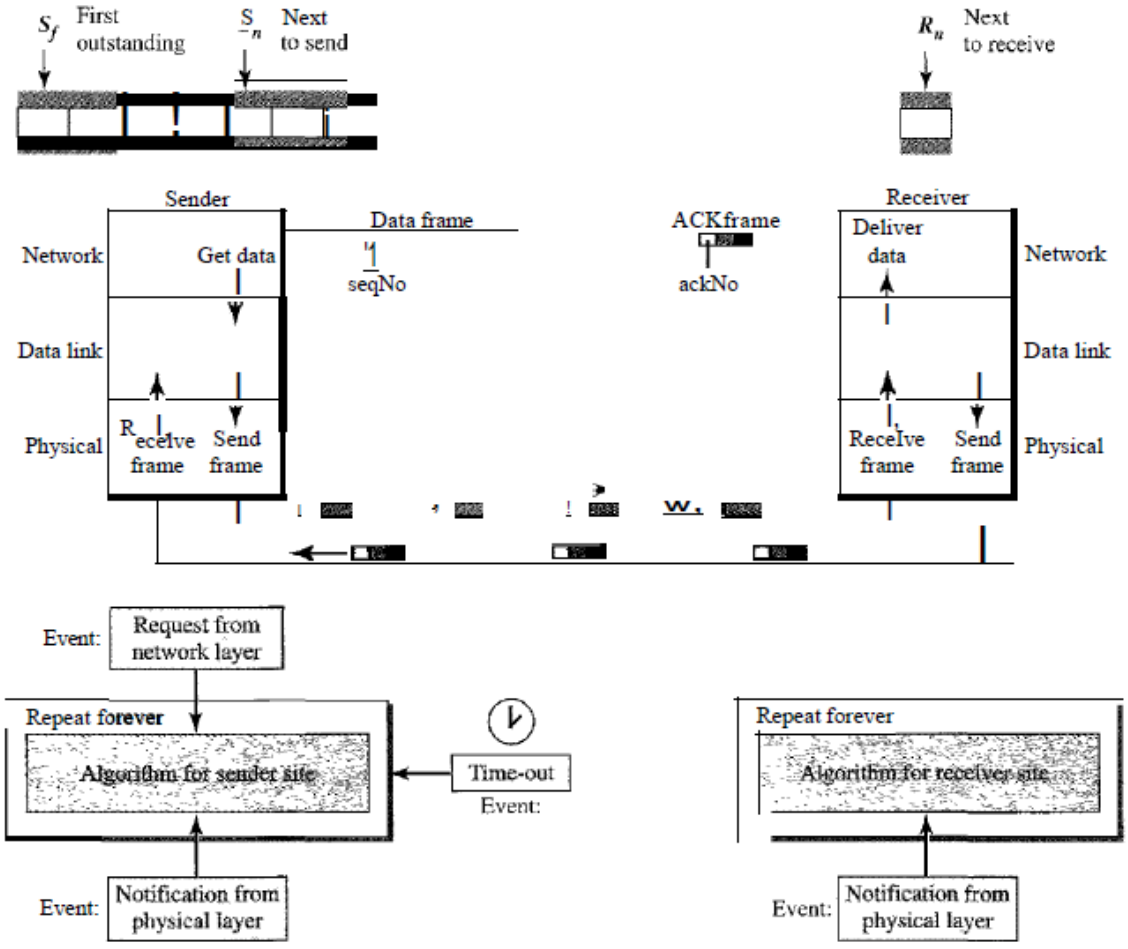

Figure 12 Design of Go-Back-NARQ

2.2.3 Selective Repeat Automatic Repeat Request

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

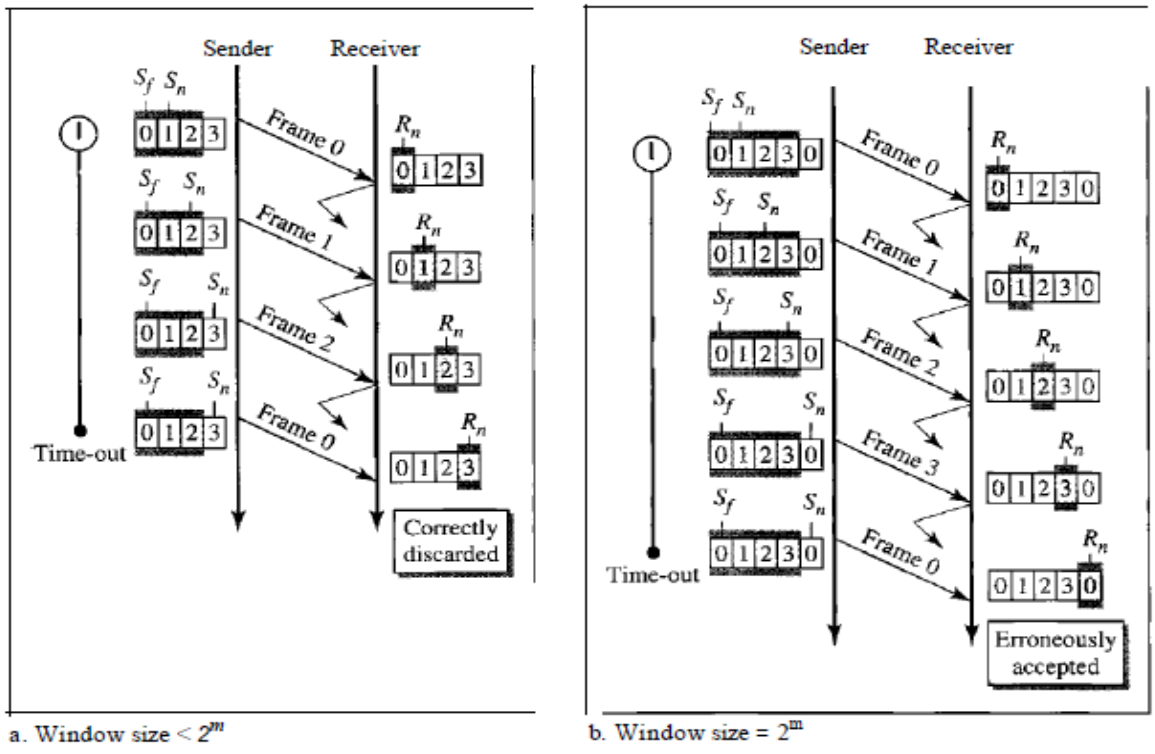

Figure 13. Window size for Go-Back-NARQ

Windows

The Selective Repeat Protocol also uses two windows: a send window and a receive window. However, there are differences between the windows in this protocol and the ones in Go-Back-N. First, the size of the send window is much smaller; it is 2m- I . The reason for this will be discussed later. Second, the receive window is the same size as the send window. The send window maximum size can be 2m- I . For example, if m = 4, the sequence numbers go from 0 to 15, but the size of the window is just 8 (it is 15 in the Go-Back-N Protocol). The smaller window size means less efficiency in filling the pipe, but the fact that there are fewer duplicate frames can compensate for this. The protocol uses the same variables as we discussed for Go-Back-N. We show the Selective Repeat send window in Figure 13 to emphasize the size.

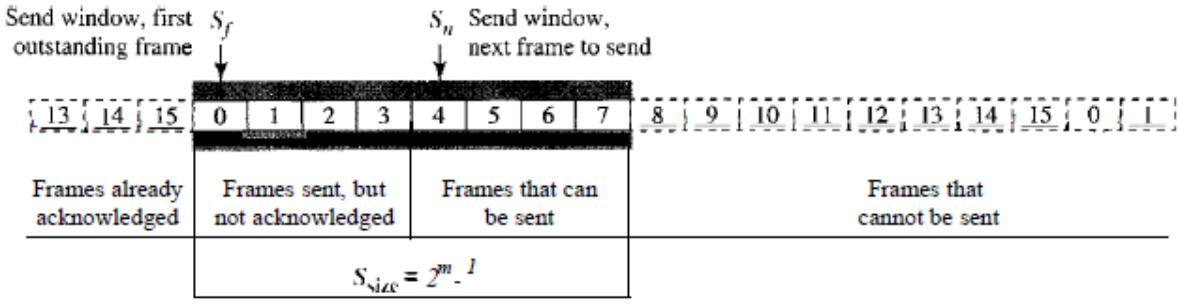

Figure 14: Send window for Selective Repeat ARQ

The receive window in Selective Repeat is totally different from the one in Go Back-N. First, the size of the receive window is the same as the size of the send window (2m- I). The Selective Repeat Protocol allows as many frames as the size of the receive window to arrive out of order and be kept until there is a set of in-order frames to be delivered to the network layer. Because the sizes of the send window and receive window are the same, all the frames in the send frame can arrive out of order and be stored until they can be delivered. We need, however, to mention that the receiver never delivers packets out of order to the network layer. Figure 15 shows the receive window in this

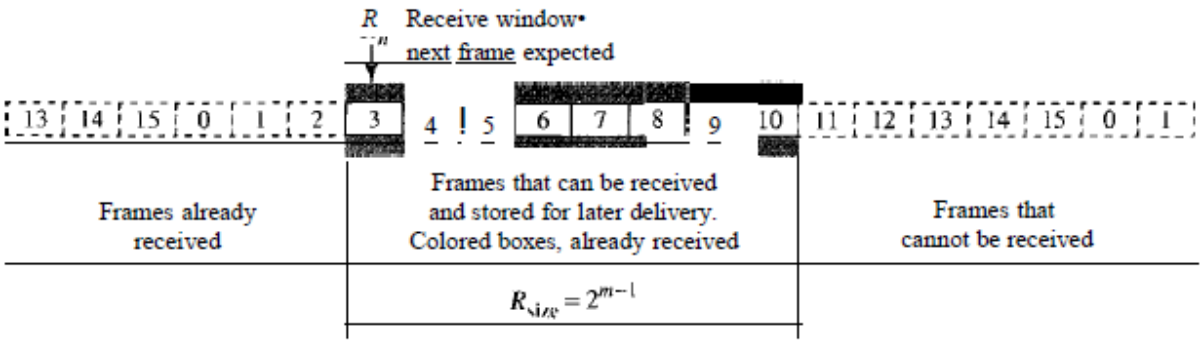

Figure 15 Receive window for Selective Repeat ARQ

Protocol. Those slots inside the window that are colored define frames that have arrived out of order and are waiting for their neighbors to arrive before delivery to the network layer.

Design

The design in this case is to some extent similar to the one we described for the Back-N, but more complicated, as shown in Figure 16

Window Sizes

We can now show why the size of the sender and receiver windows must be at most one half of 2m. For an example, we choose m = 2, which means the size of the window is 2m/2, or 2. Figure 16 compares a window size of 2 with a window size of 3. If the size of the window is 2 and all acknowledgments are lost, the timer for frame 0 expires and frame 0 is resent. However, the window of the receiver is now expecting frame 2, not frame 0, so this duplicate frame is correctly discarded. When the size of the window is 3 and all acknowledgments are lost, the sender sends a duplicate of frame O. However, this time, the window of the receiver expects to receive frame 0 (0 is part of the window), so it accepts frame 0, not as a duplicate, but as the first frame in the next cycle.

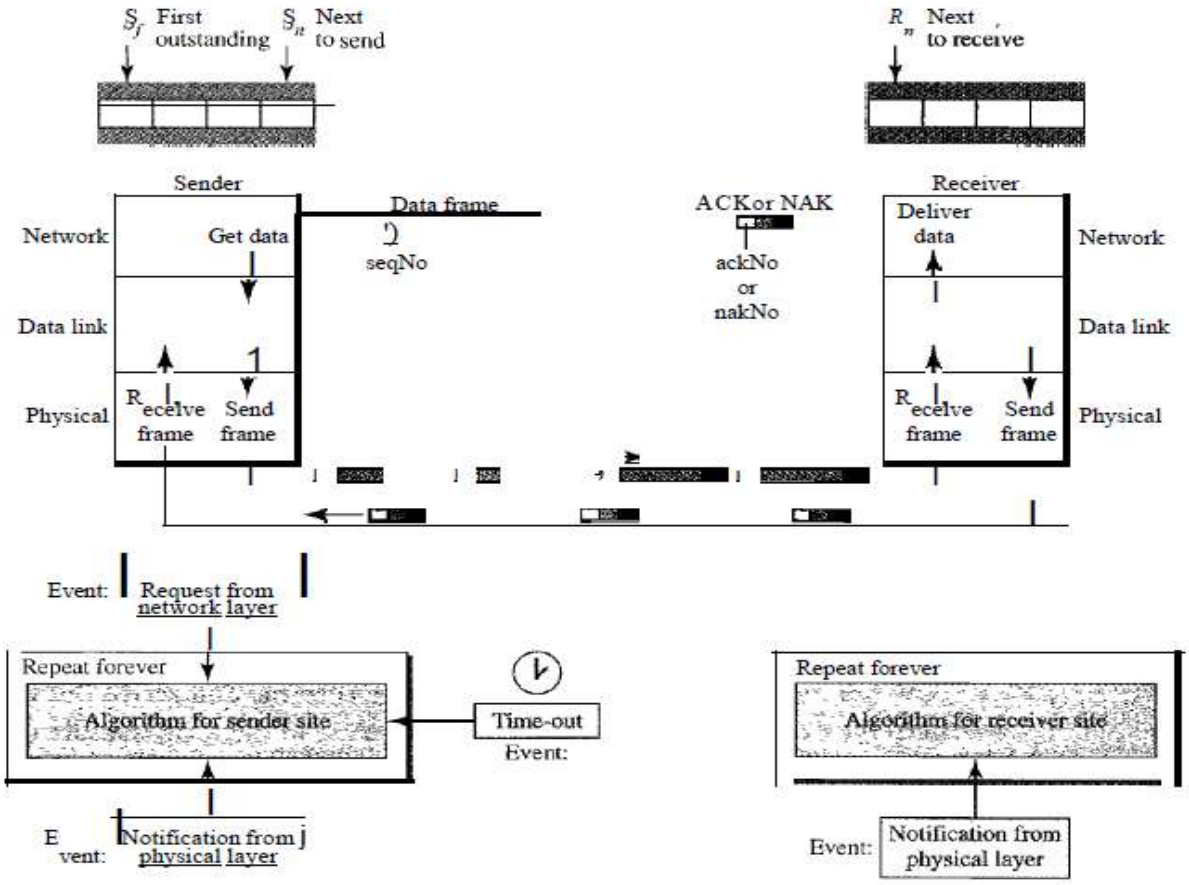

Figure 16 Design of Selective Repeat ARQ

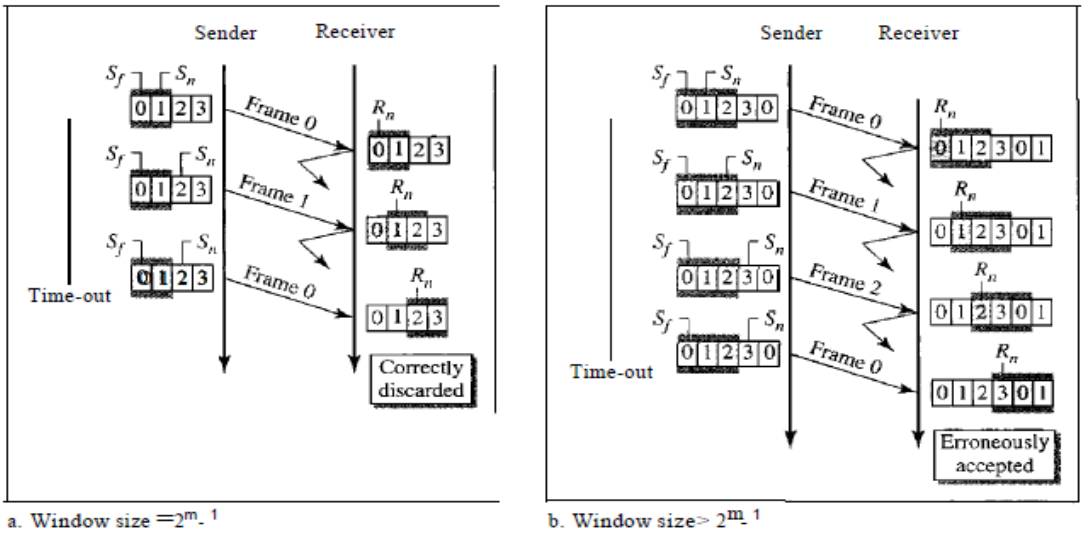

Figure 17 Selective Repeat ARQ, window size

3. Summary

In this lesson we have discussed the data link layer, which is a fundamental component of the OSI (Open Systems Interconnection) and TCP/IP networking models. It focuses on the reliable and error-free transmission of data over a physical medium, such as a wired or wireless connection. This layer is responsible for framing data into packets, detecting and correcting errors using techniques like checksums and cyclic redundancy checks, and managing access to the shared communication channel to avoid collisions. Ethernet, Wi-Fi, and Point-to-Point Protocol (PPP) are examples of data link layer protocols. Its role in ensuring efficient and accurate data transmission makes it a critical link in the communication process between devices on a network.

4. Self Check Exercise

1. What are the services provided by the data link layer?
2. Discuss error control and flow control mechanisms offered by the data link layer.
3. Explain the following:
    - Stop-and-Wait Automatic Repeat Request
    - Go-Back-N Automatic Repeat Request
    - Selective Repeat Automatic Repeat Request

5. Suggested Readings

1. Andrew S. Tannenbaum, “Computer Networks", 3rd Edition, Prentice Hall.
2. Behrouz A. Forouzan, “Data Communications & Networking”, Fourth edition, Tata McGraw Hills.
3. D.E. Cormer and D.L Stevens, "Internetworking with TCP/IP: Design implementations and Internals, “Vol II , Prentice Hall, 1990.
4. D.E. Cormer," Computer Networks and Internet”, 2nd Edition, Addison Wesley

   Publication, 2000.
5. D. Bertsekas and R.Gallagar, “Data Networks”, 2nd Edition, Prentice-Hall, 1992.

Lesson No. 1.6 **Author: Dr. Amandeep Kaur**
Converted into SLM by: Dr. Vishal Singh

MEDIUM ACCESS SUBLAYER AND LAN PROTOCOLS

Objectives

After reading this lesson you should be able to:

- Understand the concept Medium Access Sublayer.
- Understand Static and Dynamic Channel Allocation.
- Understand ALOHA protocol.
- Understand the concept of random access LAN protocols
- Understand the concept of collision free protocols and its various types
- Understand the concept of Limited-Contention Protocols and various protocols in this category

1.1 Introduction: Medium Access Sublayer

The transmission media (broadcast channel) used in LANs and MANs generally transmits frames from only one station at a time, although the media is generally shared by a number of stations. In order to overcome the difficulties which may arise through sharing of the transmission media, a **Medium Access Control** (MAC) protocol is necessary. A MAC protocol merely regulates how stations may access the medium in an orderly fashion for correct operation and also attempts to ensure that each station obtains a fair share of the channel. Broadcast channels are sometimes referred to as multiaccess channels or random access channels. The MAC protocol belongs to a sublayer of the data link layer.

The IEEE 802 specification for LANs breaks the data link layer into two sub layers: the LLC (Logical Link Control) and MAC (Media Access Control). The LLC provides a common interface point to the MAC layers, which specify the access method used. The uppermost sublayer is Logical Link Control (LLC). This sub layer multiplexes protocols running atop the data link layer, and optionally provides flow control, acknowledgment, and error recovery. The sublayer below it is Media Access Control (MAC). There are generally two forms of media access control: distributed and centralized. The MAC sub-layer acts as an interface between the Logical Link Control (LLC) sub layer and the network's physical layer. The MAC layer emulates a full-duplex logical communication channel in a multipoint network. This channel may provide unicast, multicast or broadcast communication service.

1.2 Static Channel Allocation for LANs and MANs

The traditional way of allocating a single channel, such as a telephone trunk, among multiple competing users is Frequency Division Multiplexing (FDM). If there are *N* users, the bandwidth is divided into *N* equal-sized portions each user being assigned one portion. Since each user has a private frequency band, there is no interference between users. When there are only a small and constant number of users, each of which has a heavy (buffered) load of traffic (e.g., carriers' switching offices), FDM is a simple and efficient allocation mechanism. However, when the number of senders is large and continuously varying or the traffic is bursty, FDM presents some problems. If the spectrum is cut up into *N* regions and fewer than *N* users are currently interested in communicating; a large piece of valuable spectrum will be wasted. If more than *N* users want to communicate, some of them will be denied permission for lack of bandwidth, even if some of the users who have been assigned a frequency band hardly ever transmit or receive anything.

However, even assuming that the number of users could somehow be held constant at *N*, dividing the single available channel into static subchannels is inherently inefficient. The basic problem is that when some users are quiescent, their bandwidth is simply lost. They are not using it, and no one else is allowed to use it either. Furthermore, in most computer systems, data traffic is extremely bursty (peak traffic to mean traffic ratios of 1000:1 are common). Consequently, most of the channels will be idle most of the time. Precisely the same arguments that apply to FDM also apply to time division multiplexing

(TDM). Each user is statically allocated every *N*th time slot. If a user does not use the allocated slot, it just lies fastest.

1.3 Dynamic Channel Allocation in LANs and MANs

Before considering the dynamic channel allocation methods we will consider five key assumptions underlying all these methods.

- **Station Model.** The model consists of *N* independent **stations** (e.g., computers, telephones, or personal communicators), each with a program or user that generates frames for transmission. Stations are sometimes called **terminals**. The probability of a frame being generated in an interval of length Δt is $\lambda\Delta t$, where λ is a constant (the arrival rate of new frames). Once a frame has been generated, the station is blocked and does nothing until the frame has been successfully transmitted. Thus stations are independent and work is generated at a constant rate.
- **Single Channel Assumption.** A single channel is available for all communication. All stations can transmit on it and all can receive from it. As far as the hardware is concerned, all stations are equivalent, although protocol software may assign priorities to them.
- **Collision Assumption.** If two frames are transmitted simultaneously, they overlap in time and the resulting signal is garbled. This event is called a **collision**. All stations can detect collisions. A collided frame must be transmitted again later. There are no errors other than those generated by collisions.
- The frame transmission time can be
    - **Continuous Time.** Frame transmission can begin at any instant. There is no master clock dividing time into discrete intervals.
    - **Slotted Time.** Time is divided into discrete intervals (slots). Frame transmissions always begin at the start of a slot. A slot may contain 0, 1, or more frames, corresponding to an idle slot, a successful transmission, or a collision, respectively.
- The stations may not have carrier sense or have carrier sense capability
    - **Carrier Sense.** Stations can tell if the channel is in use before trying to use it. If the channel is sensed as busy, no station will attempt to use it until it goes idle.
    - **No Carrier Sense.** Stations cannot sense the channel before trying to use it. They just go ahead and transmit. Only later can they determine whether the transmission was successful.

There are many algorithms for allocating multiple access channels. ALOHA is one such protocol which is discussed below

1.4 ALOHA

Many modern LANs evolved from a LAN known as **Aloha** which was one of the first primitive LANs to be developed. Aloha was packet based and used radio as its transmission medium. It was used for the first time in the Packet Radio System of the

University of Hawaii in 1970. It is a predecessor to the Ethernet. There are two versions of Aloha, Pure Aloha and Slotted Aloha.

ALOHA Protocol:

Aloha, also called the *Aloha method*, refers to a simple communications scheme in which each source transmit whenever there is data to send. If the frame successfully reaches the destination (receiver), the next frame is sent. If there is collusion colliding frames are destroyed and frames fail to be received at the destination. Under this protocol the sender can find out whether or not its frame was destroyed by listening to the channel, it is sent again.

Pure Aloha Protocol

With Pure Aloha, stations are allowed access to the channel whenever they have data to transmit (fig. 1). Because the threat of data collision exists, each station must either monitor its transmission on the rebroadcast or await an acknowledgment from the destination station. By comparing the transmitted packet with the received packet or by the lack of an acknowledgement, the transmitting station can determine the success of the transmitted packet. If the transmission was unsuccessful it is resent after a random amount of time to reduce the probability of re-collision.

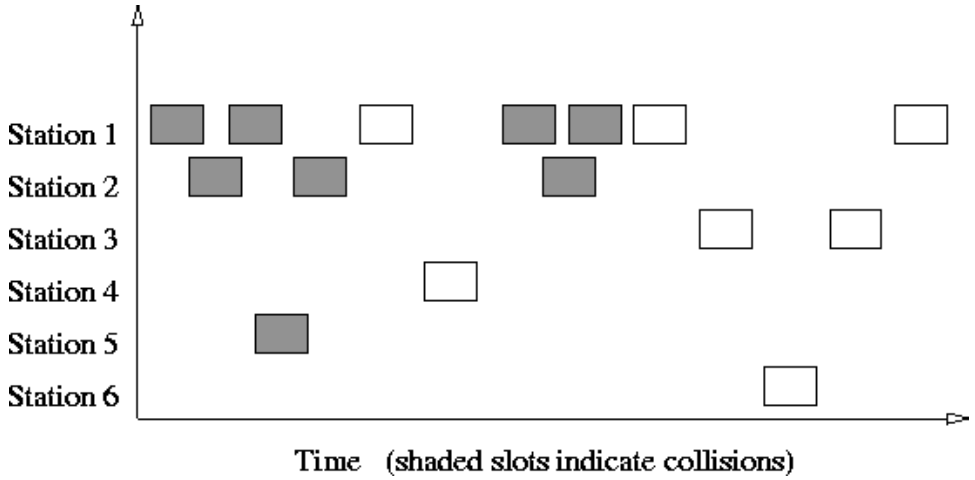

Figure 1: In pure ALOHA, frames are transmitted at completely arbitrary times

Whenever two frames try to occupy the channel at the same time there will be collision and both frames will be destroyed. Even if the first bit of a new frame overlaps with just the last bit of a frame which has almost finished, both frames will be destroyed.
Efficiency of ALOHA is given by the equation below:

$$S = Ge^{-2G}$$

Consider S as the mean new frames generated by users in a frame time (which is the amount of time needed to transmit the standard, fixed-length frame).

In addition to the new frames, the stations also generate retransmissions of frames that previously suffered collisions. Take G as the mean old and new combined frames generated per frame time. Clearly G > S. At low load (i.e., $N = 0$), there will be few collisions, hence few retransmissions, so $G = N$. A frame will not suffer a collision if no other frames

are sent within one frame time of its start. The maximum throughput occurs at $G = 0.5$, with $S = 1/2e$, which is about 0.184.

Advantages:

- Superior to fixed assignment when there is a large number of bursty stations.
- Adapts to varying number of stations.

Disadvantages:

- Theoretically proven throughput maximum of 18.4%.
- Requires queueing buffers for retransmission of packets.

Slotted Aloha

The Slotted Aloha protocol is a contention based protocol. The channel bandwidth is a continuous stream of slots whose length is the time necessary to transmit one packet fig 2. A station with a packet to send will transmit on the next available slot boundary. In the event of a collision, each station involved in the collision retransmits at some random time in order to reduce the possibility of recollision. Obviously the limits imposed which govern the random retransmission of the packet will have an effect on the delay associated with successful packet delivery. If the limit is too short, the probability of recollision is high. If the limit is too long the probability of recollision lessens but there is unnecessary delay in the retransmission.

Another important simulation characteristic of the Slotted Aloha protocol is the action which takes place on transmission of the packet. Methods include blocking (i.e. prohibiting packet generation) until verification of successful transmission occurs. This is known as "stop-and-wait". Another method known as "go-back-n" allows continual transmission of queued packets, but on the detection of a collision, will retransmit all packets from the point of the collision. This is done to preserve the order of the packets. In other cases queued packets are continually sent and only the packets involved in a collision are retransmitted. This is called "selective-repeat" and allows out of order transmission of packets.

Slotted Aloha Protocol

By making a small restriction in the transmission freedom of the individual stations, the throughput of the Aloha protocol can be doubled. Assuming constant length packets, transmission time is broken into slots equivalent to the transmission time of a single packet. Stations are only allowed to transmit at slot boundaries. When packets collide they will overlap completely instead of partially. This has the effect of doubling the efficiency of the Aloha protocol and has come to be known as Slotted Aloha. It is known through theoretical analysis of the Slotted ALOHA protocol that the maximum achievable throughput is or about 0.368 for a Poisson distributed network with uniform traffic.

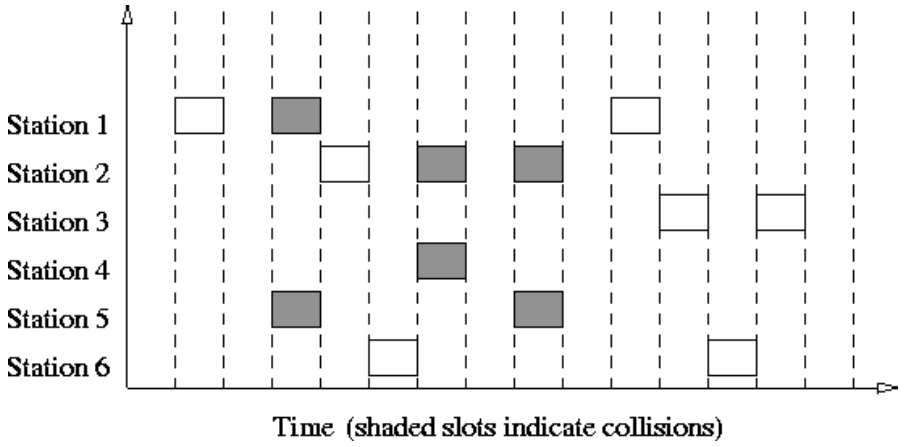

Figure 2 : Slotted Aloha Protocol

Advantages:

- Doubles the efficiency of Aloha.
- Adaptable to a changing station population.

Disadvantages:

- Theoretically proven throughput maximum of 36.8%.
- Requires queuing buffers for retransmission of packets.
- Synchronization required.

1.5 LAN Protocols A LAN is a high-speed data network that covers a relatively small geographic area. It typically connects workstations, personal computers, printers, servers and other devices. LANs offer computer users many advantages, including shared access to devices and applications, file exchange between connected users and communication between users via electronic mail and other applications.

With slotted ALOHA the best channel utilization that can be achieved is $1/e$ since in this case the stations transmit at will, without paying attention to what the other stations are doing, there are bound to be many collisions. A simple improvement which could be made to Aloha is to 'listen' to the presence of signal, prior to transmitting a frame. If there is no signal present the medium may be assumed to be idle and a station may then make an access. This is the **carrier sense** strategy. These networks can achieve a much better utilization than $1/e$. In this Lesson we will discuss some protocols for improving performance. The random access methods which we will study here have evolved from ALOHA protocol, which used a very simple procedure called multiple access (MA) fig. 1. The method was improved with the addition of a procedure that forces the station to sense the medium before transmitting. This was called carrier sense multiple access (CSMA). This method later evolved into two parallel methods: carrier senses multiple access with collision detection (CSMA/CD) and carrier sense multiple access with collision avoidance (CSMA/CA). CSMA/CD tells the station what to do when a collision is detected. CSMA/CA tries to avoid the collision.

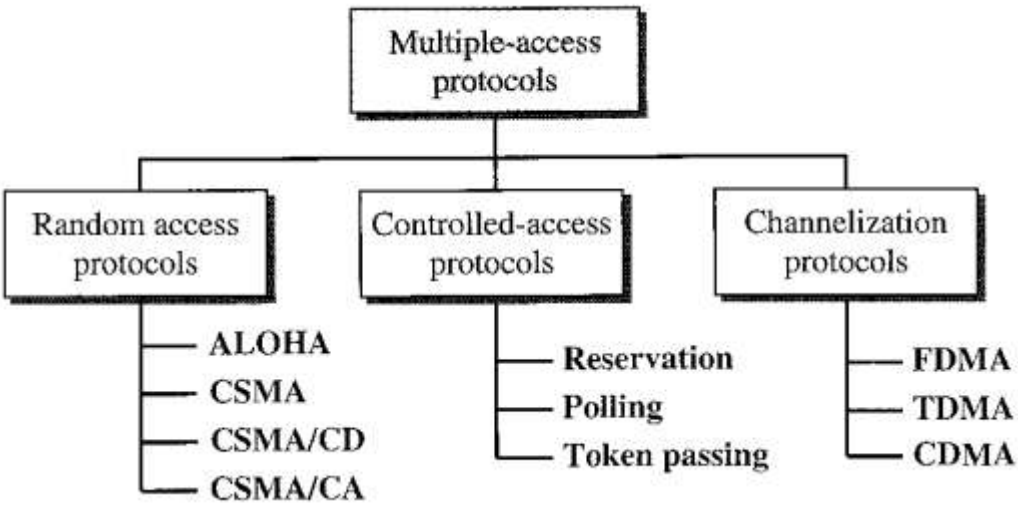

Figure 1: Multiple access protocols for LANs

1.6 Carrier Sense Multiple Access (CSMA)

To minimize the chance of collision and, therefore, increase the performance, the CSMA method was developed. The chance of collision can be reduced if a station senses the medium before trying to use it. Carrier sense multiple access (CSMA) requires that each station first listen to the medium (or check the state of the medium) before sending. In other words, CSMA is based on the principle "sense before transmit" or "listen before talk." CSMA can reduce the possibility of collision, but it cannot eliminate it.

The next section discusses various strategies or algorithms which, when used in an optimal manner, improve the performance of CSMA in terms of throughput compared with Aloha. These algorithms generally differ in terms of how they deal with a station which discovers the medium to be busy.

Persistence algorithms

The first carrier sense protocol that we will study here is called **1-persistent CSMA** (Carrier Sense Multiple Access). When a station has data to send, it first listens to the channel to see if anyone else is transmitting at that moment. If the channel is busy, the station waits until it becomes idle. When the station detects an idle channel, it transmits a frame. If a collision occurs, the station waits a random amount of time and starts all over again. The protocol is called 1-persistent because the station transmits with a probability of 1 when it finds the channel idle. The propagation delay has an important effect on the performance of the protocol. There is a small chance that just after a station begins sending, another station will become ready to send and sense the channel. If the first station's signal has not yet reached the second one, the latter will sense an idle channel and will also begin sending, resulting in a collision. The longer the propagation delay, the

more important this effect becomes, and the worse the performance of the protocol. Even if the propagation delay is zero, there will still be collisions. If two stations become ready in the middle of a third station's transmission, both will wait politely until the transmission ends and then both will begin transmitting exactly simultaneously, resulting in a collision. If they were not so impatient, there would be fewer collisions. Even so, this protocol is far better than pure ALOHA because both stations have the decency to desist from interfering with the third station's frame. Intuitively, this approach will lead to a higher performance than pure ALOHA. Exactly the same holds for slotted ALOHA

Nonpersistent CSMA

A second carrier sense protocol is **nonpersistent CSMA**. In this protocol, a conscious attempt is made to be less greedy than in the previous one. Before sending, a station senses the channel. If no one else is sending, the station begins doing so itself. However, if the channel is already in use, the station does not continually sense it for the purpose of seizing it immediately upon detecting the end of the previous transmission. Instead, it waits a random period of time and then repeats the algorithm. Consequently, this algorithm leads to better channel utilization but longer delays than 1-persistent CSMA.

P-Persistent CSMA

The last protocol is **p-persistent CSMA**. It applies to slotted channels and works as follows. When a station becomes ready to send, it senses the channel. If it is idle, it transmits with a probability p. With a probability $q = 1 - p$, it defers until the next slot. If that slot is also idle, it either transmits or defers again, with probabilities p and q. This process is repeated until either the frame has been transmitted or another station has begun transmitting. In the latter case, the unlucky station acts as if there had been a collision (i.e., it waits a random time and starts again).

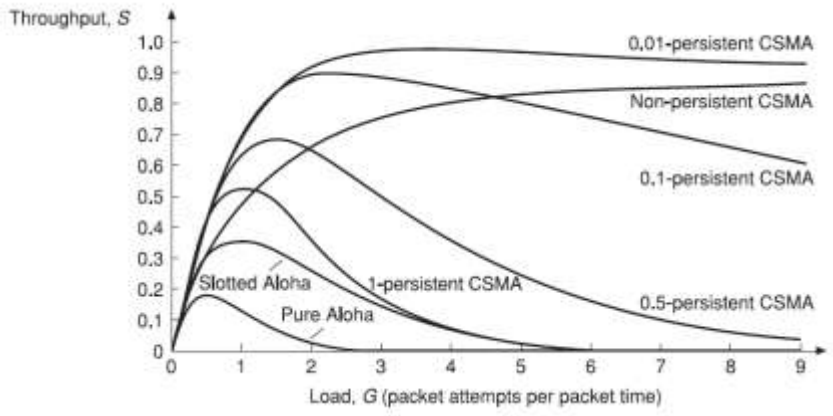

Figure 2 Throughput versus the load for various protocols is shown

1.7 CSMA with Collision Detection (CSMA/CD)

Persistent and nonpersistent CSMA protocols are clearly an improvement over ALOHA because they ensure that no station begins to transmit when it senses the channel busy. Another improvement is for stations to abort their transmissions as soon as they detect a collision. In other words, if two stations sense the channel to be idle and begin transmitting simultaneously, they will both detect the collision almost immediately. Rather than finish transmitting their frames, which are irretrievably garbled anyway, they should abruptly stop transmitting as soon as the collision is detected. Quickly terminating damaged frames saves time and bandwidth. This protocol, known as **CSMA/CD** (**CSMA with Collision Detection**) is widely used on LANs in the MAC sublayer. In particular, it is the basis of the popular Ethernet LAN

CSMA/CD, as well as many other LAN protocols, use the conceptual model of Fig. 3. At the point marked t_0, a station has finished transmitting its frame. Any other station having a frame to send may now attempt to do so. If two or more stations decide to transmit simultaneously, there will be a collision. Collisions can be detected by looking at the power or pulse width of the received signal and comparing it to the transmitted signal.

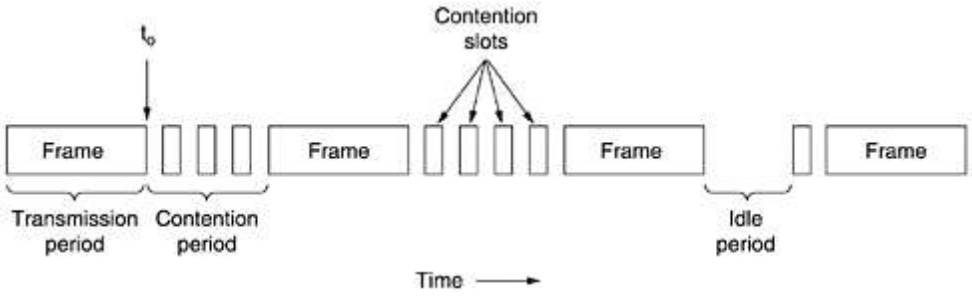

Figure 3. CSMA/CD can be in one of three states: contention, transmission, or idle.

After a station detects a collision, it aborts its transmission, waits a random period of time, and then tries again, assuming that no other station has started transmitting in the meantime. Therefore, our model for CSMA/CD will consist of alternating contention and transmission periods, with idle periods occurring when all stations are quiet (e.g., for lack of work). It is also worth noting that a sending station must continually monitor the channel, listening for noise bursts that might indicate a collision. For this reason, CSMA/CD with a single channel is inherently a half-duplex system. It is impossible for a station to transmit and receive frames at the same time because the receiving logic is in use, looking for collisions during every transmission.

The CSMA/CD algorithm may be summarized as follows:

1. Listen before talk.

2. If free, transmit and monitor transmission.

3. If busy, defer.

4. If a collision occurs during transmission, stop transmitting.

5. Send a jamming signal.

6. Random back-off.

7. Retry with LBT.

For CSMA/CD to work, we need a restriction on the frame size. Before sending the last bit of the frame, the sending station must detect a collision, if any and abort the transmission. This is so because the station, once the entire frame is sent, does not keep a copy of the frame and does not monitor the line for collision detection. Therefore, the frame transmission time T_{fr} must be at least two times the maximum propagation time τ. To understand the reason, let us think about the worst-case scenario. If the two stations involved in a collision are the maximum distance apart, the signal from the first takes time τ to reach the second and the effect of the collision takes another time τ to reach the first. So the requirement is that the first station must still be transmitting after 2τ. The concept becomes more clear from the figure 4.

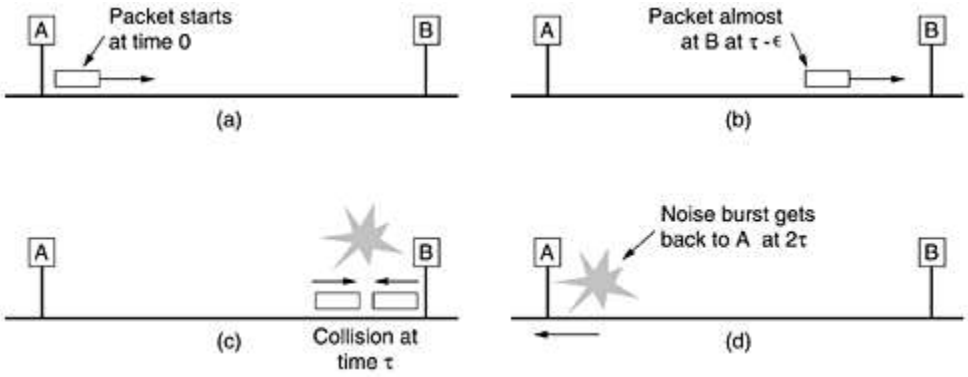

Figure 4. Collision detection can take as long as 2τ

1.8 Collision Free Protocol

Although collisions do not occur with CSMA/CD once a station has unambiguously captured the channel, they can still occur during the contention period. These collisions adversely affect the system performance, especially when the cable is long (i.e., large τ) and the frames are short. And CSMA/CD is not universally applicable. In this section, we will examine some protocols that resolve the contention for the channel without any collisions at all, not even during the contention period. Most of these are not currently used in major systems, but in a rapidly changing field, having some protocols with excellent properties available for future systems is often a good thing. In the protocols to be described, we

assume that there are exactly *N* stations, each with a unique address from 0 to *N* - 1 "wired" into it. It does not matter that some stations may be inactive part of the time. We also assume that propagation delay is negligible.

1.8.1 Bit-Map Protocol

In first collision-free protocol, the **basic bit-map method**, each contention period consists of exactly N slots fig 5. If station 0 has a frame to send, it transmits a 1 bit during the zeroth slot. No other station is allowed to transmit during this slot. Regardless of what station 0 does, station 1 gets the opportunity to transmit a 1 during slot 1, but only if it has a frame queued. In general, station j may announce that it has a frame to send by inserting a 1 bit into slot j. After all N slots have passed by, each station has complete knowledge of which stations wish to transmit. At that point, they begin transmitting in numerical order

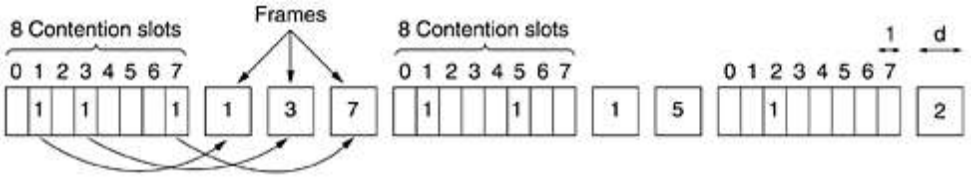

Figure 5 The basic bit-map protocol

Since everyone agrees on who goes next, there will never be any collisions. After the last ready station has transmitted its frame, an event all stations can easily monitor, another N bit contention period is begun. If a station becomes ready just after its bit slot has passed by, it is out of luck and must remain silent until every station has had a chance and the bit map has come around again. Protocols like this in which the desire to transmit is broadcast before the actual transmission are called **reservation protocols**.

Analyze the performance: Consider the situation from the point of view of a low-numbered station, such as 0 or 1. Typically, when it becomes ready to send, the "current" slot will be somewhere in the middle of the bit map. On average, the station will have to wait N/2 slots for the current scan to finish and another full N slots for the following scan to run to completion before it may begin transmitting. High-numbered stations are luckier. Generally, these will only have to wait half a scan (N/2 bit slots) before starting to transmit. High-numbered stations rarely have to wait for the next scan. Since low-numbered stations must wait on average 1.5N slots and high numbered stations must wait on average 0.5N slots, the mean for all stations is N slots. The channel efficiency at low load is easy to compute. The overhead per frame is N bits, and the amount of data is d bits, for an efficiency of d/(N + d). At high load, when all the stations have something to send all the time, the N bit contention period is prorated over N frames, yielding an overhead of only 1 bit per frame or an efficiency of d/(d + 1). The mean delay for a frame is equal to the sum of the time it queues inside its station, plus an additional N(d + 1)/2 once it gets to the head of its internal queue.

1.8.2 Broadcast Recognition with Alternating Priorities (BRAP)

The basic bit-map protocol has several drawbacks; the major drawback is the asymmetry with respect to station numbers: higher numbered stations get better service than the lower numbered ones. Another drawback is that under conditions of light load a station must always wait for the current scan to be finished before it may transmit. BRAP protocol eliminates both these problems.

In BRAP as soon as a station inserts a 1 bit into its slot, it begins transmission of its frame immediately thereafter. In addition instead of starting the bit scan with station 0 each time, it is started with station following the one that just transmitted. The permission to send rotates among the stations in a round robin fashion. The working of BRAP is explained through figure 6.

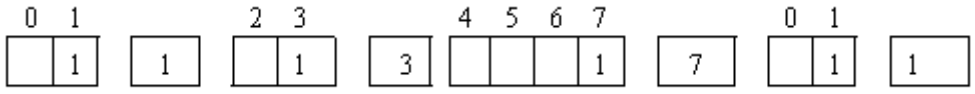

Figure 6 BRAP protocol working

1.8.3 The Multi-Level Multi-Access Protocol

The problem with BRAP is not the channel utilization, which is excellent in the case of high load, but with the delay when the system is lightly loaded. When no stations are ready there are no data frames, and the N-bit header just go on and on till some station inserts a 1 bit in its bit slot. On the average a station will have to wait N/2 bit slots before it may begin sending. In this method, a station announces that it wants to send by broadcasting its address in a particular format. If only one station attempts to transmit during a frame slot, it uses the 30-bit header to announce itself and then sends the frame. The trouble arises when more than one station tries to insert their addresses into the same header. To disambiguate all the addresses, the stations behave as follows:

The first decade in every frame slot corresponds to the hundred places in the station number. After the first decade is finished. After the first decade is finished, stations that have not transmitted a bit must remain silent until all the stations that did set a bit have transmitted their data. Call the highest occupied bit position in the first decade x. In the second decade, all stations with x as their leading digit announce their tens's place. Call the highest occupied bit here as y. In the third decade, all the stations whose addresses begin with xy may set the bit corresponding to their last digits. There are at most 10 of them. Consider the following example:

Five stations with addresses 122, 125, 705, 722, and 725 want to transmit data. Here x=7 and y=2. Finally the data is sent in numerical order of the station addresses. The Figure 7 shows how stations are recognized and put into numerical order:

9	8	7	6	5	4	3	2	1	0	
0	0	1	0	0	0	0	1	0	0	Decade 0 (122, 125, 705, 722, 725 send)
0	0	0	0	0	0	0	1	0	1	Decade 1 (705, 722, 725 send)
0	0	0	0	1	0	0	1	0	0	Decade 2 (722, 725 send)
0	0	0	0	1	0	0	0	0	0	Decade 3 (705 send)
0	0	0	0	0	0	0	1	0	0	Decade 4 (122, 125 send)
0	0	0	0	1	0	0	1	0	0	Decade 5 (122, 125 send)

Figure 7: shows how MLMA recognizes the stations

1.8.4 Binary Countdown

A problem with the basic bit-map protocol is that the overhead is 1 bit per station, so it does not scale well to networks with thousands of stations. We can do better than that by using binary station addresses. A station wanting to use the channel now broadcasts its address as a binary bit string, starting with the high-order bit. All addresses are assumed to be the same length. The bits in each address position from different stations are BOOLEAN ORed together. We will call this protocol **binary countdown**. It was used in Datakit (Fraser, 1987). It implicitly assumes that the transmission delays are negligible so that all stations see asserted bits essentially instantaneously. To avoid conflicts, an arbitration rule must be applied: as soon as a station sees that a high order bit position that is 0 in its address has been overwritten with a 1, it gives up. For example, if stations 0010, 0100, 1001, and 1010 are all trying to get the channel, in the first bit time the stations transmit 0, 0, 1, and 1, respectively. These are ORed together to form a 1. Stations 0010 and 0100 see the 1 and know that a higher-numbered station is

competing for the channel, so they give up for the current round. Stations 1001 and 1010 continue. The next bit is 0, and both stations continue. The next bit is 1, so station 1001 gives up. The winner is station 1010 because it has the highest address. After winning the bidding, it may now transmit a frame, after which another bidding cycle starts. The protocol is illustrated in Figure 8. It has the property that higher-numbered stations have a higher priority than lower numbered stations, which may be either good or bad, depending on the context.

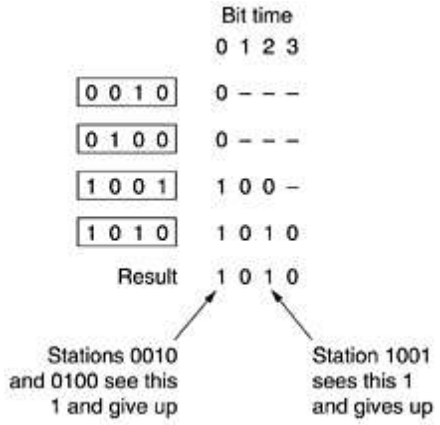

Figure 8. The binary countdown protocol. A dash indicates silence

The channel efficiency of this method is d/(d + log2 N). If, however, the frame format has been cleverly chosen so that the sender's address is the first field in the frame, even these log2 N bits are not wasted and the efficiency is 100 percent.

1.9 Limited-Contention Protocols

We have now considered two basic strategies for channel acquisition in a cable network: contention, as in CSMA and collision-free methods. Each strategy can be rated as to how well it does with respect to the two important performance measures, delay at low load and channel efficiency at high load. Under conditions of light load, contention (i.e., pure or slotted ALOHA) is preferable due to its low delay. As the load increases, contention becomes increasingly less attractive, because the overhead associated with channel arbitration becomes greater. Just the reverse is true for the collision-free protocols. At low load, they have high delay, but as the load increases, the channel efficiency improves rather than gets worse as it does for contention protocols. Obviously, it would be nice if we could combine the best properties of the contention and collision-free protocols, arriving at a new protocol that used contention at low load to provide low delay, but used a collision-free technique at high load to provide good channel efficiency. Such protocols, which we will call **limited-contention protocols**, do, in fact, exist, and will conclude our study of carrier sense networks. Up to now the only contention protocols we have studied have

been symmetric, that is, each station attempts to acquire the channel with some probability, p, with all stations using the same p. Interestingly enough, the overall system performance can sometimes be improved by using a protocol that assigns different probabilities to different stations. Before looking at the asymmetric protocols, let us quickly review the performance of the symmetric case. Suppose that k stations are contending for channel access. Each has a probability p of transmitting during each slot. The probability that some station successfully acquires the channel during a given slot is then kp(1 - p)k - 1. To find the optimal value of p, we differentiate with respect to p, set the result to zero, and solve for p. Doing so, we find that the best value of p is 1/k. Substituting p = 1/k, we get P_r (success with optimal p)

$$P_r = \left(\frac{k-1}{k}\right)^{k-1}$$

For small numbers of stations, the chances of success are good, but as soon as the number of stations reaches even five, the probability has dropped close to its asymptotic value of 1/e. thus the probability of some station acquiring the station can be increased only by decreasing the amount of competition. The limited contention protocols do precisely that.

1.9.1 The Adaptive Tree Walk Protocol

One particularly simple way of performing the necessary assignment is to use the algorithm devised by the U.S. Army for testing soldiers for syphilis during World War II (Dorfman, 1943). In short, the Army took a blood sample from N soldiers. A portion of each sample was poured into a single test tube. This mixed sample was then tested for antibodies. If none were found, all the soldiers in the group were declared healthy. If antibodies were present, two new mixed samples were prepared, one from soldiers 1 through N/2 and one from the rest. The process was repeated recursively until the infected soldiers were determined. For the computerized version of this algorithm (Capetanakis, 1979), it is convenient to think of the stations as the leaves of a binary tree, as illustrated in Fig. 9. In the first contention slot following a successful frame transmission, slot 0, all stations are permitted to try to acquire the channel. If one of them does so, fine. If there is a collision, then during slot 1 only those stations falling under node 2 in the tree may compete. If one of them acquires the channel, the slot following the frame is reserved for those stations under node 3. If, on the other hand, two or more stations under node 2 want to transmit, there will be a collision during slot 1, in which case it is node 4's turn during slot 2.

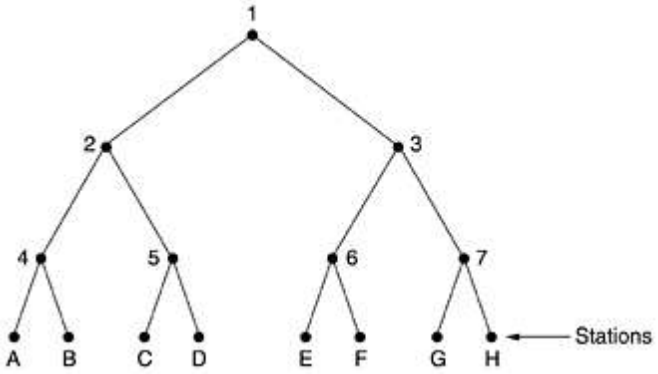

Figure 9. The tree for eight stations

In essence, if a collision occurs during slot 0, the entire tree is searched, depth first, to locate all ready stations. Each bit slot is associated with some particular node in the tree. If a collision occurs, the search continues recursively with the node's left and right children. If a bit slot is idle or if only one station transmits in it, the searching of its node can stop because all ready stations have been located. (Were there more than one, there would have been a collision.) When the load on the system is heavy, it is hardly worth the effort to dedicate slot 0 to node 1, because that makes sense only in the unlikely event that precisely one station has a frame to send. Similarly, one could argue that nodes 2 and 3 should be skipped as well for the same reason. Put in more general terms, at what level in the tree should the search begin? Clearly, the heavier the load, the farther down the tree the search should begin. We will assume that each station has a good estimate of the number of ready stations, q, for example, from monitoring recent traffic. To proceed, let us number the levels of the tree from the top, with node 1 in Fig. 9 at level 0, nodes 2 and 3 at level 1, etc. Notice that each node at level i has a fraction 2-i of the stations below it. If the q ready stations are uniformly distributed, the expected number of them below a specific node at level i is just 2-iq. Intuitively, we would expect the optimal level to begin searching the tree as the one at which the mean number of contending stations per slot is 1, that is, the level at which 2-iq = 1. Solving this equation, we find that i = log2 q. Numerous improvements to the basic algorithm have been discovered and are discussed in some detail by Bertsekas and Gallager (1992).

1.9.2 Urn Protocol

This protocol is similar to the tree walk protocol, but it uses an urn rather than a tree as its basis. Like the tree walk protocol, it limits the number of stations which are allowed to transmit during each slot in such a way as to maximize the probability of getting exactly one ready station per contention slot. In this protocol an analogy is made between the stations and the balls in an urn.

- Green ball corresponds to stations which are ready
- Red ball corresponds to stations which do not have a frame to send

- The probability of selecting exactly x green balls if we withdraw n balls without replacement is:

$$p = \frac{\binom{k}{x}\binom{N-k}{n-x}}{\binom{N}{n}}$$

The first factor in the numerator is the number of ways of selecting x green balls from among the k green balls in the urn. The second term in the numerator is the number of ways of selecting n-x red balls from among the N-k red balls in the urn. The denominator is the number of ways of selecting n balls from the N balls in the urn.

Here we are interested in the probability of drawing exactly one green ball, since that is the only way a successful transmission can occur. When x=1, the probability of success is maximized by choosing n=N/k. The mean number of green balls in the sample is equal to the sample size 'n', times the probability that a given ball is green k/N.

After determining what n should be, the next part relates to choosing the stations. The decision is made in a distributed way to which all stations agree. Several methods had been proposed in the literature. One method is outlined below: The stations are arranged in numerical order around a hypothetical circle. A window of size n rotates around the circle. During each slot those stations inside the window are given permission to send. If there was successful transmission or no transmission at all, the window is advanced n positions. If there was a collision, the window is shrunk back to half its size and the process is repeated until the collision ceases.

Now let us consider how the network works under the following two conditions:

Light Load

If the ready stations is one or fewer, the window size is N. In other words, the window will go all the way around and all stations will be allowed to send at will. Under light load its behavior is similar to slotted ALOHA protocol.

Heavy Load

Now consider if k=2, n will be N/2 and the stations will be partitioned into two groups, with half the stations operating under slotted ALOHA in the odd slots and the other half operating the same way in the even slots. Finally if k>N/2, the sample size will be one. During each slot exactly one station will be given permission to send so there will be no collisions. The position of the lucky station will rotate around the circle. In this case the system becomes identical to synchronous time division multiplexing.

All the limited-contention protocols assume that stations have an estimate of the number of stations wanting to transmit.

1.10 Summary

In this lesson we have discussed the MAC (Media Access Control) layer, which is a sublayer within the data link layer of networking models like OSI and TCP/IP. It controls how devices access and transmit data on a shared communication medium, such as a local area network (LAN). The MAC layer handles addressing, frame synchronization and collision avoidance or resolution in networks using shared access methods like Ethernet. It assigns unique MAC addresses to network interface cards, facilitating data delivery to specific devices within the same network segment. By managing access to the transmission medium, the MAC layer ensures efficient and orderly data communication among devices. We have studied various collision free protocols and its types and Limited-Contention Protocols also.

1.11 Self Check Exercise

1. MAC sublayer belongs to which layer of OSI reference model. What is its main function?
2. What are the five key assumptions underlying mostly all dynamic channel allocation methods?
3. Explain the working of Pure ALOHA and what its main disadvantages are?
4. What is meant by multi access protocols?
5. Explain CSMA protocol and bring out how it is different from CSMA/CD?
6. Write a note on any one protocol that resolve the contention for the n channel without any collisions at all?
7. Explain Multi-Level Multi-Access Protocol with the help of an example?
8. How does Urn Protocol behave when the load is light and when the load is heavy? Explain.
10. How Bit-map Protocol is different from Binary countdown?

1.12 Suggested Readings

1. Andrew S. Tanenbaum, Computer Networks, Prentice Hall India, Third Edition.
2. Forouzan, Data Communication and Networking, Tata McGraw Hill
3. D.E. Cormer and D.L. Stevens, " Inter-networking with TCP-IP: Design, Implementation and Internals," Vol II, Prentice Hall, 1990.
4. D.E. Cormer, " Computer Networks and Internet," 2nd Edition, Addison Wesley, 2000.
5. D. Bertsekas and R. Gallagar, Data Networks,"2nd Edition, Prentice Hall, 1992.
6. Larry L. Peterson and Bruce S. Davie, "Computer Networks: A Systems Approach," 3rd Edition, Morgan Kaufmann Series

Lesson No. 1.7 **Author: Dr. Dilraj Singh**
Converted into SLM by: Dr. Vishal Singh

IEEE 802 STANDARDS

Objectives

In this lesson we will discuss various IEEE standards.

1. IEEE STANDARDS

In 1985, the Computer Society of the IEEE started a project, called Project 802, to set standards to enable intercommunication among equipment from a variety of manufacturers. Project 802 does not seek to replace any part of the OSI or the Internet model. Instead, it is a way of specifying functions of the physical layer and the data link layer of major LAN protocols. The standard was adopted by the American National Standards Institute (ANSI). In 1987, the International Organization for Standardization (ISO) also approved it as an international standard under the designation ISO 8802. The relationship of the 802 Standard to the traditional OSI model is shown in Figure 1.

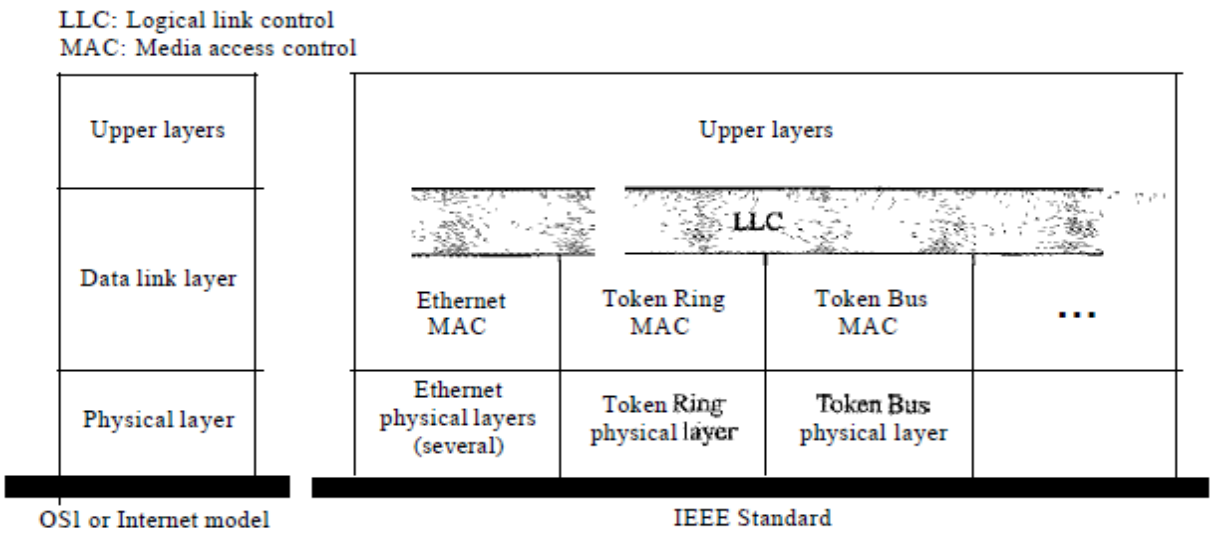

Figure 1: IEEE standard for LANs

The IEEE has subdivided the data link layer into two sub layers: logical link control (LLC) and media access control (MAC). IEEE has also created several physical layer standards for different LAN protocols.

2. IEEE 802.3

Ethernet and IEEE 802.3 specify similar technologies. Both are CSMA/CD LANs. Stations on a CSMA/CD LAN can access the network at any time. Before sending data, CSMA/CD stations "listen" to the network to see if it is already in use. If it is, the station wishing to transmit waits. If the network is not in use, the station transmits. A collision occurs when two stations listen for network traffic, "hear" none, and transmit simultaneously. In this case, both transmissions are damaged, and the stations must retransmit at some later time. Backoff algorithms determine when the colliding stations retransmit. CSMA/CD stations can detect collisions, so they know when they must retransmit. Both Ethernet and IEEE 802.3 LANs are broadcast networks. In other words, all stations see all frames, regardless of whether they represent an intended destination. Each station must examine received frames to determine if the station is a destination. If so, the frame is passed to a higher protocol layer for appropriate processing. Differences between Ethernet and IEEE 802.3 LANs are subtle. Ethernet provides services corresponding to Layers 1 and 2 of the OSI reference model, while IEEE 802.3 specifies the physical layer (Layer 1) and the channel-access portion of the link layer (Layer 2), but does not define a logical link control protocol. Both Ethernet and IEEE 802.3 are implemented in hardware. Typically, the physical manifestation of these protocols is either an interface card in a host computer or circuitry on a primary circuit board within a host computer.

Physical Connections IEEE 802.3 specifies several different physical layers, whereas Ethernet defines only one. Each IEEE 802.3 physical layer protocol has a name

that summarizes its characteristics. The coded components of an IEEE 802.3 physical-layer name are shown in Figure 2.

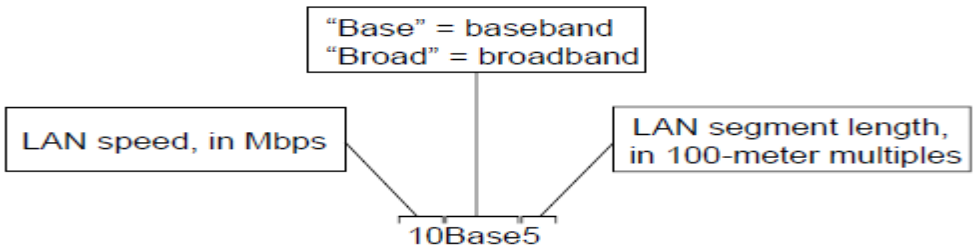

Figure 2: IEEE 802.3 Physical-Layer Name Components

Table 1. IEEE 802.3 Characteristics

		IEEE 802.3 Values				
Characteristic	**Ethernet Value**	**10Base5**	**10Base2**	**1Base5**	**10BaseT**	**10Broad36**
Data rate (Mbps)	10	10	10	1	10	10
Signaling method	Baseband	Baseband	Baseband	Baseband	Baseband	Broadband
Maximum segment length (m)	500	500	185	250	100 Unshielded twisted-pair wire	1800
Media	50-ohm coax (thick)	50-ohm coax (thick)	50-ohm coax (thin)	Unshielded twisted-pair wire	Unshielded twisted-pair wire	75-ohm coax
Topology	Bus	Bus	Bus	Star	Star	Bus

A summary of Ethernet Version 2 and IEEE 802.3 characteristics appears in Table 1. Ethernet is most similar to IEEE 802.3 10Base5. Both of these protocols specify a bus topology network with a connecting cable between the end stations and the actual network medium. In the case of Ethernet, that cable is called a transceiver cable. The transceiver cable connects to a transceiver device attached to the physical network medium. The IEEE 802.3 configuration is much the same, except that the connecting cable is referred to as an attachment unit interface (AUI), and the transceiver is called a medium attachment unit (MAU). In both cases, the connecting cable attaches to an interface board (or interface circuitry) within the end station. Frame Formats Ethernet and IEEE 802.3 frame formats are shown in Figure 3.

Both Ethernet and IEEE 802.3 frames begin with an alternating pattern of ones and zeros called a preamble. The preamble tells receiving stations that a frame is coming. The byte before the destination address in both an Ethernet and a IEEE 802.3 frame is a start-of-frame (SOF) delimiter. This byte ends with two consecutive one bits, which serve to synchronize the frame reception portions of all stations on the LAN. Immediately following the preamble in both Ethernet and IEEE 802.3 LANs are the destination and source address fields. Both Ethernet and IEEE 802.3 addresses are 6 bytes long. Addresses are contained in hardware on the Ethernet and IEEE 802.3 interface cards. The first 3 bytes of the addresses are specified by the IEEE on a vendor-dependent basis, while the last 3 bytes are specified by the Ethernet or IEEE 802.3 vendor.

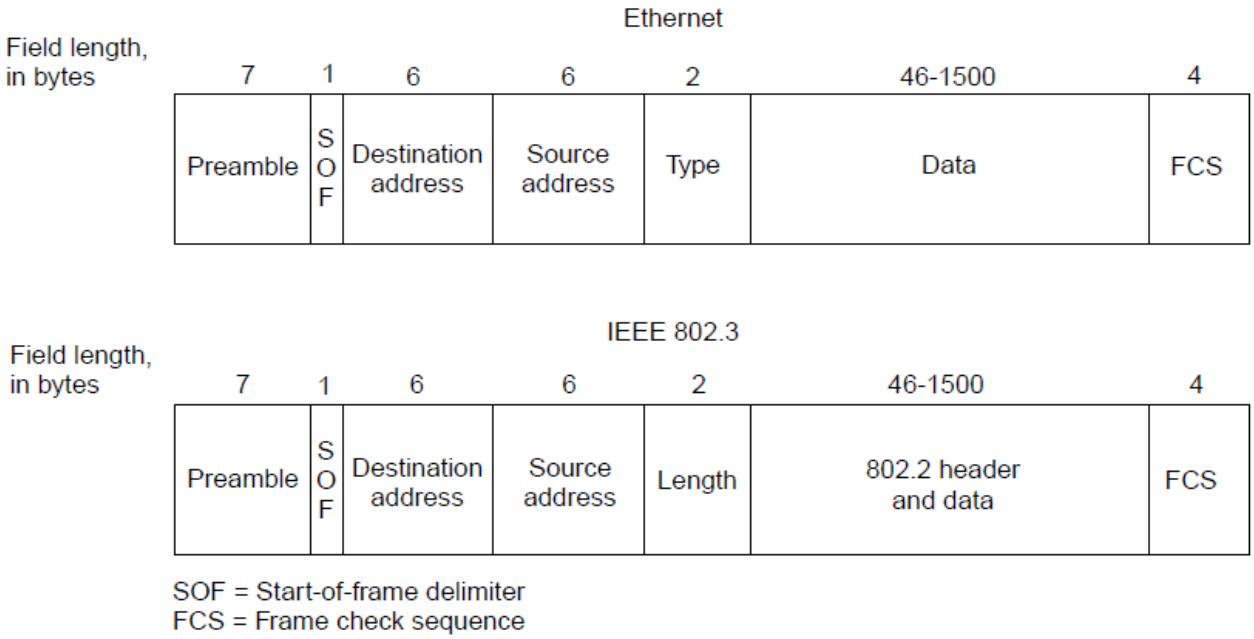

Figure 3: Ethernet and IEEE 802.3 Frame Formats

The source address is always a unicast (single node) address, while the destination address may be unicast, multicast (group), or broadcast (all nodes). In Ethernet frames, the 2-byte field following the source address is a type field. This field specifies the upper-layer protocol to receive the data after Ethernet processing is complete. In IEEE 802.3 frames, the 2-byte field following the source address is a length field, which indicates the number of bytes of data that follow this field and precede the frame check sequence (FCS) field. Following the type/length field is the actual data contained in the frame. After physical-layer and link-layer processing is complete, this data will eventually be sent to an upper-layer protocol. In the case of Ethernet, the upper-layer protocol is identified in the type field. In the case of IEEE 802.3, the upper-layer protocol must be defined within the data portion of the frame, if at all. If data in the frame is insufficient to fill the frame to its minimum 64-byte size, padding bytes are inserted to ensure at least a 64-byte frame. After the data field is a 4-byte FCS field containing a cyclic redundancy check (CRC) value. The

CRC is created by the sending device and recalculated by the receiving device to check for damage that might have occurred to the frame in transit.

3. IEEE 802.4

IEEE 802.4 Token Bus : In token bus Computer network station must have possession of a token before it can transmit on the computer network. The IEEE 802.4 Committee has defined **token bus** standards as broadband computer networks, as opposed to Ethernet's baseband transmission technique. Physically, the token bus is a linear or tree-shape cable to which the stations are attached **The topology of the computer network can include groups of workstations connected by long trunk cables.** Logically, the stations are organized into a ring. These workstations branch from hubs in a star configuration, so the network has both a bus and star topology. Token bus topology is well suited to groups of users that are separated by some distance. **IEEE 802.4 token bus networks are constructed with 75-ohm coaxial cable using a bus topology**. The broadband characteristics of the 802.4 standard support transmission over several different channels simultaneously.

When the logical ring is initialized, the highest numbered station may send the first frame. The token and frames of data are passed from one station to another following the numeric sequence of the station addresses. Thus, the token follows a logical ring rather than a physical ring. The last station in numeric order passes the token back to the first station. The token does not follow the physical ordering of workstation attachment to the cable. Station 1 might be at one end of the cable and station 2 might be at the other, with station 3 in the middle.

In such a case, there is no collision as only one station possesses a token at any given time. In token bus, each station receives each frame; the station whose address is specified in the frame processes it and the other stations discard the frame.

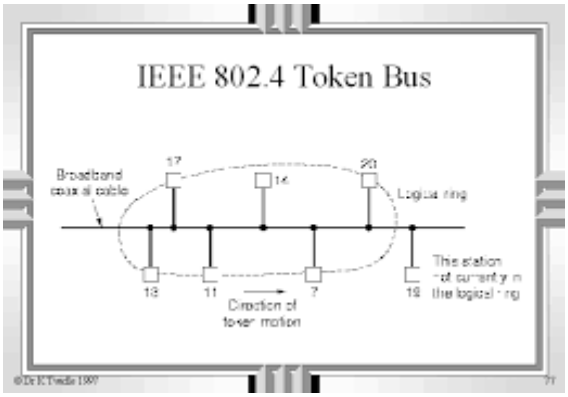

Figure 4. IEEE 8.2.4 Token Bus

3.1 MAC Sub layer Function

When the ring is initialized, stations are inserted into it in order of station address, from highest to lowest.

- Token passing is done from high to low address.

- Whenever a station acquires the token, it can transmit frames for a specific amount of time.
- If a station has no data, it passes the token immediately upon receiving it.
- The token bus defines four priority classes, 0, 2, 4, and 6 for traffic, with 0 the lowest and
- 6 the highest.
- Each station is internally divided into four substations, one at each priority level i.e. 0, 2, 4 and 6.
- As input comes in to the MAC sub layer from above, the data are checked for priority and routed to one of the four substations.
- Thus each station maintains its own queue of frames to be transmitted.
- When a token comes into the station over the cable, it is passed internally to the priority 6 substation, which can begin transmitting its frames, if it has any.
- When it is done or when its time expires, the token is passed to the priority 4 substation, which can then transmit frames until its timer expires. After this the token is then passed internally to priority 2 substation.
- This process continues until either the priority 0 substation has sent all its frames or its time expires.
- After this the token is passed to the next station in the ring.

3.2 Frame format of Token Bus

The various fields present in the IEEE 802.4 Token Ring frame format as shown in Figure 5 are

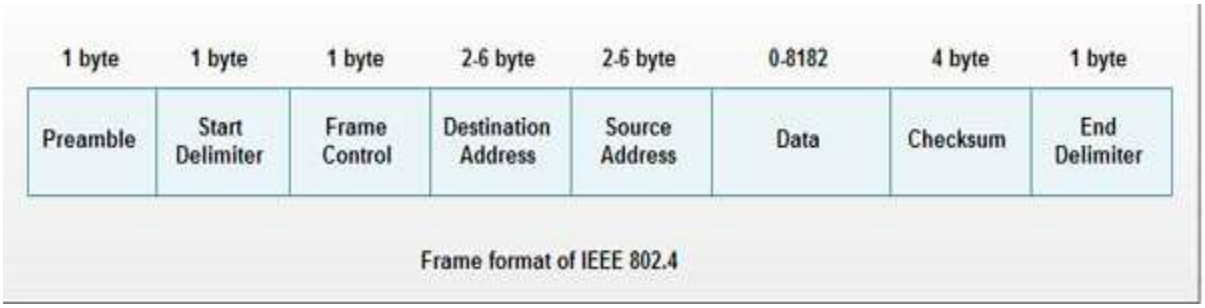

Figure 5. IEEE 802.4 Token Bus Frame

- **Preamble**: This. Field is at least 1 byte long. It is used for bit synchronization.
- **Start Delimiter**: This one byte field marks the beginning of frame.
- **Frame Control:** This one byte field specifies the type of frame. It distinguishes data frame from control frames. For data frames it carries frame's priority. For control frames, it specifies the frame type. The control frame types include token passing and various ring maintenance frames, including the mechanism for letting new station enter the ring, the mechanism for allowing stations to leave the ring.
- **Destination address**: It specifies 2 to 6 bytes destination address.
- **Source address**: It specifies 2 to 6 bytes source address.
- **Data**: This field may be upto 8182 bytes long when 2 bytes addresses are used & upto 8174 bytes long when 6 bytes address is used.

- **Checksum**: This 4 byte field detects transmission errors.
- **End Delimiter**: This one byte field marks the end of frame.

4. IEEE 802.5

The Token Ring network was originally developed by IBM in the 1970s. It is still IBM's primary local area network (LAN) technology and is second only to Ethernet/IEEE 802.3 in general LAN popularity. The related IEEE 802.5 specification is almost identical to and completely compatible with IBM's Token Ring network. In fact, the IEEE 802.5 specification was modeled after IBM Token Ring, and it continues to shadow IBM's Token Ring development. The term Token Ring generally is used to refer to both IBM's Token Ring network and IEEE 802.5 networks. This section addresses both Token Ring and IEEE 802.5. Token Ring and IEEE 802.5 networks are basically compatible, although the specifications differ in minor ways. IBM's Token Ring network specifies a star, with all end stations attached to a device called a multi station access unit (MSAU). In contrast, IEEE 802.5 does not specify a topology, although virtually all IEEE 802.5 implementations are based on a star. Other differences exist, including media type (IEEE 802.5 does not specify a media type, although IBM Token Ring networks use twisted-pair wire) and routing information field size.

	IBM Token Ring network	IEEE 802.5
Data rates	4 or 16 Mbps	4 or 16 Mbps
Stations/segment	260 (shielded twisted pair) 72 (unshielded twisted pair)	250
Topology	Star	Not specified
Media	Twisted pair	Not specified
Signaling	Baseband	Baseband
Access method	Token passing	Token passing
Encoding	Differential Manchester	Differential Manchester

Figure 6: Although dissimilar in some respects, IBM's Token Ring Network and IEEE 802.5 are generally compatible.

4.1 Physical Connections

IBM Token Ring network stations are directly connected to MSAUs, which can be wired together to form one large ring (see Figure 7). Patch cables connect MSAUs to adjacent MSAUs, while lobe cables connect MSAUs to stations. MSAUs include bypass relays for removing stations from the ring

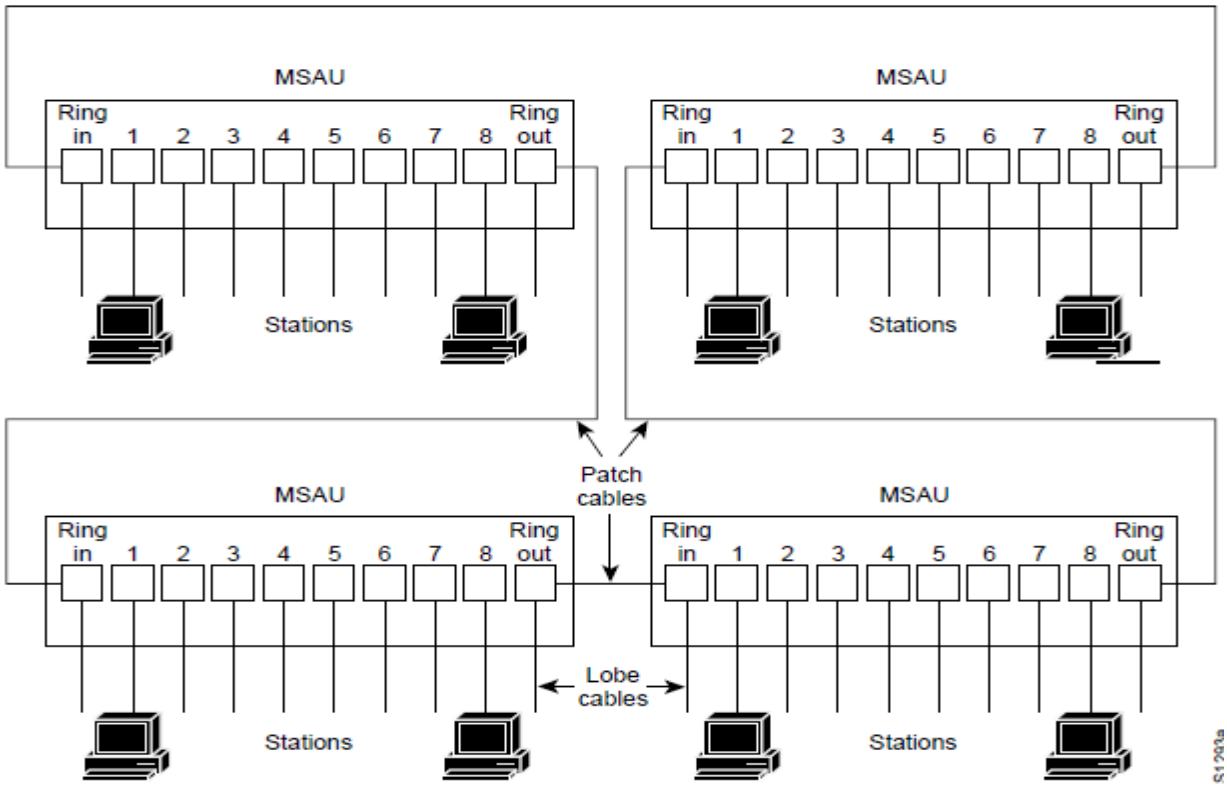

Figure 7 MSAUs can be wired together to form one large ring in an IBM Token Ring network.

4.2 Token Ring Operation

Token Ring and IEEE 802.5 are two principal examples of token-passing networks (FDDI being the other). Token-passing networks move a small frame, called a token, around the network. Possession of the token grants the right to transmit. If a node receiving the token has no information to send, it passes the token to the next end station. Each station can hold the token for a maximum period of time.

If a station possessing the token does have information to transmit, it seizes the token, alters one bit of the token, which turns the token into a start-of-frame sequence, appends the information it wants to transmit, and sends this information to the next

station on the ring. While the information frame is circling the ring, no token is on the network (unless the ring supports early token release), which means that other stations wanting to transmit must wait. Therefore, collisions cannot occur in Token Ring networks. If early token release is supported, a new token can be released when frame transmission is complete. The information frame circulates the ring until it reaches the intended destination station, which copies the information for further processing. The information frame continues to circle the ring and is finally removed when it reaches the sending station. The sending station can check the returning frame to see whether the frame was seen and subsequently copied by the destination.

Unlike CSMA/CD networks (such as Ethernet), token-passing networks are deterministic, which means that it is possible to calculate the maximum time that will pass before any end station will be able to transmit.

4.3 Priority System

Token Ring networks use a sophisticated priority system that permits certain user-designated, high-priority stations to use the network more frequently. Token Ring frames have two fields that control priority: the priority field and the reservation field. Only stations with a priority equal to or higher than the priority value contained in a token can seize that token. After the token is seized and changed to an information frame, only stations with a priority value higher than that of the transmitting station can reserve the token for the next pass around the network. When the next token is generated, it includes the higher priority of the reserving station. Stations that raise a token's priority level must reinstate the previous priority after their transmission is complete.

4.4 Frame Format

Token Ring and IEEE 802.5 support two basic frame types: tokens and data/command frames. Tokens are 3 bytes in length and consist of a start delimiter, an access control byte, and an end delimiter. Data/command frames vary in size, depending on the size of the Information field. Data frames carry information for upper-layer protocols, while command frames contain control information and have no data for upper-layer protocols. Both formats are shown in Figure 6.

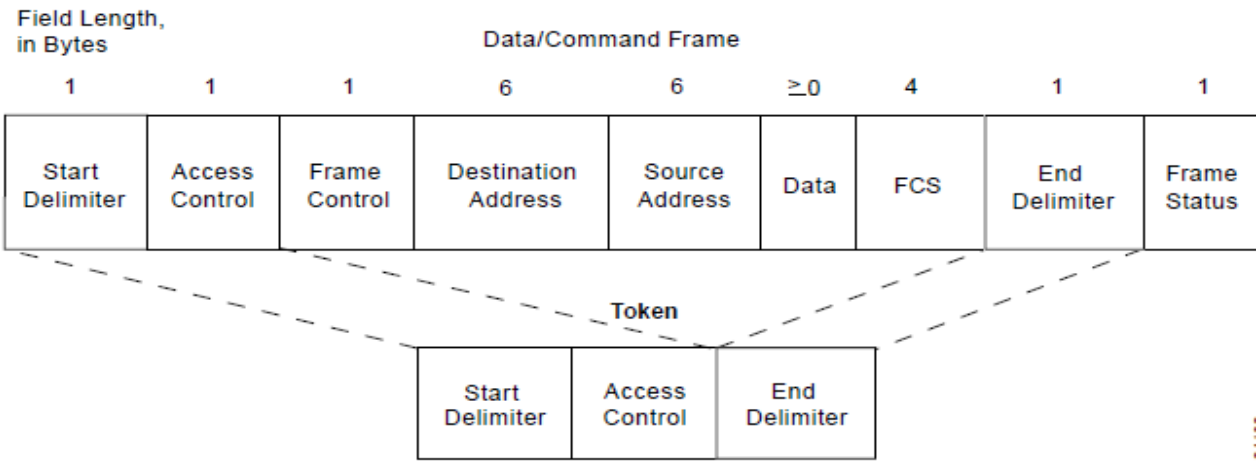

Figure 8: IEEE 802.5 and Token Ring specify tokens and data/command frames.

4.5 Token Frame Fields

The three token frame fields illustrated in Figure 6 are summarized in the descriptions that follow:

• **Start Delimiter**—Alerts each station of the arrival of a token (or data/command frame). This field includes signals that distinguish the byte from the rest of the frame by violating the encoding scheme used elsewhere in the frame.

• **Access-Control Byte**—Contains the Priority field (the most significant 3 bits) and Reservation field (the least significant 3 bits), as well as a token bit (used to differentiate a token from a data/command frame) and a monitor bit (used by the active monitor to determine whether a frame is circling the ring endlessly).

• **End Delimiter**—Signals the end of the token or data/command frame. This field also contains bits to indicate a damaged frame and identify the frame that is the last in a logical sequence.

Data/Command Frame Fields

Data/Command frames have the same three fields as Token Frames, plus several others. The Data/Command frame fields illustrated in Figure 6 are described in the following summaries:

• **Start Delimiter**—Alerts each station of the arrival of a token (or data/command frame). This field includes signals that distinguish the byte from the rest of the frame by violating the encoding scheme used elsewhere in the frame.

• **Access-Control Byte**—Contains the Priority field (the most significant 3 bits) and Reservation field (the least significant 3 bits), as well as a token bit (used to differentiate a

token from a data/command frame) and a monitor bit (used by the active monitor to determine whether a frame is circling the ring endlessly).

• **Frame-Control Bytes**—Indicates whether the frame contains data or control information. In control frames, this byte specifies the type of control information.

• **Destination and Source Addresses**—Two 6-byte address fields identify the destination and source station addresses.

• **Data**—Length of field is limited by the ring token holding time, which defines the maximum time a station can hold the token.

• **Frame-Check Sequence (FCS)**—Filled by the source station with a calculated value dependent on the frame contents. The destination station recalculates the value to determine whether the frame was damaged in transit. If so, the frame is discarded.

• **End Delimiter**—Signals the end of the token or data/command frame. The end delimiter also contains bits to indicate a damaged frame and identify the frame that is the last in a logical sequence.

• **Frame Status**—A 1-byte field terminating a command/data frame. The Frame Status field includes the address-recognized indicator and frame-copied indicator.

5. IEEE 802.11

IEEE has defined the specifications for a wireless LAN, called IEEE 802.11, which covers the physical and data link layers.

5.1 Architecture

The standard defines two kinds of services: the basic service set (BSS) and the extended service set (ESS).

5.1.1 Basic Service Set

IEEE 802.11 defines the basic service set (BSS) as the building block of a wireless LAN. A basic service set is made of stationary or mobile wireless stations and an optional central base station, known as the access point (AP). Figure 9 shows two sets in this standard.

The BSS without an AP is a stand-alone network and cannot send data to other BSSs. It is called an ad hoc architecture. In this architecture, stations can form a network without the need of an AP; they can locate one another and agree to be part of a BSS. A BSS with an AP is sometimes referred to as an infrastructure network.

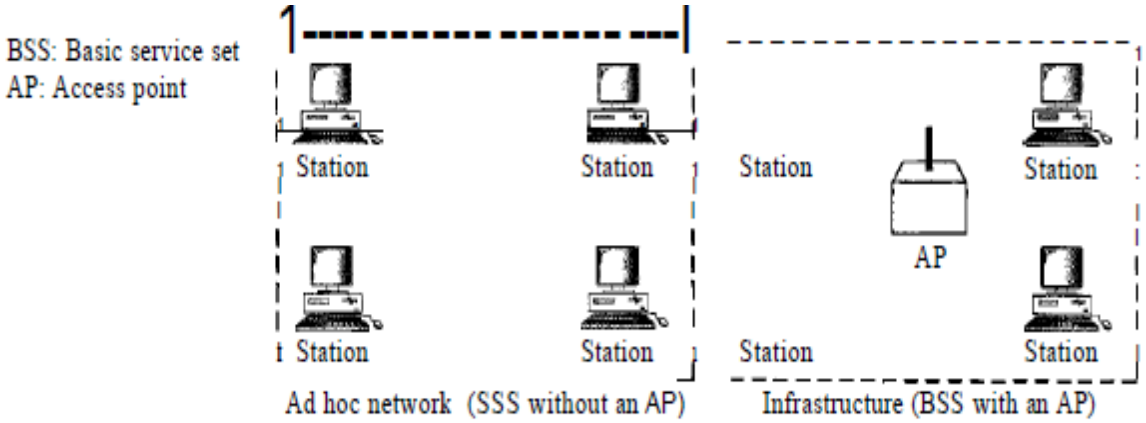

Figure 9: Basic service sets (BSSs)

5.1.2 Extended Service Set

An extended service set (ESS) is made up of two or more BSSs with APs. In this case, the BSSs are connected through a distribution system, which is usually a wired LAN. The distribution system connects the APs in the BSSs. IEEE 802.11 does not restrict the distribution system; it can be any IEEE LAN such as an Ethernet. Note that the extended service set uses two types of stations: mobile and stationary. The mobile stations are normal stations inside a BSS. The stationary stations are AP stations that are part of a wired LAN. Figure 10 shows an ESS.

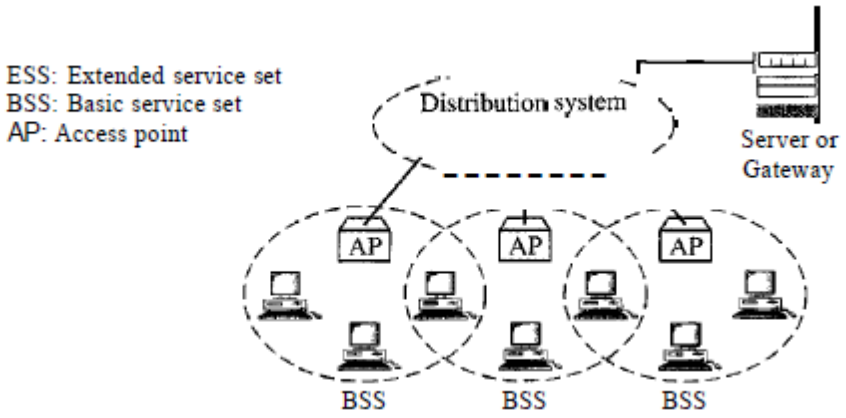

Figure 10 Extended service sets (ESSs)

When BSSs are connected, the stations within reach of one another can communicate without the use of an AP. However, communication between two stations in two different BSSs usually occurs via two APs. The idea is similar to communication in a cellular network if we consider each BSS to be a cell and each AP to be a base station. Note that a mobile station can belong to more than one BSS at the same time.

Station Types

IEEE 802.11 defines three types of stations based on their mobility in a wireless LAN: no-transition, BSS transition, and ESS-transition mobility. A station with no-transition mobility is either stationary (not moving) or moving only inside a BSS. A station with BSS-transition mobility can move from one BSS to another, but the movement is confined inside one ESS. A station with ESS-transition mobility can move from one ESS to another. However, IEEE 802.11 does not guarantee that communication is continuous during the move. MAC Sub layer IEEE 802.11 defines two MAC sub layers: the distributed coordination function (DCF) and point coordination function (PCF). Figure 11 shows the relationship between the two MAC sub layers, the LLC sub layer, and the physical layer. We discuss the physical layer implementations later in the chapter and will now concentrate on the MAC sub layer.

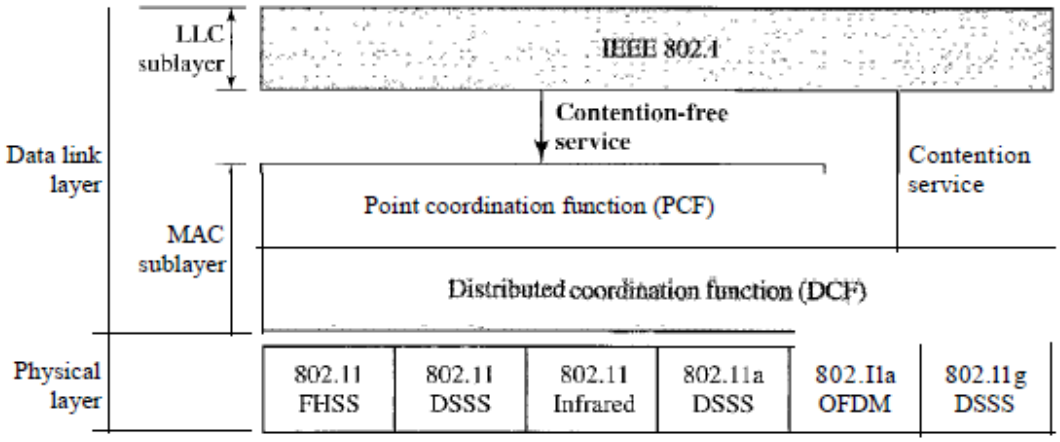

Figure 11 MAC layers in IEEE 802.11 standard

Distributed Coordination Function

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

1. For collision detection a station must be able to send data and receive collision signals at the same time. This can mean costly stations and increased bandwidth requirements.

2. Collision may not be detected because of the hidden station problem.

3. The distance between stations can be great. Signal fading could prevent a station at one end from hearing a collision at the other end.

Point Coordination Function (PCF)

The point coordination function (PCF) is an optional access method that can be implemented in an infrastructure network (not in an ad hoc network). It is implemented on top of the DCF and is used mostly for time-sensitive transmission. PCF has a centralized, contention-free polling access method. The AP performs polling for stations that are capable of being polled. The stations are polled one after another, sending any data they have to the AP. To give priority to PCF over DCF, another set of inter frame spaces has been defined: PIFS and SIFS. The SIFS is the same as that in DCF, but the PIFS (PCF IFS) is shorter than the DIFS. This means that if, at the same time, a station wants to use only DCF and an AP wants to use PCF, the AP has priority. Due to the priority of PCF over DCF, stations that only use DCF may not gain access to the medium. To prevent this, a repetition interval has been designed to cover both contention-free (PCF) and contention-based (DCF) traffic. The repetition interval, which is repeated continuously, starts with a special control frame, called a beacon frame. When the stations hear the beacon frame, they start their NAV for the duration of the contention-free period of the repetition interval. Figure 12 shows an example of a repetition interval.

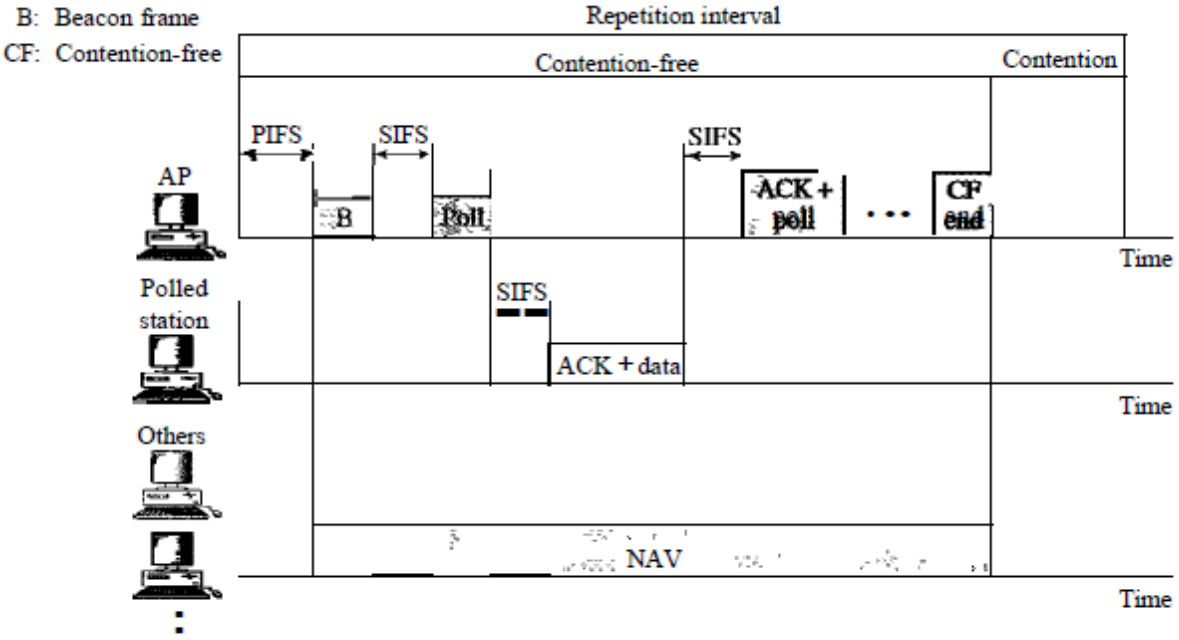

Figure 12 Example of repetition interval

During the repetition interval, the PC (point controller) can send a poll frame, receive data, send an ACK, receive an ACK, or do any combination of these (802.11 uses piggybacking). At the end of the contention-free period, the PC sends a CF end (contention-free end) frame to allow the contention-based stations to use the medium.

Fragmentation

The wireless environment is very noisy; a corrupt frame has to be retransmitted. The protocol, therefore, recommends fragmentation-the division of a large frame into smaller ones. It is more efficient to resend a small frame than a large one.

Frame Format

The MAC layer frame consists of nine fields, as shown in Figure 13.

Frame control (FC): The FC field is 2 bytes long and defines the type of frame and some control information. Table 14.1 describes the subfields. We will discuss each frame type later in this chapter.

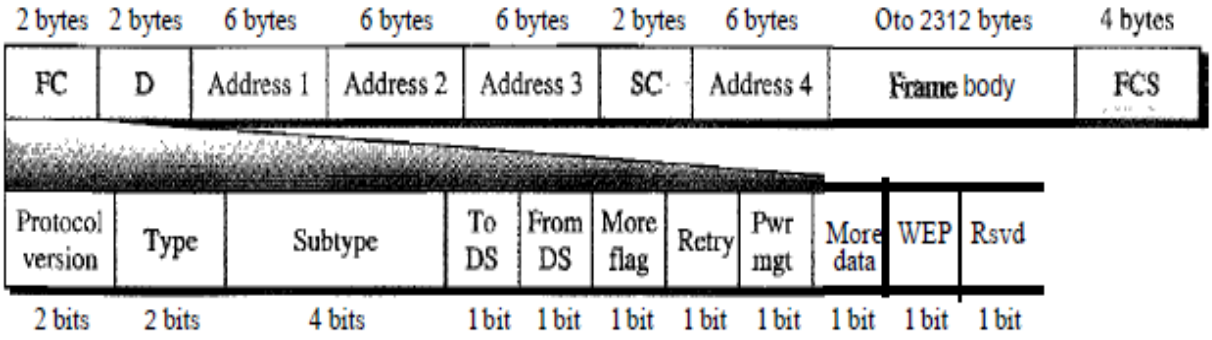

Figure 13 Frame format

Frame Types

A wireless LAN defined by IEEE 802.11 has three categories of frames: management frames, control frames, and data frames. Management Frames Management frames are used for the initial communication between stations and access points. Control Frames Control frames are used for accessing the channel and acknowledging frames. Figure 14 shows the format.

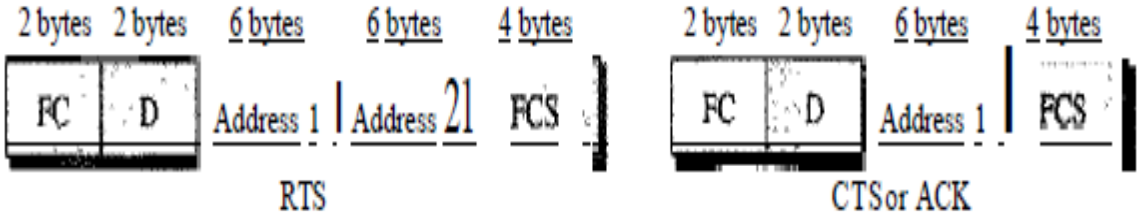

Figure 14 Control frames

Physical Layer

We discuss six specifications, as shown in Table 2.

Table 2 Physical layers

IEEE	*Technique*	*Band*	*Modulation*	*Rate (Mbps)*
802.11	FHSS	2.4 GHz	FSK	1 and 2
	DSSS	2.4 GHz	PSK	1 and 2
		Infrared	PPM	1 and 2
802.11a	OFDM	5.725 GHz	PSKorQAM	6 to 54
802.11b	DSSS	2.4 GHz	PSK	5.5 and 11
802.1Ig	OFDM	2.4 GHz	Different	22 and 54

All implementations, except the infrared, operate in the industrial, scientific, and medical (ISM) band, which defines three unlicensed bands in the three ranges 902-928 MHz, 2.400--4.835 GHz, and 5.725-5.850 GHz, as shown in Figure 15.

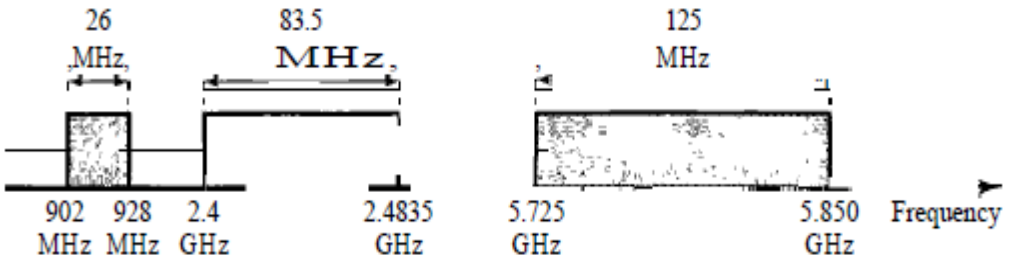

Figure 15 industrial, scientific, and medical (ISM) band

IEEE 802.11 FHSS

IEEE 802.11 FHSS uses the frequency-hopping spread spectrum (FHSS) method as discussed in Chapter 6. FHSS uses the 2.4-GHz ISM band. The band is divided into 79 sub bands of 1 MHz (and some guard bands). A pseudorandom number generator selects the hopping sequence. The modulation technique in this specification is either two-level FSK or four-level FSK with I or 2 bitsbaud, which results in a data rate of 1 or 2 Mbps, as shown in Figure 15.

IEEE 802.11 DSSS

IEEE 802.11 DSSS uses the direct sequence spread spectrum (DSSS) method. DSSS uses the 2.4-GHz ISM band. The modulation technique in this specification is PSK at 1 Mbaud/s. The system allows 1 or 2 bitsbaud (BPSK or QPSK), which results in a data rate of 1 or 2 Mbps, as shown in Figure 15.

IEEE 802.11 Infrared

IEEE 802.11 infrared uses infrared light in the range of 800 to 950 nm. The modulation technique is called pulse position modulation (PPM). For a I-Mbps data rate, a 4-bit sequence is first mapped into a 16-bit sequence in which only one bit is set to 1 and the rest are set to O.

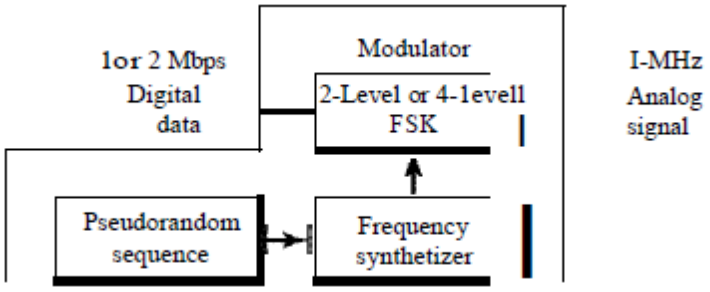

Figure 15 Physical layer of IEEE 802.11 FHSS.

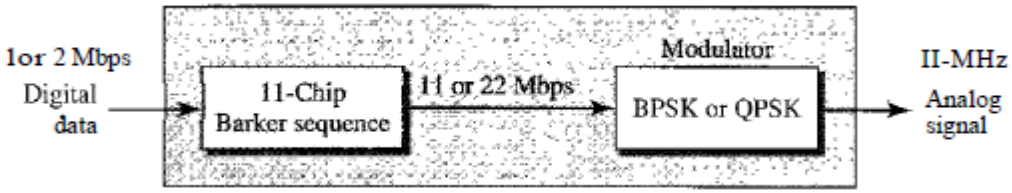

Figure 16 Physical layer of IEEE 802.11 DSSS

For a 2-Mbps data rate, a 2-bit sequence is first mapped into a 4-bit sequence in which only one bit is set to 1 and the rest are set to O. The mapped sequences are then converted to optical signals; the presence of light specifies 1, the absence of light specifies O. See Figure 17.

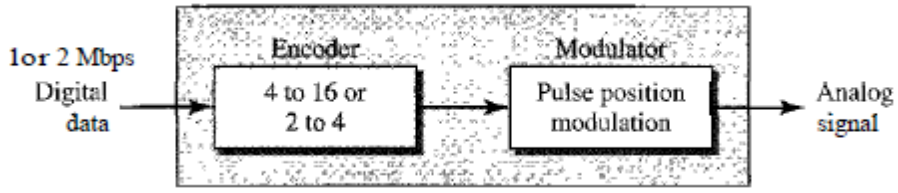

Figure 17 Physical layer of IEEE 802.11 infrared

IEEE 802.lla OFDM

IEEE 802.Ila OFDM describes the orthogonal frequency-division multiplexing (OFDM) method for signal generation in a 5-GHz ISM band. OFDM is similar to, with one major difference: All the sub bands are used by one source at a given time. Sources

contend with one another at the data link layer for access. The band is divided into 52 sub bands, with 48 sub bands for sending 48 groups of bits at a time and 4 sub bands for control information. Dividing the band into sub bands diminishes the effects of interference. If the sub bands are used randomly, security can also be increased. OFDM uses PSK and QAM for modulation. The common data rates are 18 Mbps (PSK) and 54 Mbps (QAM).

IEEE 802.llb DSSS

IEEE 802.11 b DSSS describes the high-rate direct sequence spread spectrum (HRDSSS) method for signal generation in the 2.4-GHz ISM band. HR-DSSS is similar to DSSS except for the encoding method, which is called complementary code keying (CCK). CCK encodes 4 or 8 bits to one CCK symbol. To be backward compatible with DSSS, HR-DSSS defines four data rates: 1, 2, 5.5, and 11 Mbps. The first two use the same modulation techniques as DSSS. The 5.5-Mbps version uses BPSK and transmits at 1.375 Mbaudls with 4-bit CCK encoding. The II-Mbps version uses QPSK and transmits at 1.375 Mbps with 8-bit CCK encoding. Figure 18 shows the modulation technique for this standard.

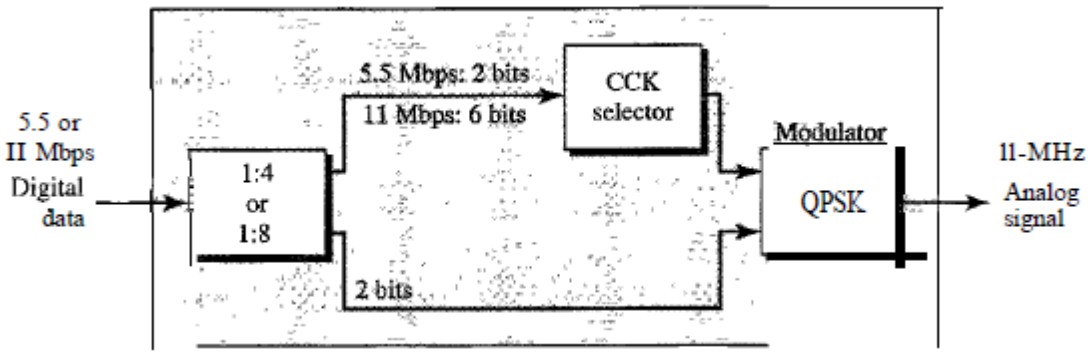

Figure 18 Physical layer of IEEE 802.11b

IEEE 802.11g

This new specification defines forward error correction and OFDM using the 2.4-GHz ISM band. The modulation technique achieves a 22- or 54-Mbps data rate. It is backward compatible with 802.11b, but the modulation technique is OFDM.

6. Summary

In this lesson we have discussed various IEEE standards, which are globally recognized specifications that govern various aspects of technology and engineering. Covering a wide range of fields, these standards ensure compatibility, interoperability, and quality across devices and systems. Notable IEEE standards include those related to networking (e.g., IEEE 802.11 for Wi-Fi), communications (e.g., IEEE 802.3 for Ethernet), and electrical engineering (e.g., IEEE 754 for floating-point arithmetic). These standards play a pivotal

role in driving innovation, enabling seamless integration, and promoting consistent practices within the industry, benefiting both manufacturers and consumers.

7. Self Check Exercise

1. What are the common IEEE 802 standards?
2. What are the benefits of IEEE 802 standards?
3. Discuss the architecture of IEEE 802 in detail.
4. Explain in detail the IEEE 802.5 standards.

8. Suggested Readings

1. Andrew S. Tannenbaum, “Computer Networks", 3rd Edition, Prentice Hall.

2. Behrouz A. Forouzan, “Data Communications & Networking”, Fourth edition, Tata Mc l7yyGraw Hills.

3. D.E. Cormer and D.L Stevens, "Internetworking with TCP/IP: Design implementations and Internals, “Vol II , Prentice Hall, 1990.

4. D.E. Cormer," Computer Networks and Internet”, 2nd Edition, Addison Wesley Publication, 2000.

5. D. Bertsekas and R.Gallagar, “Data Networks”, 2nd Edition, Prentice-Hall, 1992.

Last Updated on May 2023

Mandatory Student Feedback Form

https://forms.gle/KS5CLhvpwrpgjwN98

Note: Students, kindly click this google form link, and fill this feedback form once.