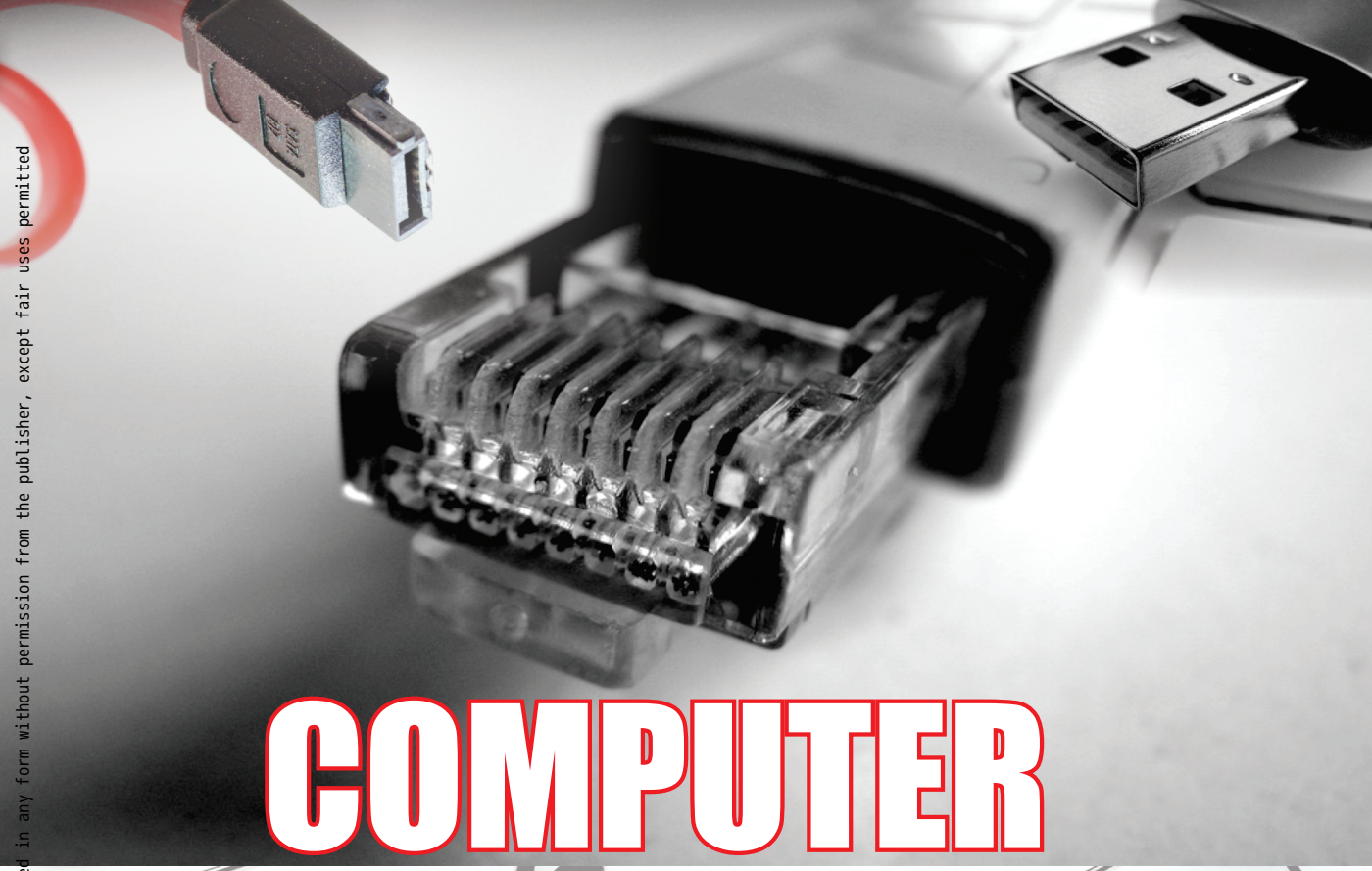


# COMPUTER NETWORKS

Er. Vikrant Vij

# COMPUTER NETWORKS

# COMPUTER NETWORKS

*For*
*B.E./B.Tech/AMIE/AMIETE examinations in Computer Science and Electronics discipline*

*By*
**Er. Vikrant Vij**
*B.tech (Hons.), M.E (ECE)*
*Ex- Faculty in Electronics and Communication Engineering*
*Jaypee University of Information Technology, Himachal Pradesh*

UNIVERSITY SCIENCE PRESS
(An Imprint of Laxmi Publications Pvt. Ltd.)
An ISO 9001:2008 Company
**BENGALURU • CHENNAI • GUWAHATI • HYDERABAD • JALANDHAR**
**KOCHI • KOLKATA • LUCKNOW • MUMBAI • RANCHI • NEW DELHI**
**BOSTON (USA) • NAIROBI (KENYA)**

**COMPUTER NETWORKS**



Printed and bound in India
*Typeset at* ABRO Enterprises, Delhi
First Edition : 2018
*ISBN* 978-93-5274-080-2



**Branches**

| | | |
|---|---|---|
| Bengaluru | 080-26 75 69 30 | |
| Chennai | 044-24 34 47 26, | 24 35 95 07 |
| Guwahati | 0361-254 36 69, | 251 38 81 |
| Hyderabad | 040-27 55 53 83, | 27 55 53 93 |
| Jalandhar | 0181-222 12 72 | |
| Kochi | 0484-237 70 04, | 405 13 03 |
| Kolkata | 033-22 27 43 84 | |
| Lucknow | 0522-220 99 16 | |
| Mumbai | 022-24 93 12 61 | |
| Ranchi | 0651-220 44 64 | |

PUBLISHED IN INDIA BY

UNIVERSITY SCIENCE PRESS
(An Imprint of Laxmi Publications Pvt. Ltd.)

An ISO 9001:2008 Company
113, GOLDEN HOUSE, DARYAGANJ,
NEW DELHI - 110002, INDIA
Telephone : 91-11-4353 2500, 4353 2501
Fax : 91-11-2325 2572, 4353 2528
www.laxmipublications.com info@laxmipublications.com

C—
Printed at:

***Dedicated to ...***

*My Loving Parents, My Living Angels, My Guides*

***Mrs. Muktesh Vij & Mr. Harsh Vij***

*For showing me light on this earth...*

# Contents

# PREFACE

*In The Name of GOD and My Angel Guides.....*

There are two sides of computer revolution: one is represented by the PC on your desktop and the second one by the device that remote-controls your TV, monitors and operates your car engine, and allows you to set up your answering machine and your microwave oven. Man's constant quest to communicate has resulted in a quantum leap in technology related to data communications. For the past quarter century the maximum obtainable transmission rate on many types of communications facilities has doubled every three to five years. During the past few years this growth rate has accelerated, with emerging technologies providing a transmission capability, an order of magnitude or more above what were considered high operating rates just a year or two ago. Accompanying this growth and, in many cases, providing the impetus for the technological developments that made such growth possible are communications-dependent applications.

This book examines a wide range of techniques, technologies and systems used in data communications and computer networks. In particular it addresses a variety of data transmission methods used to convey data between physically distant locations. A number of types of network are considered which may interconnect a few or many thousands of users on either a permanent or temporary, switched basis. In order to support successful communication, a set of rules and procedures has been developed, many of which are embodied in internationally agreed standards. We shall look at the standard bodies and their standards and recommendations and also how they are used in practice.

The organized material that resulted in this book is an attempt to provide a reasonable degree of balance between rigor, clarity of presentation and at the same time keeping the length of book at manageable level. The principal objective of the book is to provide an introduction to various computer network fundamentals and to develop a foundation that can be used as basis for research and further study in this field.

**Chapter I** is Introduction to Computer Networks highlighting Network hardware and software issues including Network models and services.

**Chapter II** is about Physical Layer and networks

**Chapter III** is about Data link layer concepts, Flow control, error control techniques.

**Chapter IV** is about MAC Layer including Ethernet, WLAN, Bluetooth concepts.

**Chapter V** is about Network layer issues, IP Address, Congestion control, Routing, etc.

**Chapter VI** is about Transport Layer and its relevant issues, TCP, UDP Protocols.

**Chapter VII** is about Application layer including DNS, WWW, e-mail, etc.

**Chapter VIII** is about Network security and cryptography.

## Readers of this book

The book has been written to position itself within the marketplace midway between a number of excellent texts in this subject area which may be regarded as comprehensive, and almost reference works, and a number of other texts which are rather brief and tend to be merely primers. Reference texts can be too detailed and large for a newcomer to the topic and the primer type of text can lack information and be rather bland for many readers. With this in mind the book is written to appeal to two broad ranges of readers:

1. Students of computer science/electronics engineering undergraduate and postgraduate courses who are studying data communications and/or computer networks.
2. Professional engineers, managers and users who work with computer networks in their everyday work. Here the book may be used either as an introductory text for personnel moving into the field or for updating in cases where previous experience has predominantly been in the area of traditional telecommunications...

While the information contained in this book has been carefully checked for accuracy, the author assumes no responsibility or liability for its use, or any infringement of patents or other rights of third parties which would result.

**—Author**

# Acknowledgements


## Credit where credit is due

I wish to express my gratitude to *My loving Grandparents* for their support and blessings throughout my life.

I would like to thank *my dear brother Er. Abhey Vij* for his wonderful ideas; that helped me in preparation of manuscript of this book.

I am grateful to my friend *Er. Tarun Mittar;* for inspiring me to work on this project.

I would like to express my sincerest thanks to *Dr. Rajneesh Arora,* Vice Chancellor Punjab Technical University, *Dr. Parijat De,* Director Technical Education, West Bengal, *Dr. A.K. Bhandari,* Registrar Punjab University, Chandigarh for their blessings and help in my academic pursuits.

I am thankful to *Sh. Manoj Gaur Ji,* Chairman, Jaypee Group, *Dr. Yajulu Medury,* COO, Jaypee Education System, *Brig (Retd.) Balbir Singh,* JUIT Solan, *Dr. Vivek Sehgal*, JUIT Solan for providing me conducive environment during my stay at Jaypee Institutes.

Lastly, I would like to extend my sincerest thanks to Management of IET Bhaddal and *Mrs. Kulwinder Gurcharan Singh,* Chairperson KFET for helping me in initial start of my teaching career.

**—Author**

# CHAPTER 1 COMPUTER NETWORK FUNDAMENTALS

*I must Create a System, or be enslav'd by another Man's; I will not Reason and Compare: my business is to Create.*

**—William Blake**

## 1.1 INTRODUCTION

The basic ideas in all types of communication are that there must be three ingredients for the communication to be effective. First, there must be two entities, dubbed a sender and a receiver. These two must have something they need to share. Second, there must be a medium through which the sharable item is channeled. This is the transmission medium. Finally, there must be an agreed-on set of communication rules or protocols. These three apply to every category or structure of communication. The effectiveness of a data communications system depends on four fundamental characteristics: delivery, accuracy, timeliness, and jitter.

1. **Delivery.** The system must deliver data to the correct destination. Data must be received by the intended device or user and only by that device or user.
2. **Accuracy.** The system must deliver the data accurately. Data that have been altered in transmission and left uncorrected are unusable.
3. **Timeliness.** The system must deliver data in a timely manner. Data delivered late are useless. In the case of video and audio, timely delivery means delivering data as they are produced, in the same order that they are produced, and without significant delay. This kind of delivery is called *real-time* transmission.

4. **Jitter.** Jitter refers to the variation in the packet arrival time. It is the uneven delay in the delivery of audio or video packets. For example, let us assume that video packets are sent every 3D ms. If some of the packets arrive with 3D-ms delay and others with 4D-ms delay, an uneven quality in the video is the result.

Figure 1.1 shows a simple communication model consisting of following key elements:

1. **Source.** This device generates the data to be transmitted; examples are telephones and personal computers.
2. **Transmitter.** Usually, the data generated by a source system are not transmitted directly in the form in which they were generated. Rather, a transmitter transforms and encodes the information in such a way as to produce electromagnetic signals that can be transmitted across some sort of transmission system. For example, a modem takes a digital bit stream from an attached device such as a personal computer and transforms that bit stream into an analog signal that can be handled by the telephone network.
3. **Transmission System.** This can be a single transmission line or a complex network connecting source and destination.
4. **Receiver.** The receiver accepts the signal from the transmission system and converts it into a form that can be handled by the destination device. For example, a modem will accept an analog signal coming from a network or transmission line and convert it into a digital bit stream.
5. **Destination.** Takes the incoming data from the receiver.

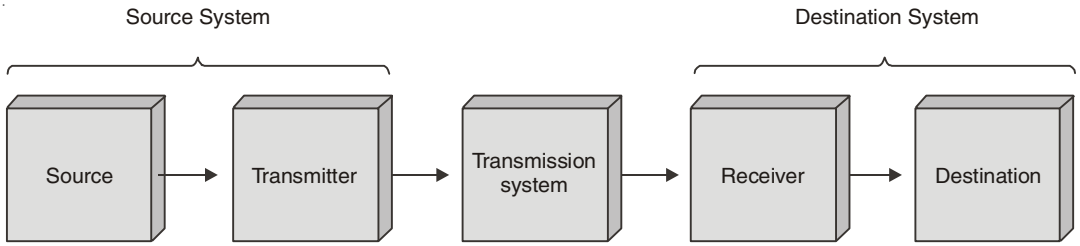



**Fig. 1.1.** *A simple communication model.*

In this chapter, we will focus on these three components in a computer network. But what is a computer network? A computer network is a distributed system consisting of loosely coupled computers and other devices. Any two of these devices, which we will from now on refer to as *network elements* or *transmitting elements* without loss of generality, can communicate with each other through a communication medium. In order for these connected devices to be considered a communicating network, there must be a set of communicating rules or protocols each device in the network must follow to communicate with another device in the network. The resulting combination consisting of hardware and software is a computer communication network or computer network in short. Figure 1.2 shows a computer network.

The hardware component is made of network elements consisting of a collection of nodes that include the end systems commonly called hosts and intermediate switching

elements that include hubs, bridges, routers, and gateways that, without loss of generality, we will call network elements.

Network elements may own resources individually, that is locally or globally. Network software consists of all application programs and network protocols that are used to synchronize, coordinate, and bring about the sharing and exchange of data among the network elements. Network software also makes the sharing of expensive resources in the network possible. Network elements, network software, and users all work together so that individual users can exchange messages and share resources on other systems that are not readily available.

Internetworking technology enables multiple, diverse underlying hardware technologies and different software regimes to interconnect heterogeneous networks and bring them to communicate smoothly. The smooth working of any computer communication network is achieved through the low-level mechanisms provided by the network elements and high-level communication facilities provided by the software running on the communicating elements.

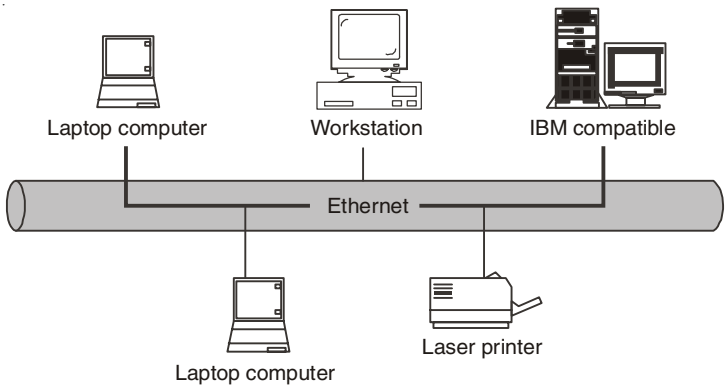



**Fig. 1.2.** *A computer network.*

## 1.2 COMPUTER NETWORK TYPES

Computer networks come in different sizes. Each network is a cluster of network elements and their resources. The size of the cluster determines the network type. There are, in general, two main network types: the local area network (LAN) and wide area network (WAN).

### 1.2.1 Local Area Networks (LANs)

A computer network with two or more computers or clusters of network and their resources connected by a communication medium sharing communication protocols and confined in a small geographical area, such as a building floor, a building, or a few adjacent buildings, is called a local area network (LAN). The advantage of a LAN is that all network elements are close together so the communication links maintain a higher speed of data movement.

Also, because of the proximity of the communicating elements, high-cost and high quality communicating elements can be used to deliver better service and high reliability. Figure 1.3 shows a LAN network.

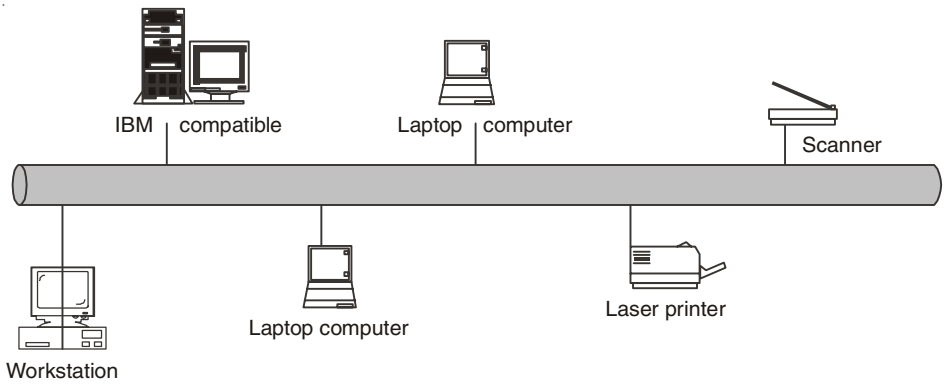



**Fig. 1.3.** *A LAN network.*

## 1.2.2 Wide Area Networks (WANs)

A wide area network (WAN), on the other hand, is a network made up of one or more clusters of network elements and their resources but instead of being confined to a small area, the elements of the clusters or the clusters themselves are scattered over a wide geographical area as in a region of a country or across the whole country, several countries, or the entire globe like the Internet for example.

Some advantages of a WAN include distributing services to a wider community and availability of a wide array of both hardware and software resources that may not be available in a LAN. However, because of the large geographical areas covered by WANs, communication media are slow and often unreliable. Figure 1.4 shows a WAN network.

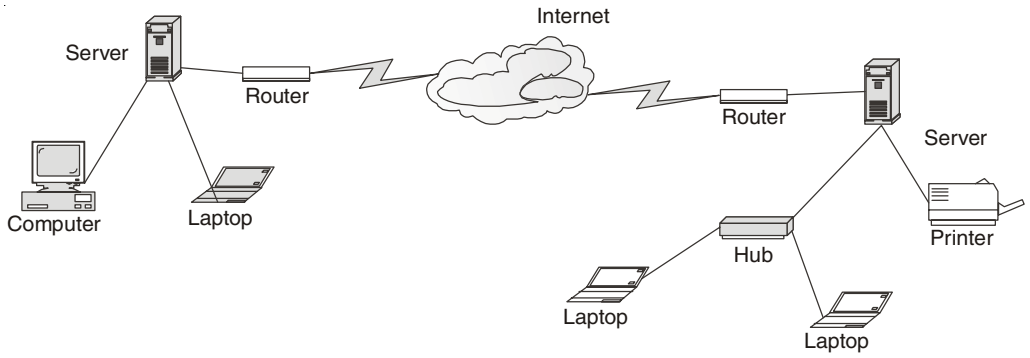



**Fig. 1.4.** *A WAN network.*

## 1.2.3 Metropolitan Area Networks (MANs)

Between the LAN and WAN, there is also a middle network called the metropolitan area network (MAN) because it covers a slightly wider area than the LAN but not so wide as to be considered a WAN. Civic networks that cover a city or part of a city are a good

example of a MAN. MANs are rarely talked about because they are quiet often overshadowed by cousin LAN to the left and cousin WAN to the right.

## 1.3 NETWORK TOPOLOGY

Computer networks, whether LANs, MANs, or WANs, are constructed based on a topology. There are several topologies including the following popular ones.

### 1.3.1 Mesh

A mesh topology allows multiple access links between network elements, unlike other types of topologies. The multiplicity of access links between the network elements offers an advantage in network reliability because whenever one network element fails, the network does not cease operations; it simply finds a bypass to the failed element and the network continues to function. Mesh topology is most often applied in MAN networks. Figure 1.5 shows a mesh network.

A mesh offers several advantages over other network topologies. First, the use of dedicated links guarantees that each connection can carry its own data load, thus eliminating the traffic problems that can occur when links must be shared by multiple devices. Second, there is the advantage of privacy or security. When every message travels along a dedicated line, only the intended recipient sees it. Physical boundaries prevent other users from gaining access to messages. Finally, point-to-point links make fault identification and fault isolation easy. Traffic can be routed to avoid links with suspected problems. This facility enables the network manager to discover the precise location of the fault and aids in finding its cause and solution.

The main disadvantages of a mesh are related to the amount of cabling and the number of I/O ports required. First, because every device must be connected to every other device, installation and reconnection are difficult. Second, the sheer bulk of the wiring can be greater than the available space (in walls, ceilings, or floors) can accommodate. Finally, the hardware required to connect each link (I/O ports and cable) can be prohibitively expensive. For these reasons a mesh topology is usually implemented in a limited fashion, for example, as a backbone connecting the main computers of a hybrid network that can include several other topologies.

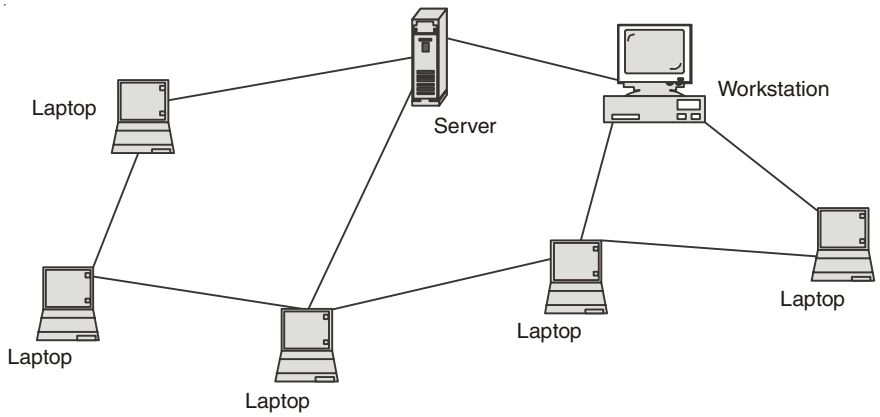



**Fig. 1.5.** *A Mesh network.*

### 1.3.2 Tree

A more common type of network topology is the tree topology. In the tree topology, network elements are put in a hierarchical structure in which the most predominant element is called the *root* of the tree and all other elements in the network share a child–parent relationship. As in ordinary, though inverted trees, there are no closed loops. So dealing with failures of network elements presents complications depending on the position of the failed element in the structure. For example, in a deeply rooted tree, if the root element fails, the network automatically ruptures and splits into two parts. The two parts cannot communicate with each other. The functioning of the network as a unit is, therefore, fatally curtailed. Figure 1.6 shows a network using a tree topology.

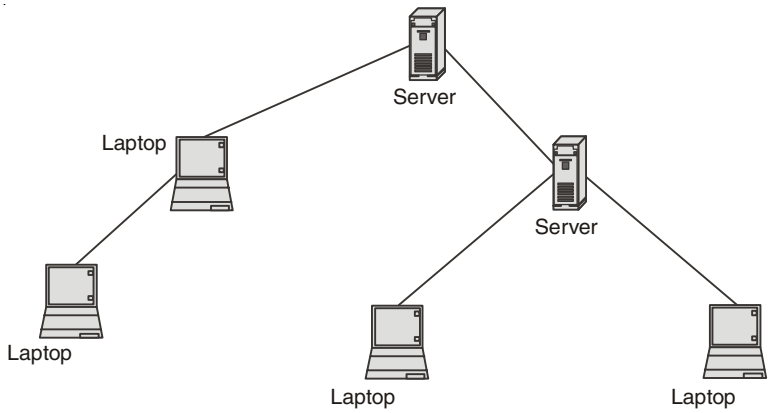



**Fig. 1.6.** *A Tree network.*

### 1.3.3 Bus

A more popular topology, especially for LANs, is the bus topology. Elements in a network using a bus topology always share a bus and, therefore, have equal access to all LAN resources. Every network element has full-duplex connections to the transmitting medium which allows every element on the bus to send and receive data. Because each computing element is directly attached to the transmitting medium, a transmission from any one element propagates through the entire length of the medium in either direction and therefore can be received by all elements in the network. Because of this, precautions need to be taken to make sure that transmissions intended for one element can be received by that element and no other element. The network must also use a mechanism that handles disputes in case two or more elements try to transmit at the same time. The mechanism deals with the likely collision of signals and brings a quick recovery from such a collision. It is also necessary to create fairness in the network so that all other elements can transmit when they need to do so. See Fig. 1.7.

A collision control mechanism must also improve efficiency in the network using a bus topology by allowing only one element in the network to have control of the bus at any one time. This network element is then called the bus master and other elements are considered to be its slaves. This requirement prevents collision from occurring in the network as elements

in the network try to seize the bus at the same time. A bus topology is commonly used by LANs.

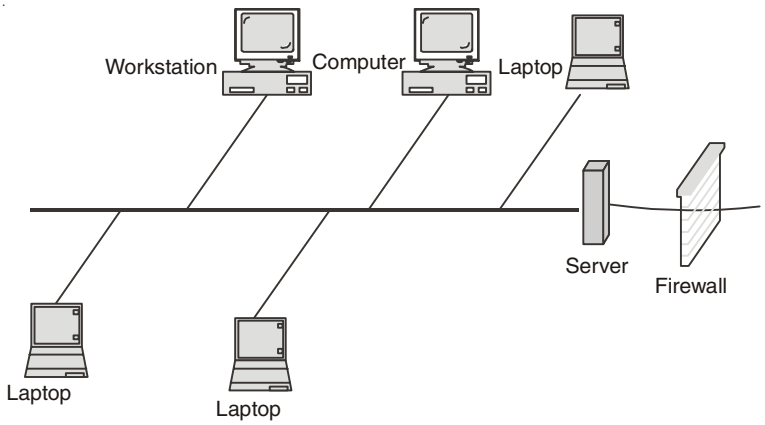



**Fig. 1.7.** *A Bus network.*

### 1.3.4 Star

Another very popular topology, especially in LAN network technologies, is a star topology. A star topology is characterized by a central prominent node that connects to every other element in the network. So, all the elements in the network are connected to a central element. Every network element in a star topology is connected pair wise in a point-to-point manner through the central element, and communication between any pair of elements must go through this central element. The central element or node can either operate in a broadcast fashion, in which case information from one element is broadcast to all connected elements, or transmit as a switching device in which the incoming data is transmitted only to one element, the nearest element enroute to the destination. The biggest disadvantage to the star topology in networks is that the failure of the central element results in the failure of the entire network. Figure 1.8 shows a star topology.

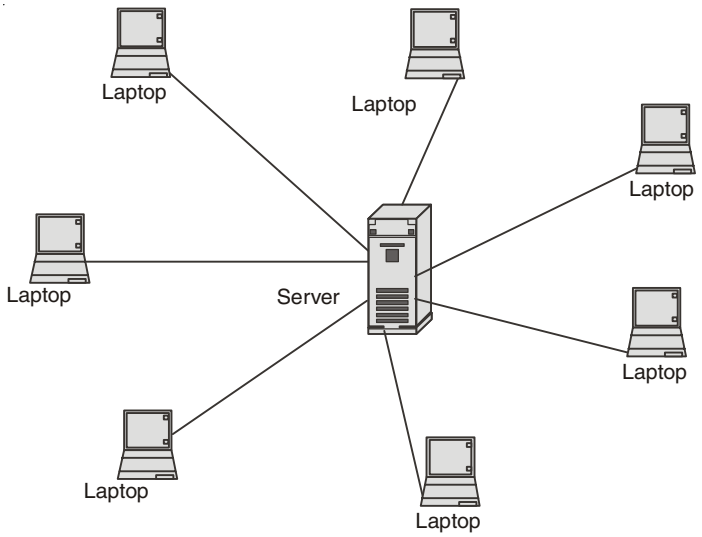



**Fig. 1.8.** *A Star network.*

### 1.3.5 Ring

Finally another popular network topology is the ring topology. In this topology, each computing element in a network using a ring topology is directly connected to the transmitting medium via a unidirectional connection so that information put on the transmission medium can reach all computing elements in the network through a mechanism of taking turns in sending information around the ring. Figure 1.9 shows a ring topology network. The taking of turns in passing information is managed through a *token* system. A token is a system-wide piece of information that guarantees the current owner to be the bus master. As long as it owns the token, no other network element is allowed to transmit on the bus. When an element currently sending information and holding the token has finished, it passes the token downstream to its nearest neighbor. The token system is a good management system of collision and fairness.

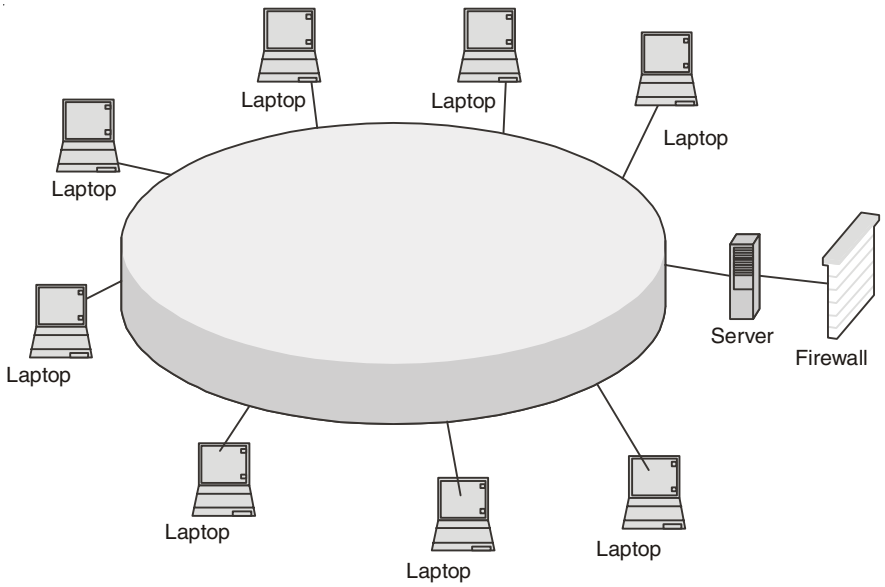



**Fig. 1.9.** *A Ring network.*

There are variants of a ring topology collectively called *hub* hybrids combining either a star with a bus or a stretched star as shown in Fig. 1.10

Although network topologies are important in LANs, the choice of a topology depends on a number of other factors, including the type of transmission medium, reliability of the network, the size of the network, and its anticipated future growth. Recently the most popular LAN topologies have been the bus, star, and ring topologies. The most popular bus- and star-based LAN topology is the Ethernet, and the most popular ring-based LAN topology is the token ring.

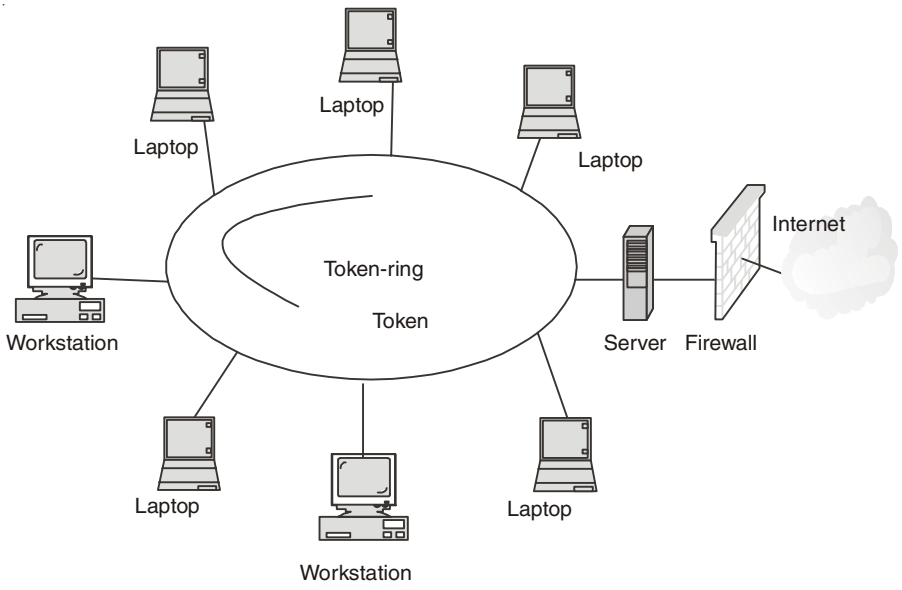



**Fig. 1.10.** *Token Ring Hub.*

## 1.4 NETWORK CONNECTIVITY AND PROTOCOLS

In the early days of computing, computers were used as stand-alone machines, and all work that needed cross-computing was done manually. Files were moved on disks from computer to computer. When computers, terminals, and/or other data processing devices exchange data, the scope of concern is much broader than the concerns we have discussed in Sections 1.2 and 1.3. Consider, for example, the transfer of a file between two computers. There must be a data path between the two computers, either directly or via a communication network. But more is needed. Typical tasks to be performed are:

1. The source system must either activate the direct data communication path or inform the communication network of the identity of the desired destination system.
2. The source system must ascertain that the destination system is prepared to receive data.
3. The file transfer application on the source system must ascertain that the file management program on the destination system is prepared to accept and store the file for this particular user.
4. If the file formats used on the two systems are incompatible, one or the other system must perform a format translation function.

It is clear that there must be a high degree of cooperation between the two computer systems. The exchange of information between computers for the purpose of cooperative action is generally referred to as *computer communications.*

Similarly, when two or more computers are interconnected via a communication network, the set of computer stations is referred to as a *computer network.* Because a similar level of cooperation is required between a user at a terminal and one at a computer, these terms are often used when some of the communicating entities are terminals.

In discussing computer communications and computer networks, two concepts are paramount:

Protocols Computer-communications architecture, or protocol architecture.

"***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. The key elements of a protocol are syntax, semantics, and timing.

- ***Syntax***. The term *syntax* refers to the structure or format of the data, meaning the order in which they are presented. For example, a simple protocol might expect the first 8 bits of data to be the address of the sender, the second 8 bits to be the address of the receiver, and the rest of the stream to be the message itself.
- ***Semantics***. The word *semantics* refers to the meaning of each section of bits. How is a particular pattern to be interpreted, and what action is to be taken based on that interpretation? For example, does an address identify the route to be taken or the final destination of the message?
- ***Timing***. The term *timing* refers to two characteristics: when data should be sent and how fast they can be sent. For example, if a sender produces data at 100 Mbps but the receiver can process data at only 1 Mbps, the transmission will overload the receiver and some data will be lost.

### 1.4.1 Standards

Standards are essential in creating and maintaining an open and competitive market for equipment manufacturers and in guaranteeing national and international interoperability of data and telecommunications technology and processes.

**Development of standards:** There are a number of advantages and disadvantages to the standards-making process. We list here the most striking ones. The principal advantages of standards are the following:

1. A standard assures that there will be a large market for a particular piece of equipment or software. This encourages mass production and, in some cases, the use of large-scale-integration (LSI) or very-large-scale-integration (VLSI) techniques, resulting in lower costs.
2. A standard allows products from multiple vendors to communicate, giving the purchaser more flexibility in equipment selection and use.

The principal disadvantages are these:

1. A standard tends to freeze the technology. By the time a standard is developed, subjected to review and compromise, and promulgated, more efficient techniques are possible.
2. There are multiple standards for the same thing. This is not a disadvantage of standards *per se*, but of the current way things are done. Fortunately, in recent

years the various standards-making organizations have begun to cooperate more closely. Nevertheless, there are still areas where multiple conflicting standards exist.

Data communication standards fall into two categories: *de facto* (meaning "by fact" or "by convention") and *de jure* (meaning "by law" or "by regulation").

- **De facto.** Standards that have not been approved by an organized body but have been adopted as standards through widespread use are *de facto* standards. *De facto* standards are often established originally by manufacturers who seek to define the functionality of a new product or technology.
- **De jure.** Those standards that have been legislated by an officially recognized body are *de jure* standards.

## 1.4.2 Standards Organizations

Standards are developed through the cooperation of standards creation committees, forums, and government regulatory agencies.

### *1.4.2.1 Standards Creation Committees*

While many organizations are dedicated to the establishment of standards, data telecommunications in North America rely primarily on those published by the following:

- **International Organization for Standardization (ISO).** The ISO is a multinational body whose membership is drawn mainly from the standards creation committees of various governments throughout the world. The ISO is active in developing cooperation in the realms of scientific, technological, and economic activity.
- **International Telecommunication Union-Telecommunication Standards Sector (ITU-T).** By the early 1970s, a number of countries were defining national standards for telecommunications, but there was still little international compatibility. The United Nations responded by forming, as part of its International Telecommunication Union (ITU), a committee, the Consultative Committee for International Telegraphy and Telephony (CCITT). This committee was devoted to the research and establishment of standards for telecommunications in general and for phone and data systems in particular. On March 1, 1993, the name of this committee was changed to the International Telecommunication Union Telecommunication Standards Sector (ITU-T).
- **American National Standards Institute (ANSI).** Despite its name, the American National Standards Institute is a completely private, nonprofit corporation not affiliated with the U.S. federal government. However, all ANSI activities are undertaken with the welfare of the United States and its citizens occupying primary importance.
- **Institute of Electrical and Electronics Engineers (IEEE).** The Institute of Electrical and Electronics Engineers is the largest professional engineering society in the world. International in scope, it aims to advance theory, creativity, and product quality in the fields of electrical engineering, electronics, and radio as well as in all related branches of engineering. As one of its goals, the IEEE oversees the development and adoption of international standards for computing and communications.

- **Electronic Industries Association (EIA).** Aligned with ANSI, the Electronic Industries Association is a nonprofit organization devoted to the promotion of electronics manufacturing concerns. Its activities include public awareness education and lobbying efforts in addition to standards development. In the field of information technology, the EIA has made significant contributions by defining physical connection interfaces and electronic signaling specifications for data communication.

#### *1.4.2.2 Forums*

Telecommunications technology development is moving faster than the ability of standards committees to ratify standards. Standards committees are procedural bodies and by nature slow-moving. To accommodate the need for working models and agreements and to facilitate the standardization process, many special-interest groups have developed **forums** made up of representatives from interested corporations. The forums work with universities and users to test, evaluate, and standardize new technologies. By concentrating their efforts on a particular technology, the forums are able to speed acceptance and use of those technologies in the telecommunications community. The forums present their conclusions to the standards bodies.

#### *1.4.2.3 Regulatory Agencies*

All communications technology is subject to regulation by government agencies such as the **Federal Communications Commission** (FCC) in the United States. The purpose of these agencies is to protect the public interest by regulating radio, television, and wire/cable communications. The FCC has authority over interstate and international commerce as it relates to communications.

## 1.5 NETWORK MODELS

As computer networks have proliferated, so the need to communicate between users located on different networks has emerged. Such intercommunicating computer systems may be termed distributed computer systems and are required to process information and pass it between each other.

A new movement was, therefore, born. It was called the o*pen system movement*, which called for computer hardware and software manufacturers to come up with a way for this to happen. But to make this possible, standardization of equipment and software was needed. To help in this effort and streamline computer communication, the International Standards Organization (ISO) developed the Open System Interconnection (OSI) model. The OSI is an open architecture model that functions as the network communication protocol standard, although it is not the most widely used one. The Transport Control Protocol/Internet Protocol (TCP/IP) model, a rival model to OSI, is the most widely used. Both OSI and TCP/IP models use two protocol stacks, one at the source element and the other at the destination element.

In this section, we give a general idea of the layers of a network and discuss the functions of each. Detailed descriptions of these layers follow in later chapters.

### 1.5.1 The OSI Model

Established in 1947, the International Standards Organization (ISO) is a multinational body dedicated to worldwide agreement on international standards. 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 not a protocol; it is a model for understanding and designing a network architecture that is flexible, robust, and interoperable.

A widely accepted structuring technique, and the one chosen by ISO, is layering. The communications functions are partitioned into a hierarchical set of layers. Each layer performs a related subset of the functions required to communicate with another system, relying on the next-lower layer to perform more primitive functions, and to conceal the details of those functions, as it provides services to the next-higher layer. Ideally, the layers should be defined so that changes in one layer do not require changes in the other layers. Thus, we have decomposed one problem into a number of more manageable sub problems.

The task of ISO was to define a set of layers and to delineate the services performed by each layer. The partitioning should group functions logically, and should have enough layers to make each one manageably small, but should not have so many layers that the processing overhead imposed by their collection is burdensome. The principles (ISO 7498) that guided the design effort are summarized below. The resulting reference model has seven layers.

1. Do not create so many layers as to make the system engineering task of describing and integrating the layers more difficult than necessary.
2. Create a boundary at a point where the description of services can be small and the number of interactions across the boundary are minimized.
3. Create separate layers to handle functions that are manifestly different in the process performed or the technology involved.
4. Collect similar functions into the same layer.
5. Select boundaries at a point which past experience has demonstrated to be successful.
6. Create a layer of easily localized functions so that the layer could be totally redesigned and its protocols changed in a major way to take advantage of new advances in architecture, hardware or software technology without changing the services expected from and provided to the adjacent layers.
7. Create a boundary where it may be useful at some point in time to have the corresponding interface standardized.
8. Create a layer where there is a need for a different level of abstraction in the handling of data, for example morphology, syntax, semantic.
9. Allow changes of functions or protocols to he made within a layer without affecting other layers.

10. Create for each layer boundaries with its upper and lower layer only.
11. Similar principles have been applied to sub layering: Create further sub grouping and organization of functions to form sub layers within a layer in cases where distinct communication services need it.
12. Create, where needed, two or more sub layers with a common, and therefore minimal functionality to allow interface operation with adjacent layers.
13. Allow by-passing of sub layers.

The ISO's OSI seven-layer reference model is shown in Fig. 1.11. The reference model has been developed based upon some of the principles discussed in general terms in the previous section. The seven layers and their boundaries have attempted to build upon other models which have proved successful and in such a way as to optimize the transfer of information between layers.

Layer 1, the physical layer, defines the electrical, mechanical and functional interface between a DCE and the transmission medium to enable bits to be transmitted successfully. The layer is always implemented in hardware. A common example used extensively in modems is the ITU-T's V.24 serial interface. No error control exists at layer 1 but line coding may be incorporated in order to match data signals to certain properties of the communication channel.

Layer 2 is the data link layer, the function of which is to perform error-free, reliable transmission of data over a link. Link management procedures allow for the setting up and disconnection of links as required for communication. Having established a connection, error detection, and optionally error correction, is implemented to ensure that the data transfer is reliable. Flow control is also performed to provide for the orderly flow of data (normally in the form of packets) and to ensure that it is not lost or duplicated during transmission.

Layer 3 is the network layer, whose principal task is to establish, maintain and terminate connections to support the transfer of information between end systems via one, or more, intermediate communication networks. It is the only layer concerned with routing, offering addressing schemes which allow users to refer unambiguously to each other. Apart from the control of connections and routing, the layer, by engaging in a dialogue with the network, offers other services such as a user requesting a certain quality of service or reset and synchronization procedures.

Layer 4 is the transport layer and separates the function of the higher layers, layers 5, 6 and 7, from the lower layers already discussed. It hides the complexities of data communications from the higher layers which are predominantly concerned with supporting applications. The layer provides a reliable end-to-end service for the transfer of messages irrespective of the underlying network. To fulfil this role, the transport layer selects a suitable communications network which provides the required quality of service. Some of the factors which the layer would consider in such selection are throughput, error rate and delay. Furthermore, the layer is responsible for splitting up messages into a series of packets of suitable size for onward transmission through the selected communications network.

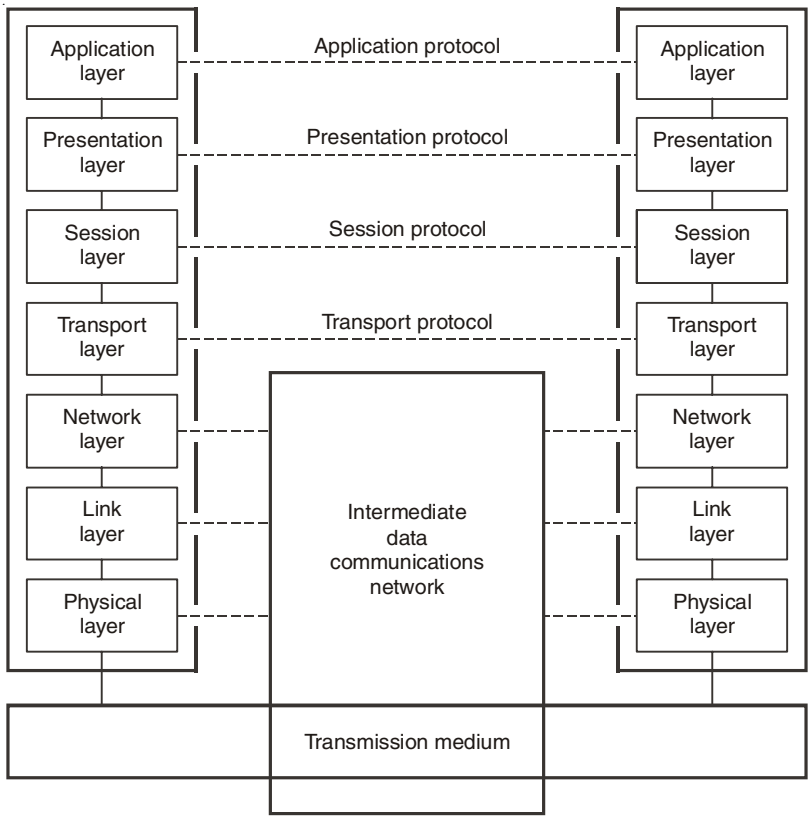



**Fig. 1.11.** *OSI Reference Model.*

Layer 5, the session layer, is responsible for establishing and maintaining a logical connection. This may include access controls such as log-on and password protection. Secondly, the session layer performs a function known as **dialogue management**. This is merely a protocol used to order communication between each party during a session. It may be best explained by way of an example. Consider an enquiry/response application such as is used for airline ticket booking systems. Although two-way communication is necessary for such an interactive application it need not be simultaneous. Suppose that the connection only provides communication in one direction at a time. The protocol must therefore regulate the direction of communication at any one instant. If, however, full simultaneous two-way communication is available then little dialogue management is required save some negotiation at set-up time. The third, and most important, function of the session layer is that of recovery (or synchronization). Synchronizing points are marked periodically throughout the period of dialogue. In the event of a failure, dialogue can return to a synchronizing point, restart and continue from that point (using back-up facilities) as though no failure had occurred.

Layer 6 is the presentation layer and presents data to the application layer in a form which it is able to understand. To that end, it performs any necessary code and/or data format conversion. In this way, there is no necessity for the application layer to be aware

of the code used in the peer-to-peer communication at the presentation layer. This means that in practice, users may operate with entirely different codes at each end and which may in turn be different again from the code used across the network for intercommunication. Encryption may also be added at layer 6 for security of messages. Encryption converts the original data into a form which ideally should be unintelligible to any unauthorized third party. Such messages may usually only be decrypted by knowledge of a **key** which of course must be kept secure.

Layer 7, the application layer, gives the end-user access to the OSI environment. This means that the layer provides the necessary software to offer the user's application programs a set of network services, for example an e-mail service. It is effectively the junction between the user's operating system and the OSI network software. In addition, layer 7 may include network management, diagnostics and statistics gathering, and other monitoring facilities.

Most standards activity has centred on the lower layers to support communications networks and their interfaces, for example ITU-T's X.25 recommendation for packet switched network operation addresses layers 1, 2 and 3, only. ISO standards have more recently addressed this imbalance with standards for some applications being available at all seven layers to support a truly open system interconnection.

### 1.5.2 Standardization Within the OSI Framework

The principal motivation for the development of the OSI model was to provide a framework for standardization. Within the model, one or more protocol standards can be developed at each layer. The model defines, in general terms, the functions to be performed at that layer and facilitates the standards-making process in two ways:

1. Because the functions of each layer are well-defined, standards can be developed independently and simultaneously for each layer, thereby speeding up the standards-making process.
2. Because the boundaries between layers are well-defined, changes in standards in one layer need not affect already existing software in another layer; this makes it easier to introduce new standards.

Figure 1.12 illustrates the use of the OSI model as such a framework. The overall communications function is decomposed into seven distinct layers, using the principles outlined in previous section.

These principles essentially amount to the use of modular design. That is, the overall function is broken up into a number of modules, making the interfaces between modules as simple as possible. In addition, the design principle of information-hiding is used: Lower layers are concerned with greater levels of detail; upper layers are independent of these details. Within each layer, there exist both the service provided to the next higher layer and the protocol to the peer layer in other systems.

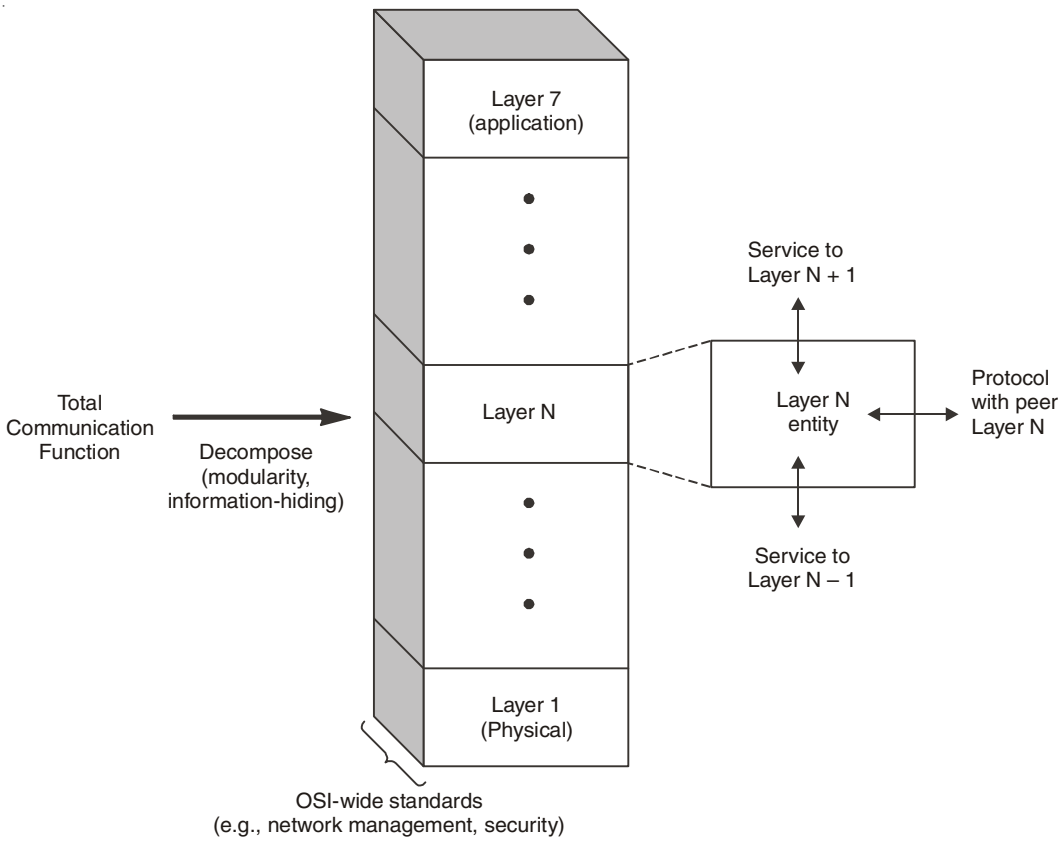



**Fig. 1.12.** *The OSI architecture as a framework for standardization.*

Figure 1.13 shows more specifically the nature of the standardization required at each layer. Three elements are key:

**Protocol specification.** Two entities at the same layer in different systems cooperate and interact by means of a protocol. Because two different open systems are involved, the protocol must be specified precisely; this includes the format of the protocol data units exchanged, the semantics of all fields, and the allowable sequence of PDUs.

**Service definition.** In addition to the protocol or protocols that operate at a given layer, standards are needed for the services that each layer provides to the next-higher layer. Typically, the definition of services is equivalent to a functional description that defines *what* services are provided, but not *how* the services are to be provided.

**Addressing.** Each layer provides services to entities at the next-higher layer. These entities are referenced by means of a service access point (SAP). Thus, a network service access point (NSAP) indicates a transport entity that is a user of the network service.

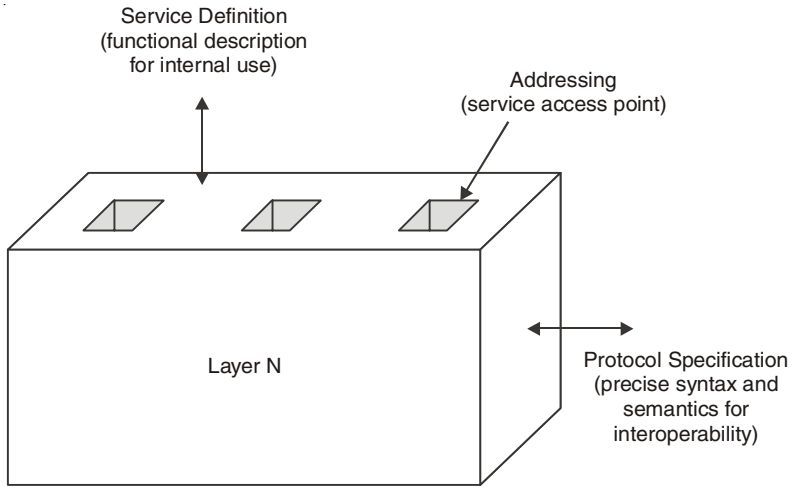



**Fig. 1.13.** *Layer-specific standards.*

The need to provide a precise protocol specification for open systems is self evident. The other two items listed above warrant further comment. With respect to service definitions, the motivation for providing only a functional definition is as follows. First, the interaction between two adjacent layers takes places within the confines of a single open system and is not the concern of any other open system. Thus, as long as peer layers in different systems provide the same services to their next-higher layers the details of how the services are provided may differ from one system to another without loss of interoperability. Second, it will usually be the case that adjacent layers are implemented on the same processor. In that case, we would like to leave the system programmer free to exploit the hardware and operating system to provide an interface that is as efficient as possible.

### 1.5.3 Service Primitives and Parameters

The services between adjacent layers in the OSI architecture are expressed in terms of *primitives* and *parameters*. A primitive specifies the function to be performed, and a parameter is used to pass data and control information. The actual form of a primitive is implementation-dependent; an example is a procedure call. Four types of primitives are used in standards to define the interaction between adjacent layers in the architecture.

#### *REQUEST*

A primitive issued by a service user to invoke some service and to pass the parameters needed to fully specify the requested service.

#### *INDICATION*

A primitive issued by a service provider to either

1. indicate that a procedure has been invoked by the peer service user on the connection and to provide the associated parameters, or
2. notify the service user of a provider-initiated action.

***RESPONSE***

A primitive issued by a service user to acknowledge or complete some procedure previously invoked by an indication to that user.

***CONFIRM***

A primitive issued by a service provider to acknowledge or complete some procedure previously invoked by a request by the service user.

## 1.5.4 Layers in the OSI Model

In this section we briefly describe the functions of each layer in the OSI model.

### *1.5.4.1 Physical Layer*

The physical layer coordinates the functions required to carry a bit stream over a physical medium. It deals with the mechanical and electrical specifications of the interface and transmission medium. It also defines the procedures and functions that physical devices and interfaces have to perform for transmission to Occur. Figure 1.14 shows the position of the physical layer with respect to the transmission medium and the data link layer.

The physical layer is also concerned with the following:

- **Physical characteristics of interfaces and medium.** The physical layer defines the characteristics of the interface between the devices and the transmission medium. It also defines the type of transmission medium.
- **Representation of bits.** The physical layer data consists of a stream of bits (sequence of 0s or 1s) with no interpretation. To be transmitted, bits must been coded into signals—electrical or optical. The physical layer defines the type of encoding (how 0s and 1s are changed to signals).
- **Data rate.** The transmission rate-the number of bits sent each second-is also defined by the physical layer. In other words, the physical layer defines the duration of a bit, which is how long it lasts.
- **Synchronization of bits.** The sender and receiver not only must use the same bit rate but also must be synchronized at the bit level. In other words, the sender and the receiver clocks must be synchronized.
- **Line configuration.** The physical layer is concerned with the connection of devices to the media. In a point-to-point configuration, two devices are connected through a dedicated link. In a multipoint configuration, a link is shared among several devices.
- **Physical topology.** The physical topology defines how devices are connected to make a network. Devices can be connected by using a mesh topology (every device is connected to every other device), a star topology (devices are connected through a central device), a ring topology (each device is connected to the next, forming a ring), a bus topology (every device is on a common link), or a hybrid topology (this is a combination of two or more topologies).
- **Transmission mode.** The physical layer also defines the direction of transmission between two devices: simplex, half-duplex, or full-duplex. In simplex mode, only

one device can send; the other can only receive. The simplex mode is a one-way communication. In the half-duplex mode, two devices can send and receive, but not at the same time. In a full-duplex (or simply duplex) mode, two devices can send and receive at the same time.

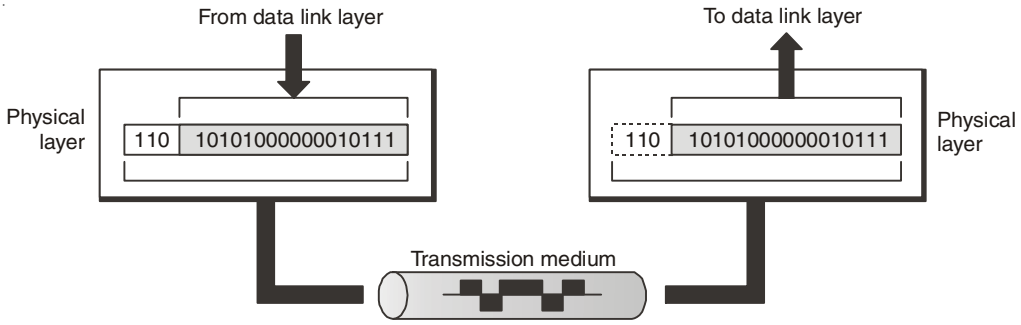



**Fig. 1.14.** *Physical Layer.*

### *1.5.4.2 Data Link Layer*

The data link layer transforms the physical layer, a raw transmission facility, to a reliable link. It makes the physical layer appear error-free to the upper layer (network layer). Figure 1.15 shows the relationship of the data link layer to the network and physical layers.

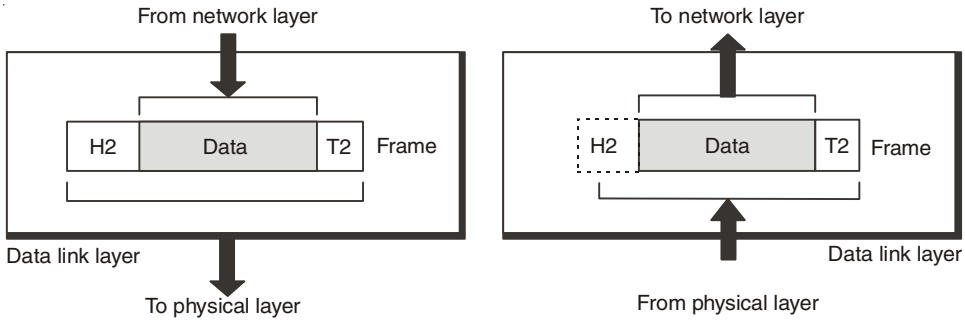



**Fig. 1.15.** *Data Link layer.*

Other responsibilities of the data link layer include the following:

- **Framing.** The data link layer divides the stream of bits received from the network layer into manageable data units called frames.
- **Physical addressing.** If frames are to be distributed to different systems on the network, the data link layer adds a header to the frame to define the sender and/or receiver of the frame. If the frame is intended for a system outside the sender's network, the receiver address is the address of the device that connects the network to the next one.
- **Flow control.** If the rate at which the data are absorbed by the receiver is less than the rate at which data are produced in the sender, the data link layer imposes a flow control mechanism to avoid overwhelming the receiver.

- **Error control.** The data link layer adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged or lost frames. It also uses a mechanism to recognize duplicate frames. Error control is normally achieved through a trailer added to the end of the frame.
- **Access control.** When two or more devices are connected to the same link, data link layer protocols are necessary to determine which device has control over the link at any given time.

#### *1.5.4.3 Network Layer*

The network layer is responsible for the source-to-destination delivery of a packet, possibly across multiple networks (links). Whereas the data link layer oversees the delivery of the packet between two systems on the same network (links), the network layer ensures that each packet gets from its point of origin to its final destination.

If two systems are connected to the same link, there is usually no need for a network layer. However, if the two systems are attached to different networks (links) with connecting devices between the networks (links), there is often a need for the network layer to accomplish source-to-destination delivery. Figure 1.16 shows the relationship of the network layer to the data link and transport layers.

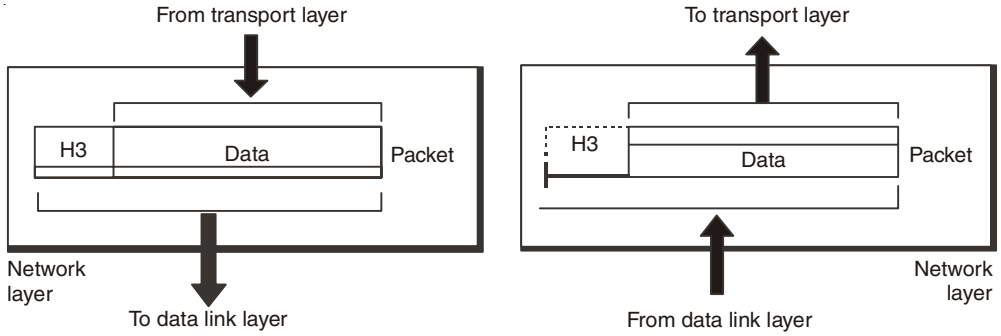



**Fig. 1.16.** *Network layer.*

Other responsibilities of the network layer include the following:

- **Logical addressing.** The physical addressing implemented by the data link layer handles the addressing problem locally. If a packet passes the network boundary, we need another addressing system to help distinguish the source and destination systems. The network layer adds a header to the packet coming from the upper layer that, among other things, includes the logical addresses of the sender and receiver.
- **Routing.** When independent networks or links are connected to create *internetworks* (network of networks) or a large network, the connecting devices (called *routers or switches) route or switch the packets to their final destination. One of the functions of the network layer is to provide this mechanism*

#### *1.5.4.4 Transport Layer*

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

source-to-destination delivery of individual packets, it does not recognize any relationship between those packets. It treats each one independently, as though each piece belonged to a separate message, whether or not it does. The transport layer, on the other hand, ensures that the whole message arrives intact and in order, overseeing both error control and flow control at the source-to-destination level. Figure 1.17 shows the relationship of the transport layer to the network and session layers.

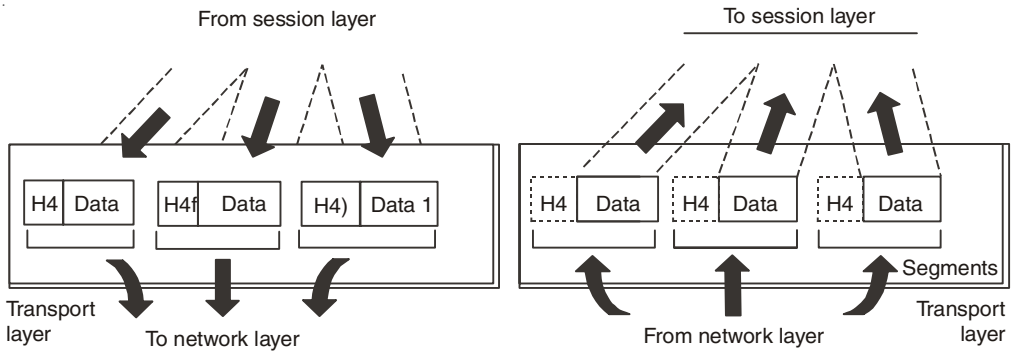



**Fig. 1.17.** *Transport Layer.*

Other responsibilities of the transport layer include the following:

- **Service-point addressing.** Computers often run several programs at the same time. For this reason, source-to-destination delivery means delivery not only from one computer to the next but also from a specific process (running program) on one computer to a specific process (running program) on the other. The transport layer header must therefore include a type of address called a *service-point address* (or port address). The network layer gets each packet to the correct computer; the transport layer gets the entire message to the correct process on that computer.
- **Segmentation and reassembly.** A message is divided into transmittable segments, with each segment containing a sequence number. These numbers enable the transport layer to reassemble the message correctly upon arriving at the destination and to identify and replace packets that were lost in transmission.
- **Connection control.** The transport layer can be either connectionless or connection oriented. A connectionless transport layer treats each segment as an independent packet and delivers it to the transport layer at the destination machine. A connection oriented transport layer makes a connection with the transport layer at the destination machine first before delivering the packets. After all the data are transferred, the connection is terminated.
- **Flow control.** Like the data link layer, the transport layer is responsible for flow control. However, flow control at this layer is performed end to end rather than across a single link.
- **Error control.** Like the data link layer, the transport layer is responsible for error control. However, error control at this layer is performed process-to process rather than across a single link. The sending transport layer makes sure that the entire message arrives at the receiving transport layer without error (damage, loss, or duplication). Error correction is usually achieved through retransmission.

### 1.5.4.5 Session Layer

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

- **Dialog control.** The session layer allows two systems to enter into a dialog. It allows the communication between two processes to take place in either halfduplex (one way at a time) or full-duplex (two ways at a time) mode.
- **Synchronization.** The session layer allows a process to add checkpoints, or synchronization points, to a stream of data.

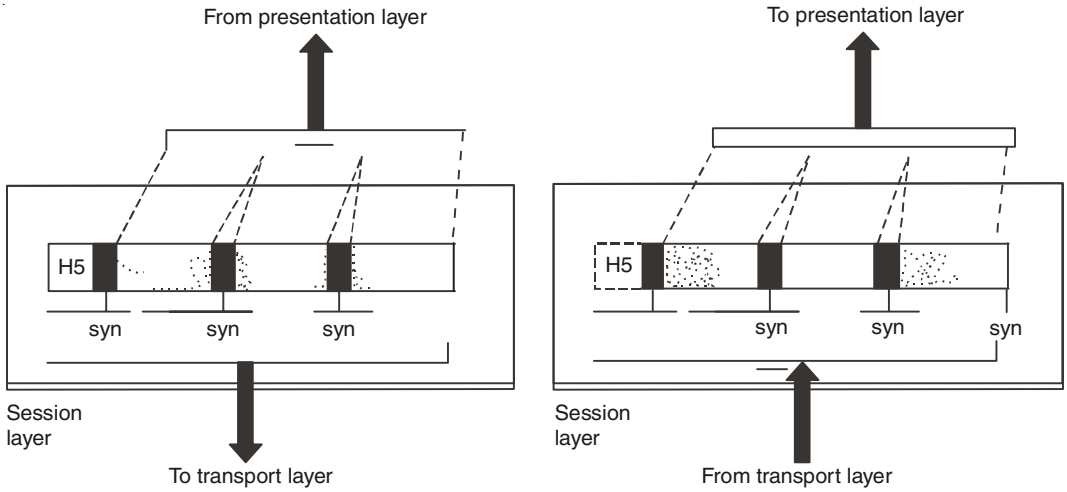



**Fig. 1.18.** *Session Layer.*

### 1.5.4.6. Presentation Layer

The presentation layer is concerned with the syntax and semantics of the information exchanged between two systems. Figure 1.19 shows the relationship between the presentation layer and the application and session layers.

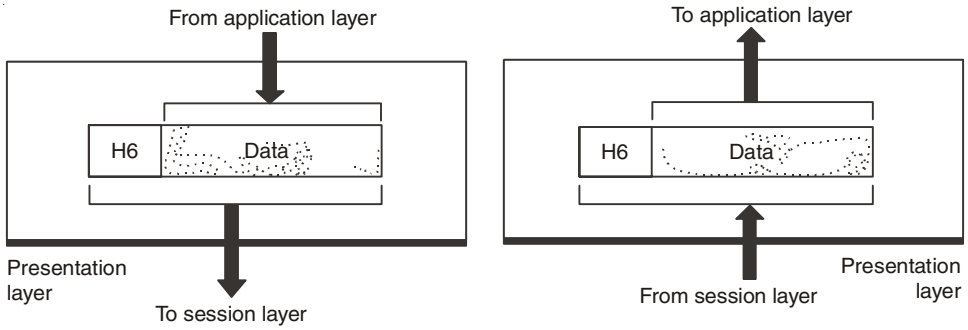



**Fig. 1.19.** *Presentation Layer.*

Specific responsibilities of the presentation layer include the following:

- **Translation.** The processes (running programs) in two systems are usually exchanging information in the form of character strings, numbers, and so on. The information must be changed to bit streams before being transmitted. Because different computers use different encoding systems, the presentation layer is responsible for interoperability between these different encoding methods. The presentation layer at the sender changes the information from its sender-dependent format into a common format. The presentation layer at the receiving machine changes the common format into its receiver-dependent format.
- **Encryption.** To carry sensitive information, a system must be able to ensure privacy. Encryption means that the sender transforms the original information to another form and sends the resulting message out over the network. Decryption reverses the original process to transform the message back to its original form.
- **Compression.** Data compression reduces the number of bits contained in the information. Data compression becomes particularly important in the transmission of multimedia such as text, audio, and video.

#### *1.5.4.7 Application Layer*

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

### 1.5.5 The TCP/IP Reference 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. Although we will give a brief history of the ARPANET later, it is useful to mention a few key aspects of it now.

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 internetworking with them, so a 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, after its two primary protocols. It was first defined in (Cerf and Kahn, 1974). A later perspective is given in (Leiner et al., 1985). The design philosophy behind the model is discussed in (Clark, 1988).

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 real-time speech transmission.

#### 1.5.5.1 The Internet Layer

All these requirements led to the choice of a packet-switching network based on a connectionless inter network 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 analogy here is with the (snail) mail system. A person can drop a sequence of international letters into a mail box in one country, and with a little luck, most of them will be delivered to the correct address in the destination country. Probably the letters will travel through one or more international mail gateways along the way, but this is transparent to the users. Furthermore, that each country (*i.e.,* each network) has its own stamps, preferred envelope sizes, and delivery rules is hidden from the users.

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. Figure 1.20 shows this correspondence.

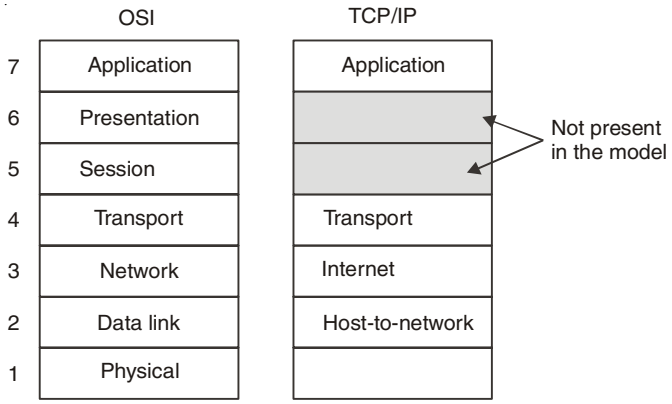



**Fig. 1.20.** *TCP/IP Reference Model.*

#### 1.5.5.2 The 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.

The relation of IP, TCP, and UDP is shown in Fig. 1.21. Since the model was developed, IP has been implemented on many other networks.

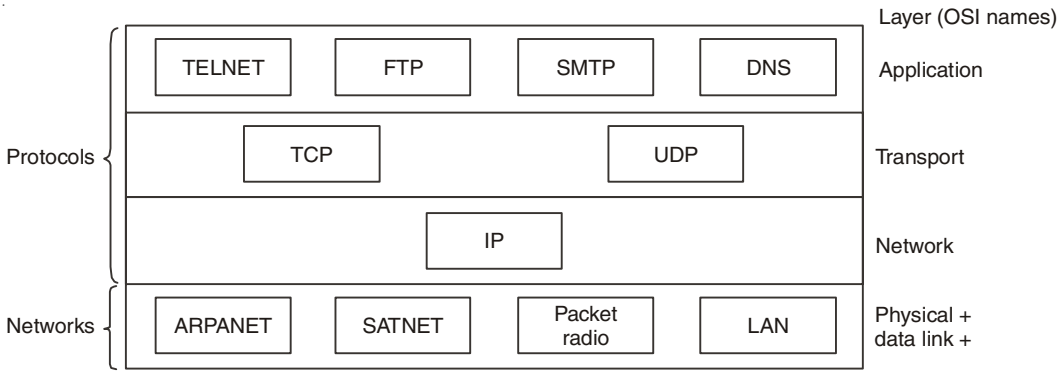



**Fig. 1.21.** *Protocols in TCP/IP Model initially.*

#### *1.5.5.3 The Application Layer*

The TCP/IP model does not have session or presentation layers. No need for them was perceived, so they were not included. Experience with the OSI model has proven this view correct: they are of little use to most applications.

On top of the transport layer is the application layer. It contains all the higher-level protocols. The early ones included virtual terminal (TELNET), file transfer (FTP), and electronic mail (SMTP), as shown in Fig. 1.21. The virtual terminal protocol 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.

#### *1.5.5.4 The Host-to-Network 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.

### *1.5.5.5 Addressing*

Four levels of addresses are used in an internet employing the *TCP/IP* protocols: physical (link) addresses, logical (IP) addresses, port addresses, and specific addresses . Each address is related to a specific layer in the TCP/IP architecture.

**Physical Addresses**

The physical address, also known as the link address, is the address of a node as defined by its LAN or WAN. It is included in the frame used by the data link layer. It is the lowest-level address. The physical addresses have authority over the network (LAN or WAN). The size and format of these addresses vary depending on the network. For example, Ethernet uses a 6-byte (48-bit) physical address that is imprinted on the network interface card (NIC). LocalTalk (Apple), however, has a I-byte dynamic address that changes each time the station comes up.

**Logical Addresses**

Logical addresses are necessary for universal communications that are independent of underlying physical networks. Physical addresses are not adequate in an inter network environment where different networks can have different address formats. A universal addressing system is needed in which each host can be identified uniquely, regardless of the underlying physical network.

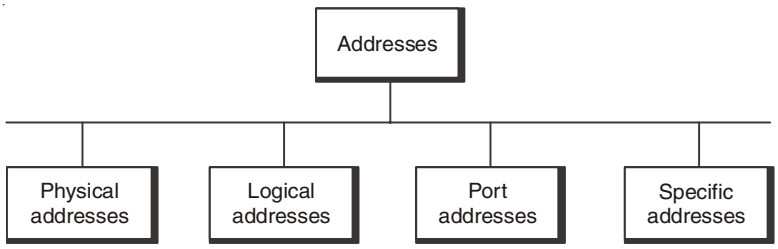



**Fig. 1.22.** *Addresses in TCP/IP.*

**Port Addresses**

The IP address and the physical address are necessary for a quantity of data to travel from a source to the destination host. However, arrival at the destination host is not the final objective of data communications on the Internet.

A system that sends nothing but data from one computer to another is not complete. Today, computers are devices that can run multiple processes at the same time. The end objective of Internet communication is a process communicating with another process. For example, computer A can communicate with computer C by using TELNET. At the same time, computer A communicates with computer B by using the File Transfer Protocol (FTP). For these processes to receive data simultaneously, we need a method to label the different processes.

In other words, they need addresses. In the TCPIIP architecture, the label assigned to a process is called a port address. A port address in TCP/IP is 16 bits in length.

**Specific Addresses**

Some applications have user-friendly addresses that are designed for that specific address. Examples include the e-mail address (for example, president@rb.nic.in) and the Universal Resource Locator (URL) (for example, www.pmofindia.nic.in). The first defines the recipient of an e-mail; the second is used to find a document on the World Wide Web These addresses, however, get changed to the corresponding port and logical addresses by the sending computer.

### 1.5.6 A 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. It is important to note that we are comparing the reference models here, not the corresponding protocol stacks. The protocols themselves will be discussed later.

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.

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.

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.

Turning from philosophical m atters to more specific ones, 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.

## 1.6 NETWORK SERVICES

For a communication network to work effectively, data in the network must be able to move from one network element to another. This can only happen if the network services to move such data work. For data networks, these services fall into two categories:

- Connection services to facilitate the exchange of data between the two network communicating end-systems with as little data loss as possible and in as little time as possible.
- Switching services to facilitate the movement of data from host to host across the length and width of the network mesh of hosts, hubs, bridges, routers, and gateways.

### 1.6.1 Connection Services

How do we get the network transmitting elements to exchange data over the network? Two types of connection services are used: the *connected-oriented* and *connectionless* services.

#### *1.6.1.1 Connected-Oriented Services*

With a connection-oriented service, before a client can send packets with real data to the server, there must be a *three-way handshake.* We will define this three-way handshake in later chapters. But the purpose of a three-way handshake is to establish a session before the actual communication can begin. Establishing a session before data is moved creates a path

of virtual links between the end systems through a network and therefore, guarantees the reservation and establishment of fixed communication channels and other resources needed for the exchange of data before any data is exchanged and as long as the channels are needed. For example, this happens whenever we place telephone calls; before we exchange words, the channels are reserved and established for the duration. Because this technique guarantees that data will arrive in the same order it was sent in, it is considered to be reliable. In short the service offers the following:

- Acknowledgments of all data exchanges between the end-systems,
- Flow control in the network during the exchange, and
- Congestion control in the network during the exchange.

Depending on the type of physical connections in place and the services required by the systems that are communicating, connection-oriented methods may be implemented in the data link layers or in the transport layers of the protocol stack, although the trend now is to implement it more at the transport layer. For example, TCP is a connection-oriented transport protocol in the transport layer. Other network technologies that are connection-oriented include the frame relay and ATMs.

#### *1.6.1.2 Connectionless Service*

In a connectionless service, there is no handshaking to establish a session between the communicating end-systems, no flow control, and no congestion control in the network. This means that a client can start communicating with a server without warning or inquiry for readiness; it simply sends streams of packets, called datagrams, from its sending port to the server's connection port in single point-to-point transmissions with no relationship established between the packets and between the end-systems. There are advantages and of course disadvantages to this type of connection service. In brief, the connection is faster because there is no handshaking which can sometimes be time consuming, and it offers periodic burst transfers with large quantities of data and, in addition, it has simple protocol. However, this service offers minimum services, no safeguards and guarantees to the sender since there is no prior control information and no acknowledgment. In addition, the service does not have the reliability of the connection-oriented method, and offers no error handling and no packets ordering; in addition, each packet self-identifies that leads to long headers, and finally, there is no predefined order in the arrival of packets. Like the connection-oriented method, this service can operate both at the data link and transport layers. For example, UDP, a connectionless service, operates at the transport layer.

### 1.6.2 Network Switching Services

Let us take a detour and briefly discuss data transfer by a switching element. This is a technique by which data is moved from host to host across the length and width of the network mesh of hosts, hubs, bridges, routers, and gateways. This technique is referred to as *data switching.* The type of data switching technique used by a network determines how messages are transmitted between the two communicating elements and across that network. There are two types of data switching techniques: *circuit switching* and *packet switching.*

### 1.6.2.1 Circuit Switching

In circuit switching networks, one must reserve all the resources before setting up a physical communication channel needed for communication. The physical connection, once established, is then used exclusively by the two end-systems, usually subscribers, for the duration of the communication. The main feature of such a connection is that it provides a fixed data rate channel, and both subscribers must operate at this rate. For example, in a telephone communication network, a connected line is reserved between the two points before the users can start using the service. One issue of debate on circuit switching is the perceived waste of resources during the so-called silent periods when the connection is fully in force but not being used by the parties. This situation occurs when, for example, during a telephone network session, a telephone receiver is not hung up after use, leaving the connection still established. During this period, while no one is utilizing the session, the session line is still open.

### 1.6.2.2 Packet Switching

Packet switching networks, on the other hand, do not require any resources to be reserved before a communication session begins. These networks, however, require the sending host to assemble all data streams to be transmitted into packets. If a message is large, it is broken into several packets. Packet headers contain the source and the destination network addresses of the two communicating end-systems. Then, each of the packets is sent on the communication links and across packet switches (routers). On receipt of each packet, the router inspects the destination address contained in the packet. Using its own routing table, each router then forwards the packet on the appropriate link at the maximum available bit rate. As each packet is received at each intermediate router, it is forwarded on the appropriate link interspersed with other packets being forwarded on that link. Each router checks the destination address, if it is the owner of the packet; it then reassembles the packets into the final message. Figure 1.23 shows the role of routers in packet switching networks.

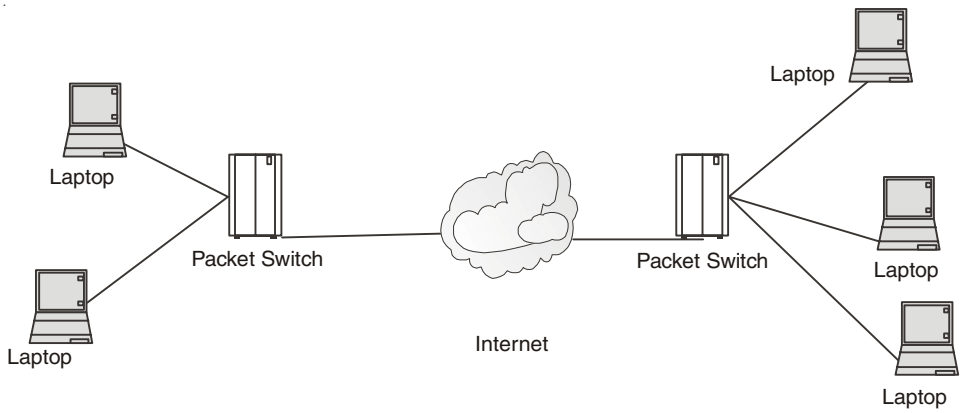



**Fig. 1.23.** *Packet switching Network.*

Packet switches are considered to be store-and-forward transmitters, meaning that they must receive the entire packet before the packet is retransmitted or switched on to the next switch. Because there is no predefined route for these packets, there can be unpredictably

long delays before the full message can be re-assembled. In addition, the network may not dependably deliver all the packets to the intended destination. To ensure that the network has a reliably fast transit time, a fixed maximum length of time is allowed for each packet.

Packet switching networks suffer from a few problems, including the following:

- The rate of transmission of a packet between two switching elements depends on the maximum rate of transmission of the link joining them and on the switches themselves.
- Momentary delays are always introduced whenever the switch is waiting for a full packet. The longer the packet, the longer the delay.
- Each switching element has a finite buffer for the packets. It is thus possible for a packet to arrive only to find the buffer full with other packets. Whenever this happens, the newly arrived packet is not stored but gets lost, a process called *packet dropping.* In peak times, servers may drop a large number of packets. Congestion control techniques use the rate of packet drop as one measure of traffic congestion in a network.

Packet switching networks are commonly referred to as *packet networks* for obvious reasons. They are also called *asynchronous* networks and in such networks, packets are ideal because there is a sharing of the bandwidth, and of course, this avoids the hassle of making reservations for any anticipated transmission. There are two types of packet switching networks:

- *virtual circuit network* in which a packet route is planned, and it becomes a logical connection before a packet is released and
- *datagram network*, which will discussed in later part of this book.

## 1.7. EXAMPLES OF NETWORK ARCHITECTURES

### 1.7.1. IBM's System Network Architecture (SNA)

IBM's System network architecture is a method for unifying network operations. SNA describes the division of network functions into discrete layers and defines protocols and formats for communication between equivalent layers. SNA describes a network in terms of a physical network and a logical network. The physical network consists of a collection of *nodes*: host node, front end communication node, concentration node and terminal node. The host node is the central processor; the front end communication node is concerned with data transmission functions; the concentration node supervises the behaviour of terminals and other peripherals; the terminal node is concerned with the input and output of information through terminal devices. Figure 1.24 depicts a simple SNA network.

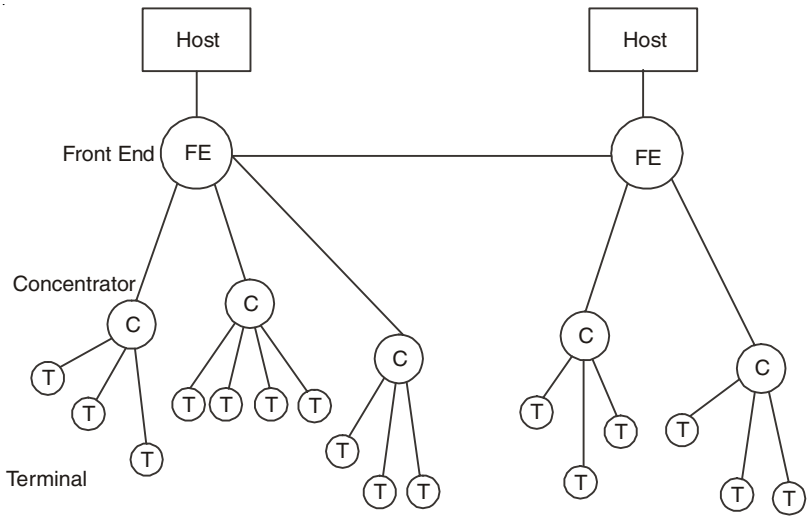



**Fig. 1.24.** *A simple SNA Network.*

The SNA logical network consists of three layers:

(*i*) transmission management;

(*ii*) function management; and

(*iii*) application.

Each node in the SNA physical network may contain any or all of these three layers. Communication between layers is as shown in Fig. 1.25 below. The application layer consists of the user's application programs and is concerned only with the processing of information.

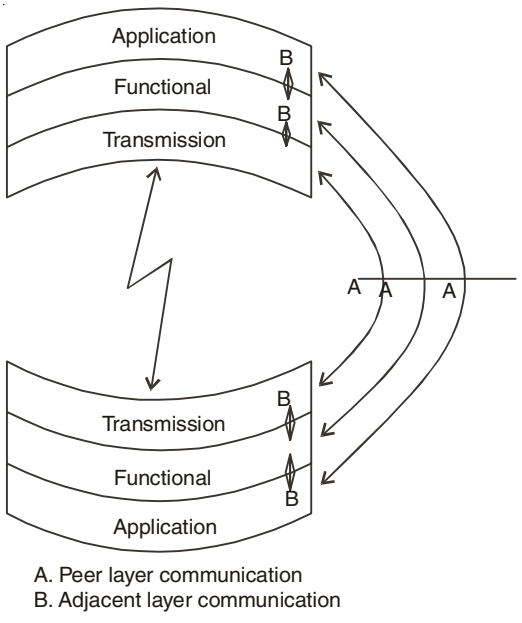



**Fig. 1.25.** *Communication between layers in SNA Network.*

The functional management layers controls the presentation format of information sent from and received by the application layer, *i.e.,* it converts th data into a form convenient to the user. The transmission management layer controls movement of user data through the network. It involves routing, scheduling and transmission functions. This layer exists in every intermediate node through which the data units flow and may utilise a variety of physical connections and protocols between the nodes of an SNA network.

Each node contains one or more *Network Addressable Units* (NAU). There are three types of NAU. A *Logical Unit* (LU) is a NAU which users use to address their process. A *Physical Unit* (PU) is a NAU which the network uses to address a physical device, without reference to which processes are using it. The third type of NAU is the *System Services Control Point* (SSCP) which has control over all front ends, remote concentrators and terminals attached to the host. The three types of NAU communicate with each other by invoking the services of the transmission management layer. The *physical link control* protocol takes care of electrical transmission of data bits from one node to another. The *data link control* protocol constructs frames from the data stream, detecting and recovering from transmission errors. This level 2 protocol is called SDLC (Synchronous Data Link Control) which we will look at later. The *path control* protocol performs the path-selection and congestion control functions within the subnet. The *transmission control* protocol initiates, recovers and terminates transport connections (called sessions) in SNA. It also controls the flow of data to and from other NAUs.

The SNA and ISO models do not correspond closely. However, a rough comparison of the architecture control levels is shown in Fig. 1.26.

<table>
<tr><th>Layer</th><th>OSI</th><th>SNA</th><th>DECNET</th></tr>
<tr><td>7</td><td>Application</td><td>End User</td><td rowspan="2">Application</td></tr>
<tr><td>6</td><td>Presentation</td><td>NAU services</td></tr>
<tr><td>5</td><td>Session</td><td>Data Flow Control</td><td>(none)</td></tr>
<tr><td>4</td><td>Transport</td><td rowspan="2">Transmission Control</td><td>Network Services</td></tr>
<tr><td>3</td><td>Network</td><td rowspan="2">Path Control</td><td>Transport</td></tr>
<tr><td>2</td><td>Data Link</td><td>Data Link</td><td>Data Link</td></tr>
<tr><td>1</td><td>Physical</td><td>Physical</td><td>Physical</td></tr>
</table>

**Fig. 1.26.** *Comparison of OSI/SNA/DECNET.*

### 1.7.2. DECNET's DNA (Digital Network Architecture)

A DECNET is just a collection of computers (nodes) whose functions include running user programs, performing packet switching or both. The architecture of DECNET is called *Digital Network Architecture* (DNA).

DNA has five layers. The physical layer, data link control layer, transport layer and network services layer correspond to the lowest four OSI layers. DNA does not have a session layer (layer 5) and its application layer corresponds to a mixture of the presentation and application layers in OSI; see Fig. 1.26 notice that DNA's level 3 is called the transport layer, not level 4 as in the OSI Model.

## Review Questions

1. Identify the five components of a data communications system.
2. What are the advantages of distributed processing?
3. What are the three criteria necessary for an effective and efficient network?
4. What are the advantages of a multipoint connection over a point-to-point connection?
5. What are the two types of line configuration?
6. What is an internet?
7. Why are protocols needed?
8. Why are standards needed?
9. List the layers of the Internet model.
10. Which layers in the Internet model are the network support layers?
11. Which layer in the Internet model is the user support layer?
12. What is the difference between network layer delivery and transport layer delivery?
13. Suppose a computer sends a frame to another computer on a bus topology LAN. The physical destination address of the frame is corrupted during the transmission. What happens to the frame? How can the sender be informed about the situation?
14. Suppose a computer sends a packet at the network layer to another computer somewhere in the Internet. The logical destination address of the packet is corrupted. What happens to the packet? How can the source computer be informed of the situation?

❑❑❑

# CHAPTER 2 THE PHYSICAL LAYER

*The hand that hath made you fair hath made you good.*

**—William Shakespeare**

## 2.1 INTRODUCTION

In this chapter we will look at the lowest layer depicted in the hierarchy of OSI Model. It defines the mechanical, electrical, and timing interfaces to the network. It is the layer that actually interacts with the transmission media, the physical part of the network that connects network components together. This layer is involved in physically carrying information from one node in the network to the next.

The physical layer has complex tasks to perform. One major task is to provide services for the data link layer. The data in the data link layer consists of 0's and 1's organized into frames that are ready to be sent across the transmission medium. This stream of 0's and 1's must first be converted into another entity: signals. One of the services provided by the physical layer is to create a signal that represents this stream of bits. The physical layer must also take care of the physical network, the transmission medium. The transmission medium is a passive entity; it has no internal program or logic for control like other layers.

The material covered in this chapter will provide background information on the key transmission technologies used in modern networks.

## 2.2 THE THEORETICAL BASIS FOR DATA COMMUNICATION

Information can be transmitted on wires by varying some physical property such as voltage or current. By representing the value of this voltage or current as a single-valued function of time, $f(t)$, we can model the behavior of the signal and analyze it mathematically. This analysis is the subject of the following sections.

### 2.2.1 Fourier Analysis

In the early 19th century, the French mathematician Jean-Baptiste Fourier proved that any reasonably behaved periodic function, $g(t)$ with period $T$ can be constructed as the sum of a (possibly infinite) number of sines and cosines:

$$g(t) = \frac{1}{2}c + \sum_{n=1}^{\infty} a_n \sin(2\pi nft) + \sum_{n=t}^{\infty} b_n \cos(2\pi nft) \quad ...(1)$$

where $f = 1/T$ is the fundamental frequency, $a_n$ and $b_n$ are the sine and cosine amplitudes of the $n$th harmonics (terms), and $c$ is a constant. Such a decomposition is called a Fourier series. From the Fourier series, the function can be reconstructed; that is, if the period, $T$, is known and the amplitudes are given, the original function of time can be found by performing the sums of equation.

A data signal that has a finite duration (which all of them do) can be handled by just imagining that it repeats the entire pattern over and over forever (*i.e.,* the interval from $T$ to $2T$ is the same as from 0 to $T$, etc.). The $a_n$ amplitudes can be computed for any given $g(t)$ by multiplying both sides of Eq. 1 by $\sin(2\pi kft)$ and then integrating from 0 to $T$.

$$\int_0^T \sin(2\pi kft)\sin(2\pi nft)\,dt = \begin{cases} 0 \text{ for } k \neq n \\ T/2 \text{ for } k = n \end{cases} \quad ...(2)$$

Since only one term of the summation survives: $a_n$. The $b_n$ summation vanishes completely. Similarly, by multiplying Eq. 1 by $\cos(2\pi kft)$ and integrating between 0 and $T$, we can derive $b_n$. By just integrating both sides of the equation as it stands, we can find $c$. The results of performing these operations are as follows:

$$a_n = \frac{2}{T}\int_0^T g(t)\sin(2\pi nft)\,dt \quad b_n = \frac{2}{T}\int_0^T g(t)\cos(2\pi nft)\,dt \quad c = \frac{2}{T}\int_0^T g(t)\,dt$$

### 2.2.2 Bandwidth-Limited Signals

To see what all this has to do with data communication, let us consider a specific example: the transmission of the ASCII character ''*b*'' encoded in an 8-bit byte. The bit pattern that is to be transmitted is 01100010. The left-hand part of Fig. 2.1(*a*) shows the voltage output by the transmitting computer. The Fourier analysis of this signal yields the coefficients

$$a_n = \frac{1}{\pi n}\left[\cos(\pi n/4) - \cos(3\pi n/4) + \cos(6\pi n/4) - \cos(7\pi n/4)\right]$$

$$b_n = \frac{1}{\pi n}\left[\sin(3\pi n/4) - \sin(\pi n/4) + \sin(7\pi n/4) - \sin(6\pi n/4)\right]$$

$$c = 3/4$$

The root-mean-square amplitudes, for the first few terms are shown on the right-hand side of Fig. 2.1 (*a*). These values are of interest because their squares are proportional to the energy transmitted at the corresponding frequency.

No transmission facility can transmit signals without losing some power in the process. If all the Fourier components were equally diminished, the resulting signal would be reduced in amplitude but not distorted [*i.e.,* it would have the same nice squared-off shape as Fig. 2.1(*a*)]. Unfortunately, all transmission facilities diminish different Fourier components by different amounts, thus introducing distortion. Usually, the amplitudes are transmitted undiminished from 0 up to some frequency $f_c$ [measured in cycles/sec or Hertz (Hz)] with all frequencies above this cutoff frequency attenuated. The range of frequencies transmitted without being strongly attenuated is called the bandwidth. In practice, the cutoff is not really sharp, so often the quoted bandwidth is from 0 to the frequency at which half the power gets through.

The bandwidth is a physical property of the transmission medium and usually depends on the construction, thickness, and length of the medium. In some cases a filter is introduced into the circuit to limit the amount of bandwidth available to each customer. For example, a telephone wire may have a bandwidth of 1 MHz for short distances, but telephone companies add a filter restricting each customer to about 3100 Hz. This bandwidth is adequate for intelligible speech and improves system-wide efficiency by limiting resource usage by customers.

Now let us consider how the signal of Fig. 2.1(*a*) would look if the bandwidth were so low that only the lowest frequencies were transmitted [*i.e.,* if the function were being approximated by the first few terms of Eq. 1. Figure 2.1 (*b*) shows the signal that results from a channel that allows only the first harmonic (the fundamental, *f*) to pass through. Similarly, Fig. 2.1 (*c*)-(*e*) show the spectra and reconstructed functions for higher-bandwidth channels.

Given a bit rate of $b$ bits/sec, the time required to send 8 bits (for example) 1 bit at a time is $8/b$ sec, so the frequency of the first harmonic is $b/8$ Hz. An ordinary telephone line, often called a voice-grade line, has an artificially-introduced cutoff frequency just above 3000 Hz. This restriction means that the number of the highest harmonic passed through is roughly $3000/(b/8)$ or $24{,}000/b$, (the cutoff is not sharp).

For some data rates, the numbers work out as shown in Fig. 2.2. From these numbers, it is clear that trying to send at 9600 bps over a voice-grade telephone line will transform Fig. 2.1 (*a*) into something looking like Fig. 2.1 (*c*), making accurate reception of the original binary bit stream tricky. It should be obvious that at data rates much higher than 38.4 kbps, there is no hope at all for binary signals, even if the transmission facility is completely noiseless. In other words, limiting the bandwidth limits the data rate, even for perfect channels. However, sophisticated coding schemes that make use of several voltage levels do exist and can achieve higher data rates. We will discuss these later in this chapter.

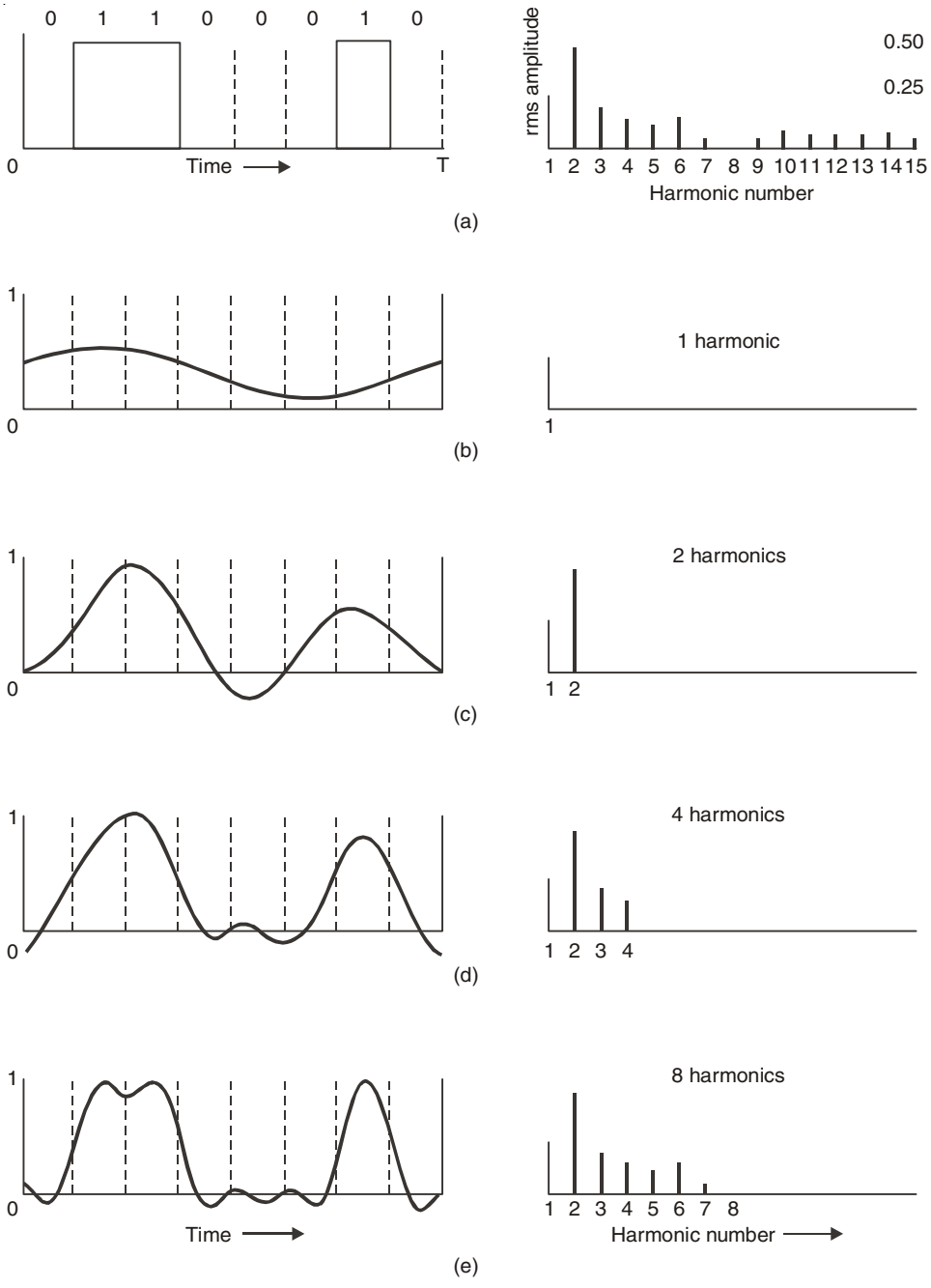



**Fig. 2.1.** *(a) A binary signal and its root-mean-square Fourier amplitudes. (b)-(e) Successive approximations to the original signal.*

| Bps | T (msec) | First harmonic (Hz) | # Harmonics sent |
|---|---|---|---|
| 300 | 26.67 | 37.5 | 80 |
| 600 | 13.33 | 75 | 40 |
| 1200 | 6.67 | 150 | 20 |
| 2400 | 3.33 | 300 | 10 |
| 4800 | 1.67 | 600 | 5 |
| 9600 | 0.83 | 1200 | 2 |
| 19200 | 0.42 | 2400 | 1 |
| 38400 | 0.21 | 4800 | 0 |

**Fig. 2.2.** *Relation between data rate and harmonics.*

### 2.2.3 The Maximum Data Rate of a Channel

As early as 1924, an AT&T engineer, Henry Nyquist, realized that even a perfect channel has a finite transmission capacity. He derived an equation expressing the maximum data rate for a finite bandwidth noiseless channel. In 1948, Claude Shannon carried Nyquist's work further and extended it to the case of a channel subject to random (that is, thermodynamic) noise (Shannon, 1948). We will just briefly summarize their now classical results here.

Nyquist proved that if an arbitrary signal has been run through a low-pass filter of bandwidth $H$, the filtered signal can be completely reconstructed by making only $2H$ (exact) samples per second. Sampling the line faster than $2H$ times per second is pointless because the higher frequency components that such sampling could recover have already been filtered out. If the signal consists of $V$ discrete levels, Nyquist's theorem states:

$$\text{maximum data rate} = 2H \log_2 V \text{ bits/sec}$$

For example, a noiseless 3-kHz channel cannot transmit binary (*i.e.,* two-level) signals at a rate exceeding 6000 bps. So far we have considered only noiseless channels. If random noise is present, the situation deteriorates rapidly. And there is always random (thermal) noise present due to the motion of the molecules in the system. The amount of thermal noise present is measured by the ratio of the signal power to the noise power, called the signal-to-noise ratio. If we denote the signal power by S and the noise power by N, the signal-to-noise ratio is S/N. Usually, the ratio itself is not quoted; instead, the quantity $10 \log_{10}$ S/N is given. These units are called decibels (dB). An S/N ratio of 10 is 10 dB, a ratio of 100 is 20 dB, a ratio of 1000 is 30 dB, and so on. The manufacturers of stereo amplifiers often characterize the bandwidth (frequency range) over which their product is linear by giving the 3-dB frequency on each end. These are the points at which the amplification factor has been approximately halved (because $\log_{10}3$ 0.5).

Shannon's major result is that the maximum data rate of a noisy channel whose bandwidth is H Hz, and whose signal-to-noise ratio is S/N, is given by

maximum number of bits/sec = $H \log_2 (1 + S/N)$

For example, a channel of 3000-Hz bandwidth with a signal to thermal noise ratio of 30 dB (typical parameters of the analog part of the telephone system) can never transmit much more than 30,000 bps, no matter how many or how few signal levels are used and no matter how often or how infrequently samples are taken. Shannon's result was derived from information-theory arguments and applies to any channel subject to thermal noise. Counter examples should be treated in the same category as perpetual motion machines. It should be noted that this is only an upper bound and real systems rarely achieve it.

## 2.3 DATA COMMUNICATION TRANSMISSION TECHNOLOGY

The media through which information has to be transmitted determine the signal to be used. Some media permit only analog signals. Some allow both analog and digital. Therefore, depending on the media type involved and other considerations, the input data can be represented as either *digital* or *analog* signal. In an analog format, data is sent as continuous electromagnetic waves on an interval representing things such as voice and video and propagated over a variety of media that may include copper wires, twisted coaxial pair or cable, fiber optics, or wireless. We will discuss these media soon. In a digital format, on the other hand, data is sent as a digital signal, a sequence of voltage pulses that can be represented as a stream of binary bits. Both analog and digital data can be propagated and many times represented as either analog or digital.

Transmission itself is the propagation and processing of data signals between network elements. The concept of representation of data for transmission, either as analog or digital signal, is called an *encoding scheme.* Encoded data is then transmitted over a suitable transmission medium that connects all network elements.

There are two encoding schemes, *analog* and *digital.* Analog encoding propagates analog signals representing analog data such as sound waves and voice data. Digital encoding, on the other hand, propagates digital signals representing either an analog or a digital signal representing digital data of binary streams by two voltage levels.

### 2.3.1 Analog Encoding of Digital Data

Recall that digital information is in the form of 1s or 0s. To send this information over some analog medium such as the telephone line, for example, which has limited bandwidth, digital data needs to be encoded using modulation and demodulation to produce analog signals. The encoding uses a continuous oscillating wave, usually a sine wave, with a constant frequency signal called a *carrier* signal. The carrier has three modulation characteristics: *amplitude, frequency*, and *phase shift.*

The scheme then uses a *modem*, a modulation–demodulation pair, to modulate and demodulate the data signal based on any one of the three carrier characteristics or a

combination. The resulting wave is between a range of frequencies on both sides of the carrier as shown below:

- *Amplitude* modulation represents each binary value by a different amplitude of the carrier frequency. The absence of or low carrier frequency may represent a 0 and any other frequency then represents a 1. But this is a rather inefficient modulation technique and is therefore used only at low frequencies up to 1200 bps in voice grade lines.
- *Frequency* modulation also represents the two binary values by two different frequencies close to the frequency of the underlying carrier. Higher frequencies represent a 1 and low frequencies represent a 0. The scheme is less susceptible to errors.
- *Phase shift* modulation changes the timing of the carrier wave, shifting the carrier phase to encode the data. A 1 is encoded as a change in phase by 180 degrees and a 0 may be encoded as a 0 change in phase of a carrier signal. This is the most efficient scheme of the three and it can reach a transmission rate of up to 9600 bps.

### 2.3.2 Digital Encoding of Digital Data

In this encoding scheme, which offers the most common and easiest way to transmit digital signals, two binary digits are used to represent two different voltages. Within a computer, these voltages are commonly 0 volt and 5 volts. Another procedure uses two representation codes: *nonreturn to zero level (NRZ-L)*, in which negative voltage represents binary one and positive voltage represents binary zero, and *nonreturn to zero, invert on ones (NRZ-I).* See Figs. 2.3 and 2.4 for an example of these two codes.

In NRZ-L, whenever a 1 occurs, a transition from one voltage level to another is used to signal the information. One problem with NRZ signaling techniques is the requirement of a perfect synchronization between the receiver and transmitter clocks. This is, however, reduced by sending a separate clock signal. There are yet other representations such as the Manchester and differential Manchester, which encode clock information along with the data.

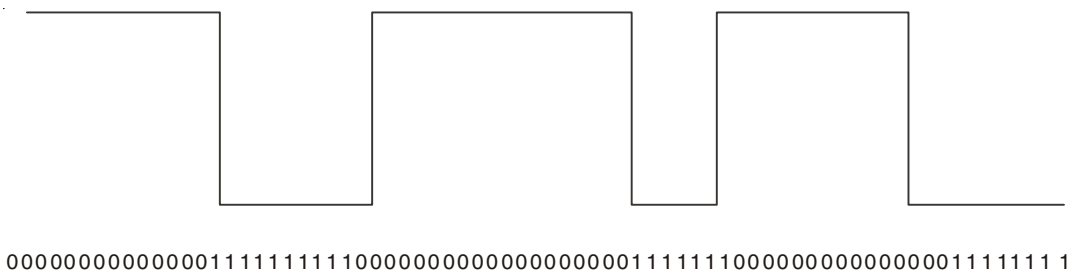
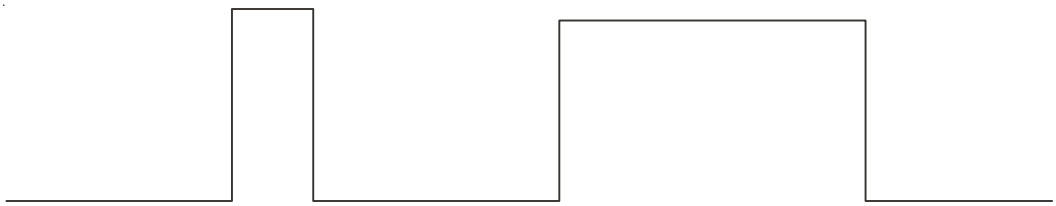



**Fig. 2.3.** *NRZ-L N Nonreturn to zero level representation code.*

One may wonder why go through the hassle of digital encoding and transmission. There are several advantages over its cousin, analog encoding. These include the following:

- Plummeting costs of digital circuitry.
- More efficient integration of voice, video, text, and image.

- Reduction of noise and other signal impairment because of use of repeaters.
- Capacity of channels is utilized best with digital techniques.
- Better encryption and hence better security than in analog transmission.



0000000000000001111100000000000000001111111111111111111000000000000

**Fig. 2.4.** *NRZI Nonreturn to zero Invert on ones representation code.*

### 2.3.3 Multiplexing of Transmission Signals

Quite often during the transmission of data over a network medium, the volume of transmitted data may far exceed the capacity of the medium. Whenever this happens, it may be possible to make multiple signal carriers share a transmission medium. This is referred to as *multiplexing.* There are two ways in which multiplexing can be achieved: time-division multiplexing (TMD) and frequency-division multiplexing (FDM).

In FDM, all data channels are first converted to analog form. Since a number of signals can be carried on a carrier, each analog signal is then modulated by a separate and different carrier frequency, and this makes it possible to recover during the demultiplexing process. The frequencies are then bundled on the carrier. At the receiving end, the demultiplexer can select the desired carrier signal and use it to extract the data signal for that channel in such a way that the bandwidths do not overlap. FDM has an advantage of supporting full-duplex communication.

TDM, on the other hand, works by dividing the channel into time slots that are allocated to the data streams before they are transmitted. At both ends of the transmission, if the sender and receiver agree on the time-slot assignments, then the receiver can easily recover and reconstruct the original data streams. So multiple digital signals can be carried on one carrier by interleaving portions of each signal in time.

## 2.4 DATA COMMUNICATION MEDIA TECHNOLOGY

### 2.4.1 Twisted-pair Cable

The common twisted-pair cable, which many readers are familiar with from wiring a telephone extension in the home, forms the basis for many measurements used in the communications field. Although we may have only one telephone in our home, the twisted-pair cable also forms the basis for large diameter cables that connect up to 25 instruments to switchboards and automatic switching equipment in an office environment.

The twisted-pair cable consists of two insulated conductors that are twisted together to form a transmission line. The pairs are combined in a cable that typically contains 2, 4 or 25-pairs, with the cable wrapped with a protective jacket of plastic or similar material. Two- and four-pair cables are used in both the home and office. From the home, the cable pair will be routed to an end office, whereas in many offices the cable pair will be routed to a switchboard or to a wiring closet.

In an office environment the wiring closet functions as a wire distribution center, permitting telephones on a floor or within a particular geographic area to be cabled to a common point. From the wiring closet a 25-pair cable is typically used to connect up to 25 telephones to a switchboard or electronic switching device known as a private branch exchange (PBX). From the PBX, calls to other telephone numbers connected to the switch are routed through the switch and never leave the customers' premises. Calls destined to telephone numbers not serviced by the PBX are first routed via trunks that connect the organization's PBX to a telephone company office. From there, the call is routed over the public switched telephone network to its destination.

### *Signal degradation*

Regardless of the method of call routing, there are numerous electrical properties associated with the twisted-pair cable that will affect the quality of an electrical signal moving through the cable. Four of the more prominent electrical properties that affect an electrical signal transmitted over a twisted pair are attenuation, capacitance, crosstalk and delay distortion.

➢ **Attenuation**

As an electrical signal travels through a cable, it becomes weaker due to the resistance offered by the cable to flow. This weakness is called attenuation and it refers to the reduction in the amplitude or height of a transmitted signal. In voice communications, attenuation reduces the loudness of a conversation. To compensate for the effect of attenuation, telephone companies install amplifiers in their facilities at selective locations to boost signal levels. Figure 2.5 illustrates the effect of attenuation on an analog signal and the use of an amplifier to rebuild the signal level. When we speak of an analog signal, we are referring to a continuous signal as opposed to a digital signal which is discrete.

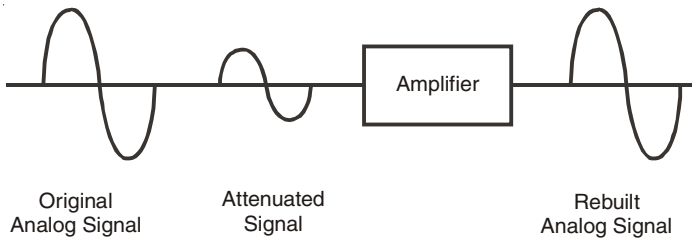



**Fig. 2.5.** *Attenuation of an analog signal. Amplifiers are used by the telephone company to boost the signal strength of an analog signal.*

A second cause of attenuation is the result of the creation of the telephone channel passband by the use of electrical filters. Telephone companies use low and high pass filters to pass only a small subset of the 20000 Hz bandwidth audible to the human ear. As a

result of the use of electrical filters at approximately 300 and 3300 Hz and the fact that high frequencies attenuate more rapidly than low frequencies, the ideal passband of a telephone channel becomes skewed. The result is an increase in attenuation as frequencies increase as well as at both filter cut-off frequencies. Figure 2.6 illustrates the typical amplitude-frequency response across a voice channel.

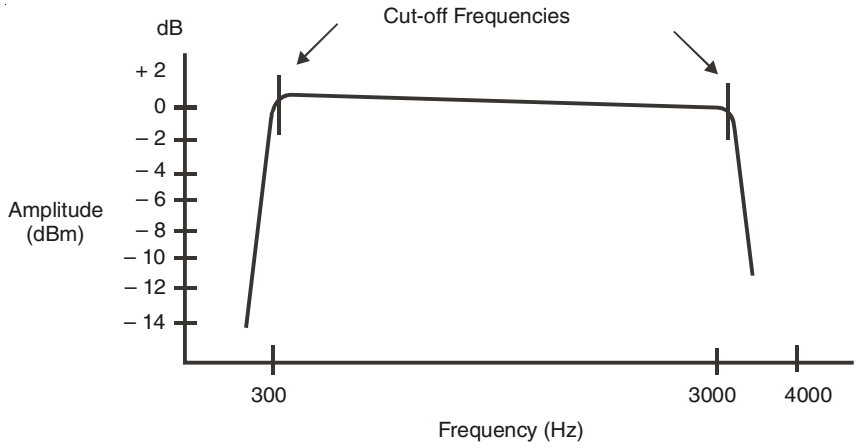



**Fig. 2.6.** *Typical amplitude vs frequency response across a voice channel.*

Due to the use of low pass and high pass filters, a large degree of attenuation occurs both below 300 and above 3300 Hz. In addition, because high frequencies attenuate more rapidly than low frequencies, the amplitude frequency response is non-linear from 300 to 3300 Hz, with the amount of signal attenuation increasing as the frequency increases.

The effect of attenuation upon a digital signal is similar to its effect upon an analog signal. That is, attenuation of a digital signal reduces the height of the square waves that form the signal.

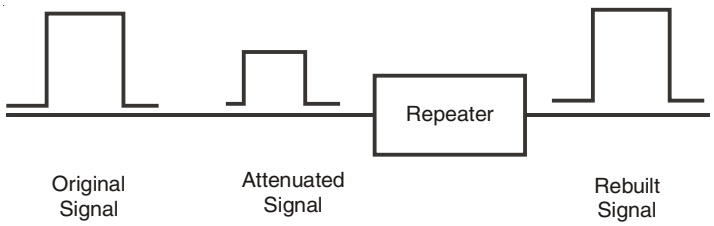



**Fig. 2.7.** *Effect of attenuation on digital signals. A digital repeater samples the line for the occurrence of a pulse and regenerates the pulse at its original height and width.*

In place of amplifiers used on analog circuits, digital repeaters, also called data regenerators, are used on a digital circuit to compensate for the loss in signal strength. The digital repeater as its name implies accepts a pulse and regenerates it at its original height and width, eliminating any previous distortion to the pulse. Figure 2.7 illustrates the attenuation of a digital signal and the use of a digital repeater to 'rebuild' the signals on a digital transmission medium.

➢ **Crosstalk**

The presence on a cable of a signal which originated on a different cable is called crosstalk. Although telephone cables always exhibit a degree of crosstalk, since a signal on an 'excited' pair always induces a signal on a 'quiet' pair, in most instances the effect of crosstalk is negligible. When the effect of crosstalk becomes relatively large, its effect upon both voice and data can become considerable.

In general, crosstalk is proportional to the dielectric constant of a cable. A cable with a lower dielectric constant will have less capacitance than a cable with a higher dielectric constant. Since the amount of capacitance on a cable is proportional to its level of crosstalk, the level of crosstalk is proportional to the dielectric constant of the cable.

## 2.4.2 Coaxial Cable

A coaxial cable consists of an inner conductor and an outer conductor that are insulated from one another. The insulation is called the dielectric. Figure 2.8 illustrates the composition of a coaxial cable.

Within the protective jacket the cable contains two conductors in concentric circles to one another. A solid wire forms the inner conductor, while the outer conductor functions as a shield and is usually grounded.

Coaxial cables can transmit signals ranging in frequency from 1 kHz to 1 GHz per second with little loss, distortion or interference to or from outside signals.

There are many types of coaxial cable that include cable with multiple conductors. However, coax, a term used to denote this type of cable, can normally be classified into those used for baseband and those used for broadband transmission. Baseband refers to the transmission of one signal at a time, and baseband coax normally has a characteristic impedance of 50 Ω. By comparison, broadband refers to the ability to simultaneously transmit two or more signals on a cable by transmitting each signal at a different frequency. Broadband coax is the 75 Ω cable used with CATV systems.

Due to the larger bandwidth of coaxial cable than of the twisted pair, they have a larger data transmission capacity. Typical usage of coaxial cable includes Community Antenna Television (CATV) and Local Area Networks (LANs).

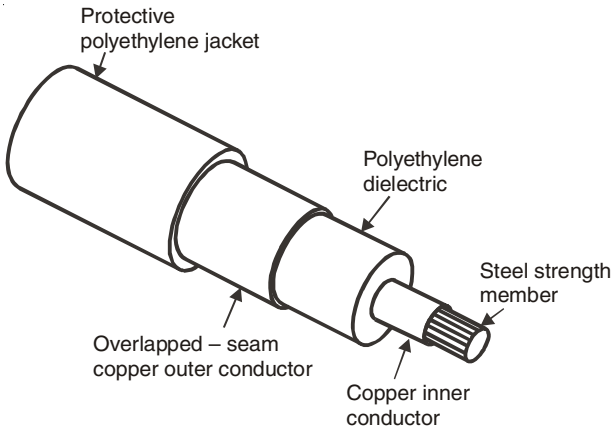



**Fig. 2.8.** *Composition of coaxial cable.*

### 2.4.3 Microwave

Microwave is a line-of-sight transmission medium that provides the bandwidth and capacity of coaxial cable without requiring the laying of a physical cable. Since microwave is a line-of-sight transmission, its use requires the construction of towers which, due to the curvature of the Earth, are limited to distances of 30 miles or less from one another. Figure 2.9 illustrates the use of microwave towers.

The large bandwidth of microwave communications permits communications carriers to simultaneously transmit thousands of calls between offices connected by microwave towers. As you travel around the globe the microwave tower is a common sight, especially in small towns and cities where its height makes it stand out as the communications link of the town or city to the rest of the world.

The transmission of a microwave signal to a point above the Earth forms the basis for satellite communications. A communications satellite contains a number of transponders (transmitters–receivers) which function as a microwave relay in the sky. The satellite receives the microwave signal transmitted from an Earth station and rebroadcasts it back to Earth.

Unlike transmission via a cable, microwave transmission can be affected by sunspots and weather: heavy rain, thunderstorms and dust storms all have an adverse effect upon microwave transmission.

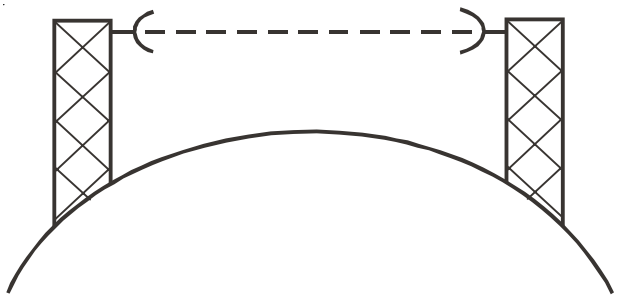


**Fig. 2.9.** *Microwave Transmission.*

During the 1950s and 1960s microwave towers literally dotted the landscape of many countries, providing the most common method for supporting inter-city communications. During the 1970s communications carriers turned to the use of fiber optic cable as a replacement for microwave. Because light transmitted over fiber is immune to electrical disturbances, by the mid—to late 1980s the use of fiber optics resulted in a substantial reduction in the use of microwave towers.

### 2.4.4 Fiber-optic Transmission

Once a laboratory and consumer product curiosity, optical transmission via fiber is now used for low cost, high data rate transmission. The major components of an optical system are similar to a conventional transmission system, requiring a transmitter, transmission medium and receiver. The transmitter, an electrical-to-optical (*E/O*) converter, receives electronic signals and converts them to a series of light pulses. The transmission medium is an optical fiber cable of plastic or glass. The receiver, an optical-to-electric (*O/E*) converter, changes the received light signals into corresponding electrical signals.

The bandwidth of a typical optical fiber can range up to 10 GHz. Normally a series of optical fibers, each shielded and bundled together, are routed between locations. To provide an indication of the transmission capability of an optical fiber, consider the potential use of two strands of fiber in a bundle. Each pair of strands can carry up to 24000 telephone calls, with calls routed in different directions on each fiber strand in a pair as until recently they were used for unidirectional transmission. An equivalent capacity established through the use of metallic twisted pair would require 48000 copper wires to establish the same number of telephone calls.

Although optical fiber has a relatively high bandwidth in comparison to copper, the growth in the use of the Internet and other applications began to tax that bandwidth. Beginning in the late 1990s a technique similar to FDM was used on many optical circuits. Based upon the subdivision of an optical fiber by wavelength, a technique referred to as wavelength division multiplexing (WDM) enables one optical fiber to transport up to 100 or possibly more simultaneous light sources, significantly increasing the transmission capacity of each optical fiber.

There are two general types of optical fiber cable used to transmit information: single-mode and multimode. A single-mode fiber has a core diameter of approximately 8 microns which is approximately the wavelength of light. This results in light flowing on one route through the fiber. In comparison, multimode fiber has a core diameter of approximately 62.5 microns, although 50, 85 and 100 micron cables are available. This diameter is large enough to permit light to travel on different paths, with each path considered a mode of propagation, hence the term multimode. Single-mode fiber is designed to transmit laser-generated light signals at distances up to 20 to 30 miles. This type of optical fiber is primarily used by communications carriers. In comparison, multimode was designed to carry relatively weak light emitting diode (LED) generated signals over relatively short distances, typically under a mile. Multimode fiber is primarily used within buildings and forms the infrastructure for optical LANs.

Optical fiber cable is typically specified using two numeric identifiers of the form *x/y*. Here x represents the diameter of the core, and y represents the cable diameter. Figure 2.10 compares the transmission of light in single and multimode fiber.

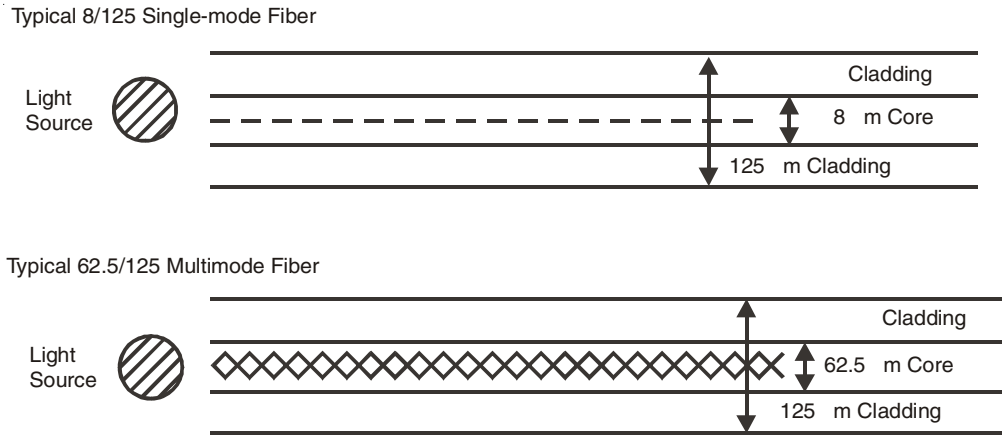



**Fig. 2.10.** *Comparison of Single-mode and multimode fibers operation.*

## 2.5 MOBILE AND CELLULAR NETWORKS AND COMMUNICATION

Cellular networks operate by dividing the service coverage area into zones or cells, each of which has its own set of resources or channels, which can be accessed by users of the network. Usually cellular coverage is represented by a hexagonal cell structure to demonstrate the concept, however, in practice the shape of cells is determined by the local topography. Sophisticated planning tools are used extensively by terrestrial cellular operators to assist with the planning of their cellular networks.

The shape and boundary of a cell is determined by its base station (BS), which provides the radio coverage. A BS communicates with mobile users through signalling and traffic channels (TCH). Signals transmitted in the direction from the BS to the mobile are termed the forward link or downlink, and conversely, the reverse link or uplink is in the direction of mobile to BS. Signalling channels are used to perform administrative and management functions such as setting up a call, while TCHs are used to convey the information content of a call. The allocation of channels to a cell is therefore divided between the TCHs, which form the majority, and signalling channels. These are allocated for both forward and reverse directions.

In order to increase the capacity of a network, there are three possibilities, either:

1. a greater number of channels are made available;
2. more spectrally efficient modulation and multiple access techniques are employed; or
3. the same channels are reused, separated by a distance which would not cause an unacceptable level of co-channel interference.

Cellular networks, which are limited in terms of available bandwidth, operate using the principal of frequency reuse. This implies that the same pool of frequencies is reused in cells that are sufficiently separated so as not to cause harmful co-channel interference. For a hexagonal cell structure, it is possible to cluster cells so that no two adjacent cells are using the same frequency. This is only achievable for certain cell-cluster sizes.

A seven-cell frequency reuse pattern is shown in Fig. 2.11. The total bandwidth available to the network is divided between cells in a cluster, which can then be used to determine the number of calls that can be supported in each cell. By reducing the number of cells per cluster, the system capacity can be increased, since more channels can be available per cell. However, a reduction in the cluster size will also result in a reduction in the frequency reuse distance, hence the system may become more prone to co-channel interference.

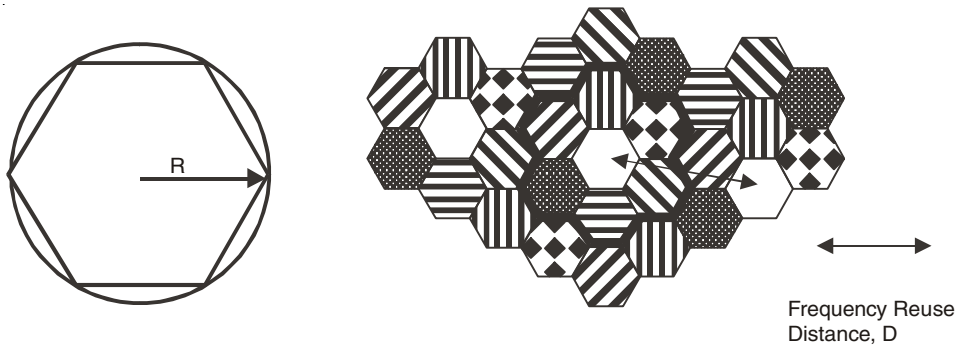



**Fig. 2.11.** *A seven cell frequency reuse pattern.*

How the mobile user gains access to the available channels within a cell is governed by the multiple access technique used by the network. Analogue cellular networks employ frequency division multiple access (FDMA), whereas digital networks employ either time division multiple access (TDMA) or code division multiple access (CDMA). For FDMA, a seven cell reuse pattern is generally employed, whereas for CDMA a single-cell frequency reuse pattern is achievable.

In a terrestrial mobile environment, reception cannot rely on line-of-sight communications and is largely dependent upon the reception of signal reflections from the surrounding environment. The resultant scattering and multipath components arrive at the receiver with random phase. The propagation channel can be characterised by a combination of a slow-fading, long-term component and a fast-fading, short-term component. As a consequence of the local terrain, the change in a mobile's position relative to that of a transmitting BS will result in periodic nulls in the received signal strength. This is due to the fact that the vector summation of the multipath and scattering components at the receiver results in a signal envelope of the form of a standing wave pattern, which has signal nulls at half-wave intervals. For a signal transmitting at 900 MHz, which is typical for cellular applications, a half-wavelength distance corresponds to approximately 17 cm. This phenomenon is known as slow-fading and is characterised by a log-normal probability density function.

As the mobile's velocity, $n$, increases, the variation in the received signal envelope becomes much more pronounced and the effect of the Döppler shift on the received multipath signal components also has an influence on the received signal.

This phenomenon is termed fast-fading and is characterised by a Rayleigh probability density function. Such variations in received signal strength can be as much as 30 dB below or 10 dB above the root mean square signal level, although such extremes occur infrequently.

In rural areas, where the density of users is relatively low, large cells of about 25 km radius can be employed to provide service coverage. This was indeed the scenario when mobile communications were first introduced into service. In order to sustain the mobile to BS link over such a distance requires the use of a vehicular-type mobile terminal, where available transmit power is not so constrained in comparison with hand-held devices. With an increase in user-density, the cell size needs to reduce in order to enable a greater

frequency reuse and hence to increase the capacity of the network. Urban cells are typically of 1 km radius. This reduction in cell size will also correspond to a reduction in BS and mobile terminal transmit power requirements. This is particularly important in the latter case, since it paves the way for the introduction of hand-held terminals.

When a mobile moves from one cell to another during the course of an on-going call, a handover (also termed handoff) of the call between BSs must be performed in order to ensure that the call continues without interruption. Otherwise the call will be dropped and the mobile user would need to re-initiate the call set-up sequence. Handover between BSs involves monitoring of the signal strength between the mobile to BS link. Once the signal strength reduces below a given threshold, the network initiates a procedure to reserve a channel through another BS, which can provide a channel of sufficient signal strength (Figure 2.12 ).

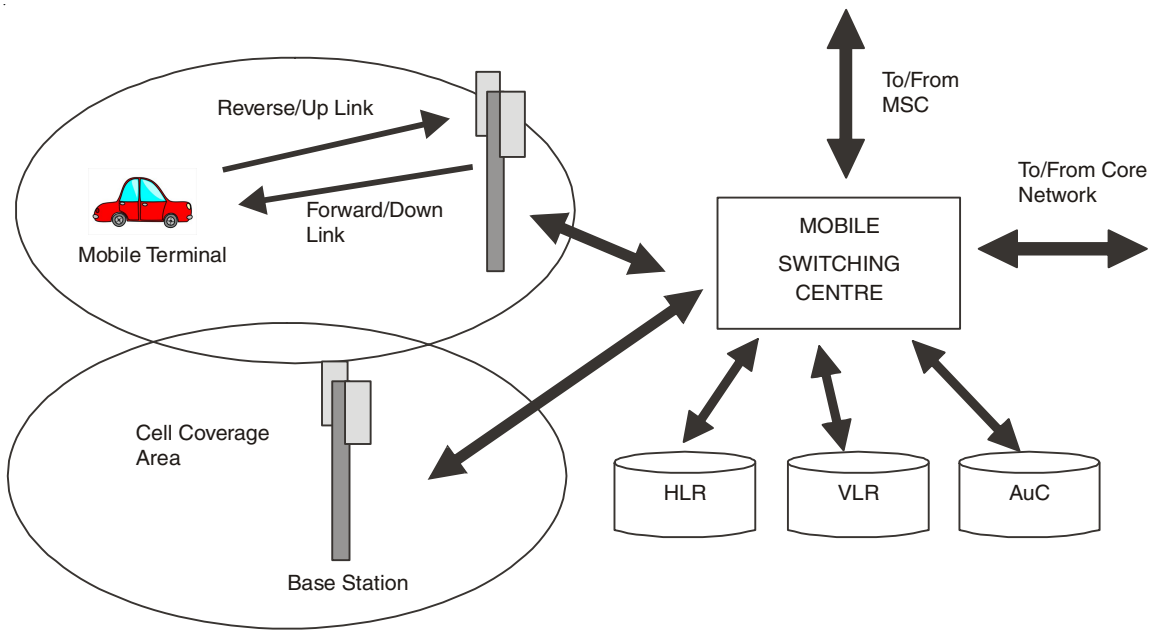



**Fig. 2.12.** *Basic Cellular Network Architecture.*

A number of BSs are clustered together via a fixed-network connection to a mobile switching centre (MSC), which provides the switching functionality between BSs during handover and can also provide connection to the fixed or core network (CN) to allow the routing of calls. The clustering of BSs around a MSC is used to define a Location Area, which can be used to determine the latest known location of a mobile user. This is achieved by associating Home and Visitor Location Areas to a mobile. Each mobile is registered with a single home location register (HLR) upon joining the network. Once a mobile roams outside of its Home Location Area into a new designated Location Area, it temporarily registers with the network as a visitor, where its details are stored in a visitor location register (VLR) associated with the MSC. Each MSC in the network has an associated VLR and HLR. The mobile's location is relayed back to its HLR, a database containing various

information on the mobile terminal, some of which is then forwarded to the VLR. The network also comprises of other databases that can be used to verify that the mobile has access to the network, such as the Authentication Centre (AuC), for example. These procedures are described later in the chapter for the GSM system.

### 2.5.1 First Generation Analog Cellular Systems

In the future mobile information society, where mobile-multimedia delivery will be the major technological driving force, analogue cellular technology has little, if any significance. Indeed, in many countries across Europe, mobile operators are now switching off their analogue services in favour of digital technology. However, analogue technologies still play an important role in many countries around the world, by being able to provide established and reliable mobile voice telephony at a competitive price. This section considers three of the major analogue systems that can still be found with significant customer databases throughout the world.

➢ **Nordic Mobile Telephone (NMT) System**

On 1 October, 1981, the Nordic NMT450 became the first European cellular mobile communication system to be introduced into service. This system was initially developed to provide mobile communication facilities to the rural and less-populated regions of the Scandinavian countries Denmark, Norway, Finland and Sweden. NMT450 was essentially developed for in-car and portable telephones. By adopting common standards and operating frequencies, roaming between Scandinavian countries was possible. Importantly, the introduction of this new technology provided network operators and suppliers with an early market lead, one that has been sustained right up to the present day.

As is synonymous of 1G systems, NMT450 is an analogue system. It operates in the 450 MHz band, specifically 453–457.5 MHz (mobile to BS) and 463–467.5 MHz (BS to mobile). FDMA/FM is employed as the multiple access scheme/modulation method for audio signals, with a maximum frequency deviation of +/−5 kHz. Frequency shift keying (FSK) is used to modulate control signals with a frequency deviation of +/−3.5 kHz. NMT450 operates using a channel spacing of 25 kHz, enabling the support of 180 channels. Since its introduction, the NMT450 system has continued to evolve with the development of the NMT450*i* (where *i* stands for improvement) and NMT900 systems.

NMT900 was introduced into service in 1986, around about the same time as other Western/European countries were starting to introduce their own city based mobile cellular-based solutions. NMT900 is designed for city use, catering for hand-held and portable terminals. It operates in the 900 MHz band with the ability to accommodate higher data rates and more channels.

The NMT system continues to hold a significant market share throughout the world and, significantly, the system continues to evolve, through a series of planned upgrades. In Europe, the NMT family has a particularly large market share in Eastern European countries, where mobile telephony is only now starting to become prevalent.

Since 1981, Nordic countries have continued to lead the way with now over 60% of the population in Finland and Norway having a mobile phone. The Scandinavian-based companies Nokia and Ericsson are world leaders in mobile phone technology and both are driving the phone's evolution forward.

➢ **Advanced Mobile Phone Service (AMPS)**

Bell Labs in the US developed the AMPS communications system in the late 1970s [BEL-79]. The AMPS system was introduced into commercial service in 1983 by AT&T with a 3-month trial in Chicago. The system operates in the US in the 800 MHz band, specifically 824–849 MHz (mobile to BS) and 869–894 MHz (BS to mobile). These bands offer 832 channels, which are divided equally between two operators in each geographical area. Of these 832 channels, 42 channels carry only system information. The AMPS system provides a channel spacing of 30 kHz using FM modulation with a 12 kHz peak frequency deviation for voice signals.

Signalling between mobile and BS is at 10 kbit/s employing Manchester coding. The signals are modulated using FSK, with a frequency deviation of ^8 kHz. The AMPS system specifies six one-way logical channels for transmission of user and signalling information. The Reverse TCH and Forward TCH are dedicated to the transmission of user data on a one-to-one basis. Signalling information is carried to the BS on the channels reverse control channel (RECC) and reverse voice channel (RVC); and to the mobile using the channels forward control channel (FOCC) and forward voice channel (FVC).

The forward and reverse control channels are used exclusively for network control information and can be referred to as Common Control Channels. To safeguard control channels from the effect of the mobile channel, information is protected using concatenated pairs of block codes. To further protect information, an inner code employs multiple repetition of each BCH (Bose–Chadhuri–Hocquenghem) code word at least five times, and 11 times for the FVC.

In order to identify the BS assigned to a call, AMPS employs a supervisory audio tone (SAT), which can be one of three frequencies (5970, 6000 and 6030 Hz). At call set-up, a mobile terminal is informed of the SAT at the BS to which it communicates. During a call, the mobile terminal continuously monitors the SAT injected by the BS. The BS also monitors the same SAT injected by the mobile terminal. Should the received SAT be incorrect at either the mobile terminal or the BS, the signal is muted, since this would imply reception of a source of interference.

Like NMT450, the AMPS standard has continued to evolve and remains one of the most widely used systems in the world. Although market penetration did not reach Europe, at least in its unmodified form, it remains a dominant standard in the Americas and Asia.

➢ **Total Access Communications System (TACS)**

By the mid-1980s, most of Western Europe had mobile cellular capability, although each country tended to adopt its own system. For example, the C-NETZ system was introduced in Germany and Austria, and RADIOCOM 2000 and NMT-F, the French version of NMT900 could be found in France. This variety of technology made it impossible for international commuters to use their phones on international networks, since every national operator had its own standard. In the UK, Racal Vodafone and Cellnet, competing operators providing technically compatible systems, introduced the TACS into service in January 1985. TACS was based on the American AMPS standard with modifications to the operating frequencies and channel spacing. TACS offers a capacity of 600 channels in the bands 890–905 MHz (mobile to BS) and 935–950 MHz (BS to mobile), the available bandwidth

being divided equally between the two operators. Twenty-one of these channels are dedicated for control channels per operator. The system was developed with the aim of serving highly populated urban areas as well as rural areas. This necessitated the use of a small cell size in urban areas of 1 km. In TACS, the cell size ranges from 1 to 10 km. TACS provides a channel spacing of 25 kHz using FM modulation with a 9.5 kHz peak deviation for voice signals. In highly densely populated regions, the number of available channels is increased to up to 640 (320 channels per operator) by extending the available spectrum to below the conference of European Posts and Telegraphs (CEPT) cellular band. This is known as extended TACS (ETACS). Here, the operating frequency bands are 917–933 MHz in the mobile to BS direction and 872–888 MHz in the BS to mobile.

Fifteen years after TACS was first introduced into the UK, the combined Vodafone and Cellnet customer base amounted to just under half a million subscribers out of a total o f 31 million. The future of analogue technology in developed markets is clearly limited, particularly with the re-farming of the spectrum for the 3G services. Nevertheless, analogue systems such as TACS have been responsible for developing the mobile culture and in this respect, their contribution to the evolution of the mobile society remains significant. Within Europe, TACS networks can also be found in Austria, Azerbaijan, Ireland, Italy, Malta and Spain. A variant of TACS, known as J-TACS, operates in Japan.

### 2.5.2 Second-Generation (2G) Systems

➢ **Global System for Mobile Communications (GSM)**

Following a proposal by Nordic Telecom and Netherlands PTT, the Group Special Mobile (GSM) study group was formed in 1982 by the CEPT. The aim of this study group was to define a pan-European public land mobile system.

In 1987, 13 operators and administrators signed the GSM memorandum of understanding (MoU) agreement and the original French name was changed to the more descriptive Global System for Mobile communications (GSM), although the acronym remained the same. By 1999, 296 operators and administrators from 110 countries had signed the GSM MoU. Significantly, in 1987, following the evaluation of several candidate technologies through laboratory and field trial experiments, agreement was reached on the use of a regular pulse excitation-linear predictive coder (RPE-LPC) for speech coding and TDMA was selected as the multiple access method.

In 1989 responsibility for the GSM specification was transferred to the ETSI and a year later Phase 1 GSM specifications were published. Commercial GSM services began in Europe two years later in mid-1991. In addition to voice services, the SMS was created as part of the GSM Phase 1 standard. This provides the facility to send and receive text messages from mobile phones. Messages can be up to 160 characters in length and can be used to alert the user of an incoming e-mail message, for example. It is a store-and-forward service, with all messages passing through an SMS centre. The SMS has proved to be hugely popular in Europe, with the transmission of in excess of 1 billion messages per month as of April 1999. In 1997, Phase 2 specifications came on-line, allowing the transmission of fax and data services.

At the end of 1998, ETSI completed its standardisation of GSM Phase 21 services high speed circuit switched data (HSCSD) and general packet radio service (GPRS). These two new services are aimed very much at exploiting the potential markets in the mobile data sector, recognising the influence of the Internet on mobile technologies. Responsibility for the maintenance and future development of the GSM standards is now under the control of the 3G partnership project (3GPP).

A simplified form of the GSM network architecture is shown in Fig. 2.13.

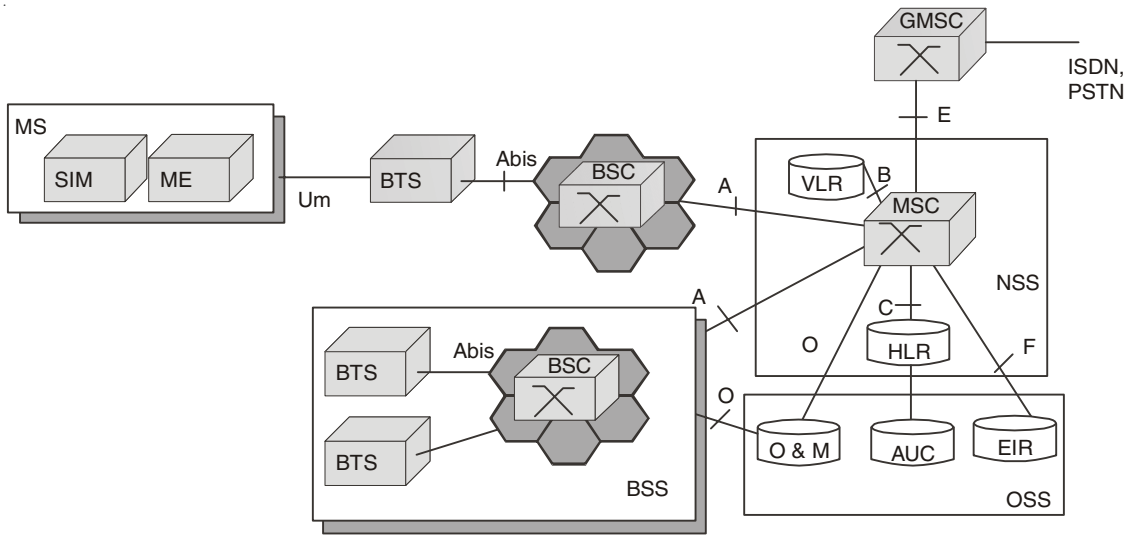



**Fig. 2.13.** *A simplified GSM Architecture.*

*Mobile Station (MS )* A subscriber uses an MS to access services provided by the network. The MS consists of two entities, the mobile equipment (ME) and subscriber identity module (SIM). The ME performs the functions required to support the radio channel between the MS and a BTS. These functions include modulation, coding, and so on. It also provides the application interface of the MS to enable the user to access services. A SIM card provides the ability to personalise a mobile phone. This is a smart card that needs to be inserted into the mobile phone before it can become operational. The SIM card contains the user's international mobile subscriber identity (IMSI), as well as other user specific data including an authentication key. Similarly, a terminal is identified by its international ME identity (IMEI). The IMSI and IMEI provide the capability for personal and terminal mobility, respectively. The radio interface between the MT and the BTS is termed the Um-interface and is one of the two mandatory interfaces in the GSM network.

Base Station System (BSS) The BTS forms part of the base station system (BSS), along with the base station controller (BSC). The BTS provides the radio coverage per cell, while the BSC performs the necessary control functions, which include channel allocation and local switching to achieve handover when a mobile moves from one BTS to another under the control of the same BSC. A BTS is connected to a BSC via an Abis-interface.

Network Management and Switching Subsystem (NMSS) The NMSS provides the connection between the mobile user and other users. Central to the operation of the NMSS is the MSC. A BSS is connected to an MSC via an A-interface, the other mandatory GSM interface. The coverage area of an MSC is determined by the cellular coverage provided by the BTSs that are connected to it. The functions of an MSC include the routing of calls to the appropriate BSS, performing handover between BSSs and inter-working with other fixed networks. A special type of MSC is the gateway MSC (GMSC), which provides connection to fixed telephone networks and vice versa. A GMSC is connected to an MSC via an E-interface. Central to the operation of the MSC are the two databases: HLR, connected via the C-interface; and VLR, connected via the B-interface. The HLR database contains management information for an MS. Each MS has an associated HLR. The information contained within an HLR includes the present location of an MS, and its IMSI, which is used by the authentication centre (AuC) to authorise a subscriber's access to the network.

Each MSC has an associated VLR. Whenever an MS is switched on in a new location area or roams into a new location area covered by an MSC, it must register with its VLR. At this stage, the visiting network assigns a MS roaming number (MSRN) and a temporary mobile subscriber identity (TIMSI) to the MS. The location of the MS, usually in terms of the VLR's signalling address, is then conveyed to the HLR. The VLR, by using subscriber information provided by the HLR, can perform the necessary routing, verification and authentication procedures for an MS that would normally be performed by the HLR. An MSC also provides connection to the SMS centre (SMSC), which is responsible for storing and forwarding messages.

*Operation Subsystem (OSS):* The OSS provides the functions for the operation and management of the network. The Network Operation and Maintenance Centre performs all the necessary functionalities necessary to monitor and manage the network. It is connected to all of the major network elements (BTS, MSC, HLR, VLR) via an O-interface using an X.25 connection. The equipment interface register (EIR) is used by the network to identify any equipment that may be using the network illegally. The MSC-EIR connection is specified by the F-interface. The AuC also forms part of the OSS.

➢ **Digital Cellular System 1800 (DCS1800)**

The first evolution of GSM came about with the introduction of DCS1800, which is aimed primarily at the mass-market pedestrian user located in urban, densely populated regions. DCS1800 was introduced under the personal communications network (PCN) concept, also known as personal communication services (PCS) in the US. In 1989, the UK Government's Department of Trade and Industry outlined its intention to issue licenses for personal communication networks in the 1700–2300 MHz band. It was recognised that the new service, which would be aimed primarily at the pedestrian user, would be an adaptation of the GSM standard. Subsequently, ETSI produced the Phase 1 DCS1800 specification in January 1991, which detailed the generic differences between DCS1800 and GSM. This was followed by a Phase 2 specification detailing a common framework for PCN and GSM. DCS1800 operates using largely the same specification as GSM, making use of the same

network architecture but, as its name implies, it operates in the 1800 MHz band. Here, parallels can be drawn with the evolution of the NMT450 to the NMT900 system from earlier discussions.

The bands that have been allocated for DCS1800 operation are 1710–1785 MHz for the mobile to BS link and 1805–1880 MHz for the BS to mobile link. Taking into account the 200-kHz guard-band, 374 carriers can be supported. Apart from the difference in the operating frequency, the only other major difference is in the transmit power specification of the mobile station. Two power classes were defined at 250 mW and 1 W peak power, respectively. As DCS1800 is intended primarily for urban environments, the cells are much smaller than those of GSM; hence the transmit power requirements are reduced.

The UK was the first country to introduce PCN into operation, through two network operators: Mercury's One2One, which was introduced into service in September 1993; and Hutchinson's Orange, which was introduced into service in April 1994. Dual-mode 900/1800 MHz terminals are now available on the market, as are triple-mode 900/1800/1900 MHz terminals, allowing roaming into North America (in the US, the 1900 MHz band is used for PCS). Towards the end of 1999, Orange and One2One accounted for a third of the UK digital cellular market at just under 5 million subscribers, with virtually an equal market share between the two of them.

### 2.5.3 Third Generation Cellular Systems

ITU has called the future cellular networks 3G networks or IMT-2000; the previous term was Future Public Land Mobile Telephone System (FPLMTS). The performance for IMT-2000 air interference can be summarized as:

– Wideband CDMA systems
– Spectrum bandwidth 5 MHz
– Full coverage and mobility for a data rate of 144 kbps to 384 kbps
– Limited coverage and mobility or no mobility for 2 Mbps
– High spectrum efficiency compared to 2G system
– High flexibility to introduce new and multimedia services

The 3G features:

– Provision of multirate services
– Packet data
– A user-dedicated pilot for a coherent uplink
– An additional DL pilot channel for beam forming
– Intercarrier handover
– Fast power control
– Multiuser detection

IMT-2000 has published a minimum performance requirement of a 3G wireless system, which is for both circuit-switched (CS) and packet-switched (PS) data:

– Data rate of 144 kbps in the vehicular environment.

– Data rate of 384 kbps in the pedestrian environment.

– Data rate of 2 Mbps in the fixed indoor and pico cell environment.

The 3G systems concentrating on the three ITU-adopted systems using CDMA technology are WCDMA-UTRA (Europe), WCDMA-ARIB (Japan), and cdma2000 (North America). 3G consists of the three systems shown in Fig. 2.14, also called three modes because in the future, the three systems can have the features of hooks and extension to make intersystem connections among them.

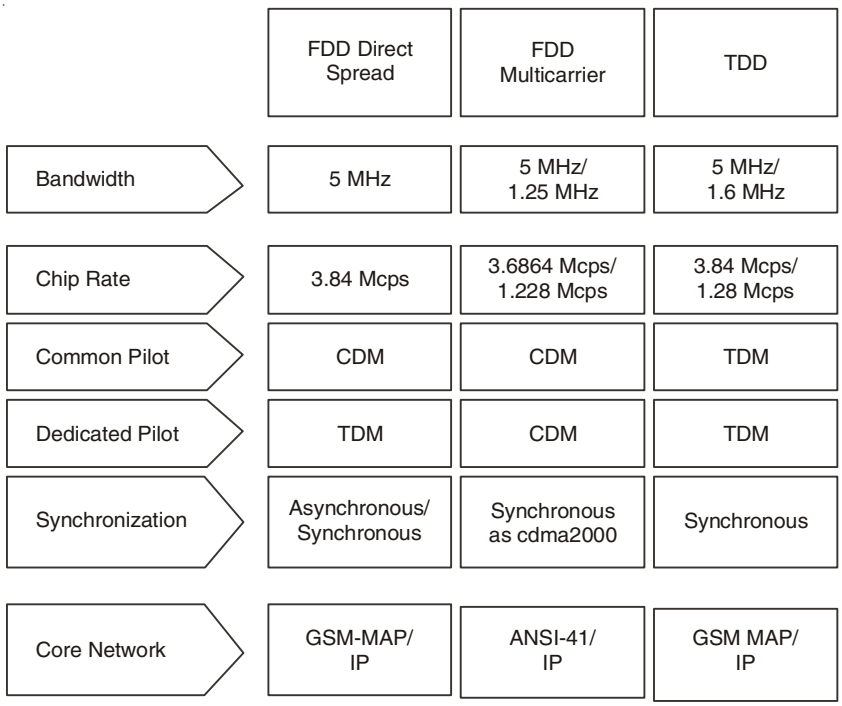



**Fig. 2.14.** *3G Standard systems.*

The wireless loop area network (WLAN)1 can offer a data speed much higher than 3G due to the different technologies.WLAN (includingWi-Fi andWiMAX) were later called the B3G (Beyond 3G) system by the industry. The comparison of the data rate with different standards in WLAN is shown in Fig. 2.15.

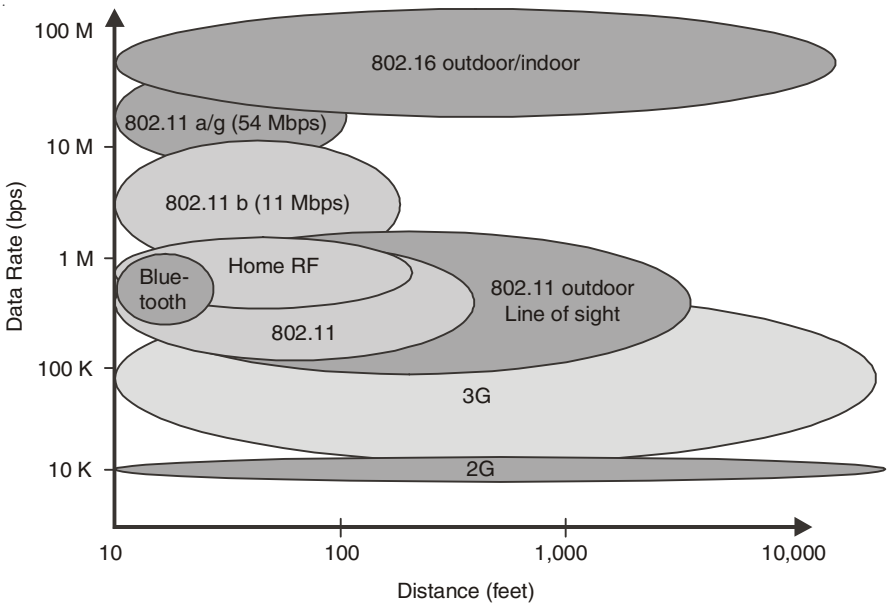



**Fig. 2.15.** *Comparison of various Wireless systems.*

## 2.6 OPTICAL SWITCHING NETWORKS

Optical fiber provides an unprecedented bandwidth potential that is far in excess of any other known transmission medium. A single strand of fiber offers a total bandwidth of 25000 GHz. To put this potential into perspective, it is worthwhile to note that the total bandwidth of radio on Earth is not more than 25 GHz Apart from its enormous bandwidth, optical fiber provides additional advantages such as low attenuation loss. Optical networks aim at exploiting the unique properties of fiber in an efficient and cost-effective manner.

### 2.6.1 Optical Point-to-Point Links

The huge bandwidth potential of optical fiber has been long recognized. Optical fiber has been widely deployed to build high-speed optical networks using fiber links to interconnect geographically distributed network nodes. Optical networks have come a long way. In the early 1980s, optical fiber was primarily used to build and study point-to-point transmission systems. As shown in Fig. 2.16 (*a*), an optical point-to-point link provides an optical single-hop connection between two nodes without any (electrical) intermediate node in between. Optical point-to-point links may be viewed as the beginning of optical networks. Optical point-to-point links may be used to interconnect two different sites for data transmission and reception. At the transmitting side, the electrical data is converted into an optical signal (EO conversion) and subsequently sent on the optical fiber. At the receiving side, the arriving optical signal is converted back into the electrical domain (OE conversion) for electronic processing and storage. To interconnect more than two network nodes, multiple optical single-hop point-to-point links may be used to form various network topologies

(*e.g.,* star and ring networks). Figure 2.16 (*b*) shows how multiple optical point-to-point links can be combined by means of a star coupler to build optical single-hop star networks. The star coupler is basically an optical device that combines all incoming optical signals and equally distributes them among all its output ports. In other words, the star coupler is an optical broadcast device where an optical signal arriving at any input port is forwarded to all output ports without undergoing any EO or OE conversion at the star coupler. Similar to optical point-to-point links, optical single-hop star networks make use of EO conversion at the transmitting side and OE conversion at the receiving side. Besides optical stars, optical ring networks can be realized by interconnecting each pair of adjacent ring nodes with a separate optical single-hop point-to-point fiber link, as depicted in Fig. 2.16 (*c*). In the resultant optical ring network, each node performs OE conversion for incoming signals and EO conversion for outgoing signals. The combined OE and EO conversion is usually referred to as OEO conversion. A good example of an optical ring network with OEO conversion at each node is the fiber distributed data interface (FDDI) standard, which can be found in today's existing optical network infrastructure.

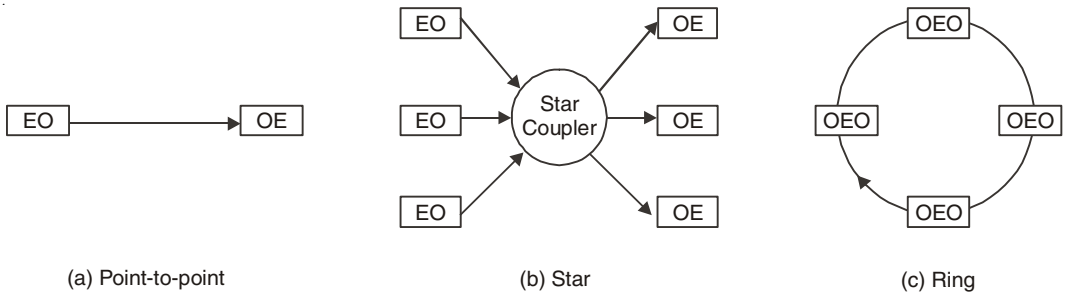



**Fig. 2.16.** *Optical single-hop connections: (a) point-to-point, (b) star, and (c) ring configurations.*

### 2.6.2 SONET/SDH

One of the most important standards for optical point-to-point links is the Synchronous Optical Network (SONET) standard and its closely related synchronous digital hierarchy (SDH) standard. The SONET standardization began during 1985 and the first standard was completed in June 1988. The goals of the SONET standard were to specify optical point-to-point transmission signal interfaces that allow interconnection of fiber optics transmission systems of different carriers and manufacturers, easy access to tributary signals, direct optical interfaces on terminals, and to provide new network features. SONET defines standard optical signals, a synchronous frame structure for time division multiplexed (TDM) digital traffic, and network operation procedures. SONET is based on a digital TDM signal hierarchy where a periodically recurring time frame of 125 μs can carry payload traffic of various rates. Besides payload traffic, the SONET frame structure contains several overhead bytes to perform a wide range of important network operations such as error monitoring, network maintenance, and channel provisioning.

SONET is now globally deployed by a large number of major network operators. Typically, SONET point-to-point links are used in ring configurations to form optical ring networks with OEO conversion at each node, similar to the one depicted in Fig. 2.16(*c*). In

SONET rings there are two main types of OEO nodes: the add-drop multiplexer (ADM) and the digital cross-connect system (DCS). The ADM usually connects to several SONET end devices and aggregates or splits SONET traffic at various speeds. The DCS is a SONET device that adds and drops individual SONET channels at any location. One major difference between an ADM and a DCS is that the DCS can be used to interconnect a larger number of links. The DCS is often used to interconnect SONET rings.

### 2.6.3 Multiplexing: TDM, SDM, and WDM

Given the huge bandwidth of optical fiber, it is unlikely that a single client or application will require the entire bandwidth. Instead, traffic of multiple different sources may share the fiber bandwidth by means of multiplexing. Multiplexing is a technique that allows multiple traffic sources to share a common transmission medium. In the context of optical networks, three main multiplexing approaches have been deployed to share the bandwidth of optical fiber: (1) Time Division Multiplexing (TDM), (2) Space Division Multiplexing (SDM), and (3) Wavelength Division Multiplexing (WDM).

- **Time Division Multiplexing:** We have already seen that SONET is an important example for optical networks that deploy TDM on the underlying point-to-point fiber links. Traditional TDM is a well-understood technique and has been used in many electronic network architectures throughout the more than 50-year history of digital communications. In the context of high-speed optical networks, however, TDM is under pressure from the so-called "electro-optical" bottleneck. This is due to the fact that the optical TDM signal carries the aggregate traffic of multiple different clients and each TDM network node must be able to operate at the aggregate line rate rather than the subrate that corresponds to the traffic originating from or destined for a given individual node. Clearly, the aggregate line rate cannot scale to arbitrarily high values but is limited by the fastest available electronic transmitting, receiving, and processing technology. As a result, TDM faces severe problems to fully exploit the enormous bandwidth of optical fiber.
- **Space Division Multiplexing:** One straightforward approach to avoid the electrooptical bottleneck is SDM, where multiple fibers are used in parallel instead of a single fiber. Each of these parallel fibers may operate at any arbitrary line rate (*e.g.*, electronic peak rate). SDM is well-suited for short-distance transmissions but becomes less practical and more costly for increasing distances due to the fact that multiple fibers need to be installed and operated.
- **Wavelength Division Multiplexing:** WDM appears to be the most promising approach to tap into the vast amount of fiber bandwidth while avoiding the aforementioned shortcomings of TDM and SDM. WDM can be thought of as optical frequency division multiplexing (FDM), where traffic from each client is sent on a different carrier frequency. In optical WDM networks the term wavelength is usually used instead of frequency, but the principle remains the same. As shown in Fig. 2.17, in optical WDM networks each transmitter $i$ sends on a separate wavelength $\lambda i$ , where $1 \leq i \leq N$. At the transmitting side, a wavelength multiplexer collects all wavelengths and feeds them onto a common outgoing fiber. At the receiving side,

a wavelength demultiplexer separates the wavelengths and forwards each wavelength $\lambda i$ to a different receiver $i$. Unlike for TDM, each wavelength channel may operate at any arbitrary line rate well below the aggregate TDM line rate. By using multiple wavelengths the huge bandwidth potential of optical fiber can be exploited. As opposed to SDM, WDM takes full advantage of the bandwidth potential of a single fiber and does not require multiple fibers to be installed and operated in parallel, resulting in significant cost savings. Optical WDM networks have been attracting a great deal of attention by network operators, manufacturers, and research groups around the world.

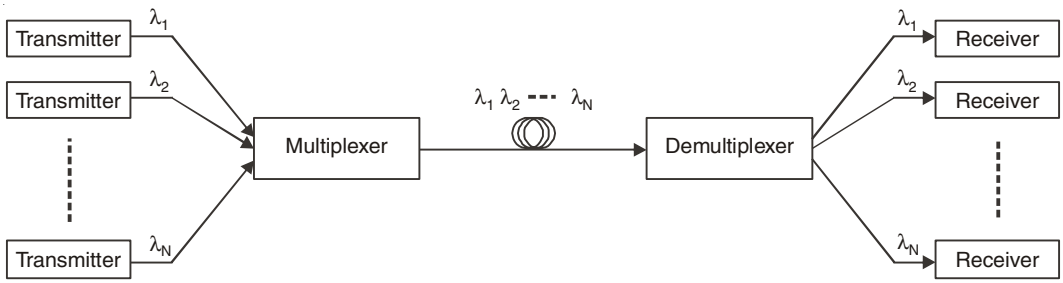



**Fig. 2.17.** *Wavelength division multiplexing.*

### 2.6.4 Applications of Optical Networks

Many of today's applications can be roughly broken down into the following two categories:

**Latency-Critical Applications:** This type of application comprises small-to medium size file transfers which require low latency. Examples of latency-critical applications are broadcast television, interactive video, video conferencing, security video monitoring, interactive games, telemedicine, and telecommuting.

**Throughput-Critical Applications:** This type of application includes large-size file transfers whose latency is not so important but which require much bandwidth. Examples of throughput-critical applications are video on demand (VoD), video and still-image email attachments, backup of files, program and file sharing, and file downloading (*e.g.,* books).

Applications have a significant impact on the traffic load and throughput-delay performance requirements of (optical) networks. Depending on the application, traffic loads and throughput-delay performance requirements may change over time. For instance, web browsing has led to rather asymmetric network traffic loads with significantly less upstream traffic than downstream traffic. Web browsing is based on the client–server paradigm, where each client sends a short request message to a server for downloading data files. In future (optical) networks, the traffic load is expected to become less asymmetric due to the fact that so-called peer-to-peer (P2P) applications become increasingly popular. Napster and its successors are good examples of P2P applications, where each client also acts as a server from which other clients may download files (*e.g.,* photos, videos, and programs). P2P applications traffic keeps growing and may already represent the major traffic load in existing access networks. The steadily growing P2P application traffic will eventually render the network traffic loads more symmetric.

The demand for more bandwidth is expected to keep increasing in order to support new emerging and future applications.

## Review Questions

1. Compute the Fourier coefficients for the function $f(t) = t$ (0 $t$ 1).
2. A noiseless 4-kHz channel is sampled every 1 msec. What is the maximum data rate?
3. Television channels are 6 MHz wide. How many bits/sec can be sent if four-level digital signals are used? Assume a noiseless channel.
4. If a binary signal is sent over a 3-kHz channel whose signal-to-noise ratio is 20 dB, what is the maximum achievable data rate?
5. What signal-to-noise ratio is needed to put a T1 carrier on a 50-kHz line?
6. What is the difference between a passive star and an active repeater in a fiber network?
7. How much bandwidth is there in 0.1 micron of spectrum at a wavelength of 1 micron?
8. It is desired to send a sequence of computer screen images over an optical fiber. The screen is 480 × 640 pixels, each pixel being 24 bits. There are 60 screen images per second. How much bandwidth is needed, and how many microns of wavelength are needed for this band at 1.30 microns?
9. Is the Nyquist theorem true for optical fiber or only for copper wire?
10. What is the essential difference between message switching and packet switching?
11. What is the available user bandwidth in an OC-12c connection?
12. Enlist applications of Optical switching Networks.

❑❑❑

# CHAPTER 3 DATA LINK LAYER

*It is a mistake to look too far ahead. Only one link in the chain of destiny can be handled at a time.*

**—Winston Churchill**

## 3.1 INTRODUCTION

The simplest network possible is one in which all the hosts are directly connected by some physical medium. This may be a wire or a fiber, and it may cover a small area (*e.g.,* an office building) or a wide area (*e.g.,* transcontinental). Connecting two or more nodes with a suitable medium is only the first step, however. There are some additional problems that must be addressed before the nodes can successfully exchange packets.

In this section we will study the design of layer 2, the data link layer (also known as the physical link control layer). The purpose of the data link layer is to transfer blocks of data without error between two adjacent devices. Adjacent devices are physically connected by a communication channel such as telephone lines, coaxial cables, optical fibres, or satellites. The implication of such a physical link is that the data bits are delivered in exactly the same order in which they are sent. The physical link has no inherent storage capacity, therefore the delay involved is the propagation delay over the link. Transmission of data over the link would be very simple indeed if no error ever occurred. Unfortunately, this is not so in a real physical link for a number of reasons:

- Natural phenomena such as noises and interference are introduced into the link causing errors in detecting the data.
- There is a propagation delay in the link.
- There is a finite data processing time required by the transmitting and receiving stations.

A data link protocol thus has to be designed to ensure an error-free transmission and also to achieve an efficiency of the data transfer as high as possible. To see the need for data link control, we list some of the requirements and objectives for effective data communication between two directly connected transmitting-receiving stations:

- **Frame synchronization.** Data are sent in blocks called frames. The beginning and end of each frame must be recognizable.
- **Flow control.** The sending station must not send frames at a rate faster then the receiving station can absorb them.
- **Error control.** Any bit errors introduced by the transmission system must be corrected.
- **Addressing.** On a multipoint line, such as a local area network (LAN), the identity of the two stations involved in a transmission must be specified.
- **Control and data on same link.** It is usually not desirable to have a physically separate communications path for control information. Accordingly, the receiver must be able to distinguish control information from the data being transmitted.
- **Link management.** The initiation, maintenance, and termination of a sustained data exchange requires a fair amount of coordination and cooperation among stations. Procedures for the management of this exchange are required.

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 Fig. 3.1. Frame management forms the heart of what the data link layer does. In the following sections we will examine all the above-mentioned issues in detail.

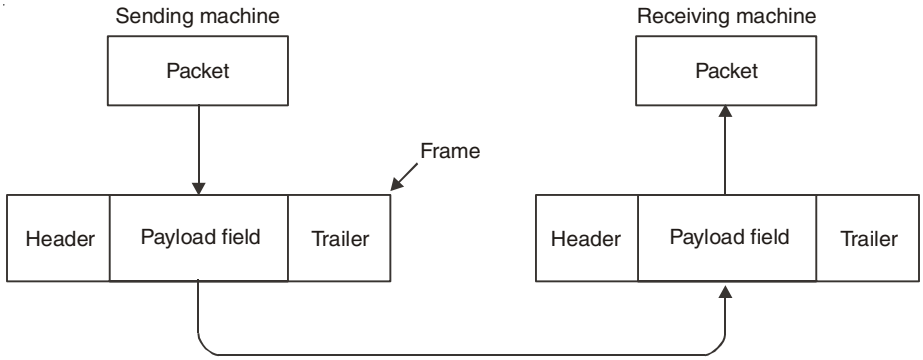



**Fig. 3.1.** *Relationship between packets and frame.*

Although this chapter is explicitly about the data link layer and the data link protocols, many of the principles we will study here, such as error control and flow control, are found in transport and other protocols as well. In fact, in many networks, these functions are found only in the upper layers and not in the data link layer. However, no matter where they are found, the principles are pretty much the same, so it does not really matter where we study them. In the data link layer they often show up in their simplest and purest forms, making this a good place to examine them in detail. It is important to mention here that data link layer provides services to network layer and in turn uses services from physical layer.

## 3.2 FRAMING

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

The data link layer, on the other hand, needs to pack bits into frames, so that each frame is distinguishable from another. Our postal system practices a type of framing. The simple act of inserting a letter into an envelope separates one piece of information from another; the envelope serves as the delimiter. In addition, each envelope defines the sender and receiver addresses since the postal system is a many-to-many carrier facility.

Framing in the data link layer separates a message from one source to a destination, or from other messages to other destinations, by adding a sender address and a destination address. The destination address defines where the packet is to go; the sender address helps the recipient acknowledge the receipt. Although the whole message could be packed in one frame, that is not normally done. One reason is that a frame can be very large, making flow and error control very inefficient. When a message is carried in one very large frame, even a single-bit error would require the retransmission of the whole message. When a message is divided into smaller frames, a single-bit error affects only that small frame.

Frames can be of fixed or variable size. In fixed-size framing, there is no need for defining the boundaries of the frames; the size itself can be used as a delimiter. An example of this type of framing is the ATM wide-area network, which uses frames of fixed size called cells. In variable-size framing, we need a way to define the end of the frame and the beginning of the next.

In this section we will look at four commonly used methods of framing:

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

### *1. Character count*

The first framing method uses a field in the header to specify the number of characters in the frame. When the data link layer at the destination sees the character count, it knows how many characters follow and hence where the end of the frame is. This technique is shown in Fig. 3.2 (*a*) for four frames of sizes 5, 5, 8, and 8 characters, respectively.

The trouble with this algorithm is that the count can be garbled by a transmission error. For example, if the character count of 5 in the second frame of Fig. 3.2 (*b*) becomes a 7, the destination will get out of synchronization and will be unable to locate the start of the next frame. Even if the checksum is incorrect so the destination knows that the frame is bad, it still has no way of telling where the next frame starts. Sending a frame back to the source asking for a retransmission does not help either, since the destination does not know how many characters to skip over to get to the start of the retransmission. For this reason, the character count method is rarely used anymore.

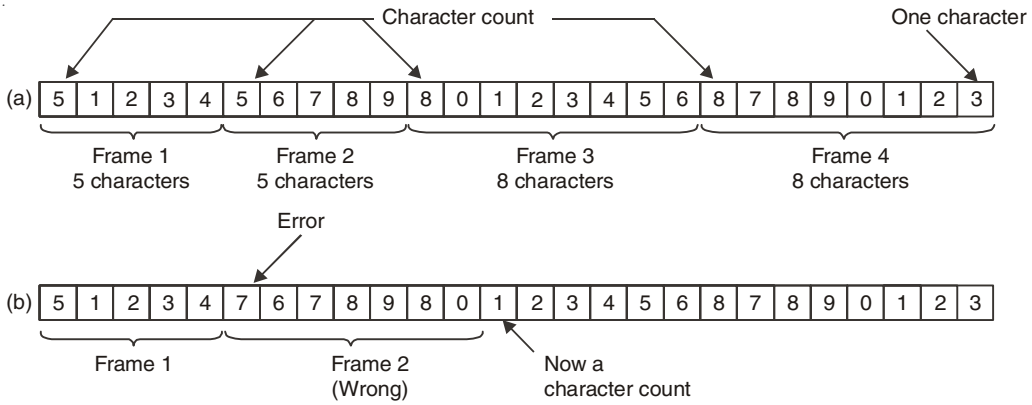



**Fig. 3.2.** *A character stream. (a) Without errors. (b) With one error.*

### *2. Flag bytes with byte stuffing*

Character-oriented framing was popular when only text was exchanged by the data link layers. The flag could be selected to be any character not used for text communication. Now, however, we send other types of information such as graphs, audio, and video. Any pattern used for the flag could also be part of the information. If this happens, the receiver, when it encounters this pattern in the middle of the data, thinks it has reached the end of the frame. To fix this problem, a byte-stuffing strategy was added to character-oriented framing. In byte stuffing (or character stuffing), a special byte is added to the data section of the frame when there is a character with the same pattern as the flag. The data section is stuffed with an extra byte. This byte is usually called the escape character (ESC), which has a predefined bit pattern. Whenever the receiver encounters the ESC character, it removes it from the data section and treats the next character as data, not a delimiting flag.

Byte stuffing by the escape character allows the presence of the flag in the data section of the frame, but it creates another problem. What happens if the text contains one or more escape characters followed by a flag? The receiver removes the escape character, but keeps the flag, which is incorrectly interpreted as the end of the frame. To solve this problem, the escape characters that are part of the text must also be marked by another escape character. In other words, if the escape character is part of the text, an extra one is added to show that the second one is part of the text. Figure 3.3 shows the situation.

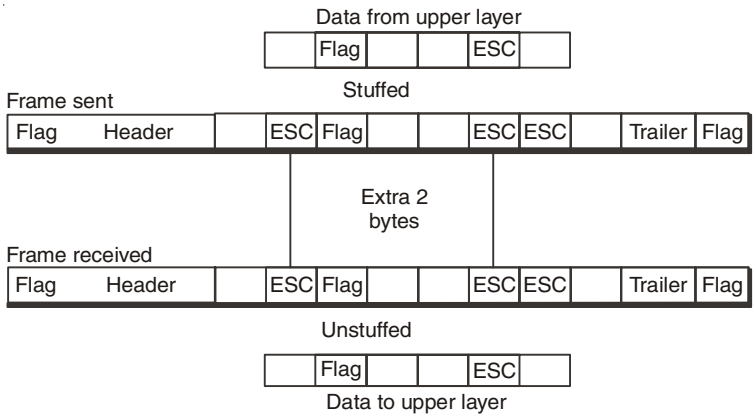



**Fig. 3.3.** *Concept of Byte stuffing.*

### *3. Starting and ending flags, with bit stuffing*

This technique allows data frames to contain an arbitrary number of bits and allows character codes with an arbitrary number of bits per character. It works like this. Each frame begins and ends with a special bit pattern, 01111110 (in fact, a flag byte). Whenever the sender's data link layer encounters five consecutive 1s in the data, it automatically stuffs a 0 bit into the outgoing bit stream. This bit stuffing is analogous to byte stuffing, in which an escape byte is stuffed into the outgoing character stream before a flag byte in the data.

When the receiver sees five consecutive incoming 1 bits, followed by a 0 bit, it automatically destuffs (*i.e.,* deletes) the 0 bit. Just as byte stuffing is completely transparent to the network layer in both computers, so is bit stuffing. If the user data contain the flag pattern, 01111110, this flag is transmitted as 011111010 but stored in the receiver's memory as 01111110. Figure 3.4 gives an example of bit stuffing.

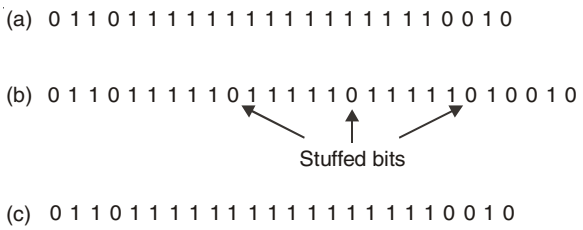



**Fig. 3.4.** *Concept of Bit stuffing.*

With bit stuffing, the boundary between two frames can be unambiguously recognized by the flag pattern. Thus, if the receiver loses track of where it is, all it has to do is scan the input for flag sequences, since they can only occur at frame boundaries and never within the data.

### *4. Physical layer coding violations*

The last method of framing is only applicable to networks in which the encoding on the physical medium contains some redundancy. For example, some LANs encode 1 bit of data by using 2 physical bits. Normally, a 1 bit is a high-low pair and a 0 bit is a low-high pair. The scheme means that every data bit has a transition in the middle, making it easy for the receiver to locate the bit boundaries. The combinations high-high and low-low are not used for data but are used for delimiting frames in some protocols.

As a final note on framing, many data link protocols use a combination of a character count with one of the other methods for extra safety. When a frame arrives, the count field is used to locate the end of the frame. Only if the appropriate delimiter is present at that position and the checksum is correct is the frame accepted as valid. Otherwise, the input stream is scanned for the next delimiter.

## 3.3 ERROR DETECTION AND CORRECTION

Once the data is dispatched over the transmission medium the characteristics of the medium normally conspire to alter the transmitted data in various ways so that the signals received at the remote end of a link differ from the transmitted signals. The effects of these adverse

characteristics of a medium are known as **transmission impairments** and they often reduce transmission efficiency. In the case of binary data they may lead to errors, in that some binary zeros are transformed into binary ones and vice versa. To overcome the effects of such impairments it is necessary to introduce some form of error control. The first step in any form of error control is to detect whether any errors are present in the received data. Having detected the presence of errors there are two strategies commonly used to correct them: either further computations are carried out at the receiver to correct the errors, a process known as **forward error control**; or a message is returned to the transmitter indicating that errors have occurred and requesting a retransmission of the data, which is known as **feedback error control**. Error control is a function of the data link layer of the OSI reference model.

### 3.3.1 Types of Errors

Whenever bits flow from one point to another, they are subject to unpredictable changes because of interference. This interference can change the shape of the signal. In a single-bit error, a 0 is changed to a 1 or a 1 to a 0. In a burst error, multiple bits are changed. For example, a 11100 s burst of impulse noise on a transmission with a data rate of 1200 bps might change all or some of the 12 bits of information.

#### *Single-bit error*

The term *single-bit error* means that only 1 bit of a given data unit (such as a byte, character, or packet) is changed from 1 to 0 or from 0 to 1.

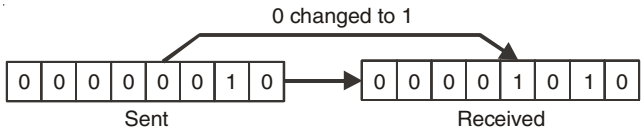



**Fig. 3.5.** *Single-bit error.*

Single-bit errors are the least likely type of error in serial data transmission. To understand why, imagine data sent at 1 Mbps. This means that each bit lasts only 1/1,000,000 s. For a single-bit error to occur, the noise must have a duration of only 1 micro sec., which is very rare; noise normally lasts much longer than this.

#### *Burst error*

The term *burst error* means that 2 or more bits in the data unit have changed from 1 to 0 or from 0 to 1. It is important to mention here that a burst error does not necessarily mean that the errors occur in consecutive bits. The length of the burst is measured from the first corrupted bit to the last corrupted bit. Some bits in between may not have been corrupted.

A burst error is more likely to occur than a single-bit error. The duration of noise is normally longer than the duration of 1 bit, which means that when noise affects data, it affects a set of bits. The number of bits affected depends on the data rate and duration of noise. For example, if we are sending data at I kbps, a noise of 11100 s can affect 10 bits; if we are sending data at I Mbps, the same noise can affect 10,000 bits.

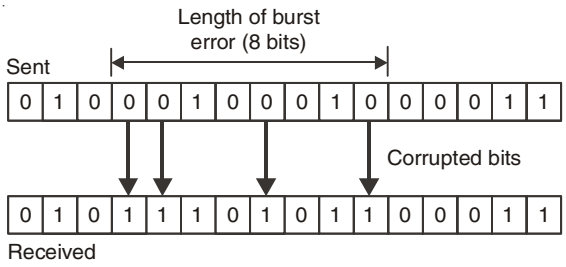



**Fig. 3.6.** *Burst error.*

### 3.3.2 Redundancy

The central concept in detecting or correcting errors is redundancy. To be able to detect or correct errors, we need to send some extra bits with our data. These redundant bits are added by the sender and removed by the receiver. Their presence allows the receiver to detect or correct corrupted bits.

### 3.3.3 Forward Error Control

The need to detect and correct errors was mentioned in Section 3.3.1 but not the means by which the detection process can be carried out. Error detection (and correction) is also known as **channel coding**. Channel coding is the process of coding data prior to transmission over a communications channel so that if errors do occur during transmission it is possible to detect and possibly even to correct those errors once the data has been received. In order to achieve this error detection/correction some bit patterns need to be identified as error free at the receiver, whereas other bit patterns will be identified as erroneous. To increase the number of identifiable bit patterns at the receiver above the bare minimum required to represent the data, additional bits, known as **redundant bits**, are added to the data or information bits prior to transmission. Various different types of code are available for use in channel coding but the most commonly used are called **linear block codes.**

### 3.3.4 Linear Block Codes

These constitute the simplest and most commonly used type of channel code. Data is transmitted as a fixed-length block. Prior to transmission the data is treated as a binary number and some form of linear mathematical process is carried out on a group of information bits so as to generate additional redundant bits which are known as **check bits.**

The check bits are transmitted along with the information bits, normally at the end of the block. At the receiver, a similar mathematical process is used to determine whether there are errors or not. Typical mathematical processes used are addition and division. A study of these codes inevitably requires some knowledge of mathematics but although the theory underlying the codes is complex, the following treatment has been kept fairly straightforward without being too simplistic.

#### *3.3.4.1 Hamming codes*

This is an important group of early error-correcting codes pioneered by R.W. Hamming in the 1950s. They involve the production of check bits by adding together different groups

of information bits. The type of addition used is known as modulo-2 addition and is equivalent to normal binary addition without any carries. The best way to see how the check bits are obtained is to consider a particular code as an example.

We shall consider a Hamming (7, 4) code, in which three check bits ($c1$, $c2$ and $c3$) are combined with four information bits ($k1$, $k2$, $k3$ and $k4$) to produce a block of data of length $n = 7$. This block of data is known as a **codeword**. A block of data of length 7 is too short to be appropriate for a practical data communications system, but the mathematics involved in longer blocks would become tedious. Three check equations are used to obtain the three check bits of this Hamming (7, 4) code as follows:

$c1 = k1 \oplus k2 \oplus k4$

$c2 = k1 \oplus k3 \oplus k4$

$c3 = k2 \oplus k3 \oplus k4$

where $\oplus$ represents modulo-2 addition. The rules of modulo-2 addition are:

$0 \oplus 0 = 0$

$0 \oplus 1 = 1$

$1 \oplus 1 = 0$ (no carry)

If we choose the information bits 1010 as an example then $k1 = 1$, $k2 = 0$, $k3 = 1$ and $k4 = 0$ and the check bits obtained from the three check equations above are as follows:

$c1 = k1 \oplus k2 \oplus k4 = 1 \oplus 0 \oplus 0 = 1$

$c2 = k1 \oplus k3 \oplus k4 = 1 \oplus 1 \oplus 0 = 0$

$c3 = k2 \oplus k3 \oplus k4 = 0 \oplus 1 \oplus 0 = 1$

The codeword is obtained by adding the check bits to the end of the information bits and therefore the data 1010101 will be transmitted (information bits first). A complete set of code words can be obtained in a similar way:

| Codeword no. | $k_1$ | $k_2$ | $k_3$ | $k_4$ | $c_1$ | $c_2$ | $c_3$ | Codeword no. | $k_1$ | $k_2$ | $k_3$ | $k_4$ | $c_1$ | $c_2$ | $c_3$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 9 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 10 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 11 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 4 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 12 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 5 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 13 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 6 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 14 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 7 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 15 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

An error that occurs in a transmitted codeword can be detected only if the error changes the codeword into some other bit pattern that does not appear in the code. This means that the code words transmitted over a channel must differ from each other in at least two bit positions. If two code words differ in only one position and an error occurs

in that position then one codeword will be changed into another codeword and there will be no way of knowing that an error has occurred. Inspection of the set of code words of the Hamming (7, 4) code reveals that they all differ from each other in at least three places. Taking code words 3 and 8 as an example, we have:

Codeword 3 0 0 1 1 1 0 0

Codeword 8 1 0 0 0 1 1 0

These two code words differ in positions 1, 3, 4 and 4 (counting from the left). The number of positions by which any two code words in a code differ is known as the **Hamming distance** or just the distance, so that the distance between these two words is four. Since all linear block codes contain the all-zeros codeword, then an easy way to find the **minimum distance** of a code is to compare a non-zero codeword which has the minimum number of ones with the all-zeros codeword. Thus, the minimum distance of a code is equal to the smallest number of ones in any non-zero codeword, which in the case of this Hamming (7, 4) code is three. If the code words of a code differ in three or more positions then error correction is possible since an erroneous bit pattern will be 'closer' to one codeword than another (this assumes that one error is more likely than two, two more likely than three, and so on). If we take codewords 8 and 10 as an example, we have:

Codeword 8 1 0 0 0 1 1 0

Codeword 10 1 0 1 0 1 0 1

The distance between these two codewords is three. If codeword 8 is transmitted and an error occurs in bit 3 then the received data will be:

1 0 1 0 1 1 0

This is not one of the other 15 Hamming (7, 4) codewords since an error has occurred. Furthermore, the most likely codeword to have been transmitted is codeword 8 since this is the nearest to the received bit pattern. Thus, it should also be possible to correct the received data by making the assumption that the transmitted codeword was number 8. If, however, a second error occurs in bit 7 then the received bit pattern will be:

1 0 1 0 1 1 1

It should still be possible to detect that an error has occurred since this is not one of the 16 codewords. However, it is no longer possible to correct the errors since the received bit pattern has changed in two places and is no longer closer to codeword 8 than any other (it is, in fact, now closer to codeword 10). Thus, this Hamming (7, 4) code is able to detect two errors but correct only one error. In general, if the minimum distance of a code is $d$, then $d$–1 errors can normally be detected using a linear block code and mod($d$–1)/2 can be corrected.

A feature of all linear block codes which arises out of the mathematical rules used to determine the check bits is that all the codewords are related by these rules. Received data which contains errors no longer conforms to the mathematical rules, and it is this fact that is used to carry out the detection and correction processes at the receiver. If we take the

example of the Hamming (7,4) code then the encoding process will restrict the number of different codewords that can be transmitted to the 16 listed above. As a result of errors that may occur in the transmission process, the data arriving at a receiver in 7-bit blocks can have any one of 27 = 128 different 7-bit patterns. This allows the receiver to detect whether errors have occurred since it is aware of the rules used in the encoding process and can apply them to see whether the received data is one of the 14 'legal' codewords or not.

Furthermore, it is the case that all Hamming codes (indeed, all linear block codes) possess the mathematical property that if we add any two codewords together (modulo-2 addition) then the resulting sum is also a codeword. For example, if we add codewords 1 and 2 from the earlier list we obtain:

**0 0 0 1 1 1**
⊕ **0 0 1 0 0 1**
**0 0 1 1 1 0** which is codeword 3

This allows us to represent a whole code by means of a small 'subset' of codewords, since further codewords can simply be obtained by modulo-2 addition. In the case of the Hamming (7,4) code this is not important, since there are only 14 codewords. However, with longer block lengths the number of codewords becomes unmanageable. For example, a short block of 32 bits involves 2^32 = 4 294 967 296 different codewords.

The subset of codewords is often expressed as a matrix known as a **generator matrix**, *G*. The codewords chosen are normally powers of 2, that is codewords 1, 2, 4, 8, . . . . A suitable generator matrix for the Hamming (7,4) code consists of the following four codewords:

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

The matrix has four rows and seven columns, that is it has dimensions $4 \times 7$ ($k \times n$). The whole code can be generated from this matrix just by adding together rows, and it is for this reason that it is called a generator matrix. A further reason for the generator matrix being so named is that it can be used to generate codewords directly. from the information bits without using the check equations. This is achieved by multiplying the information bits by the generator matrix using matrix multiplication.

### *Encoding and decoding circuits*

An attractive feature of Hamming codes in the early days of error correction was that they could be easily implemented in hardware circuits, particularly if the block length was fairly short. Since modulo-2 addition is identical to an exclusive-or (EX-OR) function, a number of multiple input EX-OR gates can be used to determine the check bits, as in the circuit for a Hamming (7,4) encoder shown in Fig. 3.7.

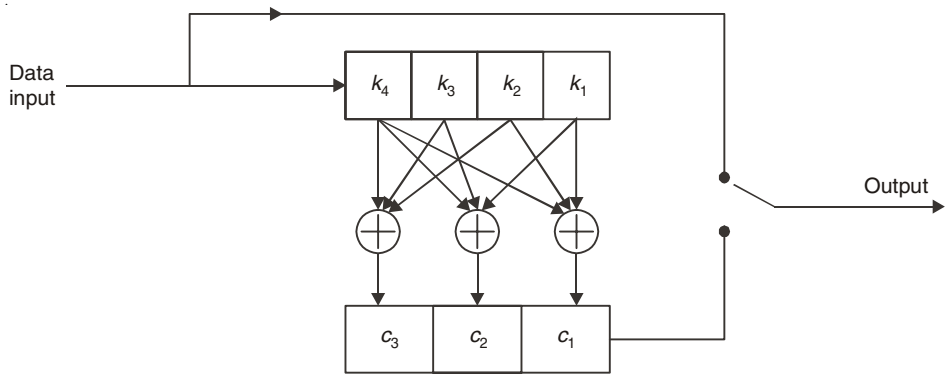



**Fig. 3.7.** *Hamming Encoder.*

The information bits are fed into a 4-bit shift register and the check bits are calculated by the EX-OR circuits and are held in a 3-bit shift register. The switch is in the up position to transmit the information bits and down for the check bits. Figure 3.8 shows a corresponding decoder.

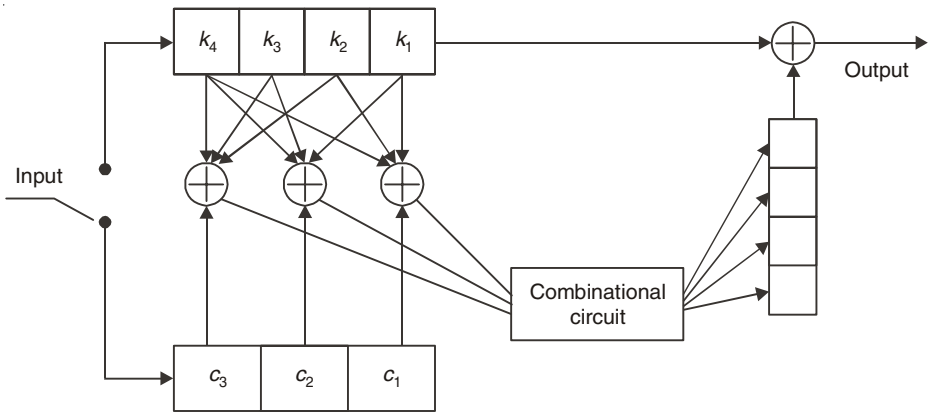



**Fig. 3.8.** *Hamming Decoder.*

With the receive switch up the information bits are received into a 4-bit shift register and with it down the check bits flow into a 3-bit shift register. The left-hand EX-OR gate works out the modulo-2 sum $k2 \oplus k3 \oplus k4 \oplus c3$. This equals zero if no errors have occurred during transmission. The output from the three EX-OR gates thus represents the syndrome which is fed into a combinational logic circuit to determine the position of any error. The error is corrected by means of a $k$-stage shift register at the output of the combinational logic circuit. This will contain all zeros apart from any erroneous position which will contain a one. The output from the shift register is added (modulo-2) serially to the received information bits and any bit position within the received data which gets added to a one will change from 0 to 1 (or 1 to 0), thus correcting the error. Unfortunately, circuits such as these, although simple in the case of the Hamming (7,4) code, become excessively complicated for the longer block lengths used in data communications networks. Hamming codes do, however, find uses in situations which do not require large block lengths, such as remote control of robotic systems.

### 3.3.4.2 Cyclic codes

As mentioned above, simple linear codes such as the Hamming code have a limitation in that if large block lengths are used, for example in data communications, then the encoding and decoding circuitry becomes very complex. Paradoxically, the circuitry can be made simpler if the mathematical structure of the code is made more complex.

A cyclic code is one in which all the codewords are related by the fact that if a codeword is rotated, it becomes another codeword. The following code is obtained from the single check equation $c1 = k1 \oplus k2$ and, although trivial, is cyclic:

| $k1$ | $k2$ | $c1$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

All four of these codewords can be rotated in either direction and will result in another codeword. Consequently, to define this code, it is only necessary to have one non-zero codeword, since all the other codewords can be obtained from it (the all zeros codeword is obtained by adding any codeword to itself). Cyclic codes are usually defined by a single codeword expressed as a polynomial, known as a **generator polynomial**. For example, the cyclic code used in the High-Level Data Link Control (HDLC) protocol has a generator polynomial $x16 + x12 + x5 + 1$, where $x = 2$ since the code is binary. This is expressed as a polynomial rather than the binary number 10001000000100001 because the latter is rather unmanageable. The highest power of a generator polynomial is called its **degree** and is always equal to the number of check bits in the code. Since cyclic codes are invariably linearblock codes, they can also be described by a generator matrix, which can be readily obtained from the generator polynomial.

The fact that the codewords of a cyclic code are obtained by shifting and adding the generator polynomial leads to the important characteristic that all codewords are a multiple of the generator polynomial. To encode incoming information bits, a cyclic encoder must therefore generate check bits which, when added to the information bits, will produce a codeword which is a multiple of $G(x)$, the generator polynomial. This is achieved as follows: firstly, the information bits, normally a lengthy bit pattern, are represented as a polynomial $K(x)$, where $x$ is, in practice, 2. Secondly, let the information bits $K(x)$, followed by $c$ zeros (*i.e.*, a codeword with all the check bits set to zero), be represented by the polynomial $F(x)$ which is in fact $K(x)$ shifted by $c$ places, that is $xcK(x)$. If $F(x)$ is now divided by the generator polynomial $G(x)$ then:

$$\frac{F(x)}{G(x)} = Q(x) + \frac{R(x)}{G(x)}$$

where $Q(x)$ is the quotient and $R(x)$ the remainder, that is: $F(x) = Q(x)G(x) + R(x)$

If the remainder $R(x)$ is now added (modulo-2) to $F(x)$, we obtain:

$$F(x) + R(x) = Q(x)G(x)$$

since addition and subtraction will give the same result in modulo-2. It is this bit sequence, $F(x) + R(x)$, which is transmitted, since it is always a multiple of the generator polynomial $G(x)$ and is therefore always a legal codeword. Thus, encoding for a cyclic code consists of adding $c$ zeros to the end of the information bits and dividing by the generator polynomial to find the remainder. (Note that modulo-2 division is used.) The remainder is added to the information bits in place of the $c$ zeros and the resulting codeword transmitted. At the receiver, a decoder tests to see whether the received bit sequence is error free by dividing again by the generator polynomial. An error-free transmission results in a zero remainder. Such a process is also known as a **cyclic redundancy check** (CRC).

### *Modulo-2 division circuits*

Modulo-2 division is achieved in an electronic circuit by repeated shifting and subtraction. This can be very easily implemented using shift registers and, bearing in mind that modulo-2 subtraction is identical to addition, EX-OR gates. A circuit that will divide by $1 + x + x3$ is given in Fig. 3.9.

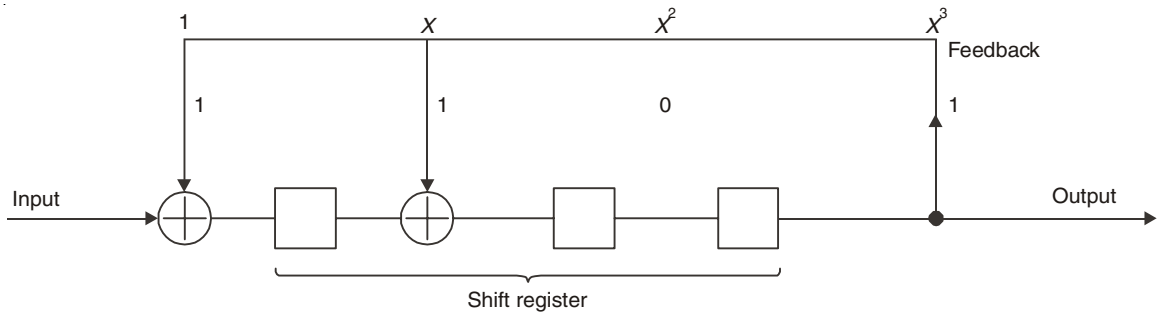



**Fig. 3.9.** *Modulo-2 Division Circuit.*

**Table 3.1. Modulo-2 Division**

| *Step J* | *Input on Jth shift* | *Feedback on Jth shift* | *Shift register after Jth shift* | *Output after Jth shift* |
|---|---|---|---|---|
| 0 | – | – | 0 0 0 | 0 |
| 1 | 1 | 0 | 1 0 0 | 0 |
| 2 | 1 | 0 | 1 1 0 | 0 |
| 3 | 0 | 0 | 0 1 1 | 1 |
| 4 | 0 | 1 | 1 1 1 | 1 |
| 5 | 0 | 1 | 1 0 1 | 1 |
| 6 | 1 | 1 | 0 0 0 | 0 |
| 7 | 0 | 0 | 0 0 0 | |

### *Cyclic encoding and decoding circuits*

Encoding for a cyclic code consists of dividing the polynomial $F(x)$ (incoming information bits with zeros in the check bit positions) by the generator polynomial $G(x)$ to find a remainder $R(x)$ which is then added to $F(x)$ to give the codeword $F(x) + R(x)$. To obtain the

zero check bits, the information bits $K(x)$ are shifted $c$ places prior to the division process. An encoding circuit, known as a Meggitt encoder, that achieves this for $G(x) = 1 + x + x3$ is shown in Fig. 3.10.

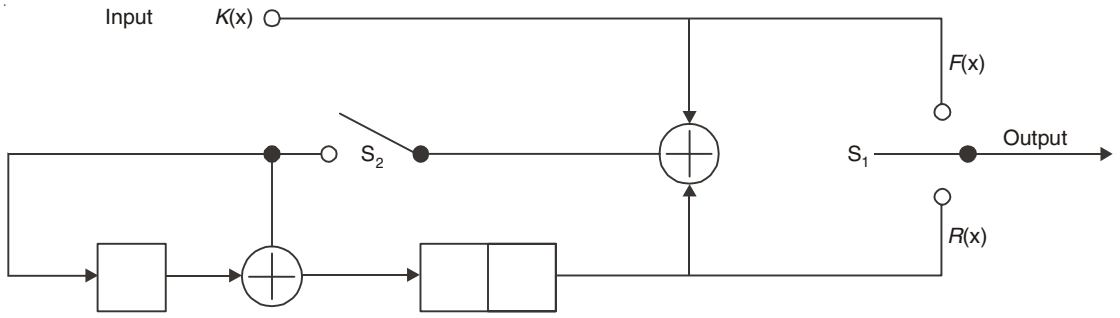



**Fig. 3.10.** *Cyclic Encoding circuit.*

Bringing the input information bits $K(x)$ into the circuit at the point shown rather than to the left of the circuit as in the division circuit of Fig. 3.9 is equivalent to shifting $c$ places ($c$ = 3 in this case). Initially S1 is in the up position and S2 is closed, while $F(x)$ is simultaneously transmitted and processed in the division circuit. After $k$ shifts, the shift register will contain the remainder $R(x)$. S1 then moves to the down position and S2 opens so that the remainder is then transmitted.

A similar arrangement is shown in the decoder circuit of Fig. 3.11. Initially the information bits are received into a $k$-stage shift register with S1 and S2 closed. At the same time, the check bits are recalculated by the division circuit and then fed into a combinational logic circuit. The check bits are received with S1 and S2 open and are fed straight into the combinational logic circuit where they are compared with the recalculated check bits to determine a syndrome. S1 and S2 are then closed again and the syndrome is used by the combinational circuit to correct the data at the output of the decoding circuit.

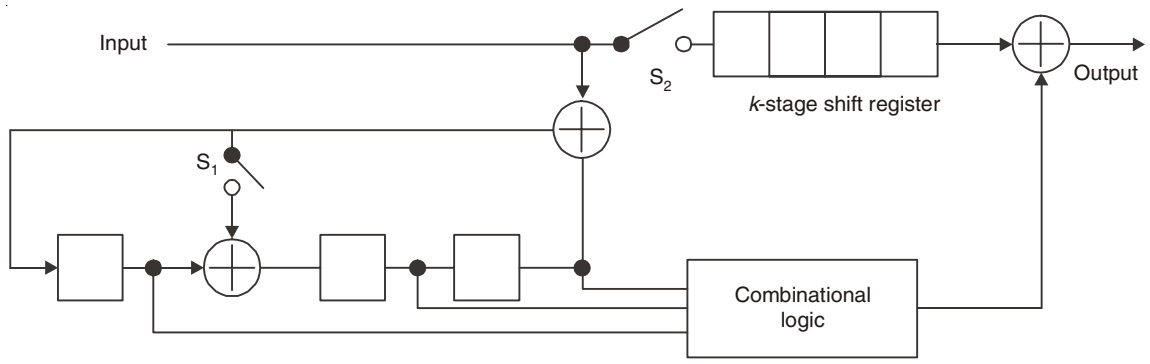



**Fig. 3.11.** *Cyclic Decoding circuit.*

### *Practical cyclic redundancy checks*

Although the material in this section appears quite theoretical, it is of great practical importance to most data communications networks. This is because most networks use a

cyclic redundancy check as part of the level 2 protocol error checking. Table 3.2 shows some commonly used generator polynomials.

**Table 3.2. Commonly used CRC**

| *Protocol(s)* | *Generator polynomial* |
|---|---|
| ATM (Header error check) | $x^8 + x^2 + x + 1$ |
| ATM (OAM cells) | $x^{10} + x^9 + x^5 + x^4 + x + 1$ |
| ATM adaptation layer 1 | $x^3 + x + 1$ |
| ATM adaptation layer 2 | $x^5 + x^2 + x + 1$ |
| Ethernet (and FDDI) | $x^{32} + x^{24} + x^{23} + x^{22} + x^{14} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ |
| X.25 (HDLC) | $x^{16} + x^{12} + x^5 + 1$ |

### *Advantages of Cyclic Codes*

Cyclic codes have a very good performance in detecting single-bit errors, double errors, an odd number of errors, and burst errors. They can easily be implemented in hardware and software. They are especially fast when implemented in hardware. This has made cyclic codes a good candidate for many networks.

### *3.3.4.3 Convolutional codes*

In order to carry out a cyclic redundancy check on a block of data, it is necessary for the complete block of data to be received into an input buffer at a network node within a data communications network. This is convenient for most data communications protocols, including IP, which transmit data in the form of blocks known as packets or frames. Convolutional codes work in a fundamentally different manner in that they operate on data continuously as it is received (or transmitted) by a network node. Consequently, convolutional encoders and decoders can be constructed with a minimum of circuitry. This has meant that they have proved particularly popular in mobile communications systems where a minimum of circuitry and consequent light weight are a great advantage. It should be noted that these considerations have become less significant with advances in electronic circuitry. As well as carrying out encoding and decoding continuously on a transmitted or received bit stream, the way a convolutional code treats a particular group of bits depends on what has happened to previous groups of bits. A simple convolutional encoder is illustrated in Fig. 3.12.

The data to be encoded is entered into a 3-bit shift register one bit at a time. For each bit that is fed in at the input two bits are transmitted from the output, one each from terminals X and Y. Table 3.3 shows the outputs that will appear as a result of different combinations of existing shift register contents and new input values.

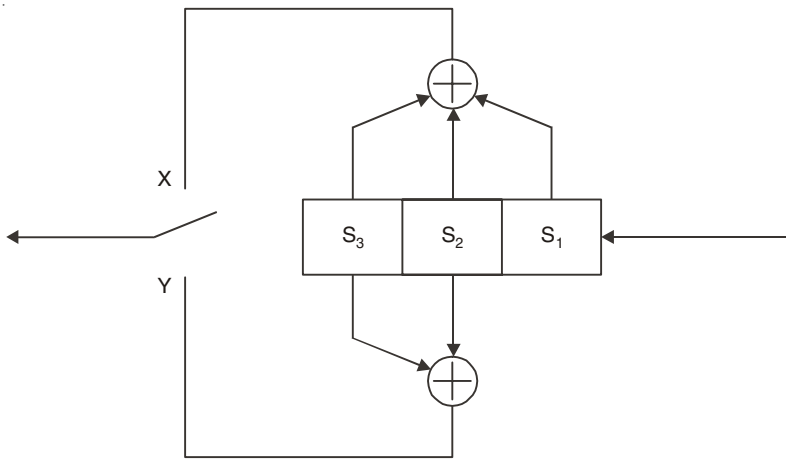



**Fig. 3.12.** *A simple convolution encoder.*

**Table 3.3. Convolution Encoder Output Values**

| *Shift register* | | | *New input* | *Output* | | *Shift register* | | | *New input* | *Output* | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *S3* | *S2* | *S1* | | *X* | *Y* | *S3* | *S2* | *S1* | | *X* | *Y* |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |

This table is of limited usefulness since it does not give any indication of the output resulting from a prolonged stream of bits at the input. In order to observe this a slightly more complicated representation, called a **Trellis diagram**, is used. There are two branches from every point in a Trellis diagram. The upper branch represents 0 input, and the lower branch represents 1 input, and the resulting output is written alongside each branch. This is illustrated in Fig. 3.13 that represents the first row of Table 3.3. At point A, it is assumed that the contents of the shift register are 000. Thus, the branch AB that represents a 0 input has an output of 00 alongside it since that is the output indicated in Table 3.3 for these values. Similarly, the lower branch AC that represents a 1 input has a value 01 alongside it. A full Trellis diagram, so called because it resembles a garden trellis, for the above encoder is illustrated in Fig. 3.14.

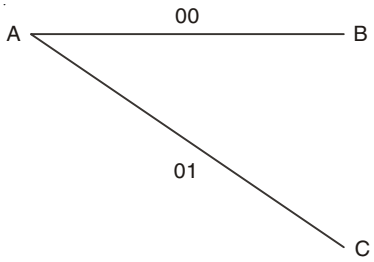



**Fig. 3.13.** *Initial Part of Trellis diagram.*

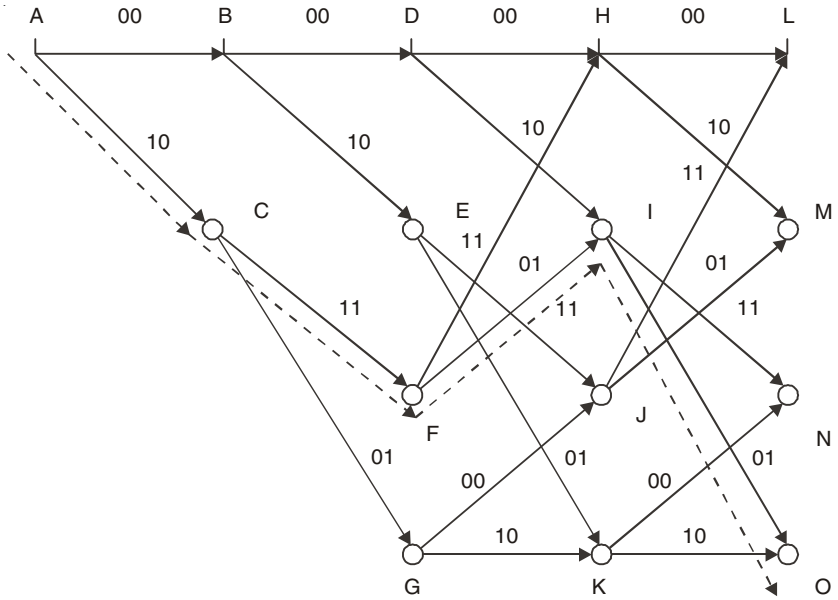



**Fig. 3.14.** *Trellis diagram.*

Note that the pattern of branches AB and AC is repeated all along the top of the trellis diagram with branches BE and BD and so on. This is because the pair of output bits will always be 00 as long as the shift register contains 000 and a zero arrives at the input. Branches CF and CG represent the second row of Table 3.3 where the contents of the shift register are 001; branches FH and FI represent the third row where the contents of the shift register are 010. The Trellis diagram can be used to indicate the output produced by a particular input bit stream by drawing a line through the trellis to represent the input bit stream. This is illustrated by the line in Fig. 3.14 that follows the path ACFIO representing an input of 1011. To obtain the resulting output, we note the pairs of digits along path ACFIO, namely 10110101.

### *Decoding and error correction*

The principle involved in decoding convolutional codes and subsequently correcting errors is the same as that involved in linear block codes; that is, error-free transmission will produce certain received bit patterns (codewords in the terminology of linear block codes)

and errors are likely to produce different bit patterns that cannot possibly be received in error-free transmissions. An inspection of Fig. 3.14 immediately reveals that certain bit patterns are not possible. For example, consider the bit pattern of 11111110 that cannot be obtained from any of the paths through the trellis. If such a pattern is received in a system using the decoder of Fig. 3.12 then it is assumed that an error (or errors) has occurred in transmission.

If no errors occur during transmission, the path taken through the trellis diagram will be the same at the receiver as at the transmitter. Errors that occur during transmission will cause the path taken at the receiver to be different, and will eventually result in an impossible path. At this point, the receiver can notify the transmitter of an error and it may be able to correct the error by deducing the path that would have been the most likely in the absence of errors. In order to do this we need to have a measure of the different paths through the trellis. In the linear block codes, we used the concept of Hamming distance, which was defined as the number of positions in which two codewords differ and the minimum distance, that is the number of positions that a codeword differs from the all-zero codeword. In convolutional codes, we retain the term **distance** to indicate the number of positions in which two paths differ and we define the **weight** as the number of bit positions in which a path differs from the all-zero path, that is the number of logical ones that appear in a particular path. Thus, in Fig. 3.14, path ACFIN is defined by the output sequence 10110111 and therefore possesses a weight of 6 since it contains six logical ones. We can also define the **free distance**, $d$free, as the smallest weight of any path that diverges from the all-zero path and subsequently returns to it. This will be comparable with the minimum Hamming distance of a linear block code and is used to define the error detecting/correcting capabilities of the code as follows:

$d$free – 1 errors can be detected.

mod($d$free – 1)/2 errors can be corrected.

An algorithm for decoding convolutional codes was developed by Viterbi (1967). In order to understand the algorithm, let us consider the received bit pattern 11111110 mentioned earlier. It is immediately obvious that this is an erroneously received bit pattern as the first two bits are not compatible with the trellis diagram. The Viterbi algorithm takes the received bit stream, compares it with the trellis diagram and calculates the distance between each line in the trellis diagram and the received bit pattern. This is illustrated in Fig. 3.15.

If we look at the first stage of the trellis we note that AB is marked 2 and AC is marked 1. This is because AB represents the output 00 which is a distance of 2 from the first two received bits (11) and AC represents 10 which is a distance of 1 from the first two received bits. The other stages of the trellis are marked in the same way. The Viterbi algorithm now computes the minimal path through this trellis. It does this in a step-by-step fashion, the details of which are beyond the scope of this text.

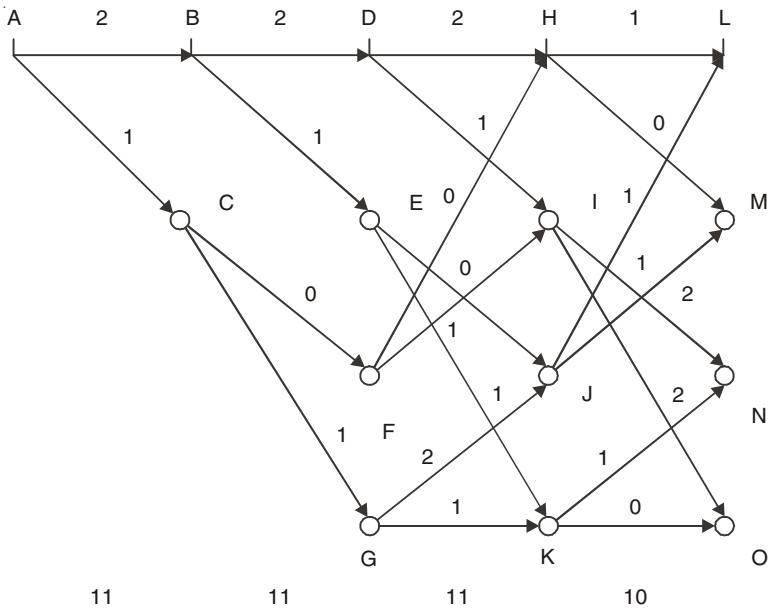



**Fig. 3.15.** *Trellis showing distance from received bits.*

In this particular case, the minimal path is ACFHM which should have a distance of 1 from the received bit sequence of 11111110. It can be seen from Fig. 3.15 that path ACFHM represents an output of 10111110, which is indeed a distance of 1 from the received bit sequence. A convolutional code receiver using the Viterbi algorithm would therefore assume that the error-free received data was 10111110 and decode it accordingly as 1001. It is worth noting that, like most other error-correcting systems, the algorithm relies on the fact that one error is more likely than two, two errors are more likely than three, and so on.

## 3.4 FEEDBACK ERROR CONTROL

Even very powerful error-correcting codes may not be able to correct all errors that arise in a communications channel. Consequently, many data communications links provide a further error control mechanism, in which errors in data are detected and the data is retransmitted. This procedure is known as **feedback error control** and it involves using a channel code to detect errors at the receive end of a link and then returning a message to the transmitter requesting the retransmission of a block (or blocks) of data. Alternatively, errors in received data can be detected and corrected up to a certain number of errors and a retransmission requested only if more than this number of errors occurs. The process of retransmitting the data has traditionally been known as **Automatic Repeat Request** (ARQ). There are three types of ARQ that have been used: namely, **stop-and-wait**, **go-back-*n*** and **selective-repeat**.

*Stop-and-wait ARQ*

This technique is the simplest method of operating ARQ and ensures that each transmitted block of data or **frame** is correctly received before sending the next. At the

receiver, the data is checked for errors and if it is error free an acknowledgement (ACK) is sent back to the transmitter. If errors are detected at the receiver a negative acknowledgement (NAK) is returned. Since errors could equally occur in the ACK or NAK signals, they should also be checked for errors. Thus, only if each frame is received error free and an ACK is returned error free can the next frame be transmitted.

**Case I:** If, however, errors are detected either in the transmitted frame or in the returned acknowledgement, then the frame is retransmitted.

**Case II:** A further (and hopefully remote) possibility is that a frame or acknowledgement becomes lost for some reason. To take account of this eventuality, the transmitter should retransmit if it does not receive an acknowledgement within a certain time period known as the **timeout interval**.

**Case III:** Finally, a frame may be received correctly but the resulting ACK may be lost, resulting in the same frame being transmitted a second time. This problem can be overcome by numbering frames and discarding any correctly received duplicates at the receiver.

Although stop-and-wait offers a simple implementation of an ARQ system it can be inefficient in its utilization of the transmission link since time is spent in acknowledging each frame.

*Go-back-n ARQ*

If the error rate on a link is relatively low, then the link efficiency can be increased by transmitting a number of frames continuously without waiting for an immediate acknowledgement. This strategy is used in go-back-*n* ARQ which is used in a number of standard protocols including HDLC (High-level Data Link Control) and, in modified form, TCP (Transmission Control Protocol), both of which are dealt with later in the text. The go-back number *n* determines how many frames can be transmitted without an acknowledgement having been received and is often referred to as a transmission **window**.

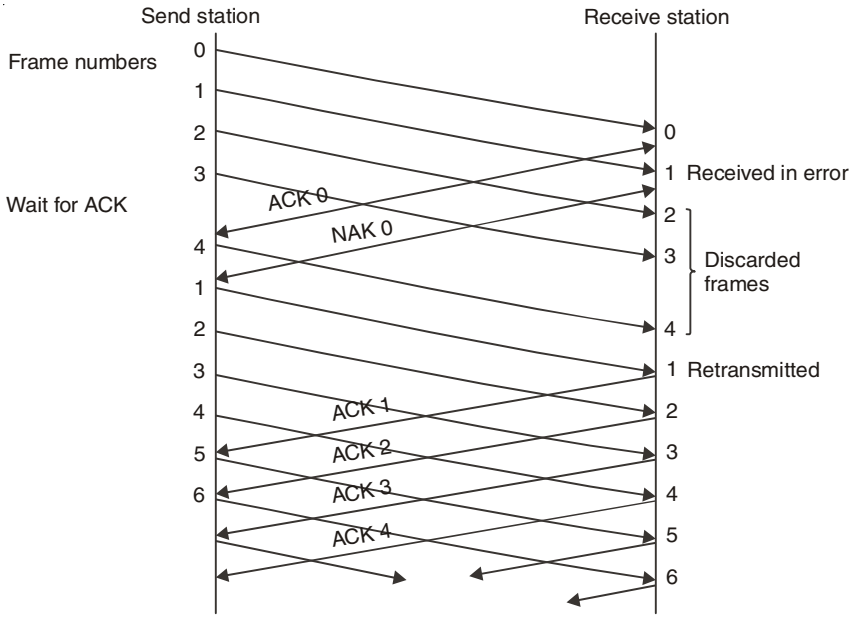



**Fig. 3.16.** *Frame transfer go-back-n.*

Frame $i + n$ cannot be transmitted until frame $i$ has been acknowledged. Figure 3.16 shows a typical transfer of frames for a go-back-4 ARQ system over a full-duplex link.

Note that the send node transmits four frames (numbered 0, 1, 2, 3) without receiving an acknowledgement. There then follows a short wait before receipt of the first ACK which contains the number of the last correctly received frame (0 in this case). Although this acknowledgement is returned immediately after the correct receipt of frame 0, transmission delays mean that it does not reach the send node until after frame 3 has been transmitted (the send node having carried on transmitting as a result of the go-back-4 strategy). If we now assume that an error occurs in frame 1, the receive node will reply with NAK 0, indicating that an error has been detected and that the last correctly received frame was frame 0. The send node will now *go back* and retransmit the frame after the last correctly received frame, which in this case is frame 1. Meanwhile the receive node will have received frames 2, 3 and 4 which, since they have not been acknowledged, need to be discarded. In the absence of further errors, the transmit node continues transmitting frames 2, 3, 4, 5, ... for as long as ACK signals continue to be returned. A go-back-*n* strategy allows for the efficient transfer of frames without the need for any substantial buffer storage by receive equipment as long as error rates are low. However, if errors occur during transmission then frames that have already been received correctly need to be retransmitted along with erroneous frames. If buffer storage is available at the receive equipment then it seems reasonable that some form of selective retransmission strategy would be more efficient than go-back-*n* in the presence of errors.

***Selective-repeat ARQ***

In selective-repeat ARQ only those frames that generate a NAK are retransmitted. Although this appears more efficient than go-back-*n*, it requires sufficient storage at the transmitter to save all frames that have been transmitted but not acknowledged in case a frame proves to be erroneous. In this system a go-back number is still used to determine how many frames can be transmitted without receiving an acknowledgement. In the past, selective-repeat ARQ was not a particularly popular ARQ strategy because of the increased memory requirements mentioned, but, as memory capabilities have become more readily available, it offers potential efficiency gains in the presence of high error rates.

## 3.5 FLOW CONTROL

Flow control is a technique for assuring that a transmitting entity does not overwhelm a receiving entity with data. The receiving entity typically allocates a data buffer of some maximum length for a transfer. When data are received, the receiver must do a certain amount of processing before passing the data to the higher-level software. In the absence of flow control, the receiver's buffer may fill up and overflow while it is processing old data. Flow control coordinates the amount of data that can be sent before receiving an acknowledgement and is one of the most important duties of the data link layer. In most protocols, flow control is a set of procedures that tells the sender how much data it can transmit before it must wait for an acknowledgement from the receiver.