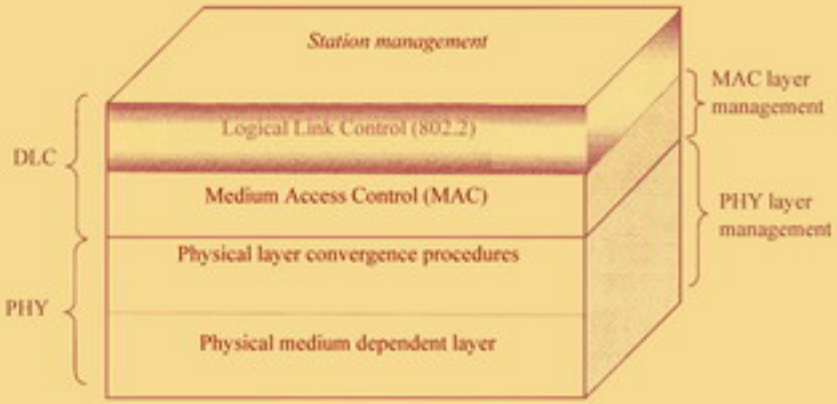
DATA COMMUNICATION PRINCIPLES

For Fixed and Wireless Networks



Protocol architecture for IEEE WLAN

Aftab Ahmad

DATA COMMUNICATION PRINCIPLES

For Fixed and Wireless Networks

This page intentionally left blank

DATA COMMUNICATION PRINCIPLES

For Fixed and Wireless Networks

Aftab Ahmad

KLUWER ACADEMIC PUBLISHERS
NEW YORK, BOSTON, DORDRECHT, LONDON, MOSCOW

eBook ISBN: 0-306-47793-9
Print ISBN: 1-4020-7328-3

Created in the United States of America

Visit Kluwer Online at: http://kluweronline.com
and Kluwer's eBookstore at: http://ebooks.kluweronline.com

NOTE TO INSTRUCTORS

As an aid to instructors interested in teaching this book as a course, exercises suitable for use in a classroom setting are available by contacting the author at *Aftab@ieee.org*.

This page intentionally left blank

To my parents

This page intentionally left blank

Table of Contents

Preface

In spite of the fact that the electronic communication systems started as data communication systems, much of their advancement has been in the field of voice. For decades, the Public Switched Telecommunications Network (PSTN) has set standards for communication of information all over the world. Things started changing only towards the closing of 80s when Internet and mobile systems offered competition in some ways. Despite the continued importance of PSTN, these two technologies have found their niche to sustain and grow. Wireless networks offer mobility as an add-on and the Internet brought the web, email and file transfer. Though PSTN could provide the Internet-like services, it has its limitations due to its circuit switched nature. For some time, we have used the term data to distinguish the store-and-forward type of information (carried by Internet) from voice. This is because the Internet uses store and forward mechanism of transmission, which is not quite suitable for interactive, real-time communication, such as voice.

In this way, PSTN, Internet and wireless networks have not quite stood in each other's way. PSTN offering toll-quality digitized voice, wireless cellular networks adding mobility to voice and data, with some degradation in quality, and the Internet allowing enormous sharing power using store-and-forward protocols for data communications. Ever since their debut, all these fields of technologies have made progress, with PSTN being steady and slow, Internet being slow first and then exploding, and the wireless technology making a steady progress at a rapid speed. It is only very recently that there is a slowing in wireless market. However, with ever-increasing products in license-free wireless band, this is projected to change soon. The emphasis, however, may shift from voice to web-based applications. In fact, with the availability of high-speed 'data' links, voice is becoming part of 'data'.

Traditionally, data was generated and processed by mainframe computers. Users accessed the computer resources through a network of dumb terminals. With developments in microchip fabrication, and reduction in memory cost, processing power shifted to terminals making them intelligent, and then as powerful as the computer itself. These days, desktop computers are mostly the processing powerhouses that mainframes once were. And these machines are capable of processing data at speeds that could easily take care of the requirements of interactive information. The introduction of IMT-2000 systems in wireless arena has reduced the gap in wireless, voice and data. Now we talk about multimedia wireless networks that could use Internet as a backbone. Consequently, the entire meaning of data and communications has changed to include real-time information and wireless networks. For a student and practitioner of data networks, the fundamental concepts of networks with

latest technology have become more important to master than ever before. Of course, by latest technologies, we mean packet switching and wireless data networks. This book has been written with these developments in mind.

The primary audience of this book is the students, senior undergraduate or first year graduate, and personnel in the fields of computer science, electrical engineering, telecommunications, information systems, and other majors that require an elementary to medium level knowledge of data communications principles. Much of the material has been used to offer graduate and undergraduate level courses in some of the above areas. The book is a compilation from lecture notes with some addition. The approach adopted is rather straightforward; define a data network as a computer communication network that could be best understood with the help of the Open System Interconnection Reference Model (OSI-RM), recommended by the International Organization for Standardization (ISO). The OSI networking standards are not nearly as prolific in use as the Internet protocols, still the pedagogical value of the reference model makes it a good choice for a first course on data communications. Besides, the main differences between the OSI and TCP/IP networks exist at layers above the data link control. The book emphasizes only on the bottom two layers, therefore making it useful for people who would work with TCP/IP, OSI or Local Area Network. Examples of protocols are chosen, among others, from the wireless data networks. This broadens the scope of the application of work. Another salient feature of the book is a chapter on performance modeling of data networks. This is a topic that results in most innovations in technology, and yet is not easy to introduce at an elementary level. Every try has been made to let the reader appreciate the models and metrics of performance measures. Separate discussions have been included about the wireless cellular network performance and simulation of networks. Here's the organization of chapters.

A major part of the book is dedicated to the understanding of data and its transmission across a single link. However, to put things in perspective, the first two chapters discuss the protocol architectures in general. Examples of OSI-RM, TCP/IP suite and IEEE Wireless LAN are discussed under this topic. Chapter 3 discusses data in most of its forms, from analog form to characters and as data exists within the network (baseband signal, passband modulated signal, to protocol data unit). Chapter 4 discusses physical layer characteristics and protocol examples. Among the protocol examples, a discussion on the physical layers of IEEE Wireless LAN has been included. Simple descriptions of some important concepts have been provided here, such as the need of physical medium dependent (PMD) sublayer, and terms such as spread spectrum communications. Chapter 5, a rather lengthy one, is devoted to the discussion of functions and duties of the data link control layer. Chapter 6 builds on Chapter 5 using example protocols. HDLC, ATM and IEEE 802.11 MAC have been discussed. Chapter 7 is on multiplexing and

carrier systems, that make the backbone of transmission systems. T-1, though receding in its deployment, still makes an excellent case of studying a carrier system. Discussions on SONET/SDH and DSL are also included in this chapter. Chapter 8 provides a one shot treatment of the layers above the link layer. Terminology pertinent to these layers in introduced here. Chapter 9 completes the understanding, albeit at an elementary level, by discussing the topic of performance. The material in the book, proposed to be covered in one semester, could be adjusted according to the specialty of the audience.

In the end, I would like to thank and acknowledge numerous anonymous people who have contributed to this field in many ways. I have used many books, articles, websites and documents of numerous companies and standardization agencies in learning the subject, some repeatedly. Thanks are due to the reviewers of the manuscript. Also, thanks are due to the companies that designed software and hardware that went in preparing the manuscript. I also take this opportunity to thank Alex Green of the Kluwer Academic Publishers for his persistence in making this book a reality. Thanks are due to Melissa Sullivan and Deborah Doherty of the Kluwer Academic Publishers for help with formatting the manuscript. Most of all, I am much indebted to my wife for her patience during the preparation of the manuscript, especially during the final stage.

Aftab Ahmad

This page intentionally left blank

1. Computer Communications Networks - Introduction

The purpose of a computer communications network is to allow moving information from one point to another inside the network. The information could be stored on a device, such as a personal computer in the network, it could be generated live outside the network, such as speech, or could be generated by a process on another piece of information, such as automatic sales transactions at the end of a business day. The device does not necessarily have to be a computer; it could be a hard disk, a camera or even a printer on the network. Due to a large variety of information to be moved, and due to the fact that each type of information has its own conditions for intelligibility, the computer network has evolved into a highly complex system. Specialized knowledge from many areas of science and engineering goes into the design of networks. It is practically impossible for a single area of science or engineering to be entirely responsible for the design of all the components. Therefore, a study of computer networks branches into many areas as we go up from fundamentals to the advanced levels.

Advancements in communication of speech have long been matured in the form of public switched telephone network (PSTN). However, design of store-and-forward type of networks, such as the Internet, is far from matured - perhaps due to proliferation of the ways in which such networks are used. The integration of the two types of networks is the culmination of telecommunications technology. It is not futuristic to imagine telecommunications networks meeting the needs of live traffic (e.g., phone calls) as well as store-and-forward data (e.g., email) traffic according to the desired quality of service.

In this chapter, we look at a computer network as a whole, from both an application point of view and a design point of view. In fact, the design and application influence each other so much that a study of the fundamentals is practically impossible by leaving either one out.

1.1. Main Components

As described above, a computer network is composed of a number of independent components. Three main components are:

1. The Computer System
2. The Communications System
3. The Networking System

Data Networks

There is a myriad of terms used to describe a computer communications network. Computer Networks, Networks, Communications Networks, Telecommunications Networks, Packet Switched Networks, Networking Systems and Data Networking Systems are all among the terms used. The terms data network used to be more descriptive when live speech and video could not be transmitted over such networks. At that time, data meant store and forward type of information that had no real-time content. Examples of such information are file, email and logon programs. With time, definition of data has changed (see Chapter 3), and so has the definition of data networks. A data network now includes all of the above. The telephone network could be seen as an exception, even though it is also been used as data network in many of its applications. However, due to its circuit-switched nature, it is not projected to belong to computer networks.

1.1.1. The Computer System

Computer systems are stand-alone systems, along with peripheral devices, capable of performing information input, output, storage and processing. The study and design of computer systems is the job of computer scientists and engineers.

Computer systems usually consist of hardware (processor, memory, storage devices and input and output devices), system software for user interface and resource management, such as operating system and special purpose software such as programming languages, database management system, text-processing systems etc. Developments in microchip have led to the utilization of processor technology in everyday appliances, making all networkable devices operating like a computer system.

Examples of computer systems are: personal computers, notebook computers, and data acquisition systems.

What is a Terminal?

The computer communications network is an evolution of many networking stages; all for sharing software and hardware resources. Not long ago, the computer used to be lying awayfrom users in a big, air-conditioned room and the users used it through a network connecting a cluster of terminals connected to the main processor and input/output devices. Terminals were also used wherever information needed to be prepared for computing, transmission and storage. With the advancement in microchip, some processor functions started to be included in the terminals. This was the intelligent terminal as compared to the previously dumb terminal. With further improvements, much of the processing, memory and software could be available inside the terminal and it could be used without connecting to a big computer - it became personal computer. However, the network caught up with it very soon, and now all computers are essentially network terminals as well. Other terms used for computers are stations, nodes, hosts, devices, and whatnot! Other network terminals include printers, scanners, facsimile machines and disk drives. Some specialized devices are used for dedicated networking function, such as routers, bridges, repeaters and gateways.

1.1.2. The Communications System

The communications systems provide a vehicle of carrying information from one point to another by conditioning it appropriately. The conditioning may include changing the actual shape of the information, or even adding to and removing parts of it. Example of changing the actual shape of the information is in speech communication devices that take speech signal in the form of mechanical energy and generate an equivalent electrical signal suitable for transmission media. Examples of adding to the information is error control coding in which extra information is added in order to combat errors that might have entered the information during its movement inside the network. Example of removing information is data compression in which the size of the information is reduced yet preserving the amount of intelligence it represents.

The challenges in designing a communications system relate to the efficient usage of available network resources (bandwidth, etc.), reliable communication in the wake of channel noise, and special purpose requirements owing to applications that generate information or are the users

of the information (e.g., security) or other conditions (e.g., wireless, underwater).

A device known as MODEM (Modulator/demodulator) is an example of a communications system. In designing a MODEM for telephone line, the main challenge comes in utilizing the limited telephone bandwidth to transfer information at a maximum possible rate. The job of communication system design lies with the communications engineer. A communications engineer has to study the characteristics of information, the channel and the environment in order to design a system to meet specified performance criteria.

1.1.3. The Networking System

Networking systems provide the capability of efficient use of transmission and switching resources and provide with the rules that govern communication among computer systems and software programs. The design of networking systems is the job of the network engineer. A network engineer has to study the characteristics of the communication systems (designed by communications engineer) and computer systems (designed by computer scientist and engineers) to devise mechanisms of physical and logical interconnection of various computer systems via the communications systems.

Some of the challenges faced by a network engineer include the efficient use of communications link (using, for example, multiplexing), study of the characteristics of the information to be exchanged and its peculiar requirements of timing and bandwidth. Networks are designed to share communications resources and network engineer designs switching mechanisms for this purpose. Because of link sharing, security of information becomes very critical in networking systems. In this way, solving one problem raises another. The area of network engineering started as a conglomerate of computer science and communications engineering. However, it has fully grown into a field of knowledge by itself. Sometimes it means different things to different types of people involved - users, providers, and designers.

1.1.3.1. Communication Systems Versus Networking Systems

The communication systems and networking systems are two *different* fields of study altogether. Sometimes, confusion may arise as to if the design of a component of a computer network is part of the networking system or the communications system due to close interaction of the two. Let's look at two examples of such components: one relating to a hardware component and the other relating to a function. A hardware example is the MODEM. A MODEM design is the job of communications engineer and not the network engineer. The network engineer is a *user* of modem. An example of a function is error recovery. A communications system is design to be

robust in the presence of random errors that may interfere with electrical energy signals. The receiver design employs efficient *signal detection* mechanisms to minimize the probability of error. The communication engineer usually adds additional hardware or mechanism called a CODEC (from Coder/Decoder) to substantiate recovery from errors. A network engineer can add other layers of coding or use different methods for error control, such as, retransmission of information with errors. However, the concern of a network engineer is not at signal detection, but after the signals have been received. In this way, the end product may have many layers of error control, as part of both the networking and communication systems.

1.2. Network Development Example

In this section, we will consider an example of a computer from a user point of view. Later, we will see how these requirements of user translate to a language suitable for the designer of such a network. Consider a multi-national organization with offices and personnel computing needs in the following hierarchy.

1. Every employee (or at least office) needs to have a computer on every location.
2. Many people are working in multistory buildings at each location.
3. The company has many locations in metropolitan areas.
4. A large number of transactions with other national/international locations are carried out on a regular basis.
5. All computers must have some general-purpose software.
6. Many select computers require special purpose software.
7. Software and hardware sharing on each floor is desirable.
8. Software sharing among all floors in each building is desirable.
9. Some software sharing among all locations at the end-of-business-day is a requirement.
10. Some sharing in locations nationwide is desirable.
11. Transaction capability among international locations and with other businesses is also desirable.

1.2.1. Three Role Players

In order to solve the networking problem for the above business, three types of services/staff are involved. These are: user with the information technologists (IT staff), the network provider and the network vendor/designer.

User/IT Staff: The business in question is the user in this example. The IT staff is a permanent staff closely aware of the business needs and is expert of

the available software for special and general purpose. The IT staff provides what is called the information system services.

Network Provider: Another company will provide those networking services that are not required to be owned or/and are too high level, technically, to maintain by the IT staff. The network providers may not design their own systems, in general - they could be simply a carrier or operator company. Instead, the system design is a separate task not related to the business organization directly.

Network Designers/Vendors: A variety of equipment and services vendors may be involved directly or indirectly. Some may not directly interact with the business organization. Such a vendor designs and manufactures equipment to be operated by user and network provider.

We will look at their role further in the next section.

1.2.2. Network Design

With help from the IT staff and network provider, the company may end up with the *hierarchy* of networking systems shown in Figure 1-1.

The double-sided arrows show a bi-directional communications capability for sending and receiving information. To each user, the network connection should look transparent and direct with all networking levels, as shown in Figure 1-2.

This transparency of intermediate networks is a very important issue in the design of data communication networks. It is taken care by many layers of software and hardware. That will be the subject of many chapters; right now we focus on challenges for the three role players:

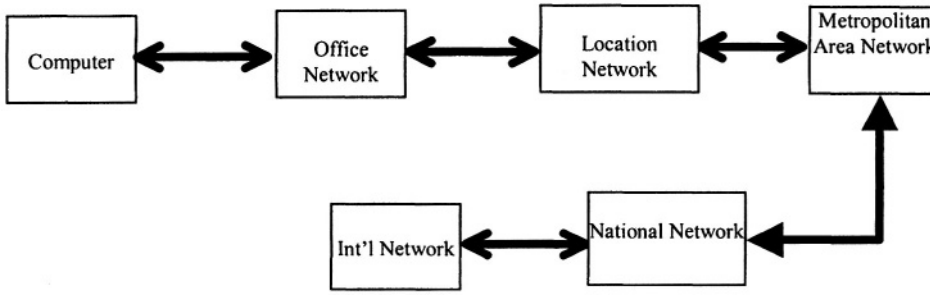


Figure 1-1. The hierarchy of networks to meet all the business needs for the example in section 1.2.

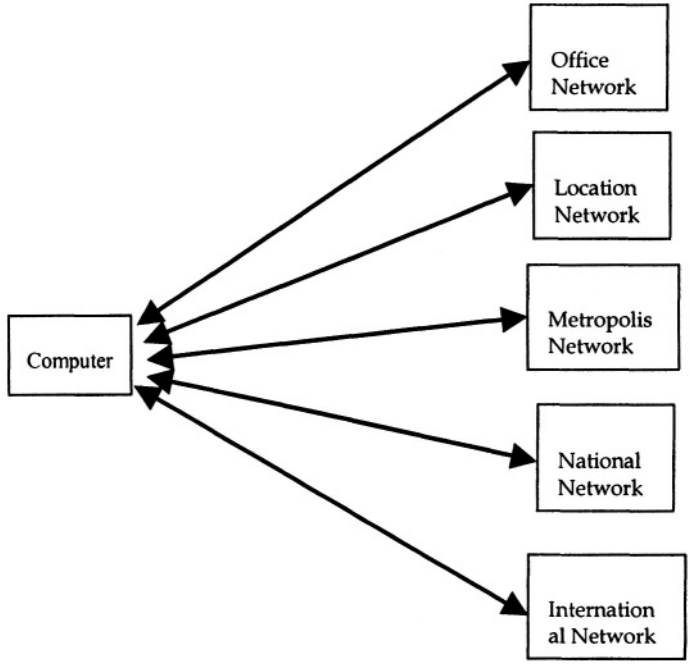


Figure 1-2. The logical connections between a user and the hierarchy of networks

1.2.2.1. User/IT Staff

User must own or be the sole controller of all the equipment responsible for internetworking of the computers and other devices within a floor, or even the whole building. This is for the security purposes as well as for the fact that things may change from time to time and computers and software may need updating frequently.

Therefore, the challenging task for the user is to decide from which vendor to make purchases so that the current and foreseeable, or otherwise future, needs may also be incorporated in the network design. The IT staff is hired for this purpose.

1.2.2.2. Network Provider

The network connection from building to building, nationally and internationally may be too expensive and inefficient for the user to own. Many companies provide such connections on subscription or leasing bases. The challenge for the network provider is to provide some guarantee of a reliable service in a cost effective manner.

1.2.2.3. Network Designer/Vendor

The network vendor designs the equipment as required by network provider and user for a diverse user base. The challenge for the designer is to have ready-made designs to meet most or all of the current and near future needs. The research and development department of the vendor must be able to predict future needs and, possibly, prototype future equipment.

1.2.2.4. Relevance of the text to the above role players

This text covers some of the fundamental essentials for all the three groups stated above. However, it may be sufficient only for the user part (IT professionals). The network provider must have the capability of having the design knowledge in order to customize and fully understand all possible and associated problems. The equipment vendor is the one who needs professionals with much deeper and diversified knowledge of all the three sub-systems. While both network provider and network designer usually need research and development departments, the emphasis of network provider is on the application of technology to be able to predict near or further future needs. The vendor and designer companies need people who could research the fundamentals of knowledge so as to bring about breakthroughs in technology. However, it is obvious that no amount of breakthroughs can satisfy all the users exactly according to their needs. This is because the equipment design process is rather long. New equipment cannot be designed every time a need arises for every user. There has to be equipment available in

the market from where the provider and user can choose. The process of standardization solves this problem.

1.3. Standardization

Standardization of networks and their components works just like standardization of any thing else: doors, nails, papers, pens and all. The purpose is also the same, namely, to make the vendors of related products know what 'dimensions' the hardware and software should have. Through the process of standardization of networking hardware and software, it is possible for the vendor to design equipment without consulting the network provider – such consultation is done during the standardization process.

Standards for networks encourage competition among vendors by allowing enhanced services. They provide user with the chance to shop around every time a new need of software, hardware or service arises. Most of all, it allows for *interoperability,* the capability of the equipment by one vendor to interact with the equipment designed by another vendor.

Once we understand the necessity of standardization, we can look into how the job of a computer network can be broken down into functions that could be standardized. In essence, the communication has to be broken into well-defined functions so that each function can be standardized. However, due to the diverse nature of the data to be communicated, it may not be a simple task to break the needs into same parts for all type of data. Let's take two examples in the next section: voice and file transfer.

1.3.1. Example 1 - Communication of Voice

In a voice communications network, such as the public switched telecommunications network (PSTN), the following is an important sequence of events in order for the voice to carry over the network.

`Key: type_of_signal (process/function) → next_type_of_signal`

Speech (Microphone)→ Electrical Signal (Analog/Digital Conversion)→ Digital Bit stream (Data Compression) → Shorter Bit stream (Channel Coding)→ Longer bit stream (Multiplexing/Synchronization)→ Transmission capable bit stream (Switching) → Routing { Opposite functions starting with multiplexing} → Speech at the receiving end

Following is a simple explanation of the key terms and functions mentioned above.

Analog/Digital Conversion is done in order to generate a digital signal (typically bit stream) from the analog speech signal.
Data Compression is performed in order to remove redundant part of speech signal for reducing the need of channel resources.
Channel Coding is done in order to detect/correct errors due to random noise.
Multiplexing/Synchronization are required when a number of speech signals are to be transmitted on a single, high-speed link. Multiplexing is the process of combining multiple signals on a single link and synchronization is the process of keeping track of each signal, its part or even each bit for later retrieval.
Switching is necessary when many signals destined for different recipients are being transmitted together on a single line. At many points, these signals are supposed to take a different route from one another. This route change is performed by switching the signals towards their desired directions. The way it is done is by having a device with several signals coming *into* it and being *switched* to different *outputs* depending upon their destination.
{ Opposite Functions starting with multiplexing } refers to the fact that at one stage the destination of signal arrives and the original speech signal is desired. This asks for the reversal of all the processes/functions. For examples, demultiplexing, decoding, etc.

1.3.2. Example 2 - File Transfer

File transfer may require several or even all of the above functions and procedures, but that may not be sufficient. In fact, there is an entirely different way of looking at the file transfer owing to the following chief differences between a file and speech contents:

1. The file contents are data-specific in their original form. There is no analog signal to be approximated into a digital signal. The bit stream is rigorously defined.
2. If something is not clearly understood in speech communication, recipient guesses or requests to repeat the sentence. In file transfer, the recipient is not a human being, but another file. Therefore a mechanism is needed to have equivalence of requesting a repeat or guessing a correct bit or string of bits.
3. An error can amount to a big loss depending upon where it occurs.
4. A big file in transit could cause network congestion.
5. There are issues such as which type of software program (called application program) will be used to process / look at the file contents and what format and language will be used by that application program.

So, here is an approximate breakdown of the procedure for file transfer.

Large File → Language/format comparison with recipient → Break up into manageable slices → Sequencing and information integrity embedded in each chunk → Routing → Combat channel/Link problem → physical transmission

In this example, instead of emphasizing on signal type, we care more about "readability" of the transmitted document. There is no need of format comparison, breaking up in manageable chunks, sequencing etc. in voice communication. Even functions common to both voice and file communications, such as routing, could have different implementations, and following paragraph explains how.

For speech communication, the gaps in talk spurts occur naturally and form an essential part of information. It is necessary that these gaps be maintained at the receiving point. However, such is not the case for a file that is stored in a directory at the sending end and would be stored at the receiving end in another directory. In other words, if we use a different path for each data block of the file with a sequence number stamped on it, we will not lose any information by having each chunk using different route. We can always look at the sequence numbers of the received data blocks and put them back in order. Not only that, if we make quite small, manageable chunks of file, we can process them individually, as if each one is from a separate user. So, if one of the chunks is in error, it can be requested again from the sending computer. In essence, even though both voice and file transfer need routing, the most suitable mechanism can be substantially different for the two.

The most favorable way for routing voice data is what is called circuit switching. File like data, on the other hand can best use 'chunk-based-switching' called packet switching. Here's a brief account of each (a detailed discussion will follow in Chapter 2).

1.3.2.1. Circuit Switching

In this switching mechanism, a circuit is allocated to every piece of complete information (called a call). This circuit allocation is all the way from the sending to the receiving computer or terminal. It stays in place throughout the duration of the call until the sending (or receiving) side signals that it is not needed any more. In more formal terms, we say that a fixed bandwidth is guaranteed throughout the communication session.

Circuit switching can be used for voice or file communication. However, it is easily seen that for file communication, it is best to send one part of file at a time. These chunk or data blocks are called packets. Each packet of the file may be transmitted via the same or different route. This allows for the number of additional functions and procedures that can be performed on each packet. Moreover, it arises a new type of switching, called packet switching for obvious reasons.

1.3.2.2. Packet Switching

In this type of switching, data is broken into smaller data units, called packets. Inside the network, each packet may be treated as if it were a small, complete message. The bandwidth can either be guaranteed for all packets or not. It is best suited for file transfer. However, if enough guarantees can be provided about the inter-gap times of voice signals, it can be used for voice as well (called packet voice).

With the brief description of the nature of circuit and packet switching, it is easy to imagine that packet switching is most suitable for a data network capable of transporting all kinds of information together. However, this will result in a lot of processing of information before and during communication. Classification of these processing functions and networks makes it easier to keep track of all the communications issues and their study. Computer communications networks can be classified in many ways: their geographic scope being one that could be easily described at this stage.

It must be noted that the following classification is not definition. It is meant to understand networks mainly from application point of view.

1.4. Classification of Networks

There are several ways of classifying data networks, such as, geographic scopes, protocol architectures and type of service. Following is a classification based on (roughly) the geographical scope.

1.4.1. Local Area Networks (LANs)

LANs are (usually) small networks that provide a high-speed physical and logical connection among a group of stations. They typically encompass a walk-able geographic area, owned and administered by the user and are mainly used either for hardware sharing or as access networks for greater geographical scale. Most commonly used LAN is the Ethernet.

In the example in section 2, the network on each floor or building is typically a LAN. Combining a few other LANs can also result in a LAN.

1.4.2. Wide Area Networks (WANs)

WANs cover a general geographical area that may vary from a small office area to the whole world (or even more!). Usually, network providers and big businesses own such networks. WANs are mostly *heterogeneous,* meaning, a large variety of LANs and equipments or other WANs can constitute a single WAN. An example of a WAN is the Internet. Internet spans much of the populated world, is administered by different groups at different locations, and has many other WANs as part of it.

1.4.3. Metropolitan Area Networks (MANs)

MANs are networks between a LAN and WAN. They are a type of interconnecting networks for big businesses in a metropolitan area. Usually, they have interconnecting (switching) devices instead of user desktop computers as their nodes, but it is possible to have user computers directly attached to a MAN.

One way to differentiate among LANs, WANs and MANs is the way transmission resources are accessed. Typically, LANs have uncontrolled shared medium, MANs are controlled shared medium access, and WANs have address-based, switched medium access through a separate network.

There are other types of networks in this classification. More recently, the term personal area networks (PANs) has got a legitimacy in networking literature due to the fact that they can be distinguished from other three types. PANs are networks interconnecting the devices of personal use that could, in general, be carries around. Examples of such devices are personal digital assistant (PDA), various wireless phones, remote control, etc. Usually, the design of these networks entails replacing the wire only, as they aren't expected to be interconnected via WANs and MANs in near future.

1.5. Network Protocol Architecture

In addition to classifying a network as LAN, MAN or WAN, there is a structured terminology to describe and identify various parts of the hardware and software making up a computer communication network. Three most important terms of this terminology are protocols, standards and network architecture.

1.5.1. Protocols

Protocols are rules of communication. It is through protocols that computers can exchange information. Just like humans obey certain rules of communications, so must the computers. Computers are specific about rules and cannot guess like humans. They have protocols as part of their software or hardware interaction and can’t change that unless the software or hardware is changed or modified.

1.5.2. Standards

Standards are the protocols that have gone through a standardization process. They are documented by some agency or organization so that a large number of vendors can get those documents and design systems based on the same protocols. This takes care of the *interoperability* issue and helps both vendors and users. Examples of standardization agencies are; the Internet Society, International Organization for Standardization (ISO), Institute of

Electrical and Electronic Engineers (IEEE) and American National Standards Institute (ANSI), European Telecommunications Standards Institute (ETSI) and International Telecommunications Union (ITU).

1.5.3. Protocol Architecture

Every computer and network needs a large number of protocols in order to complete data communications. The number of protocols can easily grow into several hundreds for a network. Besides, protocols take many different forms, from software to hardware, manufactured and designed by many companies. Different networks may have entirely different sets of protocols for every function of communications. Therefore, it may be helpful to classify protocols in groups in order to streamline a network layout. Automatically, this will help all sections of role players, user, provider and designer. A set of protocols specific to a network is sometimes called a *protocol suite.* When a subset of a protocol suite could be grouped together to perform functions that can be related to each other in communication terms, such a subset is often called a *layer or level.*

1.5.3.1. A Protocol Layer

A protocol layer is a set of protocols that perform a common (larger) function. Usually, a protocol layer consists a number of protocols. The concept of layering helps arrange the protocol suite as a set of layers. Then the job of defining a computer network is really taken in the following steps:

1. Define protocols in each layer.
2. Define all the layers needed.
3. Define interaction among layers in the same computer.
4. Define interaction among layers on different computers, intermediate and end stations.

By specifying the above guidelines, all the network communication can be defined as a set of protocol layers. Such a set of protocol layers is called as the network architecture.

In essence, a network architecture or *protocol architecture* is the set of layers and associated protocol specifications that can achieve complete communications among two or more computers connected via a network.

1.6. Example of a Protocol Architecture

Example architecture, and by far the most attractive (at least academically) in networking books, is the Open System Interconnection (OSI) reference model. This model was recommended by International Organization for Standardization for open system interconnection (OSI).

1.6.1. Open System

The term open system in OSI refers to the fact that the computer systems using OSI architecture will be open to communications to all systems designed by any vendor as long as they implement the same protocol architecture. Thus the specifications of the computer or hardware or operating systems play no role in interoperability of all the computers using the OSI architecture.

The OSI reference model (OSI-RM) breaks communications into seven layers. Each layer has a well-defined scope of its functions clearly identifiable from other layers. User information enters one layer at a time. Only one layer is responsible of actually sending the bit stream on the channel. Layers on the same computer can communicate only with the adjacent layers. Layers on different computers can communicate only with their peer layers. With these rules set aside, the user has the flexibility of shopping around for different layers and adding equipment from many vendors to an existing network.

1.7. Summary

A computer communications network is a complex system designed by many different, independent, software, hardware, and communication and networking professionals. The networking part mainly consists of designing efficient resource management methods and protocols to effect successful and reliable communication. Due to a large number of functions expected from protocols, their organization is very important according to their place in the process of communication. This may be helped by defining layers and network architectures. Usually, the design of layers that are closest to physical transmission is the subject of communications engineering. Logical functions of communications that are above the physical functions are typically for the network engineer to resolve. Software professionals deal with the application developments for stand-alone and networked systems. The applications make use of the networking protocols to get confidence in the exchanged information. The information is exchanged through physical circuits (or air) by either using a fixed path (circuit switching) or some less rigorously defined path (packet switching).

The study of data communications pertains to the study of all the layers of a network architecture from wires and cables to signal characteristics, to protocol definition, specification and coding, to management of networks. For this book our main emphasis is on the protocols relating to the physical transmission of bits, logical interpretation of the exchanged information between directly connected computers, and part of switching and routing mechanisms to route information through a network of inter-connected nodes.

1.8. Review Questions

1: Define a communications *protocol?*
2: What is the difference between a *protocol* and a *standard?*
3: What is the difference between a computer *operating system* (OS) and *network operating system* (NOS)?
4: What is a protocol *layer* and what is the chief benefit of defining layers?
5: What is *network architecture?*
6: How can a LAN and WAN be differentiated from each other?
7: What does *ISO* stand for and what is its purpose?
8: What does *OSI-RM* stand for and what is an *open system?*

2. Network Architectures - Examples

As seen towards the end of Chapter 1, protocols are organized into layers and architectures. We define and distinguish among networks from the protocol architectures used in their design and operation. In this chapter, we will look at some of the important reference models and actual protocol architectures. The first two models considered are most popular academically as well as in practice. The first is called the open system interconnection reference model (OSI-RM) and the second is called transmission control protocol/ Internet protocol (TCP/IP) suite. Perhaps, the reason why it is better to call TCP/IP as protocol suite instead of network architecture is due to the lack of strict definition of the lower levels leaving TCP and IP as the most important protocols. Many protocols of the TCP/IP suite have *evolved* rather than being documented in a well-defined layered paradigm. The process of evolution continues as the Internet outgrows itself and we keep welcoming new protocols and new versions of existing protocols.

The OSI model is a different story. The International Organization for Standardization (ISO) proposed this architecture. The ISO was created in 1946 for standards in trade and manufacturing. It has no limit to the items and categories under its jurisdiction of specifications and has a well-defined procedure for obtaining them. OSI reference model (OSI-RM) was developed in prediction of wide use of computer networking in future (which happened to be the case). Arguably, OSI really set up computer networking as an area distinct from communications and computer science. However, OSI network architecture is not as widely implemented as TCP/IP. In spite of that, the OSI-RM still provides an excellent platform for understanding of networks at elementary level. There is another reason to include the OSI-RM and TCP/IP in this chapter, that is, the main protocol examples considered in this text are proposed by ISO as part of OSI network and are also used with TCP/IP protocol suite.

In the rest of the chapter, we will look at the characteristics of OSI-RM, the TCP/IP suite and the protocol architecture for wireless LANs. Following the examples, we will have a brief discussion on the working of ISO, Internet Society and some other standardization organizations. In the end, we will draw a framework for protocol study that may help in understanding a given protocol, software or hardware, and at any layer or level.

2.1. The OSI Reference Model (OSI-RM)

In the OSI-RM, the network architecture consists of the following seven layers: layer seven being closest to the user interface.

1. *Physical Layer* (PHY) is the protocol layer responsible for physical interface between a computer and an OSI network.
2. *Data Link Control Layer* (DLC) is responsible for specifications of the logical connection across a physical link that directly connects two communication stations in an OSI network.
3. *Network Layer* provides specifications for options, mechanisms and algorithms for routing data through the OSI network.
4. *Transport Layer* (TL) takes care of the imperfections of network and lower layers by providing end-to-end reliability functions.
5. *Session Layer* provides specifications for managing the communication session between two applications across an OSI network by facilitating the dialogue and inserting checkpoints in a large sequence of data bits.
6. *Presentation Layer* provides information syntax and formatting specifications to facilitate communication between applications that could otherwise be using different formatting structures.
7. *Application Layer* provides specifications to design application program interfaces for OSI networks.

2.1.1. OSI-RM Characteristics and Terminology

The reference model assumes a packet-by-packet communications. In the same stack, a layer can communicate only with the adjacent layers. In other words, layer number N can exchange messages with either layer number N+1 or N-1. Each layer has an address within the computer. This address is called a service access point (SAP). A layer interaction with adjacent layers is shown in Figure 2-1. A different vendor may provide each layer as long as the protocol specifications are not violated.

The communication via SAPs occurs through the use of programs called *primitives.* A primitive is a software program (better yet a procedure or function) that contains data and other parameters to be transferred to the next adjacent layer. In OSI terminology, a layer invokes or requests services from the layer below and provides services to the layer above. Thus, in Figure 2-1, layer number N provides services to layer number N+1 while it requests services from layer number N-1.

Interlayer communication among communicating computers is provided as follows. Each layer attaches additional bits to the data that it receives from the layer above. These bits are variously called *header* or *trailer, protocol information, protocol header* etc. The headers and trailers are used in communication between peer layers. In other words, the *peer-to-peer protocols* are imbedded in the header or trailer of a packet. A data packet

along with the header and/or trailer is sometimes called a protocol data unit or simply PDU.

Protocol Data Unit (PDU) is a combination of data and protocol information *for the peer-to-peer protocols.*

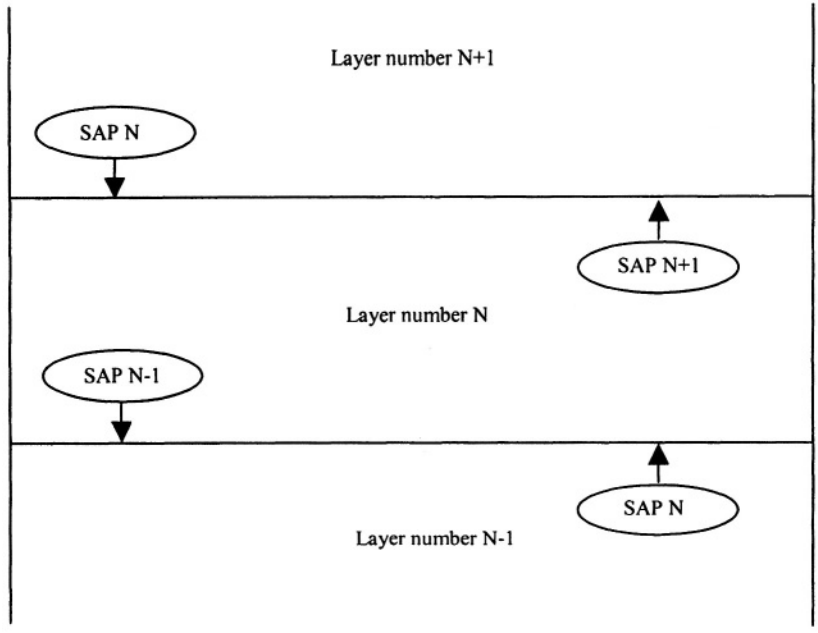


Figure 2-1. Interlayer communication in OSI-RM

2.1.2. Communications Model within an OSI Node

With the above definitions and rules understood, we can imagine the flow of data in an OSI network. The first observation to be made is that the user interacts with only the application software. The second observation is that the actual bit transmission occurs only at the PHYsical layer. Layers other than the PHYsical communicate with their peers only through protocol headers/trailers. The following is a communications model for an OSI network.

User data enters the application layer via the application program. The application layer attaches protocol information in the form of a header. In Figure 2-2, the resulting PDU is called application data or application PDU (A-PDU). The A-PDU is passed onto the presentation layer that is the layer directly below application. The presentation layer adds its own protocol information resulting in presentation data or (P-PDU) and passes it on to the next lower layer. In this way, information data travels all the way down to the PHYsical layer. It is the PHYsical layer that actually transmits data on a medium. On the receiving computer, it is again the PHYsical layer that receives data from the medium and passes it on to the data link layer as DL-PDU or data link layer protocol data unit. The data link layer strips off the data link header/trailer that was added by the sending data link layer, decodes the protocol information in its header/trailer, performs the relevant functions and passes on the remaining (network layer) PDU up to the network layer. It appears as a N-PDU to the network layer. The network layer strips off the network layer header, performs requisite tasks and sends up the remaining portion to the transport layer (T-PDU), which sends all but the transport header to the session layer as S-PDU and so on. In this way, the user gets only the user data from the application layer. After receiving the data from the application layer, the user is free to store or process the data with the help of appropriate application software.

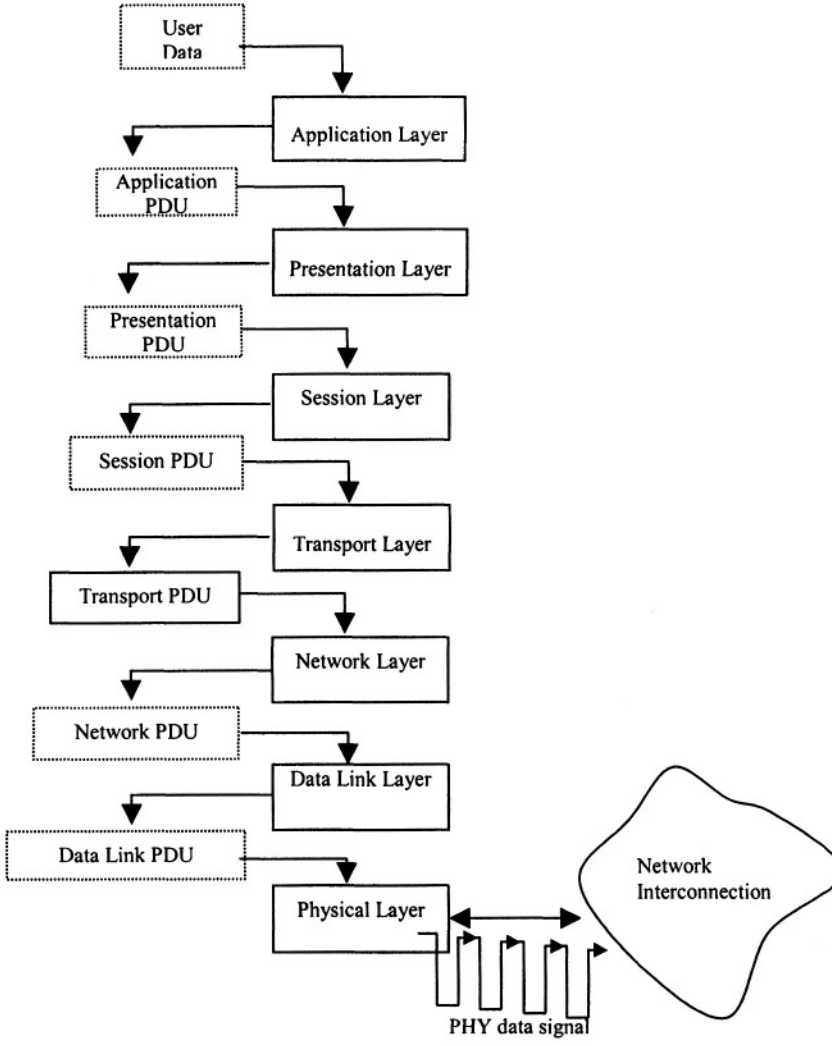


Figure 2-2. The working of OSI layers.

2.1.3. Communications Across the OSI Network

With the above communications model, we see how user data leaves and enters a computer connected to the OSI network. Next, let's see how it travels across a network. We will look at the network example of one intermediate node that is representative of the network.

Figure 2-3 shows different connections and the flow of data as it goes from one application in a sending computer, through an interconnecting node, to another application in another computer. Several observations are in order:

1. Most importantly, there are only three bottom layers in the intermediate node. That does not stop the intermediate node from implementing all the layers. However, in its role as the intermediate node, these three are the only ones needed.
2. The connection is shown as solid lines between the PHYsical layers and dotted lines between all other layers. This is to remind us that the actual physical connection exists only at the PHYsical layer. The higher layers are said to form a logical connection. That is to say, they communicate through protocol header/trailer information only.
3. The third important observation from the figure is that layers above the network do not have to know anything about the operation of network and lower layers. This is an important job of the network layer, namely, to provide network independence to the higher layers.

Note: The N-PDU entering the intermediate N-Layer is in general different from the N-PDU exiting it. The reason for this is that as soon as the network layer receives the N-PDU, it strips off all the protocol information that tells it what procedures need to be performed on the data. After executing the required functions, the network layer adds its own header for the next network layer.

The layers 4 and above are sometimes referred to as the end-to-end layers.

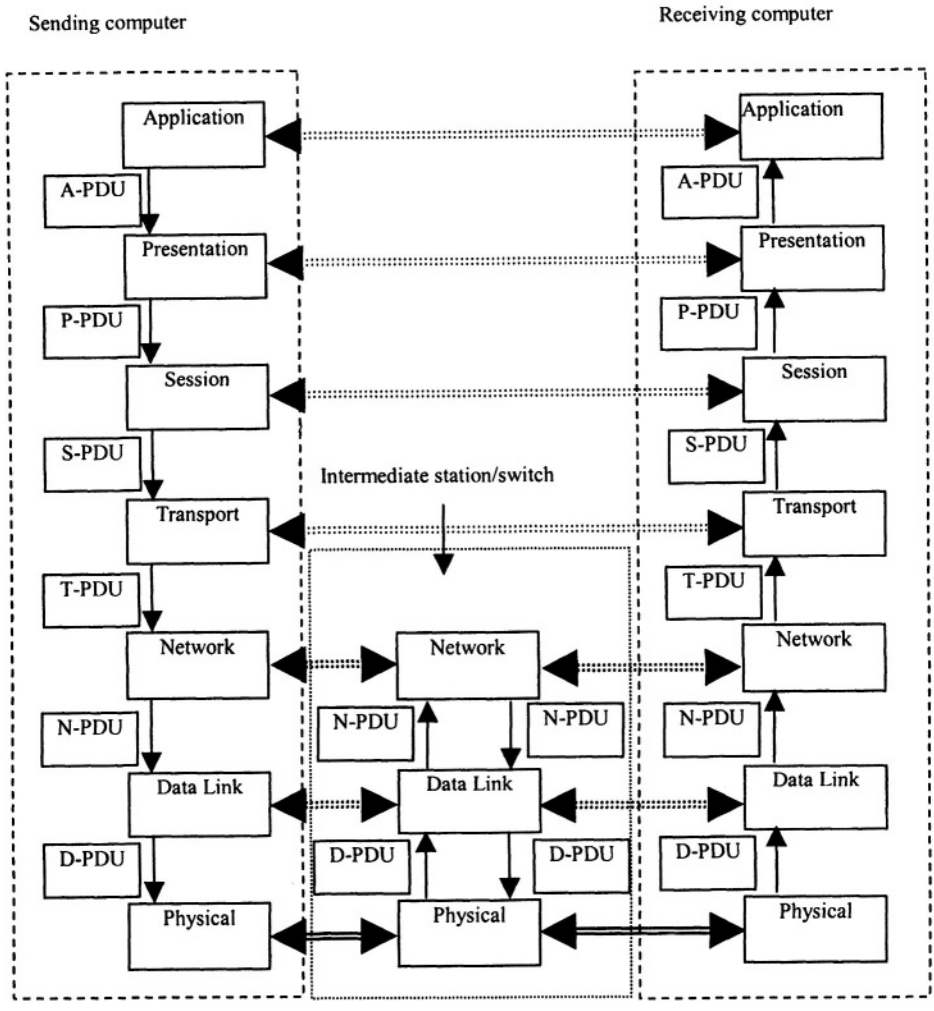


Figure 2-3. Communications flow across an OSI network

2.1.4. Inter-layer communication

As said earlier, a layer can communicate with adjacent layers *(only).* This communication occurs through primitives. Primitives are software procedures used to request a service, indicate a request, respond to and

confirm the execution of a service. There can be several types of primitives, each one used for a particular task. Usually, the name of a primitive is chosen to indicate its task. There are four types of primitives defined in the OSI-RM: Request, Indication, Response and Confirm.

The *Request* and the *Confirm* primitives are used at the sending station while the *Indication* and the *Response* primitives are used in the receiving station as shown in Figure 2-4. Following is a brief account of the meaning of each:

Request primitive is initiated by a layer in the sending station to invoke or request a service from the layer below.

Indication primitive is issued by layer (N) at the receiving station to indicate to the layer above (layer N+1) that a request has been placed by the layer (N+1) of the sending station.

Response is the primitive used by a layer at the receiving station to indicate to the layer below whether the requested service can be provided or not. In other words, this primitive is used in response to the Indication primitive.

Confirm is the primitive initiated by a layer (layer N) at the sending station to the layer above (layer N+1) informing the response of the peer layer (layer N+1) of the receiving station.

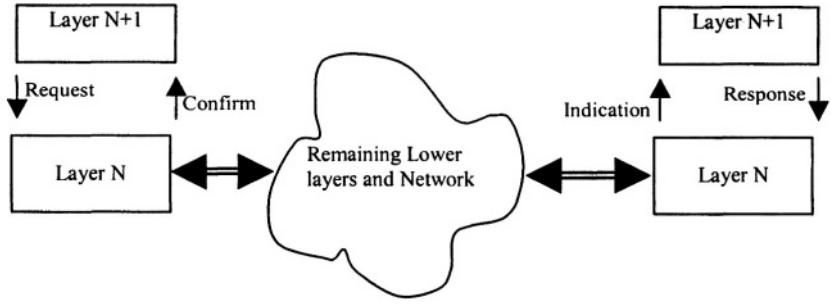


Figure 2-4. Use of primitives in the OSI-RM.

2.1.4.1. The Role of the Lower Layers

As see from Figure 2-4, when a layer receives a Request primitive, it has to communicate the information to its peer layer across the link or in the receiving computer across the network. It employs the services of lower layers for this purpose. The lower layers, working in the same paradigm, communicate the request, through the network, to the peer layer of the

destination computer. Therefore, what was a Request primitive sent by layer N+l at the sending computer appears as Indication primitive to the same layer at the receiving station. Similarly, the Response primitive from layer N+l at the destination computer is routed through the network to appear as Confirm primitive to the same layer at the sending station.

If all the four primitives are employed in a service invocation and provision, such a service is said to be a *confirmed service.* Alternatively, a *non-confirmed service* will make use of only the Request and Indication primitives. In such cases, a layer after requesting the service, will assume that the service is available. In case there is a problem in complying with the request, it will be known at a later stage.

2.1.5. OSI-RM Layer Definitions and Functions

In the following, we will describe the functions of each of the seven layers of OSI-reference model.

2.1.5.1. The Physical Layer

The physical layer provides network interface for a successful transmission and detection of bits. Protocols on this layer specify the mechanical, electrical, functional, procedural and transmission characteristics of the interface.

The mechanical characteristics pertain to the types of connector and socket. Since the pin functions will be the same for the connector and the socket, it is immaterial which is on the computer or network except for vendor independence. Electrical characteristics of the specifications pertain to the voltage levels, signal duration and decision regions for signal detecting equipment. Decision regions are ranges of voltage values that define the existence of a particular signal on the cable. Decision regions help the bit-detecting device to decide whether the received bit is a zero or a one. Functional characteristics of the physical interface specify the capabilities of the interface. These functions can be several which can be either all provided or a subset thereof. Synchronization is an example of a function. This is a function widely implemented in synchronous as well asynchronous transmission systems and helps the receiver keep track of the beginning and ending of bits, or blocks of bits. Procedural characteristics define how functions should be implemented or realized. The example of function above can also be extended here. Specification of bit synchronization as a function would say that bit synchronization would be provided in a particular standard. The specification of the procedure will specify how it will be provided and what are the tolerances to be employed. Transmission characteristics of a physical layer are related to the transmission medium and the layers above the physical layers. These characteristics would include specifications of signal shape, speed and signal form (optical, radio etc.).

Generally, the physical layer specification is the most complex part of the network architecture. The complexity depends, however, on factors such as bit rate, levels of security, performance sought from the network and cable type used for physical connection. Generally speaking, higher the bit rates, more complex the physical layer. Also, for unreliable transmission medium, such as the air, complex functions and procedures are specified for reliable communication. These functions include transmission characteristics such as, modulation, coding, interleaving and so on. For high-speed networks, physical layer is sometimes divided into two or more parts, one to address issues related to the medium (medium dependent) and the other to conform physical layer data to the higher layer format (convergence sublayer). In cases like these, physical layer transmission may include both bit-by-bit and block transmission.

Examples of mechanical specifications are the data connectors for Local Area Networks, e.g., RJ45 and BNC (or the T-) connectors. Example specifications of electrical characteristics of an interface may read something like this "A voltage pulse of value less than -3 volts is interpreted as a binary '1' while a voltage level greater than 3 volts is interpreted as a binary '0'".

2.1.5.2. The Data Link Control Layer (DLC)

The DLC layer is located right above the physical layer. It provides for a logical connection between directly connected computers and other devices via a wire or through air/space. The meaning of directly connected can be confusing as sometime the data link layer operation exists between two computers separated by many other computers. An example of such connection is a Local Area Network (LAN). However, the data received is not changed by an intervening computer – thus ascertaining the 'direct connection'.

While data may be transmitted at the physical layer bit-by-bit, the data link layer looks at it block by block. These blocks of data at the DLC are sometimes called *frames.* DLC layer provides functions for frames instead of individual bits. This could essentially be the separating line between circuit-switched and packet-switched networks. For a circuit-switching network such as the telephone network, there is no need of data link control layer as there is no need to frame or process information on intermediate switches.

Each link in a network data path consisting of many links can have its own noise and bandwidth limits. Therefore, functions such as error recovery and flow control are essential to the specification of DLC layer protocols. One of the important functions of any DLC layer is the addressing mode that it supports. There are three potential addressing mechanisms that are generally needed in all networks. These are:

(i) Point-to-point (unicast) addressing in which two processes or computers communicate with each other without any other station having access to the user information.

(ii) Point-to-multipoint broadcast in which all the machines on a link are the receivers of the information whether they all need it or not.

(iii) Point-to-multipoint multicast in which a specified group is the recipient of the information.

In addition to addressing mode, the DLC layer takes care of the type of physical link with respect to the direction of transmission. For example if the link is *simplex* allowing communication only in one direction, DLC layer sends and received data on different physical links. If the link is *full duplex* then it may provide for simultaneous data exchange on the same link. If the link is *half-duplex* allowing transmission in one direction at a time, then DLC layer may have the capability of using the same link by multiplexing information in the two directions.

Sometimes, a DLC layer is simply called a link layer. In some network architectures, the link layer is divided into two sublayers each performing a task separate from the other. An example of this is a shared medium Local Area Network (LAN). For such LANs, a medium access control (MAC) layer is defined as a sublayer of the link layer. Its exclusive job is to define mechanisms and limits on accessing the shared medium. A second sublayer, above the MAC sublayer, would provide the standard functions of a DLC layer.

Examples of data link layer protocols are the ISO's high-level data link control (HDLC) protocol and IEEE802.2 Logical Link Control (LLC).

2.1.5.3. The Network Layer (NET)

Layer three in the OSI reference model is the network layer. Its main function is networking and internetworking. Networking pertains to routing of data through an interconnection of transmission facilities under a common network administration. Inter-networking implies networking of networks. Internetworking is routing of data usually based on decisions and agreements reached by the administrations of the networks involved. *Switching* is another term used for routing of data. When data consists of packets, as in OSI networks, 'packet switching' is preferable to route N-PDUs through the network. Each PDU in this case could be treated independent of all other PDUs as if it were a complete message. Alternatively, all packets of one call can be treated in the same manner for routing purpose. In other words, packet switching is of two types. We describe the difference between the two in the following.

2.1.5.3.1. Datagram or Connectionless Switching

This is a packet switching mechanism in which all the packets are not required to take the same route in the network. Since different network paths could be of different length, the packets are not guaranteed to arrive in order in which they were transmitted. In many protocols even the delivery is not guaranteed. In fact, sometime we refer to connectionless routing as *unreliable datagram delivery.* An example of a protocol using datagram switching is the Internet Protocol (IP).

A network layer providing datagram service treats each packet of data just like another call. A packet in connectionless network service is sometimes called a datagram. We will discuss attributes of a connectionless packet switching further when we talk about the Internet suite of protocols later in this chapter. At this point, we describe the second type of packet switching called virtual circuit (VC) switching.

2.1.5.3.2. Virtual Circuit (VC) Switching

Virtual circuit is the term used for packet switching route that is fixed in some sense for all the packets of a call for the duration of the call. It implies in-sequence delivery of packets. A virtual circuit, once established, provides a guaranteed path for the call duration. When a call using VC is completed, the network path used as VC must be released and marked as 'available'. Therefore, there are three phases of a call using a VC connection (VCC). In phase I, the *connection setup* phase, a VC to the destination is requested by the computer that is the sender of data. The network layer on receiving the request finds out a path between the two stations. This path is identified as a number, called *virtual circuit identifier* (VCI). If the VC consists of more than one intermediate links - or *hops* - each hop may have a different VCI.

On receiving a VCI, the sending station goes into phase II of VC, the *data transfer phase*. In this phase, user data is transmitted in the form of packets. Each packet has the VCI stamped on it. Network layer on the intermediate nodes needs VCI information to switch the packets to appropriate destination or another intermediate node. Each intermediate node (*switching nodes*) may stamp a new VCI on the outgoing packets. The switching nodes maintain a table with assigned outgoing VCI for each given incoming VCI.

Once all the data packets have been exchanged, the third phase of VC switching takes effect. In this phase, the VC is terminated. The termination procedure can be protocol specific. In some cases, the sending and receiving stations (or one of them) informs the network layer that the data transfer phase is complete. The network layer simply marks the affected VC links as 'available' again. Alternatively, an absence of packets on a VC for some specified time could be taken as a signal of completion of data transfer phase.

In this case, the network layer will automatically release the circuits on expiry of this time.

Sometimes, packet switched networks are called store and forward networks. This is because a packet is stored in a buffer on all intermediate nodes before switching (*forwarding*) to the next leg of the VC (or datagram path). An inevitable result of the store-and-forward switching is a variable amount of delay introduced in packet delivery to destination. This is because every switching node could be serving a different number of VCs on the way. Thus, the time of waiting (*store*) before being forwarded could be different for different packets at different intermediate nodes. Some services are not delay sensitive, such as, email. Some others are delay sensitive, such as any real-time event. In order to maximize the use of network resources and to guarantee user satisfaction, the network layer has to take such requirements of user data into consideration. This is the most important issue in any network that is to provide multiple services (called multimedia or integrated services). Research on the next generation of Internet is overwhelmed with this issue, as the market for such services already exists. However, until now, the most successful way of guaranteeing users' delay needs is by circuit switching. It is, therefore, considered appropriate to include a discussion on circuit switching at this point.

2.1.5.3.3. Circuit Switching (CS)

Circuit switching is similar to VC except that the packetization of data is not necessary in this case. In circuit switching, a network path (and bandwidth) is dedicated to a call for the complete duration. The sending station can send data as it is generated without necessarily packetizing or storing it. Therefore, there is no store-and-forward mechanism needed in a circuit switched network. Consequently, the only delay that data suffers is the propagation delay. Additionally, before data transmission may take place, there is a call setup phase in which the network layers find out the path to be used for data in data transmission.

An example of circuit-switched network is the public telephone network. When a user in a telephone network dials the number of the called party, the network uses that number to find out the path for voice communications. Once the path is established and allocated to the pair of calling and called parties, it is dedicated for the call duration. When either party wants to terminate the call (by hanging up), or the network control system detects that the allocated time is over, the path is released and can be used by other users. Thus, the CS has the same three phases as VC.

2.1.5.3.4. A Comparison of Switching Schemes

The three switching mechanisms discussed above have some distinct and some overlapping characteristics. Figure 2-5 is one way of looking at the differences and commonalities among the three.

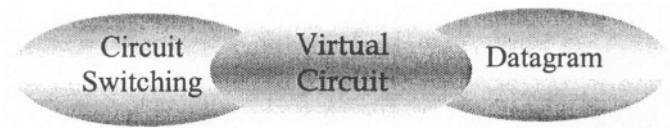


Figure 2-5. Logical relation among switching mechanisms.

As depicted in this figure, circuit switching has something in common with VC switching. These commonalities include: the same route (or VCI) for all data in a call, and call setup and circuit release phases. Similarly, there are many features common to the VC switching and datagram switching, including packetization and implementation of reliability functions of flow and error control (to be discussed in detail in chapter on functions of data link layer).

2.1.5.3.5. Quality of Service (QoS)

From a user's point of view, the differences in the three types of data switching appear in the form of quality of service (QoS). For voice communications, QoS may be defined as the intelligibility and clarity of speech. This asks for retaining the natural gaps between adjacent talkspurts during transmission. In other words, voice cannot tolerate too much delay. For data, such as, email and file transfer, QoS may be defined as data integrity. This means that there should be little or no data loss. For other applications, such as streaming video, it could be both the clarity as well as integrity. Due to the store-and-forward nature of packet switching, it is obvious that CS is the most suitable for speech communications.

Packetization of data in VC and datagram allows for adding the capability of retransmission of packets. The transmitting station implements retransmission mechanism by retaining a copy of all packets in a buffer even after they have been transmitted. If the receiving station is satisfied with the quality of the received packet, it will accept that packet. Otherwise, it requests the sending node to retransmit another copy of the packet. This makes it more suitable for reliable data transmission. The user applications that require high data integrity are sometimes are called as *loss-sensitive.* File transfer is an

example of loss-sensitive applications. Voice, on the other hand is an example of a *delay-sensitive* application. It is safe to say that CS is suited to delay-sensitive applications while packet switching is best for loss-sensitive data traffic.

When a path is dedicated to a call (e.g., speech) in a circuit switched network, no other data or information can be transmitted on that path until the call is disconnected. It is well known that during conversation, the actual speech is present only about 40% of the time. Thus, the circuit remains unused for almost 60% of the time. Such does not have to be the case with packet switching. In packet switching, a circuit exists only in a virtual sense. There is not a physical path associated with either datagram or VC. Therefore, when a path is unused, it can be shared by another call or packet. Consequently, the path can be utilized more efficiently in packet switching than in circuit switching. Since each link on a network path has a certain bandwidth, we can claim that packet switching potentially has higher bandwidth utilization than circuit switching. Thus, the QoS guarantee of the circuit switching is at the cost of inefficiency of bandwidth utilization.

In the following, we explain the differences among the three types of switching with the help of an example.

Example 2-1: Consider the network shown in Figure 2-6(a). A 9,000-bit file is to be transferred from node A to node E

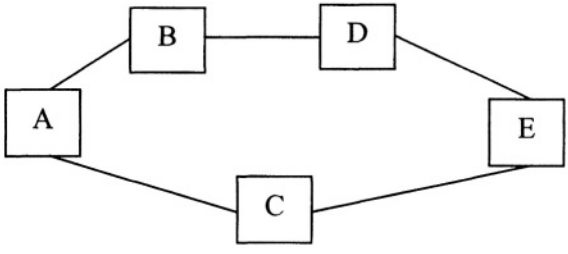


Figure 2-6(a). Network in Example 2-1.

Suppose that the time to setup a call for CS and VC is 5 sec each and it takes 1 second for a 1000 bit data packet to be transmitted from one node to another (transmission time plus store-and-forward time). For VC and datagram, the file is broken into 1000 bit data packets, with 900 bits of file data and 100 bits of protocol overhead (headers and/or trailers). Compare the total delay for transferring the file from A to E. For datagram, assume that

node A sends packets alternatively through B and C. Assume no propagation time.

Solution: Let's consider the three types one-by-one.
CS: Call setup time = 5 sec
Assuming that the transmission time increases linearly with the data size (not always a valid assumption!), the transmission time = 9000/1000 = 9 secs.
The general expression for the transmission time for data of size L = L/R, where R is the data rate (amount of data transmitted in 1 sec), also variously called the transmission arte, link capacity, etc.
Total data transfer time = 5 + 9 = 14 secs.
Note: Since data is not stored or forwarded, it is immaterial in this case what is the exact path for a CS. There are two possibilities: $A \rightarrow C \rightarrow E$ and $A \rightarrow B \rightarrow D \rightarrow E$. Both will give the same result.
VC: In case of VC, there will be a total of 9000/900 = 10 data packets, each of 900 data bits and 100 overhead bits. So, each packet is 1000 bit size and will spend 1 sec on each intermediate node. However, since more than one packets can be in transit together, we may have to follow the trail of packets carefully.

There are two possibilities of route, $A \rightarrow C \rightarrow E$ and $A \rightarrow B \rightarrow D \rightarrow E$. The two routes will result in different delays due to store-and-forward nature of the VC connection. We assume the shortest route is selected as shown below in Figure 2-6(b):

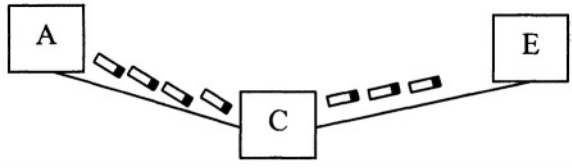


Figure 2-6(b)

We make the observation that before a packet reaches E, it is stored and forwarded twice. Therefore the total time can be split into following categories:
Call setup time = 5 secs.
Total transfer time for first packet to go from A to C = 1 sec
Total time for first packet to go from C to E *and* second packet from A to C = 1 sec
Total time for first packet to go from A to E = 1 + 1 = 2 secs.

Additional time for each of the second (and later) packets to go from A to E = 0 + 1 = 1 sec
Total time for all packets to go from A to E = 2 + 9 × 1 = 11 secs.
Total time for call setup and data transfer = 5+ 11 = 16 secs.
Datagram switching: Since it is given that node A sends packets on the two outgoing links alternatively, we can visualize the packet paths shown in

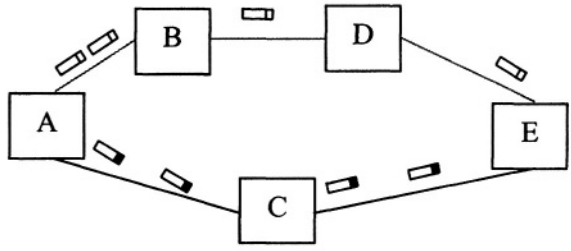


Figure 2-6(c)

Figure 2-6(c).
Here's an analysis of the total time to transfer all the data from A to E.
Call setup time = 0 sec (always so by definition of datagram switching)
Time for first packet to go from A to C = 1 sec
Time for first packet to go from C to E and second packet from A to B = 1 sec
Time for second packet to go from B to D and third packet from A to C = 1 sec
Time for second packet to go from D to E, third packet from C to E and fourth packet from A to B = 1 sec
Since each route will take five packets, we reach the following conclusion:
First packet (#1) total transfer time = 2 secs (route A to C to E)
Additional time for two packets (#2,#3 and #4,#5 and #6,#7 and #8,#9) together (one from each route) = 2 secs.
Additional time for packet #10 =2 secs.
Total time for file transfer = 2 + 4×2 + 2 = 12 secs.

Putting our analysis results together, we get the total data transfer time as follows: Delays for,
Circuit Switching = 14 secs., Virtual Circuit = 16 secs., Datagram switching = 12 secs.

Point of discussion: What details are missing in this example and how will they affect the results?

2.1.5.4. The Transport Layer (TL)

The transport layer works independent of the network infrastructure. TL communicates end-to-end across the network to ascertain reliability of the exchanged packets. Besides data integrity, it provides the functions of data link layer on the network path. Such functions include multiplexing, addressing and flow and congestion control. In directly connected stations, TL does not provide any significant service. However, as the size of network expands and communicating computers get further from each other, the need of TL becomes obvious. OSI network has set of five different protocols at this layer each one implementing a different set of capabilities. These are named as TP0, TP1, TP2, TP3 and TP4. Since the functions provided by TL are similar to the functions of DLC layer, we will not go into the detail of these functions at this point.

Just like the network layer, the transport layer protocols can either be connectionless or *connection-oriented.* In a connection-oriented protocol, the TL protocols on the sending and receiving computers first establish an understanding (called as a *logical connection*) that the two are going to exchange data, through a call setup phase. It is only after having established a connection that the two protocols can send and receive data packets for each other. In a connectionless TL protocol, a packet may be transmitted at any time without prior notification or understanding. A connectionless transport protocol is suitable for short (typically single-packet) messages for a quick task or query made by a client to a server. An example of such task would be logging on to a remote computer. Internet suite of protocols has the User Datagram Protocol (UDP) that provides end-to-end functions similar to the transport layer of the OSI-RM.

2.1.5.5. The Session Layer

The session layer manages long sessions of data communications between logically connected applications in computers across a network. It would perhaps not be critically needed by real-time applications. However, in file transfer when the file size is very large, it could provide useful functions. It controls the dialogue between the communicating stations so that communication occurs with proper ‘turns’. It also provides other functions, such as, check pointing by allowing for stops in a long communication sessions to check for any problems.

2.1.5.6. The Presentation Layer

Emulation, syntax coordination and format exchange are done at the presentation layer. These functions are to be agreed upon by the two communicating sides before the actual transmission starts. As an example, if one of the computers necessitates encryption of all data, then this layer could

provide a common format for encryption. Another example would be data compression.

2.1.5.7. The Application Layer

The application layer specifies the network interface for user applications. The application program prepares data for communication and the application layer prepares the application header to be understood by the receiving application layer. The receiving application, after reading the header, provides information to the receiving application about how data has been arranged in the application packet. An example would make it a little clearer.

Electronic mail is an application program. The user might prepare email by using text-editing or word-processing software. The mail protocol defines how to combine addresses, file attachments, and graphic and text information for transmission. This gives the users the flexibility of using any of the several mail utility packages available in the software market. No matter how different they look to a user, they all must obey a common protocol for interoperability. The same applies to file transfer and all other network and distributed applications.

We will, at this point, move on to the second example of network architecture and look at the main components of the Internet suite of protocols, sometimes called TCP/IP suite of protocols. The reason why we chose to describe OSI-RM as the very first example is that it provides a more systematic and comprehensive view of the networking functions. Some reference models introduced later had included lessons learnt from the development of OSI itself. These lessons related to some vagueness in the boundaries between upper layers and a lack of the network management infrastructure as part of network architecture. Examples of such architectures are the broadband integrated services digital network (B-ISDN) and IEEE802.11 wireless local area network (WLAN). It is customary in networking literature to consider networks other than the OSI as having equivalent OSI-RM layers. This is also necessary sometimes due to the lack of a reference model for some very popular network architectures. The one discussed next stands out in this respect. Being the most popular architecture, TCP/IP suite really changed the way we live. Yet, there is no reference model for this architecture. That does not, however, alter the need to study and understand it. The fact of the matter is that we use OSI-RM to study TCP/IP more often than a protocol suite using entirely OSI protocols.

Point of discussion: What is the difference between a reference model and a protocol suite?

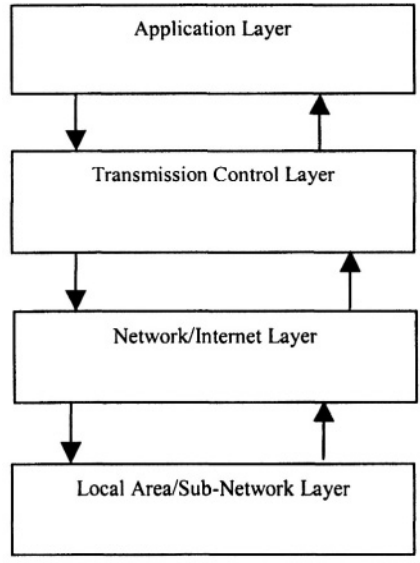


Figure 2-7. A possible layered model for the Internet protocol suite.

2.2. The TCP/IP Protocol Suite

The main power of the Internet lies in two protocols, the Transmission Control Protocol (TCP) and the Internet Protocol (IP). Because of this, the Internet suite of protocols is referred to as the TCP/IP protocol suite. TCP is very close to the definition of transport layer of the OSI-RM. It provides a connection-oriented delivery of data between two distributed applications in the Internet. TCP provides this reliable service on the top of an unreliable IP. IP is responsible for the routing and switching of datagrams through the Internet. The exact location of TCP and IP protocols with respect to the OSI-RM has been the subject of some debate among the networking community. This could be due to the application protocols being located directly above TCP. In addition, there are protocols parallel to TCP as well as IP. It would be safe to say that there is no reference model followed in the development of the Internet suite of protocol[1]. There is one that came into

[1] For a discussion on ARPANET reference model (ARM), see the Internet Request for Comments Number 871.

being owing to the need for defining interlayer and layer-to-layer relation. It is shown in Figure 2-7.

Note: Some of the authors of books on computer networking compare OSI-RM with TCP/IP suite of protocols instead of a reference model for TCP/IP suite. The reason perhaps is the same as stated above, there was no reference model developed for the Internet. It just kept evolving until a certain point when it was realized that its popularity is not just outnumbering the address space needed for each machine to be on the Internet, but also a need of understanding it in light of the OSI-RM. In spite of an overwhelming popularity of the Internet and a large number of people learning it, there is no real reference model.

The terms used for the layers in Figure 2-7 are not standard and only describe a common larger function except, may be, for the application layer, which is more clearly defined. However, the task/s of each box shown are very well defined and documented into standards.

A standard practice is to show the protocol stack for the TCP/IP suite and compare it with the OSI-RM. Such a stack is shown in the Figure 2-8.

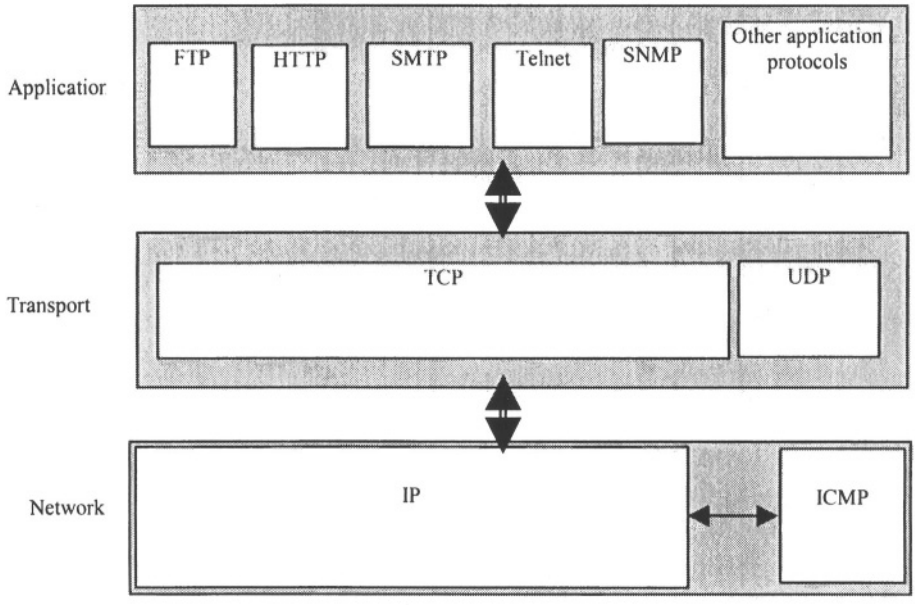


Figure2- 8. Partial protocol stack for the Internet.

Here is a list of acronyms used in Figure 2-8:
FTP: File Transfer Protocol
HTTP: Hyper Text Transfer Protocol
SMTP: Simple Mail Transfer Protocol
SNMP: Simple Network Management Protocol
TCP: Transmission Control Protocol
UDP: User Datagram Protocol
IP: Internet Protocol
ICMP: Internet Control Message Protocol

Here are some of the observations that can be made immediately from Figure 2-8:

1. Most prominently the lower layers are missing. Even though there are protocols developed just for lower levels of the Internet, e.g. the point-to-point protocol (PPP), IP allows a large number of proprietary and non-proprietary infrastructures to be used underneath itself. A set of protocols, called Address Resolution Protocol (ARP) and

reverse ARP (RARP) is specified to dynamically collect information about the addresses of the lower layers.

2. There are two transport layer protocols. TCP is for a reliable, connection-oriented delivery while UDP is a connectionless protocol for short messages that don't require too much overhead. A late addition to the transport layer is the real-time transport protocol (RTP) that is hard to place in the reference model. RTP uses the services of UDP and provides some additional transport layer services. With the introduction of real-time communications over the Internet, the protocol architecture is going to evolve even more and introduce what are called 'service architectures'.
3. ICMP is at the same level as IP, however as indicated by arrows, the beneficiary of ICMP is IP itself. ICMP is used for diagnostic purpose because IP datagram is unreliable and may get lost without a trace. There are new developments for resource reservation protocols, such as the reservation protocol (RSVP) for multicast resource reservation. It is not shown here, as it is more of what would be a 'signaling protocol'.

We will not delve into the discussion on each of the boxes labeled in the above 'model'. However, we will say a few words about the TCP/IP protocols.

2.2.1. The Internet Protocol (IP)

The indefinite sharing capability of the Internet is due to the Internet Protocol (IP). It was based on a novel idea proposed by a student at the Massachusetts Institute of Technology (MIT)[2]. IP could be regarded as an alternative to point-to-point networking for a multi-hop network. In point-to-point networks, the concatenation of link layers is set up for multi-hop communications. Once a path is setup, the data has no choice but to use the same path. In case of a link failure or congestion on a link, the data may be lost a new end-to-end path needs to be set up again. To change the path, the whole process of communications must start anew.

IP brought with it the concept of datagram or connectionless switching. In general, the IP datagram leaves a computer without any pre-programmed route. The node with an arriving IP packet calculates the best next link dynamically, in general. The result is that multiple IP datagrams from the same message may arrive out of order at the destination. However,

[2] Now, the much renowned Dr. Leonard Kleinrock whose multiple volume work 'Queueing Systems' is so deep-rooted in network performance literature that some people spell 'queuing' incorrectly.

as seen in an example earlier, this may quicken the packet delivery due to savings in call setup time.

Each IP datagram carries enough information to identify it uniquely within the network. Consequently, IP packets from many senders and to many destinations can share the same links and switches. This capability of IP to mix traffic is perhaps responsible for Internet commerce boom. The user application from a computer entrusts its data packets to the lower layers and then IP makes its way to the other end.

The services provided by the IP are encoded as various fields in the packet header. For example, a four bit field is use to designate the protocol version so that the node that receives the packet knows what to expect in this packet. The latest IP version is 6, called IPv6. This version is not only backward compatible with the existing version 4 (IPv4), but also provides a lot more functions than IPv4.

Other services provided by IP include (i) destination and source addresses (IP addresses) for many kinds of communications; unicast, multicast, loopback, broadcast and so on, (ii) datagram *fragmentation* to break a datagram into smaller packets if necessary, and (iii) header error checksum to provide a check on the header content.

2.2.2. The Transmission Control Protocol (TCP)

The data part of the IP packet is the TCP packet. TCP takes care of the imperfections of IP and protocols below IP. As mentioned earlier, the IP protocol may result in packets arriving out-of-sequence at the destination. This is due to the possibility that each IP packet could be taking different route to destination. TCP provides the capability of re-arranging the packets back in sequence. IP packet may get lost and TCP would request a duplicate copy of the lost packet. Besides, error checking in IP is provided only for the packet header. The TCP provides for the error checking of the complete packet.

If the reliability functions (in-sequence delivery, error checking and congestion control) are embedded in IP, then the IP would become very reliable. However, the data transfer rate will suffer because IP header is processed at every node. The TCP is an end-to-end protocol and therefore is processed only at the sending and destination computers.

TCP is connection-oriented. This means that the protocols on two communicating computers set up a connection before actual data transfer takes place. This is a very desirable property of TCP. However, if the actual data message is short, then connection setting up and error checking adds too much overhead. Besides, if the TCP at the receiving computer finds a packet in error, it requests the TCP at the sending station to retransmit a copy of the packet. This is not a good error recovery mechanism for delay sensitive applications. Due to these two inefficiencies of TCP, the Internet suite of

protocols includes a connectionless protocol at the same level. It's called the user datagram protocol (UDP). UDP is a connectionless protocol and does not require the use of retransmission of packets. However, the capability of error control is included as an option in UDP.

2.2.3. The Application Protocols for the Internet

The ultimate objective of any network architecture is to facilitate communication of user data across the network. The users of computers generate and process data with the help of application programs. For transparent data exchange, all application programs must use common network interfaces and data compiling mechanisms. The application layer protocols are the guidelines used by application programs for this purpose. The Internet suite of protocols has many protocols at this level.

Most commonly known application protocols are the simple mail transfer protocol (SMTP), the file transfer protocol (FTP) and hypertext transfer protocol (HTTP). Most email applications are based on the specifications outlined in SMTP. HTTP is used in web data transfer and FTP for file transfer. These protocols are specialized for transfer of data stored in files on a computer's hard disk. Internet was designed only for such data files. With the popularity of Internet in e-commerce, the need for multimedia has given new impetus to having protocols suitable for live and streaming data. During the 1990s work in this direction has progressed substantially. New transport protocol for real-time data (that is, RTP) is a new breed of Internet protocols that are part transport and part application level. The introduction of RTP has not entirely resolved the QoS issue with a mixture of real-time and stored data, such as live video conferencing. The Internet Society is aptly following on the demand of various services to be provided at the application level. The success of these services will have a major impact on the way the Internet is viewed by the users. In future, users will be able to choose from a set of services provided by the Internet Service Providers (ISP). The internal working of IP may have to be modified greatly as well. Consequently, the future Internet is bound to be a mixture of connectionless and connection-oriented data delivery.

2.2.4. Lower Layers of the Internet

The success of the Internet has, at least in part, been a separate development in the area of networking – the origination of Local Area Networks (LANs), especially the Ethernet. Not only a LAN provides an excellent way of sharing hardware and software, it has also provided high-speed access mechanism for the TCP/IP networks. The Internet and Ethernet are so closely related that Ethernet is a de facto business standard for the lower layers of TCP/IP. However, one must not forget that Internet was

invented before the Ethernet. Ethernet was developed by a company (Xerox) as a LAN only. The marriage took place at a later time and has been the much unstated but significant event.
LANs provide the access architecture for IP layer and above. However, they are usually used for a group of computers that share many hardware and software resources locally. For a single terminal, with a dial-up line there is the point-to-point protocol (PPP). PPP is advancement in an earlier dial-up line protocol for the Internet called serial line Internet protocol (SLIP). PPP is implemented in home computers that use a point-to-point dial-up connection with an Internet server, such as a computer by an Internet Service Provider (ISP).
When we study LANs, we essentially study what would be equivalent to the two lower layers of TCP/IP suite. We will have a look at the architecture of a local area network in the next section. We will restrict ourselves to the reference model and highlight those parts of the model that define the local area network separated from other networks. We have chosen the Wireless Local Area Network (WLAN) for this purpose as recommended by the IEEE802.11 committee.

2.3. The IEEE Wireless Local Area Network (IEEE WLAN)

2.3.1. Local Area Networks

LANs are an integral part of business computing environments these days. A LAN provides not only an access infrastructure for Wide Area Networks (WAN), such as the Internet; it is also the best way of sharing hardware and software resources of a business. A LAN is much like a connectionless data Public Branch Exchange (PBX) with one or more connections to the (outside) WAN and a high-speed shared transmission mechanism for internal communication. LANs are implemented in a variety of configurations. One rather popular layout is using a central hub with multiple ports on it. These ports are data sockets. A wire from a network interface card (NIC) in a computer is extended to the hub and one of the standard connectors is used to insert the wire in the hub port. RJ45 is a most commonly used connector for LANs. Two or more hubs are interconnected though another hub or through a cable that runs as a backbone network. Figure *2-9* shows a LAN connecting four computers via a single hub.

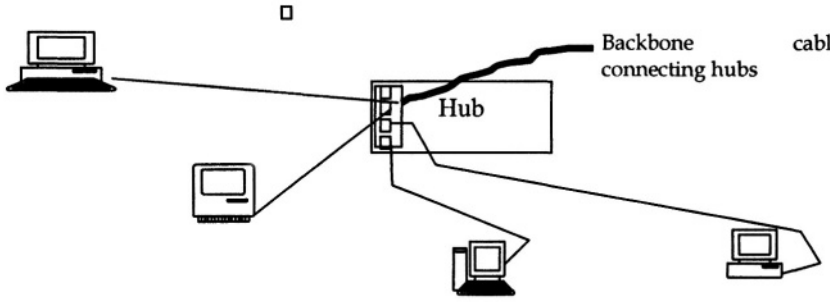


Figure 2-9. A LAN connection

2.3.2. Wireless Local Area Networks

A wireless LAN (WLAN) is much like a wired LAN except that it employs little or no wiring. An *access point* replaces the hub. All wireless terminals communicate with each other via the access point. Sometimes, we distinguish between two types of WLANs based on the use of an access point. A WLAN with an access point is called an *infrastructure network.* A WLAN without an access point is called *independent network* or *ad hoc network.* In an independent WLAN, the terminals communicate directly with each other without the access point.

There are many WLAN architectures used worldwide, some are proprietary and others are industry standards. Examples of standards are the HIPERLAN proposed by European Telecommunications Standards Institute (ETSI) and IEEE802.11 recommended by the 802 committee of the Institute of Electrical and Electronic Engineers (IEEE802). In this third example of network architectures, we will have a look at the reference model of the IEEE802.11. Figure 2-10 shows a general schematic of an infrastructure WLAN and Figure 2-11 shows the protocol architecture of the IEEE802.11 WLAN.

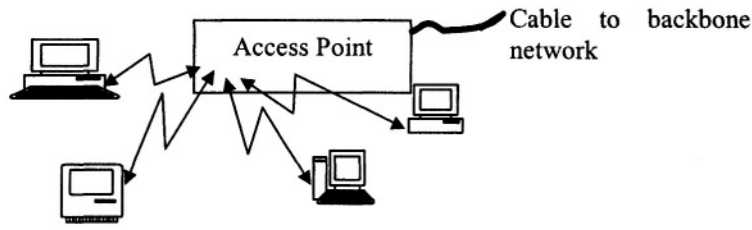


Figure 2-10. Infrastructure WLAN using a single access point.

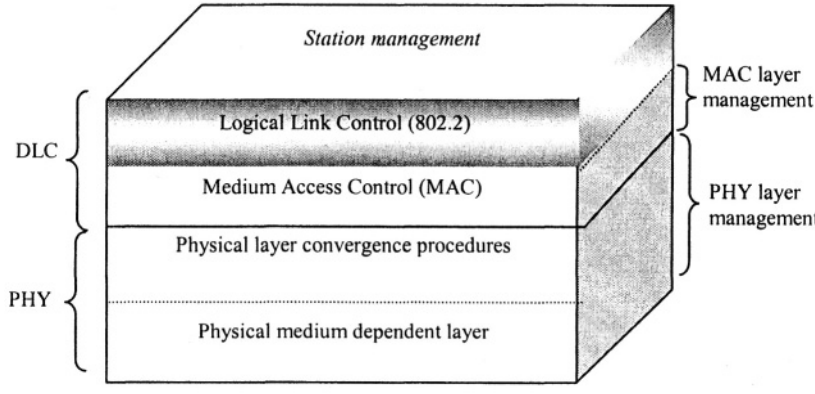


Figure 2-11. Protocol architecture for IEEE WLAN.

We mentioned towards the end of section 1 of this chapter that the OSI-RM did not include network and layer management as part of the reference model. The IEEE WLAN (and some other models after the OSI-RM) took care of this part as well for more comprehensive protocol architectures. The WLAN reference model is a three dimensional model. The front plane of the reference 'cube' defines the user data protocol layers. In the IEEE802.11 documents three sublayers are defined and specified. The MAC sublayer is a part of what is called the data link control layer in OSI terminology. The other two sublayers, namely, physical medium dependent (PMD) sublayer and physical layer convergence procedures (PLCP), constitute equivalent to the physical layer (PHY) of the OSI model. The other part of DLC, that is, the logical link control (LLC), is defined in a separate document and has a different designation (IEEE802.2). LLC is independent of the medium (air or wires) and provides interoperability between stations on the fixed or wireless LANs. Its use in both fixed and wireless LANs also saves protocol processing in the access points. Access points do not have to implement LLC and can act simply as relay stations for MAC PDUs.

The side and the top planes of the reference model take care of management functions. The sublayer management functions are specific to the protocols defined on these layers. For example, the MAC sublayer management power management and access control management. The PHY layer management function is used to define a management information base (MIB) for adapting to various link conditions. Sometimes the PHY and MAC management layers need to share information. This is taken care by the station management plane.

With this brief introduction, we will look at the layer definitions of the IEEE WLAN next.

2.3.3. The Physical Layer (PHY)

The physical layer is divided into two sublayers, the physical medium dependent (PMD) sublayer and the physical layer convergence procedures (PLCP). The PMD defines signal characteristics for various options of medium use. The medium is the frequency band and options include modulation, coding and ways of distributing bandwidth among different wireless terminals. The PMD defines the infrared signal and spreading mechanisms for the spread spectrum signals. The PLCP defines frame types for physical transmission, data rates and header error check etc.

There are two frequency bands available for use. The unlicensed ISM band allocated for research and development of wireless networks in Industrial, Scientific and Medicine (ISM) fields. Most typical ISM band, with about 80 MHz bandwidth, is allocated around the 2.4GHz frequency range. The second frequency band is the Infrared (IR) light. IR band is located just

below the visible light and provides an affordable and high bandwidth medium for communications at short distances.

2.3.3.1. Spread Spectrum Communications

Due to the unlicensed nature of the frequency bands in WLANs, there is the possibility of interference from other users with equipment using the same frequency spectrum. Therefore, data signal is modulated using special interference suppression techniques. Spread spectrum (SS) is one such scheme. In SS systems, all users share the bandwidth such that uniqueness is maintained for each user signal. In *frequency hop spread spectrum* (FH-SS), the system bandwidth is divided into many narrowband channels. The uniqueness of each user signal is maintained by rapidly changing user frequency channels - the changing pattern being unique for each user. This pattern of hopping among channels (or code) is allocated to each user. The code is known only to the receiver and transmitter pair, thus excluding all other stations from the knowledge of hopping pattern. In direct sequence spread spectrum (DSSS), each user bit is transmitted as a combination of smaller bits, called *chips,* thus spreading the narrow band bit stream to a wider band chip stream. Again, the uniqueness of spreading mechanism helps the actual receiver to be able to receive the signal.

2.3.4. The Medium Access Control (MAC) Layer

The medium access control (MAC) sublayer is the most important layer in any LAN. It is the differentiating layer among various LAN types. The MAC layer for IEEE 802.11 has been defined keeping several things in view, such as,

(i) The wireless medium is potentially different from wires in that the signal strength is lost quickly in the air.

(ii) There is an increasing demand for multimedia data communications.

(iii) Power conservation is necessary in wireless terminals.

The MAC layer has been defined for power conserving, high throughput multimedia wireless access. The channel access protocol specified in the IEEE802.11 document defines random channel access with priorities for delay bound and expedited data packets. An example of delay-bound traffic is the voice traffic. Example of expedited data packet is a small packet sent by the receiver of data packet to the sender acknowledging the receipt of the data packet.

2.3.4.1. Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)

The channel access mechanism requires that each wireless station with a packet to send must sense the channel and make sure that the channel is idle for a specified amount of time. This channel sensing mechanism is called as *carrier sense multiple access with collision avoidance (*CSMA/CA). The time that the station must wait before transmission is called the *interframe spacing* (IFS). Several IFS types have been defined in order to provide relative priority to different terminal and data types. The part of MAC implemented in every wireless station is called the *distributed coordination function* (DCF) and the part to be implemented only in access points is called as the *point coordination function* (PCF).

2.4. Framework for Studying a Protocol

Before closing this chapter, let's look at the possibility of a common framework in which we can project all or at least most protocols. If we are successful, we can structure the study of protocols by simply fitting it in the framework. Here is one possible approach.

All protocols provide some substantial service that determines its equivalence to one or more OSI layers. So, the very first thing we look for could be where a certain protocol (or suite of protocols) fits in relation to the OSI-RM. Alternatively, if we have that knowledge, it gives us a hint as to what should be our expectations. For examples, if a protocol provides logical functions for data exchanged between two directly connected stations; we know that it is a DLC protocol. Alternatively, if we know that a protocol is a DLC protocol, we know what type of functions to expect from it. Secondly, each protocol performs a set of functions. Each function may be performed by a (set of) procedure(s). Therefore, the second thing that we may want to look in the study of a protocol is what functions it provides and how. Usually, these functions are obvious from the protocol header/trailer. The procedure in order to implement each of these functions is also important to know. For example, there are several ways of providing a flow control mechanism. If it is known that a protocol provides flow control, this is not enough information in fully understanding the protocol. The third and important characteristic of any protocol is its performance. Given the layer information and set of functions that it can perform, how well does it actually perform these functions? This is perhaps the most important part of protocol study, and sometime most neglected. It must be noted that many developments in protocols and new standards are required just because of this reason – the existing ones can't perform as well. However, the performance study of a protocol is admittedly a complex subject and should be studied at a higher level.

In summary, the following knowledge can help one fully understand and even design of a protocol.

1. Its level in the OSI network architecture in order to understand the main functions of the protocol.
2. Functions and procedures provided by the protocol.
3. Performance bottlenecks and efficient implementations.

2.5. Standardization of Protocols

The developments in data communications seem to be headed towards the evolution of a global network requiring same protocols no matter where we are and with any computing equipment. However, the innovations and competition have a natural trend of obstructing this goal. Due to this reason, standardization process has seen more and more co-ordination internationally. In spite of all efforts of coordination, many organizations will have to continue their work in a particular direction due to their peculiar needs and circumstances. These needs and circumstances have something to do with factors such as:

1. National economy of a country is booming and it decides to go ahead with the next generation of networks. An example of this situation is the Republic of Korea becoming a member of the Organization of the Economically Developed Countries (OECD) in the 90's and spearheading for the latest developments in wireless networking.
2. Sometimes, special interest groups (SIG) of vendors and users would like to spur the process by making their own forums. This is a very common trend these days. An example is the ATM Forum that led the developments in ATM technology ahead of already established organizations.
3. There are different needs of security and level of trading relations of all nations. For example, encryption algorithms, developed in one country may not always be available in some other countries.
4. Then, there is a different infrastructure in every country. Some countries have optical fiber networks, other emphasize on wireless technologies.
5. Sometimes, a nation lacking a certain infrastructure would like to bypass some technologies and go directly to a later stage. An example is in going wireless. Many nations find it difficult to build a wired network infrastructure, resulting in an emphasis on standardization in wireless communication.

As a result of the above and perhaps other reasons, the standardization process will remain a heterogeneous one and the question of interoperability will always be there. The following organizations are at the forefronts of providing standards on national and international levels. There are numerous national, vendors and consumers organizations that provide input and/or make their own standards.

1. International Telecommunications Union (ITU)
2. Internet Society

3. International Organization for Standardization (ISO)
4. European Telecommunications Standards Institute (ETSI)
5. American National Standard Institute (ANSI)
6. Institute of Electrical and Electronics Engineers (IEEE)

2.5.1. International Telecommunications Union (ITU)

The ITU's standards are perhaps most widely used in the global telecommunications industry. It's organized into sectors, committees and sub-committees. For telecommunications and switching, it has the ITU-T sector, formerly known as Consultative Committee for International Telegraph and Telephone (CCITT). Formed in 1865, ITU has the UNO member countries as its members. Traditionally, standardization process at the ITU process has been slow, coming up with revisions every four years after going through several stages of confirmation. At the end of the four-year period a series of standards and recommendations used to be published. This multi-volume series was known by the color of its binding (such as green book or blue book). But the administration sensed the unnecessary delay in the process and overhauled it in early 90s. The procedure is still long but the restriction of waiting till the end of four years period is not there any more. Most of the European and Asian countries follow ITU standards in the telecommunications infrastructure network. In USA, AT&T laid down most of the infrastructure resulting in a hierarchy of network switching and transmission infrastructure slightly different from ITU recommendations.

2.5.2. The Internet Society

Like Internet itself, the Internet Society is an evolution of many groups working towards the development of the Internet. With the worldwide explosion of TCP/IP networking, there is more and more organizing in the way protocols and standards are developed for Internet. Specifications for Internet are available for free of cost at all stages of development. Any one can participate in the protocol development. Thus, a large number of universities, businesses and even users participate in Internet Society in many ways. However, this should not be taken as a lack of an organized way of standard development and approval. There are many groups under the Society that are responsible for specific tasks. Some of the key groups are as follows:

2.5.2.1. Internet Architecture Board (IAB)

IAB sets up the future targets for Internet protocols and oversees all the developments. It makes critical decisions that impact the future of Internet.

2.5.2.2. Internet Engineering Steering Group (IESG)

IESG oversees the standardization development process. It sets milestones and performs technical management of protocols.

2.5.2.3. Internet Engineering Task Force (IETF)

IETF is responsible for the actual development of standards for the Internet. Task force interacts with the steering group on one hand and with the scientists and engineers on the other hand. It is responsible for getting initial proposals, critique and suggesting whether a document is ready to be announced as a standard.
There are four stages of protocol development, Internet Draft, Request For Comment (RFC), Draft Standard and Internet Standard.

2.5.3. International Organization for Standardization (ISO)

The ISO is an international organization for the development of standards for all types of goods and equipment. Established soon after the World War II, ISO has either governments or public utility agencies as members. From USA, the representative member is ANSI, the American National Standard Institute. ISO performs works on joint projects for standards in electrical and computer engineering with another organization, International Electrotechnical Commission (IEC). IEC and ISO sometime make joint groups called Joint Technical Committee (JTC). Each JTC is entrusted with a specific job for standard development. The standardization process of ISO/JTC is quite well defined and goes through six stages from proposal to publication.

2.5.4. European Telecommunications Standards Institute (ETSI)

ETSI has been formed by a number of operators, manufacturers, users and administrative companies in Europe. The main objective of ETSI is to make standards that could be used all across Europe and beyond. It has several hundred members from over 50 countries.

2.5.5. American National Standard Institute (ANSI)

ANSI is a US-based private, not-for-profit organization that acts a facilitator for industry standards. It provides a two-way pipe for importing and exporting standards to and from the USA. ANSI represents USA in some International organizations for this purpose. An example of facilitating standardization is the ANSI accreditation of the Telecommunications Industries Association (TIA). TIA is a group of private industry that makes standards together with Electronic Industry Association (EIA) and promotes

these standards through International events. The standards developed by the joint efforts of TIA and EIA are prefixed as TIA/EIA.

2.5.6. Institute of Electrical and Electronic Engineers (IEEE)

Started as an organization of electrical and electronics engineers for sharing the respective research, development and academic activities, IEEE has advanced in great deal in its form and objectives. Constituted of 'individuals' as against industry or countries, the IEEE has its standardization making its mark on the International scene. The standards by committee 802 have long been popular in networking community. The IEEE802.3 specifications for the Medium Access Control (MAC) sublayer of the so-called Ethernet is perhaps the most recognized LAN standard worldwide. Since the late 90s, the specification of wireless LAN (IEEE802.11 series) has gathered an acclaim reminiscent of the Ethernet. IEEE makes standards in many areas of networking and software development.

2.6. Summary

In this chapter, we looked at some examples found quite often in the study of layered models of network protocol architectures. The OSI-RM is a well-defined and well-designed reference model that breaks up the communications task into seven layers. Each layer performs a distinct set of functions requested by the layer above. The physical layer provides the network interface to the communicating stations. The application layer provides protocols for application programs to exchange user data. All other layers interface a layer above and a layer below through service access points (SAP)s. Adjacent layers exchange primitives to request, indicate, respond to and confirm the provision of a service. In different computers, layers can communicate with their peers only. They do so by attaching extra information to the received data, called header, trailer or sometimes protocol control information (PCI).

The second example we saw was the Internet suite of protocols, popularly known as the TCP/IP protocol suite. Even though there is no reference model for this suite of protocols, there are many ways used to stack the protocols together. Due to its popularity, the Internet suite of protocols is evolving quite rapidly adding new protocols.

In the end, we had a brief account of a local area network protocol architecture, the IEEE WLAN. This three dimensional architecture defines a protocol plane and two management planes, one for each protocol layer and another for coordination between the layers.

2.7. Review Questions

1: How can we compare the telephone network with OSI-RM?
2: Distinguish between the following: (a) Primitives, (b) Protocol header.
3: What is (a) good about (b) not so good about having too many or too less layers in a protocol architecture?
4: If in an OSI network it is allowed that any layer can communicate with any other layer, what is (a) advantage (b) disadvantage over the current reference model?
5: What are the equivalent of SAPs in the TCP/IP suite of protocols?
6: What is the main difference between layer 2 and layer 4 of the OSI-RM?
7: What are the main ingredients of a protocol framework?

This page intentionally left blank

3. Network and User Data

In the description of network protocol architectures, we have made liberal use of the term data. Alternative terms, such as, signals, codes, pictures, etc. throw some light on the diversity of the types of data. In this chapter, we will have a comprehensive look at this term. Like many other scientific terms, there is no all-inclusive definition of data. Luckily, there is no such need. The meaning of data is generally obvious from the context. However, it has different meanings depending upon the scope. For example, when we talk of user data, we imply the information content that the user will be interested in. On the other hand, network data will be something that a network is entrusted to transport, store or process. The differentiation among various data types does not stop here; there are many forms of user data. The same is true about the network data. In fact, one of the terms used for a PDU in Chapter 1 was the layer data. The application layer PDU was also called as the application data. The use of layer data to imply PDU may sound a bit different from many other texts. We will describe this difference next in section 1. In this section, we will also say a few words about the layer data in general. Following this discussion, we will go on to the main topic of this chapter that deals with data at the application and the physical layers. The reason for importance of data at these layers will be explained towards the end of section 1.

3.1. The Network Data

A PDU has been described as having two parts, the data part, that is the higher layer PDU, and the protocol control information part in the form of a header/trailer. For example, the network PDU (N-PDU) consists of the network layer header and the transport layer PDU (T-PDU). We call N-PDU as the network (layer) data while it is also said that the T-PDU is the data part of N-PDU. As long as the reader understands from the context, there will be no confusion. However, following is a justification for calling N-PDU as the network layer data.

As we learnt earlier, layers talk to their peers. For example, the N-PDU transmitted by the network layer of a sending computer will be processed only by the network layer of the next receiving computer. Thus, for the layer 3 of the receiving computer, the N-PDU is the data to be delivered by the network. However, T-PDU is a part of a primitive sent by the transport layer in the same computer asking the network layer to convey and send it to the transport layer of the destination computer. In computer science terminology, T-PDU will be the parameter of a function (the *primitive*) passed to the network layer. It can be called the data part of the N-PDU. However, for the network layer the information that is important comes from another network layer in the form of N-PDU. Thus, we differentiate between the network layer data (N-PDU) and the data part of N-PDU, which is T-PDU unless otherwise specified.

Data is processed by all layers in ways defined by the respective protocols on those layers. It has the same format on all layers except the PHY: consisting of the header and the higher layer PDU as shown in Figure 3-1. Data in this format is also called a packet. *A packet is a protocol data unit.* There are various other terms used to describe a packet, such as, datagram,

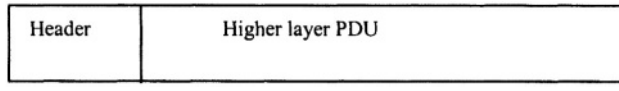


Figure 3-1. Typical network data format

frame, segment, fragment and cell.

There are two exceptions to Figure 3-1 that need to be mentioned. These are in the application layer and the physical layer data. The application layer does not have another layer above it. It gets data from the user or user application software. Since user can have data in various forms, the journey of data from user to the application can take many forms. Another exception is the physical layer data, which is transmitted without being encapsulated by a lower layer. The PHY sends data 'down the wire'. So, for the receiving PHY,

the incoming data first consists of the signal on the wire, and after the signal has been received, it may consist of a data packet with a header and a DLC packet in it. Also, before the data reaches from user to the application layer, it may have to go through various processing functions depending on following factors:

(i) In what form the data was generated?

(ii) In what application program it will be used by the receiving computer?

(iii) In what format is it stored?

Before data reaches the destination PHY, it may have to go through several processing functions depending upon factors such as:

(i) What type of medium is used for transmission, e.g., copper cable, optical fiber or air?

(ii) What distance it traveled? Usually data takes different forms (of modulation) depending upon distance.

(iii) In what format it was coded?

In the next sections, we will have a discussion on various forms of data when it exists before entering the network (application) or before the destination physical layer receives it. We will start by a natural sequence of events to put things in perspective.

3.2. The Physical Layer Data

3.2.1. Sequence of Events and Definitions

As the information is exchanged between two computers on a network, the data passes through several stages.

Data is any piece of information, *analog* or *digital.* Though a plural (of datum) in form, it is extensively used as a singular, such as in this book.

Analog Data consists of information that may assume values from a continuum. There is not a unique way of being analog. An analog data does not have to have infinite number of data points. It could simply be a finite number of data values that are chosen from an indefinite set of possibilities. Examples of analog data are temperature, speech signal and air pressure.

Digital Data consists of information whose values are chosen from a finite number of values. If the data vary as a function of time then time may advance either continuously or in steps. Examples of digital data are number of students enrolling every year and stock market shares sold during a certain day. In the former example, time varies in steps (of one year) while in the later the time varies continuously (throughout the day).

Modern computers and networks process data only in digital form. If the user data is in analog form then it must be converted into an equivalent digital form before a computer can accept it. The process of converting analog

data into digital form is called analog to digital (A/D) conversion. The reverse process is called D/A conversion - read as D to A conversion.

The network transmission links have the ultimate responsibility of taking data from one point to another. Transmission occurs in the form of signals traveling from one point to another. A signal is a representation of data as a function of time (or other dependent variables[3]).

A **Signal** may be defined as a graphical, functional, tabular, electromagnetic or optical realization of data.

Signaling is the analog or digital transmission of (usually electrical or optical) signals.

We know from a theorem of mathematics that each signal can be represented as a weighted sum *of sinusoidal* functions (sine, cosine, etc.).

Sinusoidal functions are single frequency functions with three attributes as shown in Figure 3-2, namely, the amplitude, the frequency and the phase. The mathematical form of a sinusoidal function (called the sine function) *x(t)* is given as:

$$x(t) = A \sin(2\pi ft + \theta) \qquad (3\text{-}1)$$

Whereas,

A is called the amplitude of the signal, defined as the maximum value of the signal,

f is the frequency of the signal, defined as the number of cycles (or wavelengths) per second, and

θ is called the phase of the signal, defined as the angle of the sinusoid when $t = 0$. Figure 3-2 shows the origination of the sinusoidal functions.

A phasor is a line drawn from the origin at an angle θ from the horizontal axis. Suppose that the length of the phasor is *A*. If we draw a line from the tip of the phasor perpendicular on the *x-axis* (shown in dotted), we can draw an arrowhead at the point where this line meets the *x-axis.* The length of the *x-axis* segment from the origin to the point where the dotted line meets the axis is equal to $A\cos\theta$. Similarly $A\sin\theta$ is a segment on the vertical axis as shown. $A\sin(\theta)$ and $A\cos(\theta)$ are the values of the sinusoidal functions when the phasor is stationary (or Time $t = 0$). If the phasor starts rotating in a circle (suppose anticlockwise), the values of the sinusoidal functions vary. For example, when the phasor has moved to a point such that it makes an angle of 90° with the *x-axis,* then $A\cos(90°) = 0$ and $A\sin(90°) = A$. In this way, the values of the sinusoidal functions fluctuate between $-A$ and $+A$ as the phasor rotates. The pattern of values is repeated after every complete rotation. Figure

---

[3] Even though we assume that data varies as a function of time; time is not always or the only variable of data. Other variables such as length, share types (in stock market), surface area, etc., can also be used to replace time in the definition of data.

3-3(a) and (b) show the general shape of the two sinusoidal functions when the rotation speed is 1 cycle per second.

Hertz is the unit of frequency of rotation (called simply the frequency). One complete rotation of the phasor in one second is called one Hertz (one cycle per second), usually abbreviated as Hz. for example, if a phasor makes 1000000 rotations per second, its frequency is 1000000 Hz or 1000 kHz, or 1 MHz. the prefix k stands for thousands or 'kilo' and M stands for million or 'mega'. So, 1000 kHz is 1000 kilo Hertz and 1 MHz is one mega Hertz.

If we plot the sine and cosine functions separately, we get graphs similar to ones shown in Figure 3-3 (a) and (b).

The graphs plotted in Figure 3-3 show $\sin(2\pi ft)$ and $\cos(2\pi ft)$ with f = 1 Hz and $\theta = 0^o$. It is obvious from this figure that if we shift the sine function one-fourth of a rotation to the left (equal to subtracting 90° from its angle), we get the cosine function. In other words,

$$\sin(2\pi ft + \theta - 90^o) = \cos(2\pi ft + \theta) \qquad (3\text{-}2)$$

Another observation that can be made from Figure 3-3 is the following:

If we increase f or the frequency of a sinusoid, there will be an increased number of complete rotations in one-second plot, thus this will be like compressing the waveform. Similarly, decreasing f is like expanding the waveform.

The knowledge of sinusoidal functions is very useful in the study of data and data communications because we use mathematical functions to represent data. According to a theorem of mathematics, any function can be represented as sum of sinusoidal functions. We take two simple examples to elaborate this point: one of a repeated function and the other of a function bound in time.

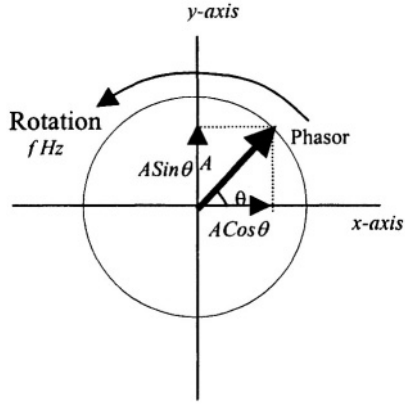


Figure 3-2. The Figure shows various terms used to describe sinusoidal functions (sine and cosine).
Phasor is an imaginary vector whose length is the *amplitude* of the sinusoids and its angular position with respect to x-axis is the *phase* of the sinusoids. If the phasor is rotating anticlockwise, the number of complete rotations (cycles) in one second is the frequency of the sinusoids.
The sine function is the component of the phasor on y-axis.
The cosine function is the component of the phasor on the x-axis.

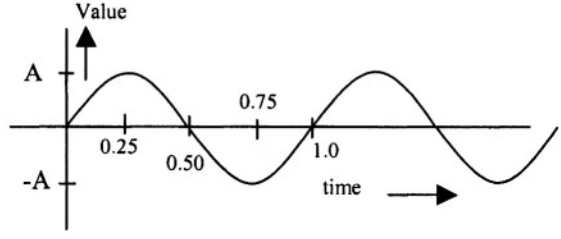


Figure 3-3(a) sine($2\pi ft$) with $f = 1$ Hz

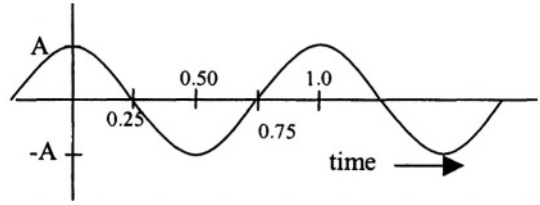


Figure 3-3(b) $\cos(2\pi ft)$ with $f = 1$ Hz

Example 3-1: The square-wave function

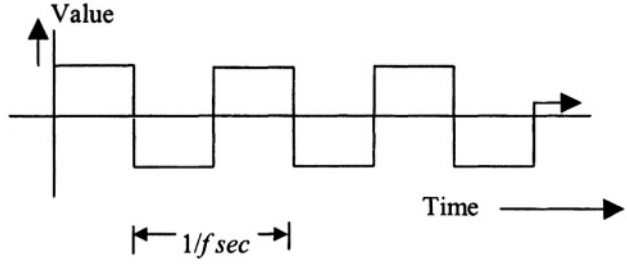


Figure 3-4(a): The square wave function. *1/f* cycle time equals to a cycle repetition or *frequency* equal to *f* Hz

The square-wave function is shown in Figure 3-4(a). It is easily proven that such a square-wave function is equal to weighted sum of sine functions of all odd frequencies, that is, *f*, *3f*, *5f*, and so on. In other words, if SQ(*f*,t) is the square wave function with frequency *f* and amplitude of unity, then it can be written in the following form.

$$SQ(f,t) = \sin(2\pi f) + 1/3 \sin(2\pi.3ft) + 1/5 \sin(2\pi.5ft) + 1/7 \sin(2\pi.7ft) + \quad \ldots \quad (3\text{-}3)$$

Figure 3-5(a) shows plots of the above sum with increasing number of sinusoidal functions used for each graph. It demonstrates the point that as the number of terms increases; the sum closes in to the square wave function.

Example 3-2: A gate function

The gate function is quite different from the square wave function discussed

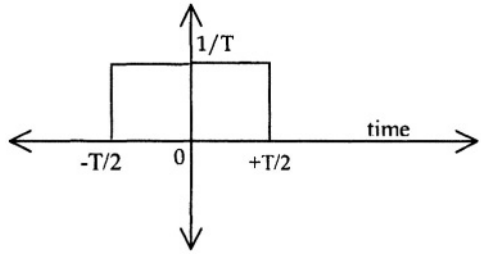


Figure 3-4(b). A gate function of width T and height 1/T centered at the origin. In general, the width, height and center point all can be different.

above in that it is time limited. In other words, the function exits only within (-T/2,T/2) and its value is zero outside this interval of time as shown in Figure 3-4(b). It turns out that this simple function, too, have an infinite number of sinusoids in it, with frequencies ranging from $-\infty$ to ∞. The frequencies of the adjacent sinusoids in a gate function are so close to each other that it is difficult to construct a figure like Figure 3-5(a) for this.

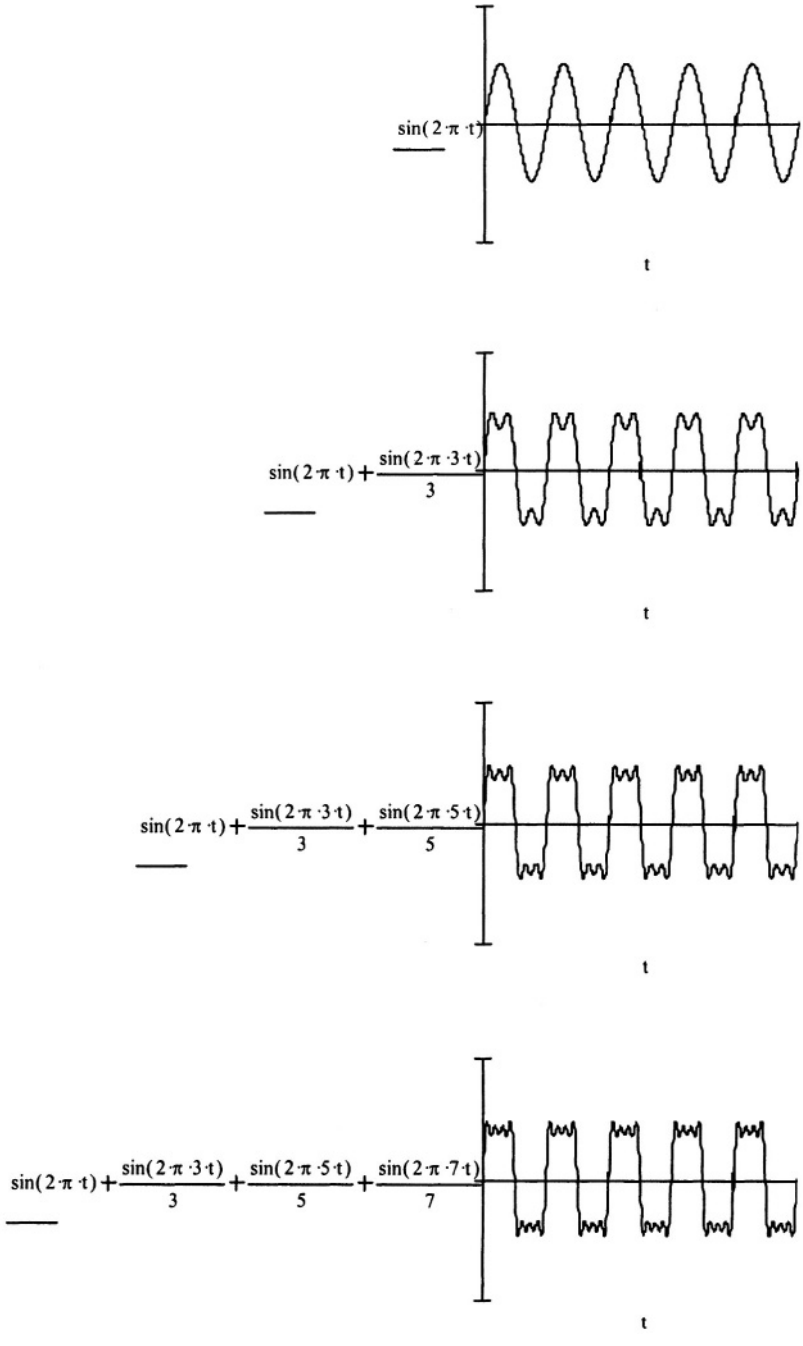


Figure 3-5(a). A demonstration of the representation of a square wave function as sum of sinusoidal functions

For the square wave function, the frequencies of neighboring sinusoids were apart by 2*f*. In the case of time limited functions, however, we can represent the signal in frequency domain easily as is shown in Figure 3-5(b). The frequency domain representation of a signal is a profile of the sinusoids whose sum if equal to the signal. This profile is called as the *frequency spectrum* of the signal.

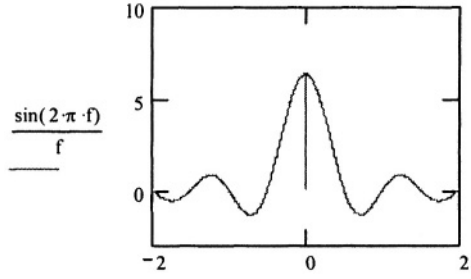


Figure 3-5(b). The frequency profile of the sinusoids in example 3-2.

Fourier Analysis

The Fourier Analysis is a branch of mathematics that helps scientists and engineers to determine what frequency components a signal is made of. In other words, it is a signal analysis technique. Fourier series is the sum of sinusoids that constitute a periodic function - like the one in example 1 above. A periodic function consists of periodic repetitions of part of it. The smallest time of repetition is called the period of the function. For non-periodic functions, Fourier transform helps us determine the profile of the sinusoids (called the frequency spectrum) that constitute a function. Usually the aperiodic nature of such functions results in an infinite number of sinusoids between any two frequencies values, thus making it difficult or impossible to have a simple (finite or infinite) sum of sinusoids representing a function. The example of a non-periodic or aperiodic function is the gate function.

Signal bandwidth is the frequency range of significant sinusoidal functions in a signal. Roughly, it is difference between frequencies of the highest frequency and lowest frequency (significant) sinusoid of a signal.

As a signal propagates through a medium it deteriorates due to equipment imperfections, channel attenuation and other natural factors. Different frequencies deteriorate with different intensities. Usually, the channel *impairments* are worst at the lowest and highest end of allowable frequency range.

Channel is the term used to describe the physical or logical medium used to transmit signal. Examples of physical channels are: copper cable, glass fiber and free space. Example of logical channel is an identification number that could be mapped into some physical channel.

Channel Bandwidth is the range of frequencies that could be transmitted through a channel without significant impairments. Roughly, it could be said to be the difference between the highest and the lowest frequency sinusoids that can be propagated through a channel without significant impairment.

For successful communication, the channel bandwidth must be greater than or equal to the signal bandwidth.

Digital transmission is the transmission of analog or digital data in the form of digital signals. Digital transmission speed is usually measured in bits per seconds or symbols per second, either one called as *data rate.* The data rate can be either in bits per second (bps) or in bauds (or simply the baud rate).

Baud rate is defined as the number of symbols per second. Data rate in bits per seconds is also called the *bit rate.*

Bit (**bi**nary digi**t**) is a data symbol (physical or logical) that could have one of the two possible values. In general, a data symbol does not have to be a bit. It may represent one or more bits. If a symbol has four possible values, these values can represent four combinations of 2 bits. The following table (Table 3-1) shows one way of mapping the four symbol values to the four two-bit combinations.

Table 3-1 Mapping of symbol values to bit values

Symbol value (arbitrary)	Bit patterns
0	00
1	01
2	10
3	11

The symbol values chosen in Table 3-1 are arbitrary, and so are the equivalent bit patterns. The two do not have to be equal in decimal values. However, it is a common practice to use the decimal equivalence. In general,

if the number of symbol values can be represented as power of 2, then a simple relation can be defined between the number of symbol values and the number of bits needed to represent all the values. Let M be the number of symbol values, and k be the minimum number of bits needed to represent these values. Then $M = 2^k$. Alternatively, $k = log_2 M$. This relationship also holds for the bit rate and baud rate.

$$bit\ rate = baud\ rate \times log_2 M \qquad \ldots (3\text{-}4)$$

$log_2(.)$ is the logarithm to the base 2.

There is a direct relation between the bandwidth of a channel and the maximum data rate that it allows. In general, the greater the bandwidth, the greater will be the data rate. For exact relation between a signal bandwidth and its transmission bit rate, we have to know how the signal has been processed, coded, modulated and what type of equipment is used for its transmission or reception.

Channel capacity of a communications channel is the maximum data rate that can be obtained from that channel. For channel with only one type of impairment (white noise), the capacity is given by

$$C = B\ log_2(1 + SNR)\ \text{bps} \qquad \ldots (3\text{-}5)$$

The SNR in Equation (3-5) is the signal to noise power ratio, B is the channel bandwidth and $log_2(.)$ is the logarithm to the base 2. Channel capacity is different for different channel types, depending upon the impairments. For example, in a wireless channel where the received signal is the sum of all the reflected components, the channel capacity may be different from Equation (3-5).

Note

The terms frequency and bandwidth do not have the same meanings. Usually, any signal consists of many sinusoidal components. The bandwidth is the difference in the frequencies of the highest frequency and lowest frequency sinusoidal components.

The result of imperfections in the channel bandwidth and the impairments is that the data, once transmitted, loses at least some of its shape and content. Consequently, what is received over a transmission link is not exactly what was transmitted. This is due to several impairments of a

communications channel. These impairments will be discussed in the next chapter. It should suffice to say at this point that due to the imperfections of the transmission medium, data undergoes many transformations before it is ready to be transmitted. The result of these transformations is to represent data in signal form that could be reliably carried over the medium. A general term for this transformation is the *modulation* of data into transmission signals.

3.2.2. Modulation of data and signals

Modulation can be defined as transformation of data or signals into a form better suited to transmission. The data (or signal) that is transformed is called the *modulating data* (or *signal*). After modulation, the data (or signal) becomes *modulated data* (or *signal*). A simple example would illustrate this concept. Suppose our data consists of human speech to be broadcasted live on a radio station. The data is originally in the form of mechanical energy (speech). The first step in the transmission of this data is to convert it into an electrical form because the network (which is a radio network) takes only electronic signals. For this purpose the speech signal is modulated into an equivalent electrical signal by passing it through a microphone. The modulating signal is the mechanical speech signal, the modulated signal is the electrical speech signal and the microphone is the modulator. However, the bandwidth of the electrical speech is such that it cannot be transmitted through air because it gets attenuated quickly. It would be possible to transmit the signal if its frequency bandwidth were transformed to a much higher value. For this purpose the electrical speech signal is modulated into an equivalent signal with higher frequency components suitable for transmission in the air. An amplitude modulator (AM) or frequency modulator (FM) is used for this purpose. In this case, the electrical speech signal is the modulating signal and the equivalent high frequency AM or FM signal is the modulated signal. The modulator will be a device that converts the low frequency signal into a high frequency signal suitable for transmission. See Figure 3-6 for an illustration.

Figure 3-6(a). Mechanical voice signal

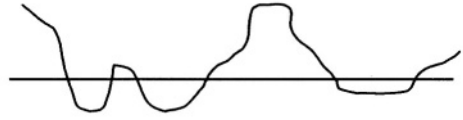


Figure 3-6(b). Equivalent electrical signal after using the microphone.

Figure3- 6(c). Equivalent high frequency electrical signal after amplitude modulation.

This simple example illustrates what is called as analog modulation. For transmission over computer networks, the analog data must be converted into a digital form before it could be modulated. We will discuss this topic under the section on user data. For now, we explain two terms used to differentiate between low and high frequency modulated signals.

3.2.2.1. Baseband and Passband Modulations

If the frequency spectrum of a signal is close to zero Hz, such a signal is called a baseband signal. An example of a baseband signal is the electrical speech signal. If we use Fourier analysis to find the frequency spectrum of such a signal, the spectrum vanishes at a frequency values greater than 4 kHz. We say that the bandwidth of speech signal is 4 kHz (meaning that the

frequencies in it range between 0 and 4000Hz[4]). A data representation in the form of baseband signal is called baseband modulation. Baseband signal usually cannot be transmitted to relatively longer distances.

To transmit a baseband signal to a long distance, it is modulated with a high-frequency signal, with much higher frequency than the highest frequency component of the baseband signal. The resulting modulated signal is called the passband signal. The passband modulation and the baseband modulations exist for analog and digital signals. Example of a digital baseband signal is a voltage pulse similar to the gate function discussed in a section above. Figure 3-7 shows what happens to the spectrum of a baseband signal when it is modulated with high frequency sinusoid - called as a *carrier signal.*

[4] These values are mainly for illustration purpose, the actual spectrum of voice is less than 4 kHz and ranges from 300 Hz to about 3400 Hz. Most of the literature on communications, however, uses the nominal value of 4 kHz.

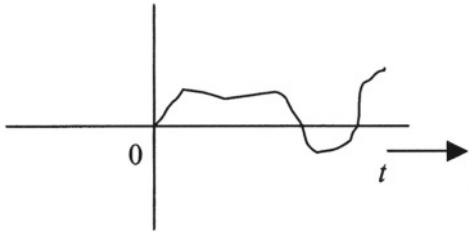


Figure 3-7(a). The baseband signal

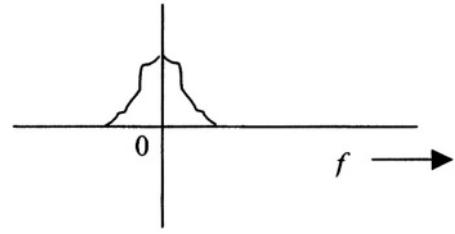


Figure 3-7(b). Spectrum of the baseband signal

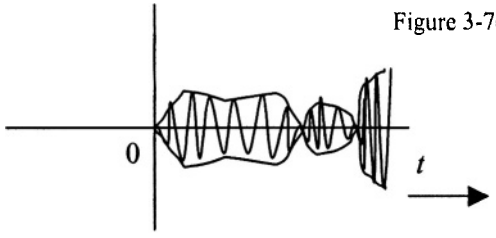


Figure 3-7(c). The passband signal

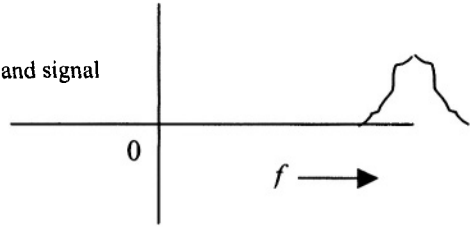


Figure3- 7(d). Spectrum of the passband signal

3.2.3. Digital Encoding of Data

The digital baseband modulation is the mapping of data into digital baseband signals. Since digital data can assume values from a finite set of values, we need an equivalent number of signals to represent such data unit (the data unit for binary data is a bit). For example, if we have binary data

having only two possible values, we need a binary signal having two possible forms. These forms could be in signal values, signal shapes or any other characteristics of the signals. An example of a binary signal is the squarewave function with two values of amplitudes.

Since data in most computers is stored in binary form, the most popular baseband modulation schemes result in the form of binary signals. Since there is no unique way of representing data in signal form, there are many types used and recommended.

Data encoding or **digital encoding of data** are other terms used for digital baseband modulation of data. We will use these terms interchangeably. Many computer networks are short range, such as data acquisition and storage systems and the LANs. For such networks, baseband data encoding is sufficient for transmission and there is no need for modulating the baseband data signal with a high-frequency carrier.

We will discuss three of the most commonly used and discussed encoding schemes in this section. These are:

1. Non-Return-to-Zero (NRZ) Coding
2. Multilevel Coding
3. Manchester Coding

The reason why we have a (large) number of coding schemes is that each one has its advantages over the other in terms of performance with respect to the bandwidth used, robustness to noise and detection imperfections and cost. Depending upon these performance attributes, each could be preferable under a certain set of conditions.

3.2.4. Non-Return to Zero (NRZ)

In this type of data encoding, voltage pulses of constant amplitude and fixed duration are used to encode binary data. One level of voltage, e.g., a positive voltage, may be considered as a '1' and the another level (e.g., negative or no voltage) may be considered as a '0'. In either case, the bit duration time (or the bit time) should be the same for both types of pulses. NRZ can be divided into two categories depending upon how the data are coded and interpreted.

In **NRZ-Level (NRZ-L)**, it is strictly the voltage level that determines the data values.

In **NRZ-Invert-on-one (NRZ-I)** it is not the voltage level that determines whether a pulse represents a '0' or '1', instead, a *transition of voltage at the beginning of the pulse* determines the associated data value. For example, a transition in the beginning may be used to represent a binary '1' while the lack of transition may be interpreted as a binary '0'. See Figure 3-8 for an example.

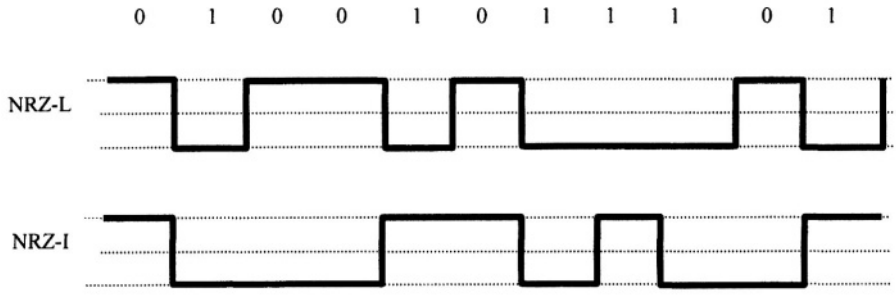


Figure 3-8. Example encoding of binary data using NRZ types

In both cases, the waveforms will appear as constant level voltage pulses. The level remains constant throughout the bit duration. NRZ codes are simple in concept as well as implementation. Depending on the combination of '1's and '0's in a data stream, there may be a strong dc component that could saturate the electronic devices, such as the amplifiers. Therefore their application is usually limited to very short distance communication, such as data storage and recording.

NRZ-I is an improvement over NRZ-L, as it is easier to detect transitions than a level. In fact, if the channel noise adversely affects the whole pulse except the edge, the detection could still be made. In addition, a reversal of the connections between the positive and negative voltage wires will reverse all zeros and ones in NRZ-L but does not affect NRZ-I as the information is imbedded in the transitions which are unaffected by such a reversal. Both of the NRZ coding types have the disadvantage of the potential of running out of clock information given a long string of same bit. The clock information is embedded in the transition points. When a long string of one data type (1s or 0s in NRZ-L and 0s in NRZ-I) occurs, there are no transitions. Thus, if transmitter and receiver lose synchronization of each other's clocks during this long string, they may be unable to communicate properly. NRZ-I is used in the FDDI (fiber distributed data interface) networks.

3.2.5. Multilevel Encoding

In multilevel coding schemes, more than two signal levels are used for binary data. This gives the added benefit of annulling dc bias by employing complementary levels. We will mention two of these, the Bipolar-AMI that is widely used in T-1 and E-1 lines worldwide. The other is the multi-level-3 that is used in CDDI (copper distributed data interface).

3.2.5.1. Bipolar-AMI (Alternative Mark-Inversion) Coding

In Bipolar-AMI encoding scheme, there are three levels used for binary data. A zero level voltage represents a binary '0' while a binary '1' is represented by alternative polarity (negative or positive) pulse. This takes care of one of the problems with NRZ encoding, namely, the dc accumulation. By alternating the voltage polarity (hence, the word 'Bipolar'), each pulse cancels the dc effect of the previous pulse. Sometimes, Bi-polar AMI or its invert form is called pseudo-ternary coding due to the use of three values of pulses representing two values of data. Figure 3-9 shows an example encoding for

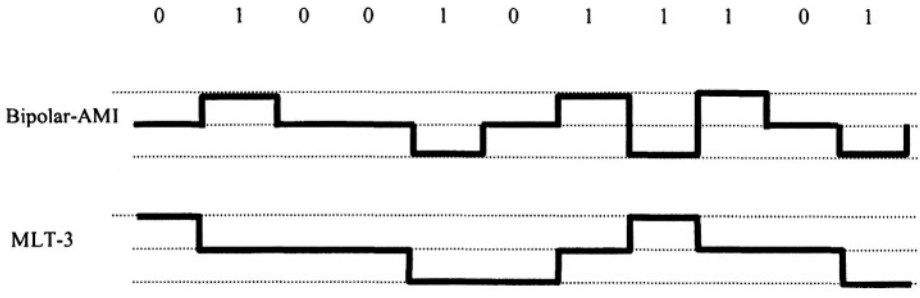


Figure 3-9. Data encoding using Bipolar-AMI and MLT-3

the same data as used above for NRZ encoding. For a long string of 0s, there is no signal transmitted. This has the potential of dragging the transmitter and receiver out of synchronization.

3.2.5.2. Multi-level 3 (MLT-3) Coding

The MLT-3 is a differential data encoding scheme with three levels of transitions. A transition indicates a binary 1 and a lack of transition a binary 0. There are three levels but a transition from high-to-low or vice versa is not allowed. The transitions occur only between one extreme (high/low) and zero voltage as shown in Figure 3-9. MLT-3 is used in 100-Base-TX, the 100 Mbps Ethernet.

3.2.6. Manchester Coding

Manchester coding provides regular clock information within each bit by inserting a mid-bit transition. The direction of transition is used to interpret

whether the intended bit is a '0' or a '1'. A low-to-high transition represents a '1' and a high-to-low transition represents a '0'.

Differential Manchester Coding is the differential version of Manchester coding in which a transition at the beginning of the pulse is also used to code data, instead of the mid-bit transition. A transition in the beginning of a bit represents a '0' while the absence of a transition at the beginning represents a binary '1'. Figure 3-10 shows example encoding for the two Manchester coding types. Manchester coding is employed in the various 10 Mbps Ethernet standards while Token Ring (4 Mbps and 16 Mbps) uses differential Manchester coding.

Differential Coding techniques are a class of coding techniques in which the data is coded in the pulse transitions. NRZ-I, MLT-3 and differential Manchester are examples of such coding types. They are more robust than non-differential techniques. The cost is in the complex circuitry to encode and decode such schemes.

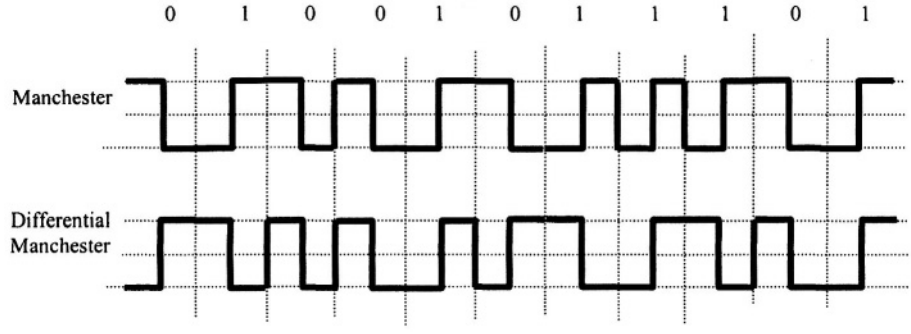


Figure 3-10. Example of Manchester coding

3.2.7. General Characteristics of Bit Encoding

The NRZ and AMI coding schemes require comparable bandwidths. The two could offer problems in locking the transmitter and receiver clocks for long strings of same data values. In Manchester coding schemes mid-bit transition inserts a clock signal within each bit. However, these mid-bit transitions require higher bandwidth. NRZ has another disadvantage of accumulation of dc voltage in the receiver. AMI schemes take care of the dc accumulation by alternating polarity of pulses. There is no dc accumulation in Manchester coding as one half of every pulse is opposite in polarity to the other half.

3.2.8. Zero-substitution and *n*B/*N*B Translation

There are ways to inset signal transitions for clocking information in NRZ and AMI schemes without requiring the bandwidth needs of Manchester coding. These mechanisms increase the complexity of coding device. One of such mechanisms is called zero-substitution in which a long string of '0's is coded such that there is a certain minimum number of pulses in the string. For example, in B8ZS (Bipolar with 8-zero substitution) eight zeros are coded together by inserting pulses in violation of the Bipolar AMI codes. The violation of AMI codes pertains to inserting pulses with the same polarity as the previous pulses occurrence. The violations are inserted in such as way as to avoid dc accumulation. Thus, if the polarity of the last pulse before the string of zeros was positive, then the eight zeros are coded as 0 0 0 + - 0 - +. Therefore, the two 'violations' of the Bipolar AMI code help decode the correct string of zeros. If the last pulse before the occurrence of eight 0's is negative, the polarity of the pulses representing the eight zeros is reversed making it 0 0 0 - + 0 + -. This is mostly used in USA for some long distance communications networks. Other coding schemes using zero-substitution are HDB3 (high density binary coding with 3 zeros), B3ZS and B6ZS. Figure 3-11 shows plots of B8ZS and HDB3 encoding schemes.

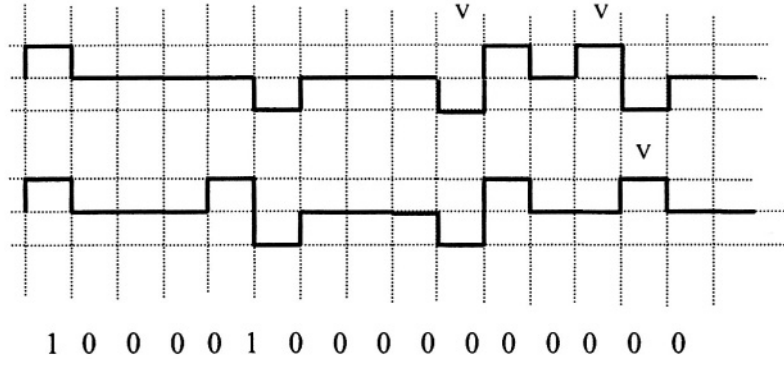


Figure 3-11. Zero substitution B8ZS (above) and HDB3 for a given string of data bits. A 'V' indicates the violation of AMI codes.

The rules for HDB3 are given in Table 3-2 and for B8ZS in the the rules of encoding can be described as follows: If the polarity of the last pulse before the string of zeros is positive, then the eight zeros are coded as 0 0 0 + - 0 - +. Therefore, the two 'violations' of the Bipolar AMI code help decode the correct string of zeros. If the last pulse before the occurrence of eight 0's is negative, the polarity of the pulses representing the eight zeros is reversed making it 0 0 0 - + 0 + -.

Table 3-2: HDB3 encoding.

POLARITY of LAST PULSE	Last Substitution Sequence	
	000+ or +00+	000- or –00-
+	-00-	000+
-	000-	

In another mechanism of coding with clocking information, a table is used, that has two columns. The first column consists of 2^n patterns of n bits. The second column consists of 2^n patterns of N bits. The value of N is chosen to be greater than n, so that the 2^n patterns of N bits have a specified number of both data bits. For example, in 4B/5B translation, all the $2^4 = 16$ patterns of the second column have at least two '1's to guarantee clock information in a string of every 5 bits. The coder looks at the transmitting data 4 bits at a time. Then, using the table, it transmits the 5 bits pattern corresponding the 4 bits to be transmitted. In this way, 20% of the transmitted data constitutes the overhead for inserting clocking information.

3.3. Passband Modulation

The carrier frequency is central to the concept of passband modulation. The job of carrier is to shift the frequency spectrum of a baseband signal using modulation so that it can travel longer distances.

3.3.1. The Carrier Signal

A carrier signal or a 'carrier' is a high frequency signal, sufficiently high to travel on a guided or wireless medium. The way a carrier signal is used is by *modulation of a baseband or lower frequency signal to it.* Thus, when we use a carrier for digital transmission, we usually have two layers of modulation, first the baseband modulation and then the passband modulation.

Carrier can also be used to transmit analog data represented by an analog *baseband signal.* In communication system terminology, the former is simply called as *digital modulation* and the later as *analog modulation.*

3.3.2. Analog Modulation

In analog modulation, both the modulating and the carrier signals are of the analog form. We know from an earlier section in this chapter that a sinusoid (either type of the sinusoids) has three attributes, the amplitude, the frequency and phase. When we generate a sinusoidal signal, we simply rotate a phasor with amplitude *A* anticlockwise for positive values of frequency. A clockwise rotation yields negative frequency. A carrier generated thus is shown in Figure 3-12. It is a sine function with frequency $f = 400$ Hz[5] plotted between 0 to 0.05 seconds showing 20 cycles. The figure also shows the analog data signal *m(t).*

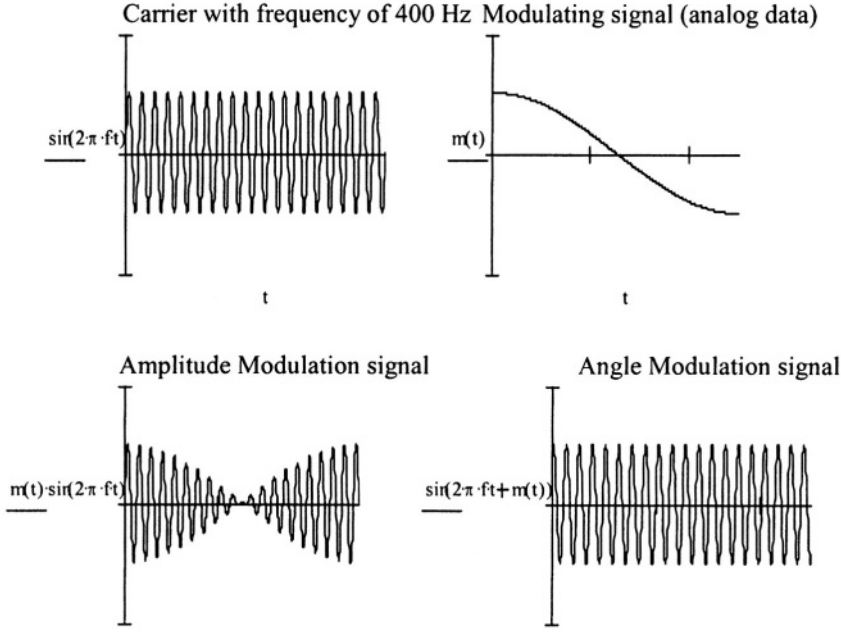


Figure 3-12. The impact of modulation on carrier.

[5] In practice, the carrier frequencies are higher than this value by several orders of magnitude.

The data signal *m(t)* could be modulated with the carrier $sin(2\pi ft)$ in a variety of ways. Each different way of mixing *m(t)* with the carrier signal defines a type of analog modulation. Here are two major types.

3.3.2.1. Amplitude Modulation (AM)

The AM signal is generated by varying the amplitude of the carrier signal in proportion to modulating signal *m(t).* This is easily achieved by multiplying the carrier amplitude with *m(t).* The AM modulator is sometimes called a multiplier or mixer for this reason. As seen in Figure 3-12, the amplitude of the carrier makes an envelope that resembles the modulating signal. The receiver of this signal only needs to detect this envelope. In fact, one type of AM receiver is called an envelope detector. This is the simplest form of amplitude modulation and has its own name - double sideband with suppressed carrier (DSC-SC). The sideband refers to the each half of the frequency spectrum about the origin. Each side carries all the signal information and differs only in the sign of frequencies in them. The side band on the right side of the origin has positive frequencies while the other sideband has negative frequencies. Naturally, one may ask if only one of the sidebands can be used in AM. The answer is yes, as is done in single sideband AM systems. The reason why it is called as suppressed carrier is explained next. The AM signal shown in Figure 3-12 can not be exactly detected by an envelope detector. This is because the envelope detector would detect only the positive values of the signal while the actual signal had both, the positive and negative values. The result of envelope detector is shown in Figure 3-13. The envelope in fact receives the absolute value of the modulating signal.

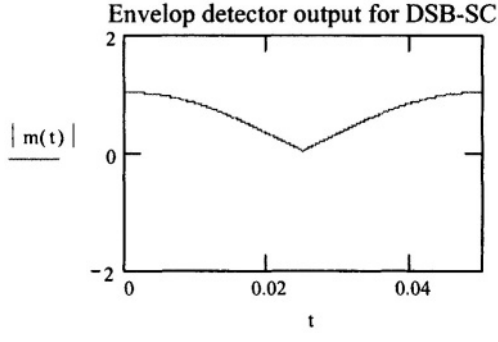


Figure 3-13. The envelope of DSB-SC signal.

One solution to this problem is to add extra carrier with the AM signal and instead of transmitting $m(t)sin(2\pi ft)$, transmit $[A+m(t)]sin(2\pi ft)$. This additional carrier with an amplitude *of A* is called the pilot carrier and it helps the receiver get both positive and negative values of the signal on the positive (or negative) side of the envelope by adding a dc bias of *A* to the envelop. Then by removing the dc bias, the original signal can be recovered.

See Figure 3-14 below for the final result. This is equivalent to receiving $|1+m(t)| - 1 = m(t)$

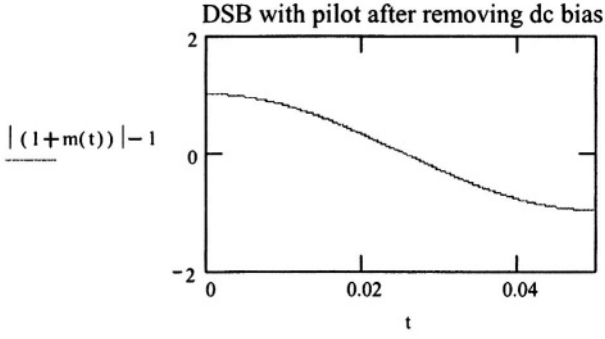


Figure 3-14. DSB with pilot, A = 1.

3.3.2.2. Angle Modulation

A second mechanism of mixing the analog data signal with a high frequency carrier signal is to generate the carrier signal by varying the angle of the phasor according to *m(t)*, the modulating signal. In other words, if the carrier signal was $sin(2\pi ft)$, then the modulated signal will be either $sin[2\pi ft+m(t)]$ or $sin[2\pi ft + \int m(t)dt]$ depending upon whether *m(t)* is used to change the phase or the frequency of the carrier, respectively. Sometimes, we multiply *m(t)* or its integral by a factor $0 \le n_i \le 1$ to vary the extent of modulation. Due to this reason n_i is called the index of modulation.

When the phase of the modulated carrier is $[2\pi ft+m(t)]$, the resulting angle modulation is also called as the *phase modulation.* Alternatively, when the phase of the resulting modulated carrier is $[2\pi ft + \int m(t)dt]$, the modulation is called the *frequency modulation* because the differentiation of $[2\pi ft + \int m(t)dt]$ has the units of radians, the angular frequency. In other words, frequency of the modulated signal in Hz $= (1/2\pi).\ d[2\pi ft + \int m(t)dt]/dt$. Figure 3-12 shows an example of phase modulation.

As noted from Figure 3-12, the amplitude of the carrier is not affected in phase (and frequency) modulation. This is a big advantage of angle modulation over amplitude modulation and results in reduced noise interference. The interference most easily occurs in the amplitude and least affects the carrier angle. The cost of this noise reduction property of angle modulation comes in the form of a complex receiver required to extract the modulating signal from the carrier phase. A simple receiver, such as an envelope detector is not usable in this case. Instead, the angle or the frequency of the received signal has to be estimated to extract the signal from it.

The analog modulation has ruled the world of broadcast communications ever since its inception. The latest trend is, however, towards the more robust digital modulation schemes. In data networks and wireless communications systems, digital communications has almost replaced analog to the point that analog modulation is not even considered as an option in many new systems. We devote the next section to the digital modulation schemes.

3.4. Digital Modulation

The baseband signals (NRZ, Manchester, etc.) discussed earlier have applications in local area networks, ISDN and data storage devices. The signal spectra of these schemes contain low frequency components that are attenuated quickly with distance in most transmission media. In order for long haul communications over wire and wireless media, we usually need to shift the signal spectrum to fit within the channel bandwidth. This applies equally to digital and analog signals. Digital modulation is performed by mixing the baseband digital version of the data signal with a high frequency carrier and using the resulting signal for communications. There are a number of ways in which a baseband data signal can be 'mixed' with one or more high frequency carriers. Following are three most commonly used mechanisms. Parallels can be easily drawn between the analog and digital modulations. This is due to the use of a carrier signal for both and the fact that information signal is used to modify one or more of the three attributes of the carrier, the amplitude, the frequency and the phase. The difference between the analog and digital modulation types is that in digital modulation the modification in carrier attributes is in the form of discrete steps while in analog modulation, it is in continuous form. In the remaining of this section, we will look at three basic types of digital modulation schemes.

3.4.1. Amplitude Shift Keying (ASK)

In ASK, the baseband digital signal is used to modulate the amplitude of an analog carrier. One way to perform ASK would be to transmit the carrier signal with constant amplitude for one level of digitally encoded baseband signal (e.g. NRZ) voltage and sending nothing for the other level. This is a direct mapping of NRZ into a high frequency carrier, as shown in Figure 3-15. ASK is a very simple mechanism and is a frequent choice in optical communications where a large amount of bandwidth is available. In optical communications, it is also called the intensity modulation (the light intensity varied between two limits).

For binary ASK (two types of pulses used to represent binary zero and one), the baud rate (pulses per second) is equal to the bit rate. It is possible to use more than two values for the amplitude of the modulated

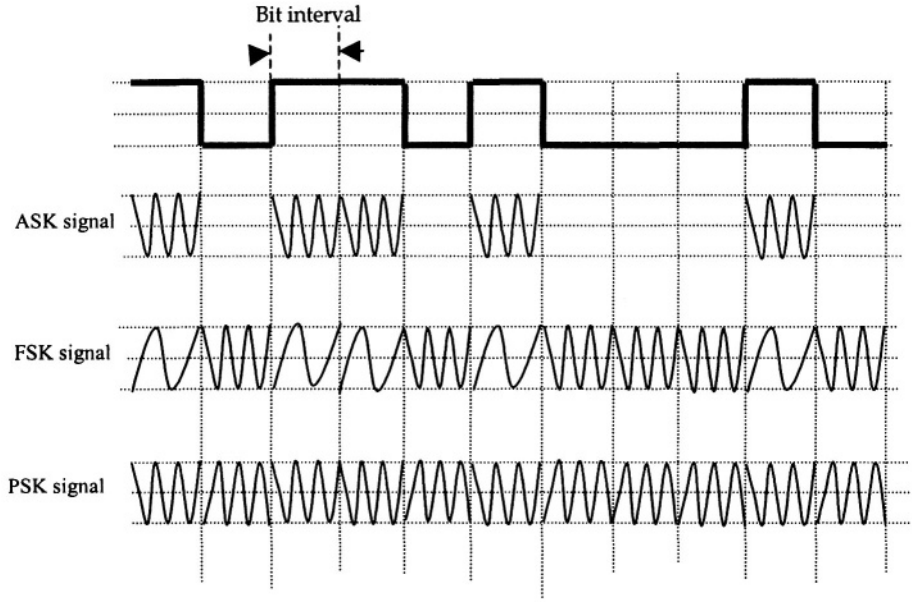


Figure 3-15. Digital modulation schemes

carrier. If that is the case, then each modulated pulse may represent *k* bits instead of one bit. If the number of amplitude values used in ASK is *M,* then each carrier signal interval could be used to represent $k = log_2(M)$ bits of data. The bit rate is *k* times the baud rate in this case. In Figure 3-15, binary ASK signal is shown that modulates NRZ-like baseband signal. As seen from this figure, a carrier with constant amplitude is transmitted for a positive voltage pulse and nothing (carrier with zero amplitude) is transmitted for a negative pulse. In other words, the baseband signal acts as a gate to turn the carrier signal on or off. For this reason, ASK is also called on-off keying (OOK). For a multi-level ASK, a table such as Table 3-1 is used by the transmitter and receiver to encode and decode bits from the carrier signal.

3.4.2. Frequency Shift Keying (FSK)

In binary FSK, two signals with different carrier frequencies are used to represent binary data. Since these carriers have different frequencies, they can easily be distinguished at the receiver. It has the added benefit over the ASK that if a bit is lost during transmission, it is known that the bit is lost, since there should always be a carrier signal for zero or one.

The baud rate in this case is again equal to the bit rate. As compared to ASK, signal synchronization is easier to maintain in FSK. In ASK, it is possible for the receiver to lose sync with the transmitter when a long string of no-carrier

signals occurs. In FSK, the difference between the carrier frequencies is kept large enough so that energy is not trapped in their byproducts. If M carriers are used for FSK with $M > 2$, then we have the M'ary FSK with the baud rate equal to $1/k^{th}$ of the bit rate, $k = log_2(M)$. The binary FSK shown in Figure 3-15 modulates a positive baseband pulse with a low frequency carrier and a negative baseband pulse with a high frequency carrier signal.

3.4.3. Phase Shift Keying (PSK)

In this type of digital modulation, the data information is imbedded in the phase of the carrier. The same carrier frequency is used for both types of bits (0 and 1) but the phase is inverted for one or the other. There are a number of ways of representing baseband data/signal in PSK. In simple binary PSK (or BPSK), two carrier signals are defined with same frequency and same amplitude but opposite phases. In differential BPSK, instead of defining two carrier signals with opposite phases, phase inversion is used to modulate one of the two types of binary signals. Figure 3-15 shows simple BPSK in which a positive baseband pulse is modulated into a cosine function and a negative baseband pulse is modulated in the negative of a cosine function. In BPSK, the baud rate and bit rate are equal. This is because the carrier phase (360° or 2π radians) is equally divided into two halves. In a multi-phase PSK (called M'ary PSK or simply MPSK), the carrier phase may be divided into M equal parts. For M equal divisions of the phase, each of the M resulting signals will have one of the M values of phases between 0° and 360°. In MPSK, each symbol can represent as many as Log_2M bits. Thus, the bit rate would be k times the baud rate, $k = log_2(M)$.

3.4.3.1. Quadrature Phase Shift Keying (QPSK)

A special case of MPSK is the QPSK (Quadrature PSK) in which $M = 4$. Therefore, the carrier phase is divided into four phases with difference of $[360/M]^0 = 360/4 = 90^0$ between adjacent signal phases. In this case, each symbol represents $Log_24 = 2$ bits. Thus the baud rate of QPSK is half the bit rate. Because of this, it is very popular in systems where bandwidth is at a premium, such as wireless communication systems and telephone network. With advancements in microchip technology, systems using MPSK for $M > 4$ are easily available these days.

As said earlier, ASK is the simplest type of digital passband modulations. The complexity increases for FSK and further for the PSK. However, most of the channel noise affects the carrier amplitude, which makes FSK and PSK less susceptible to noise than ASK. The performance of PSK is the best for the additive noise types.

3.4.3.2. Signal Constellation

For MPSK, there are M signals with equal amplitudes and separated by an angle of $2\pi/M$ radians from the nearest neighbors. Thus, as M increases, the closeness between adjacent signals increases. This makes it very easy for a short noise signal to blur the difference between two neighboring symbols, making it difficult for the receiver to distinguish between the two. This can be seen by viewing the signal constellation of MPSK scheme. A signal constellation is a graphical representation of signals as points on the $\{x, y\}$ plane. The distance of a point from the origin is the strength of the signal (e.g., the amplitude of the carrier) and its angle from the horizontal axis is its phase. Figure 3-16 shows a signal constellation for MPSK with $M = 8$. It is also called 8-PSK. Each signal can be used to transmit $\log_2(8) = 3$ bits. The

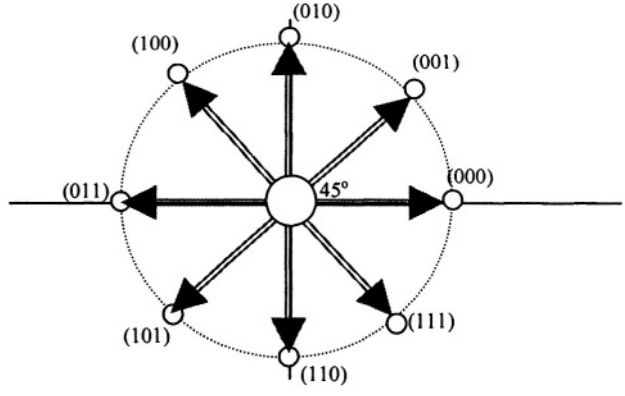


Figure 3-16. Example constellation for 8-PSK modulation. Each signal is represented by a point (shown as a small circle for clarity) on the circumference of the large circle. The bit rate is $\log_2(8)$, that is 3 times the baud rate. The angles between two adjacent symbols is $2\pi/8$ radians or $360°/8 = 45^0$. The radius of the large circle is the energy of each symbol.

figure also shows a tentative labeling of symbols.

The data at the PHY layer is transmitted in either the baseband or the passband modulation form. LANs typically use baseband modulation, such as Manchester or differential Manchester coding. Passband modulation is used in dialup connections, carrier systems for long haul transmission and for wireless cellular systems.

Just as the PHY layer does not pass data to a layer below it, the application layer does not get data from another layer above it. The data passed on to application layer could be processed in many ways before it

becomes available to the application. This processing depends on many factors, such as, whether the data is originated as analog or digital, live or stored in a file, or, in what format it is stored if that is the case. We discuss these issues under the user data in section to follow.

3.5. The User Data

The data part of the application protocol data unit (A-PDU) is what the users of data networks are really interested in exchanging. This data is in logical form, that is, it consists of strings of bits. When the application layer delivers this data to the destination computer user, the recipient may be able to look at it, process it or even change its format for obtaining useful information from it. The user of the 'user data' does not have to be a human being: it could be a software program, such as a Telnet server program or a hardware device, such as a robotic arm that responds to remote commands. When the user is a human being then data could have been originally in analog form in many instances. Examples of such data are the speech and pictures. Alternatively, many data types have digital form, such as amounts, numbers; or text form such as email, memos etc. The application layer, however, does not distinguish among analog and digital, or text and numbers forms. It needs the data to be available in binary form, from some application program or a file. Consequently, all data are converted into a string of binary numbers before being network ready. Let's look at the three variant forms of user data and how they are represented in binary form. We first consider an example of analog data: human speech signal in the next section. Following the example of analog signal, we will discuss the representation of numbers and text in a computer and network.

3.5.1. Digital Transmission of Voice

Transmission of voice in digital form is an interesting and most typical example of digital transmission of analog data. This type of user data has brought about most of the developments in the PSTN and continues to dominate the research for multimedia wireless networks and the Internet. In this example, we will consider the salient stages of voice transmission in a digital network. Voice, being an analog signal, first goes through digitization with the help of sampling and quantization. Then, the quantized voice signal is converted into binary form. Following the binary form, it is ready for storage, processing and transmission through a network as a digital baseband or passband signal.

At the core of the digital transmission of analog voice is the sampling theorem. A simple description of the sampling theorem is as follows.

3.5.2. The Sampling Theorem

Consider a continuous signal with a frequency spectrum between 0 and B Hz. According to the sampling theorem, all the information in this signal is preserved in its samples taken at a rate of 2B samples per seconds (bauds) or higher.

Example 3-1: *Speech signal is analog with bandwidth within 4 kHz. If a speech signal is sampled at a rate of 8000 samples per second or higher, these samples contain all the information in the original signal.*

Now we look at the example of transmission of analog speech (data) using digital signals.

Figure 3-17 shows the sequence of events before a speech signal is completely converted into binary form. Figure 3-17(a) shows an analog signal, such as speech. Let B be the highest frequency component in it. Then, according to the sampling theorem, samples taken at a rate of 2B bauds would preserve the information content of the signal. This gives an inter-sample time of 1/2B seconds. If B = 4 kHz, then 1/2B = 0.000 125 seconds or 125 microseconds (μsec). These (analog) samples can have any value from an infinite number of possibilities. Therefore, the next step is to limit these samples in having a value from a set of finite values (digital samples). This process is called quantization and is explained further when we discuss pulse-coded modulation. After the signal samples have been quantized, the next step is to represent their values in binary form. This results in the conversion of an analog (speech) signal into a string of binary numbers. This binary data can be transmitted digitally using many of the baseband or passband modulation schemes. One particularly interesting scheme is the pulse-coded modulation (PCM). PCM and its variants, such as adaptive PCM (APCM) and adaptive differential PCM (ADPCM) have been extensively used for voice communications on digital links since the 1960s and continue to be popular in today's multimedia networks. It may be recalled, however, that for a computer communications network, the voice digitization simply provides the data part of the application PDU. Actual transmission does not occur until the physical layer. In the following, we will describe PCM in greater detail.

3.5.3. Pulse Coded Modulation (PCM)

For this discussion we assume that the speech signal has a bandwidth of 4 kHz. Then, according to the sampling theorem, the minimum sampling rate of speech signal should be 8,000 bauds (samples per second). Suppose that the electrical analog voice signal has voltage levels between 0 and 5 volts. Accordingly, each sample has a value within the same range (0 to 5

volts). To simplify matters, the first step we do is called *normalization* of the signal value. Normalization is done by dividing all signal values by the maximum (5 volts in this case). The result of normalization is that the normalized signal (always) has magnitudes between 0 and 1. Next step is *quantization* of the signal samples.

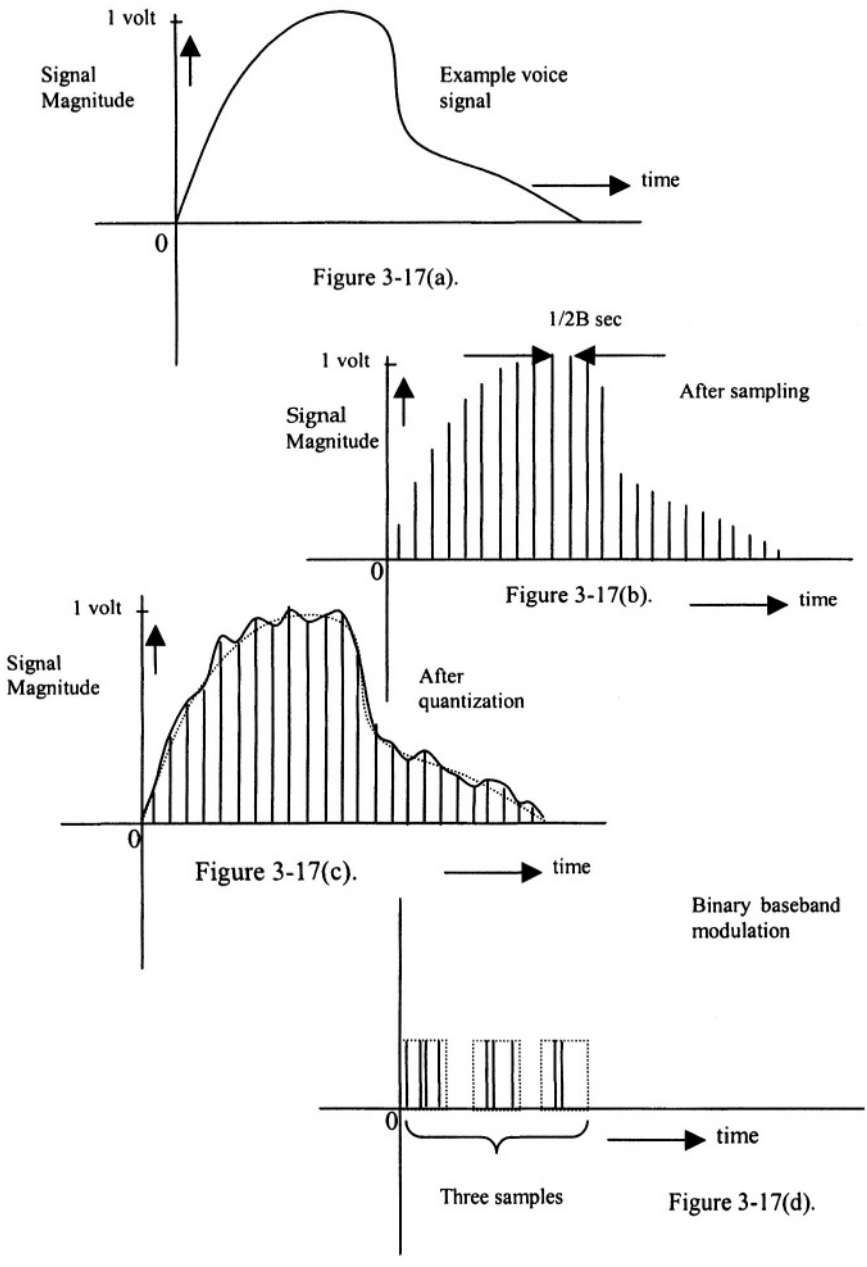


Figure 3-17. Conversion of analog signal into a bit stream using sampling theorem. Figure(a) shows an example analog signal; (b) shows the signal samples with an inter-sample time of 1/2B seconds. Sample values are still analog; in (c) quantization results in digital set of samples, meaning that the samples can assume one of a finite set of values. Value of ach sample can be converted into an equivalent binary number. Binary encoding schemes such as NRZ can be later used to digitally encode this bit stream, as shown in (d).

Quantization is the process of equating the analog samples to their closest digital values. For this purpose, we choose a set of (digital) values. It is customary to choose the number of values that can be represented as power of 2. Examples are 2^4 or 2^5 or 2^k where k is an integer. For $k = 4$, we have 2^4 or 16 values for sample magnitudes. The reason for choosing the number of values to be a power of 2 is that these values can all be represented as binary numbers using k bits. For example, 16 values can be represented by 4 bits with binary representation (0000) through (1111). In this way, Table 3-3 can be used to map actual sample values into digital values.

Table 3-3 A quantization example for 4-bit digitization

Sample value range	Closest digital value	Equivalent 4 bit binary value	Sample value range	closest digital value	Equivalent 4 bit binary value
0-1/16	1/32	0000	8/16-9/16	17/32	1000
1/16-2/16	3/32	0001	9/16-10/16	19/32	1001
2/16-3/16	5/32	0010	10/16-11/16	21/32	1010
3/16-4/16	7/32	0011	11/16-12/16	23/32	1011
4/16-5/16	9/32	0100	12/16-13/16	25/32	1100
5/16-6/16	11/32	0101	13/16-14/16	27/32	1101
6/16-7/16	13/32	0110	14/16-15/16	29/32	1110
7/16-8/16	15/32	0111	15/16-16/16	31/32	1111

Figure 3-18 shows the implication of the use of Table 3-3.

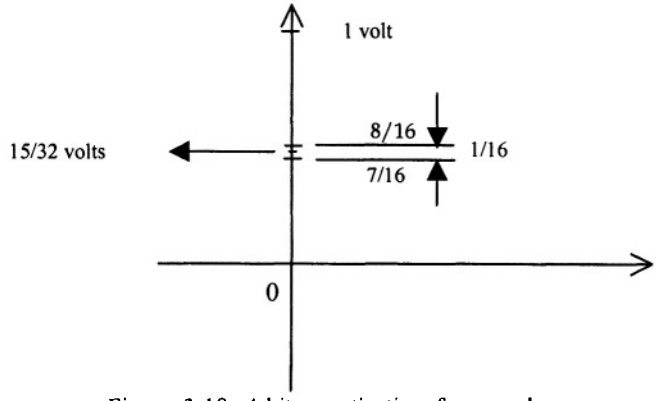


Figure 3-18. 4-bit quantization for samples with normalized values (between 0 and 1 volt). A sample with value anywhere between 7/16 and 8/16 volts is assigned a quantized value of 15/32 volts

Here is a summary of the quantization procedure.

Step 1. If not given, decide how many bits will be used for quantization, for example k bits per sample.

Step 2. Normalize the analog signal by dividing it by its maximum value. This makes unity as the maximum signal magnitude.

Step 3: Divide the interval between the minimum and maximum signal values into 2^k-1 small intervals. If the signal values are between 0 volt and 1 volt, then each step will be $1/(2^k-1)$ volts. This is assuming that all steps are of equal size.

Step 4: Let the midpoint between two steps represent the value of a quantized sample whose actual value may lie anywhere between plus or minus half the step value. For example, a value of $(L+1)/2^{k+1}$ volts is assigned to all samples with actual value between $L/2^k$ and $(L+1)/2^k$ volts, for $0 \le L < 2^k$. In Figure 3-18, $L = 7$ and $k = 4$. If this rules is followed then a binary signal representing 0111 (from Table 3-3) is always received as equivalent to an analog signal of 15/32 volts. In reality it could have any value between 7/16 and 8/16 volts.

Note

The above is one of the several possible ways of quantization procedure. There can be variations depending upon factors such as the minimum normalized signal being zero or non-zero, step size constant or variable, and where to define the quantized values, the middle of steps or on the step boundaries.

Once the quantization procedure is complete, the next step is to convert the quantized values to binary code. The binary code will be the data for generating baseband PCM signal or any of the passband signal types.

Quantization noise (error) is the difference between the actual value of an analog sample and the assigned digital value. For example, from Table 3-3 we know that if the analog value of a sample is in the range 4/16-5/16, the assigned value is 9/32. For a sample whose value is 0.26 volts the assigned value is 9/32 = 0.281. Thus, there is an error of 0.281-0.26 = 0.021 volts due to quantization. The maximum quantization error is the half of the step size. If step size is Δ, then the maximum quantization error is $\Delta/2$. For k-bit PCM, there are 2^k-1 steps per volt (normalized signal) Therefore, the step size is $1/(2^k-1)$ volts and the maximum quantization error is half of that.

Example 3-2: For an 8-bit PCM system to be used to transmit an analog signal with range between 0 and 6 volts and bandwidth of 4 kHz, we have the following parameters.

Normalized voltage level range: 0 to 1 volt

Normalized step size = $1/(2^8-1)$ = 1/255 volts

Absolute step size = 6 x 1/255 = 6/255 volts

Maximum quantization error = ½ × 1/255 = 1/510 volts normalized or 6/510 volts absolute

Maximum signal size for mid-step resolution = 1 - 1/510 = 509/510 volts normalized or 6×509/510 volts absolute

Minimum sampling rate = 8000 bauds

Bits per sample = 8 (given)

Minimum bit rate = 8 × 8000 = 64 kbps

The above is a straightforward description of PCM. Actual systems are much more complex due to certain characteristics of the voice signal, such as the information is carried more in one part of the signal spectrum than the other. The quantization is performed using methods that minimize the quantization error. Also, some signal levels are more sensitive to noise than others. The lower signal levels, for example, are affected by noise more heavily than the higher amplitudes. Again, we can manipulate the

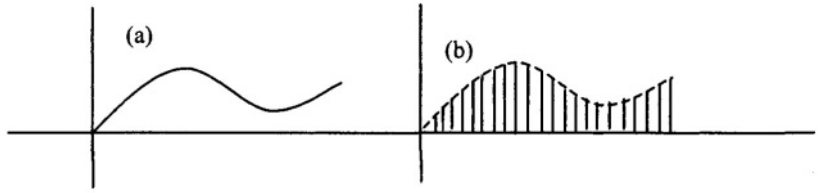
quantization in such a way that finer resolution can be used to represent lower signal amplitudes so that it is easier to detect them at the receiving side.

3.5.3.1.1 . Channel Bandwidth Requirements for Digital Transmission

Among other things, the above example highlights the relation between digital transmission and the required channel bandwidth. For analog transmission of a B Hz signal, the channel bandwidth must be greater than or equal to B Hz. However, if the signal is digitized using k bits per sample, then the channel must be able to sustain a bit rate of $2\times k\times B$ or higher. For most of the binary modulation schemes discussed, this translates approximately to $k\times B$ Hz of channel bandwidth, thus requiring k times as much bandwidth as the analog transmission. This is the cost of digital transmission. In communications via cables, this requirement is easily met these days by advancements in optical fiber manufacturing. In wireless communications, however, low bit rate digital schemes are sought after because the bandwidth is always at a premium. Modulation schemes are investigated with high bandwidth efficiency. The bandwidth efficiency may be defined as the number of bits carried per Hz of the channel bandwidth. Combinations of MPSK and ASK usually provide good solutions for such systems.

3.5.4. Delta Modulation

Another technique to convert an analog signal into digital bit stream using signal samples is the delta modulation. Recall that in PCM, each sample is converted into a bit combination. Contrarily, in DM, only the difference in values of consecutive samples is coded into binary number. The simplest form of DM will use just one bit to code each sample depending on whether the sample value is greater than or less than the previous sample. An example is shown in Figure 3-19.



01111111110000000011111

(c)

Figure 3-19. (a) Analog signal, (b) Sampled signal, (c) Single bit DM data.

After a bit stream representation has been obtained, any of the baseband or passband modulation mechanisms can be used for digital transmission. For one-bit DM, here's a list of steps after sampling the analog signal.

Step 1. Represent the first sample of analog signal in binary form.

Step 2. If the next sample is higher than the previous code the next sample as a binary '1'.

Step 3. If the next sample is lower than the current code the next sample as a binary '0'.

Step 4. If the next sample value is equal to the previous one, code the first such occurrence opposite to the previous bit and then alternate to '0' and '1' for later occurrences.

The above rules can be changed in many ways without affecting the output too much.

In this way, one does not need k bits per sample as in PCM. One bit per sample will suffice. However, sampling rate for delta modulation is supposed to be much higher than the PCM for comparable quality of speech. Consequently, the advantage in saving bandwidth in sampling is lost in rendering the quality equal to the PCM. If DM uses sampling rate of about 2B as in PCM, the result will be in *slope overload* or *slope underload.* In *slope overload,* the slope of the samples is sharper than the actual signal due to abrupt rise in the analog signal values. In *slope underload* condition, the slope of the quantized signal is less sharper than the analog signal, due to the analog signal rising too slowly or falling too fast.

The discussion in this section applies equally to video data, or any analog signal. The differences between speech and video signals are mainly related to the signal bandwidth. Each type of video application, from still pictures to high-motion interactive video, requires a conversion process from analog signal to digital data representation. Other processes to enhance quality

or save bandwidth play important roles as well. The example of speech signal, however, makes a general point about how an analog signal can be converted into a digital bit stream.

3.6. Text and Numerical Data

A major part of user data consists of documents, numerical quantities, and symbols. The basic unit of these data types is a character. Letters of an alphabet, digits and punctuation marks are all examples of characters. In fact, space between two consecutive words is also a character. If we have a look at a computer keyboard, we see the numbers, letters and punctuation marks along with a space bar as printable characters. Each time we hit a key, electrical signals travel through a set of wires to the computer memory or the processor. These signals are accepted by the computer as a string of bits. Inside the computer is a software program that interprets this string of bits as the character represented by the keyboard stroke in question. If the user wants to see this character on a monitor screen then the software program draws the character just as it should look like in actual writing.
Glyph is what we see on the screen.
Code point or *code* is the bit string that defines a character.
Font is one of the many styles use for writing a character.

For interoperability among computer and communications equipment, the codes to represent characters are standardized. Just like other standards, character codes have proprietary, national and international significance. There are several reasons for having each type of significance. National codes usually care more about the national language and the number system used nationally. For example, American Standard Code for Information Interchange (ASCII) is a worldwide popular code for text and numerical data storage and interchange in computers and networks. Its worldwide popularity is mainly due to a lack of competition at the time of its inception and due to the fact that English language and computing machinery produced in the USA have dominated the world scene for much of the computing history. ASCII combined character processing with data transmission. It is this capability of data communications characters that has really brought fame and dominance to ASCII. Other nations, too, have their own character sets and respective codes. Some languages like the Chinese, Japanese and Korean have characters that represent a whole word. These languages have to have codes for words instead of alphabetic characters.

There are proprietary codes as well, even though they are more of a part of history now. IBM has been at the forefronts in defining its own character sets to be used in its mainframe computers. Due to the popularity of ASCII, the IBM standards (e.g., EBCDIC - extended binary coded decimal for information change) have not been widely used and studied. However, they do exercise influence over ASCII and other sets.

The globalization of computer networking has resulted in codes that have code points for many languages, mathematical symbols and even the ideographic characters of Chinese, Japanese and Korean languages. In the two sections to follow, we will say a few words about ASCII and an evolving International code, called the *Unicode.*

3.6.1. ASCII (American National Standard Code for Information Interchange)

The ASCII code was laid down for computers with the following characteristics:

1. They use English as the text editing language.
2. They could be used to code not just letters and digits, but also invisible punctuation marks.
3. Computers need to transfer data to other computers across a link or to another device, such as a printer connected directly through the parallel port or across a network.

At the time of ASCII specification, it was obvious that all of the characters could be accommodated with 7-bit codes. There are a total of $2^7 = 128$ combinations of 7 bits, giving the potential of 128 codes. This could easily take care of all alphabets, digits and punctuation marks. In fact, it leaves a number of codes to be used for characters that do not have a glyph. Such characters have descriptive name, such as NUL or EOT (end of transmission). The ASCII table is easily available in many books on data communications and on the Internet. Following is an example of ASCII code definition for English language.

Table 3-4 The 7-bit ASCII code examples (with equivalent character in EBCDIC)

Note: The leading 0 is required only in 8-bit EBCDIC code set.

Character	ASCII code	EBCDIC equivalent	Character	ASCII code	EBCDIC code
A	01000001	A	!	00100001	SOS
B	01000010	B	"	00100010	FS
a	01100001	A	$	00100100	BYP
b	01100010	B	0	00110000	0
NUL	00000000	NUL	1	00110001	1
SOH	00000001	SOH	9	00111001	9
ETX	00000010	ETX	DEL	01111111	"

NUL = Null; SOH = Start of header; ETX = End of text; SOS = Start of significance

FS = Form separator; BYP = Bypass

The third and sixth columns of Table 3-4 show the equivalent characters for the same codes in IBM's Extended Binary Coded Decimal

Interchange Code (EBCDIC). The ECBCDIC code (also known as CP1047) is not identical to ASCII as seen and was primarily designed for file exchange between IBM mainframe computers and terminals. Another difference between the ASCII and EBCDIC is that EBCDIC is by definition an 8-bit code as compared to the 7-bit ASCII. Therefore if two computers one using ASCII character set and the other EBCDIC exchange data, there is a need for translating the codes in both directions.

3.6.1.1. ASCII and Other Standard Organizations

One of the main reasons for the success of ASCII is the presence of device control and data transmission characters, such as start of header (SOH), end of text (EOT) and many others. Due to the International scope of ASCII, other organizations also adopted the same character set as standard. For example, the 7-bit code ISO 646 (International Reference Version) is identical to the ASCII 7-bit character set. Most communication and computer processing occurs in bytes that is equal to 8 bits. ASCII characters can be expanded to 8 bits each by using an additional bit as a parity-check bit. The use of this bit adds the capability of detecting an odd number of errors (see Chapter 5 for details).

The indispensability of ASCII has been transported in the form of other codes adopting ASCII as part of it. With the spread of personal computing worldwide, the one deficiency of ASCII surfaced prominently, that is, the lack of characters of other languages and mathematical and Greek symbols. One way to accommodate other characters was to use a full 8-bit ASCII code and define an additional 128 characters. This adds another layer of configuration to the terminals using ASCII characters set. Yet another and more useful way was to define a more comprehensive set of codes. Two such codes will be discussed next, the ISO 8859-1 based on glyphs called ISO Latin-1 and the ISO 10646, the Universal multiple-octet Coded Character Set (UCS). ISO 8859-1 has practically replaced ASCII by making it a subset and adding another 128 characters. UCS is more universal and has embodied characters from all languages of the world.

3.6.2. ISO 8859-1 (ISO Latin -1)

Even though ISO 8859-1 is considered to be the same thing as Latin - 1, there is a difference of opinion among the scientific and engineering community on this issue. The difference is due to the fact that Latin-1 is a set of glyphs while the ISO 8859-1 is the codes for Latin - 1. As defined earlier, a character has many attributes, e.g., glyph, code, font, and description. Here is an example illustrating the differences among these attributes.

‘A’ is the *glyph* of the first letter in the English alphabet
‘ 1000001’ is the *code* for A as defined in ASCII

'A' is 'A' written in Century Gothic *font*
The *description of* A is "the first letter of the English alphabet"

Another term *repertoire* is used to describe a set of glyphs on display. The International Organization for Standardization (ISO) together with the International Electrotechnical Committee (IEC) has defined a set of glyphs as ISO Latin - 1. The codes for ISO Latin - 1 are defined in the document ISO 8859-1 (ISO 8859 part one). The larger (ISO 8859) standard has a total of 15 parts that cover all the languages using Latin alphabets, Cyrillic alphabet, Arabic alphabet, Greek alphabet and Hebrew alphabet and lot more. ISO 8859 -1 is the part one of the standard.

The Latin -1 consists of all the characters defined in 7-bit ASCII with identical codes using leading zeros. The remaining 128 characters consist of new mathematical, International and accented characters added to ASCII. In one way, OSI 8859-1 has completed the ASCII set but not quite internationalized it. It is used by most of the computer software, such as, HTML, Microsoft Windows™ and X-Windows. The inclusion of all international characters has been achieved through another ISO standard, the ISO 10646, another joint project of ISO and IEC.

3.6.3. UCS (Universal multiple-octet coded Character Set)

UCS consists of two formats: a 32-bit coding scheme called UCS-4 (four-octet UCS) or the *canonical form* and a 16-bit format (UCS-2) called the Basic Multilingual Plane (BMP). UCS integrates all the codes that have been agreed upon nationally or international into one unified code set. Thus, the first 127 characters of BMP are the 7-bit ASCII code set. Similarly, ISO Latin -1, too, is a subset of BMP.

With the extended use of computers and international community's trust in it, more and more characters are needed to be coded. These included all languages not coded as yet, industrial symbols and ancient scripts. With a wide range of possibilities of 2^{32} characters, UCS has taken care of all the current and (perhaps) future needs of characters. There are even some space left for user-defined characters. For example, drink manufacturers sellers may define their own set of characters to label various types of drinks, depending on e.g., sugar content etc. The codes already defined include, among existing codes, phonetic alphabet, all forms of Arabic characters (four in all), Indic languages (including Bengali, Gujrati, and Tamil and a lot more), South Asian languages (Khmer, Lao, Thai, etc.), ideographic languages (Chinese, Japanese, Korean), aboriginal syllables (Cherokee etc.), box and line drawing symbols, special optical character recognition (OCR) set (used, e.g., on checks) and a lot more. If you see a software application in which one can print any printable character, it is most likely using UCS.

Unicode®

Unicode is an international character-encoding standard for processing and exchange of characters. Fully compatible with ISO 10646, it is sometime considered to be ISO 10646 itself. However, Unicode goes beyond the ISO 10646 characterization of glyphs and defines characters as abstract entities. In fact it does not define glyphs at all. The compatibility with ISO 10646 is only in the sense that both sets use the same codes for same characters. Like ISO 10646 allows for coding of more than a million characters (2^{32}) that include BMP, about 8000 unused code points (for user defined characters) and another over 97000 code points. The currently available version 3.0 has definitions for 49,194 characters. At this time another 46,000 characters are ready to be included in the next version. The most authentic source of Unicode information is the official publication "The Unicode Standard, Version 3.0", published by Addison Wesley Longman Publisher, 2000. Information can also be obtained from the official web page of Unicode: www.unicode.org.

3.7. Summary

Data means different things to the users and the network. Within the network, data changes its forms as it travels from the application layer protocols to the physical layer. For all layers, except the application and the physical, the data consists of the PDUs. At the physical layer, data goes through many forms, from logical form (bits), to encoded form (series of pulses), to modulated form. Standards in coding and modulation techniques are developed so that the physical layers at two communicating computers can understand each other. Modulation and coding can be done at two levels, the baseband level and the passband level. In baseband encoding of data, the logical form of data is converted into an equivalent set of baseband waveforms, such as NRZ, Bipolar-AMI and Manchester coding. In passband modulation, the baseband signal is modulated with a high-speed carrier signal for transmission to longer distances. Local area networks typically use baseband modulation while wireless systems or telecommunications carrier systems and DSL use both modulation types. Manchester and differential Manchester coding schemes are used in Ethernet and Token Ring standards. MPSK and FSK are used in the phone line modems.

The application layer data represents the user information in bit form. User data could be originally in analog or digital form. If it is in analog form then sampling theorem provides the basis for its digitization. According to the sampling theorem, the samples of a continuous signal retain all the information if it is bandwidth limited and the samples are taken at a rate twice the highest frequency in the signal. Quantization can be used to convert samples into digital form. This digital form will have a binary equivalent as well. User data in the form of text and numerical information is converted into bit form by defining binary codes for each character of data. Such codes have been defined by industry [such as IBM's Binary Coded Decimal (BCD) or its extended form EBCDIC], nations [such as American National Standards Institute (ANSI)'s American National Standard Code for Information Interchange (ASCII)], or internationally [such as ISO/IEC 10646]. The ISO 10646 and the Unicode® are designed taking into consideration all the characters used by all nations and leaving space for future expansion.

3.8. Review Questions

1: Define *frequency* of a communication carrier signal?
2: Define the *bandwidth* of a communications signal?
3: Which one of the following data encoding schemes is the most efficient in terms of (i) bandwidth, (ii) dc bias, (iii) self-clocking capability: NRZ, AMI, Manchester?
4: What are violations of AMI code and what is their use?
5: What are differential coding schemes and what is their advantage over non-differential schemes?
6: With the help of phasors, indicate where information is stored in the carrier for (i) ASK, (ii) PSK.
7: What is the relation between the baud rate and bit rate for 8-PSK?
8: Why is it easier to detect an error due to additive noise occurred in FSK than ASK signal?
9: The highest frequency component in a band-limited signal is 200 kHz, What should be the minimum sampling rate of the signal so that no information in the signal is lost?
10: A 10 volt analog signal is normalized for communications using 8 bit PCM. What is:
(i) The step size?
(ii) Maximum Quantization noise?
(iii) The maximum normalized voltage level?

This page intentionally left blank

4. The Physical Layer

In the study of data communications, functions and procedures at the physical layer play the most important role. It is this layer that deals with the uncertainties of the channel media. At all other layers information and connection exist only in a logical sense. Information on this layer exists in logical as well as physical form. So does the connection. The limitations due to channel bandwidth, channel noise and external interference are all taken care by the physical layer protocols.

In the OSI reference model, the general responsibilities of the physical layer protocols deal with the mechanical, electrical, functional, procedural and transmission characteristics of the interface between a computer (terminal) and a network (user-network interface node). In other words, specifying the connector shape and pin configuration, voltage levels and tolerances, signal types and signaling sequences are all needed at the physical layer. With the advent of high-speed and wireless networks, responsibilities of the physical layer protocols have increased. The multiplexing and synchronization functions of high-speed networks and bandwidth limitation along with the statistical nature of the wireless channel have led standardization agencies break the physical layer protocols into two sublayers: one for the physical media dependent (PMD) characteristics and the other for mapping physical layer logical functions to the higher layer and backwards. The later sublayer is called the physical layer convergence procedures.

In this chapter, we will start with a discussion on the bandwidth and transmission limiting factors of a physical medium - that is, the channel impairments. We will then describe channel types in common use including metallic and optical cables. Wireless channels will be considered together as a single category, including air and space as well as the indoor and outdoor wireless media. We will close this chapter with two examples of physical layer protocols: a rather popular protocol for low to medium speed communications EIA232 and a wireless physical layer protocol specified by IEEE802.11 committee.

4.1. Channel Impairments

A transmission channel is the path of information signal. It could be made of matter, such as metals, water and air, or it could be the space. A channel is characterized by the speed at which information could pass through it without significant deterioration. Sometimes, a channel is defined as a certain amount of bandwidth of the medium. In fact, the term 'transmission medium' is more popularly used to convey the matter type (metal, glass or air) rather than the channel. However, for this section, we use the terms transmission medium and channel interchangeably.

Channel affects the signal in many ways. Some of the effect is related to the radiation of energy that results in weakening of the signal as it moves away from its origin *(attenuation).* Other effect is because of the frequency selective behavior of the channel that results in different frequency components of the same signal arriving at the destination with different strengths. Yet, some of the channel impairments are related to the electronic noise created due to the random motion of electrons, atoms, and molecules. Here is a brief account of each.

4.1.1. Signal Attenuation

Due to the signal spreading and the resistance of the medium, the signal strength reduces as it travels on a cable or in the air. Such reduction in signal strength is called as attenuation. For each medium, the attenuation can usually be predicted from the knowledge of medium characteristics. In general, there is less attenuation in cables than free space. Atmosphere is worse than free space and usually causes significant amounts of attenuation.

Cables are sometimes called as guided media because of the fact that a signal traveling on a cable has most of its energy guided in the direction of the cable. Space, however, is an example of an unguided media in which the direction of energy propagation is not strictly controlled. There are transmission systems with directed transmission in the air, but even for those systems, the energy is directed to an angle instead of a specific direction. This is done by using directional antennas for signal radiation.

4.1.1.1. Attenuation and Propagation Loss

There are more factors that impact the received signal energy in an unguided medium than in guided media. In guided media, the resistance of the cable increases as a function of length, resulting in an increase of signal attenuation. The effect of distance is more pronounced in wireless media though. Usually, the attenuation increases (signal strengthen reduces) in proportion to the square of distance or worse.

Figure 4-1 shows how a signal will be attenuated if the attenuation is proportional to the n^{th} power of distance (d). The figure shows a plot of a function $f(d,n) = 1/d^n$. The effect of distance is similar to shown in Figure 4-1. In reality, there are many other factors, such as the signal frequency, the antennas type and weather etc. Free space is assumed to have $n = 2$ and some wireless telephone design use a value of 4 for n. The ordinate of Figure 4-1 is $10\log_{10}f(d,n)$. It is customary to represent and plot signal powers and loss on a logarithmic scale.

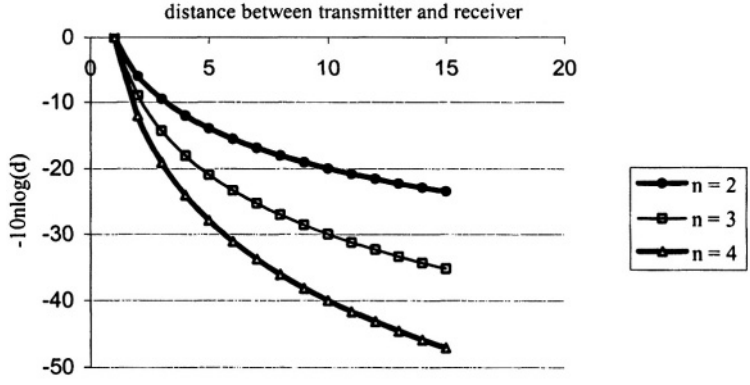


Figure 4-1: Signal stregth in dB as a function of distance

Logarithmic Scale:

We define the propagation loss as the ratio of the powers of the received and the transmitted signals when the signal powers have the units of Watts. This ratio translates to difference of powers when represented on logarithmic scale (in dBw or dBm). If the units of the received power (P_R) and transmitted power (P_T) are in Watts, then the loss L is unit-less and is equal to P_R/P_T . To represent loss in decibels (dB)s, we have the following expression:

$$L \text{ (in dBs)} = 10\log_{10}(P_R/P_T) = 10\log_{10}(P_R) - 10\log_{10}(P_T)$$

$$L \text{ (in dB)} = P_R \text{ (in dBw or dBm)} - P_T \text{ (in dBw or dBm)}$$

The definitions of dBw and dBm are given below:

x *Watts of power* $= 10.\log_{10}(x)$ dBw
$= 10.\log_{10}(1000.x)$ dBm.

Dependence of loss on frequency

A more general expression for propagation loss is $L = P_R - P_T = -10.n.G.log(4\pi d/\lambda)$ dB, where G is a factor that depends on antenna design in a wireless channel and $\lambda = c/f$, where c is the speed of light given as 3×10^8 m/s. λ is called the wavelength of a signal. P_T and P_R are the transmitted and received powers respectively. As mentioned earlier, for free space $n = 2$.

Example 4-1

Suppose that a signal has a power of 10 mW at the transmitter. If the receiver is at a distance of 100 meters what will be the power received given an attenuation of 1 dB per meter. Assume that the attenuation in dB increases linearly with distance.

Solution: An attenuation of 1dB per meter implies a total attenuation of 100 dB. In other words $P_R - P_T = -100$ dB. Then the power received will be simply $W_R = W_T - 100 = 10\log(10) - 100 = -90$ dBm $= 10^{-9}$ mW.

Note: Attenuation in dB generally does not increase linearly with distance.

Example 4-2

Suppose that the same signal has a power of 10 mW at the transmitter. If the receiver is at a distance of 100 meters what will be the power received given an attenuation proportional to the square of the distance.

Solution: An attenuation proportional to the square of distance implies that n = 2. A total attenuation of $10\times2\times\log(4\pi\times100/\lambda)$ dB. Thus we need the frequency or the wavelength of the signal in order to calculate the received power.

Example 4-3

If the signal frequency is 300 KHz in the above case, find the power of the received signal?

Solution

In this case $10\times2\times\log(4\pi\times100/\lambda) = 10\times2\times\log(4\pi\times100\times f/c) = 10\times2\times\log(4\pi/10)$

Thus the propagation loss is about 2dB. The received power is 10 dBm - 2dB = 8dBm = 0.1 mW.

4.1.2. Delay Distortion

A signal can be represented as sum of a number of sinusoids. For a periodic signal, there is a fundamental frequency and its higher harmonics. For an aperiodic function, there are infinite number of frequency components between any two frequencies. When a signal is transmitted from one point to

another point, each sinusoidal component of the signal arrives with a phase different from other sinusoidal components of the signal. This difference in phases of the arriving sinusoids causes their sum to be different from the transmitted signal. Thus, as the signal travels from one end to another, the delay suffered by each sinusoidal component in the signal is different,

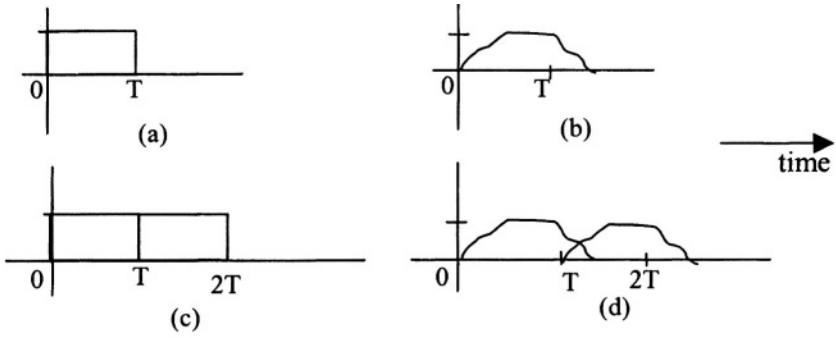


Figure 4-2. Effect of channel distortion; (a) voltage pulse of duration T; (b) the same pulse after dispersion; (c) two adjacent voltage pulses; (d) the same two pulses after dispersion.
Intersymbol Interference (ISI) is the interference of two pulses in (d).

resulting in a distortion of phase delays at the receiving end.

The result of delay distortion is 'dispersion' of the signal energy out of its time-limited range. If many signals are transmitted one after the other, then each signal spreads in time. This causes adjacent signals to overlap as they propagate along a medium. For example, in binary transmission, where each signal represents a zero or one, the adjacent pulses may overlap to the extent that the receiver cannot separate them. This phenomenon is called inter-symbol interference (ISI) and is a major concern on lines with limited bandwidth, such as telephone lines. Figure 4-2 shows the concept of dispersion. In order to design a communications system to combat ISI, we have to do several things, such as design pulse shapes and perform equalization.

Equalization restores the delay distortion in such a manner that all signal components seem to be arriving at the same time. *Equalizers* do this by simulating the inverse channel behavior.

4.1.3. Noise

Noise in electronic communications plays a key role in the system design. It can be defined as the spurious signals added to the communications

signal by channel, equipment, electromagnetic coupling, and clicking of switches. There are several types of noise that affect the quality of reception. Some of these are as follows.

4.1.3.1. Thermal Noise

The thermal noise arises naturally from the thermal agitation of electrons in all electronic devices. Thermal noise is a statistical quantity and is represented by its probability distribution. It is well known that the thermal noise has a Gaussian distribution and exists at all frequencies (hence also called *white* noise). The property of Gaussian distribution allows it to be completely specified in terms of its power density (= power per Hz), usually denoted by N_o (= kT), k is the Boltzmann's constant and T is the temperature in ^oK (degrees Kelvin).

4.1.3.2. Crosstalk

Crosstalk is a type of noise originally observed in the form of phone conversation spilling from one user (circuit) to another. It results from electromagnetic coupling of wires at close proximity from one another. There are two types of crosstalk noises, the near-end crosstalk (NEXT) and the far-end crosstalk (FEXT) observed in telephone local loop. The NEXT is due to the coupling of signal with another signal on the same side of communications circuit. FEXT is the coupling of a signal with another signal from the far side of communications circuit. The coupling signals are usually reflections of the same signal. Another term used for digital communications networks to describe crosstalk is *echo.*

4.1.3.3. Impulse Noise

This is another type of noise that plays a significant role in data communications. It is due to the 'clicking' of switches and results in either audible click sounds during conversation or spikes in digital bit transmission. Another source of impulse noise could be sparking due to imperfect insulations.

4.1.4. Multipath

Multipath is very similar to *echo* in concept. It is the phenomenon in which a signal arrives at the receiver from reflections and sometime including the direct line of sight (LoS). The resulting signal is a sum of its replicas with different arrival times. The study of multipath is important especially in the wireless communication systems and offers the most significant challenge to the system designers. The uniqueness of this noise type is that it could be used constructively by combining the signal replications in a constructive way.

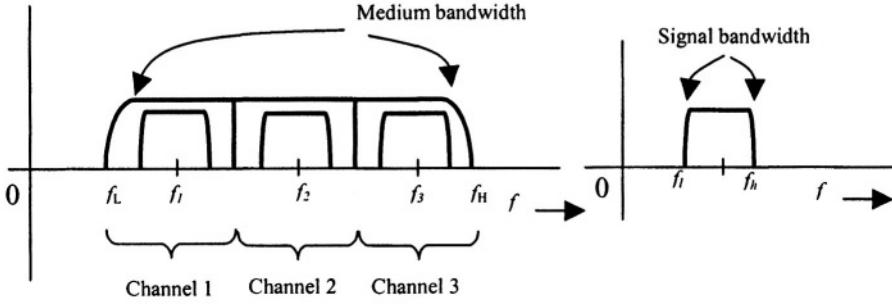


Figure 4-3. Medium bandwidth $B_M = f_H - f_L$ is much larger than the signal bandwidth $B_S = f_h - f_l$. The medium is divided into three channels, centered at frequencies f_1, f_2 and f_3, each with bandwidth greater than B_S.

4.2. Transmission Media

In the discussion on transmission channel, we had tried to differentiate between the channel and the medium. The difference between a channel and medium is most obvious when the medium can consist of more than one channel. Such is the case almost with all transmission media. Even in today's telephone subscriber's lines, there could be more than one channel on the subscriber's line. In DSL, the twisted copper wire pair provides separate channels for telephone and the Internet. In general, if the medium consists of a certain bandwidth, say $B_M >> B_S$ where B_S is the signal bandwidth, then there is the possibility of having more than one channel in the medium. Figure 4-3 shows how it is possible to do so.

Besides bandwidth, the transmission media can be classified in many ways depending on factors such as material, cost and proximity to the interference. Generally, we classify transmission media with respect to the propagation mechanism of information signal. With that in mind, we have four types of media that figure prominently in data networks. These are solid conductor, such as twisted copper pair, hollow conductors, such as co-axial cable, light carriers, such as optical fiber and wireless media such as atmosphere and space.

In twisted copper pair, coaxial cable and atmosphere the information travels in the form of electromagnetic signals. However, the exact nature of propagation is different in all the three media. The bandwidth characteristics are different as well. We will briefly describe each in the sections to follow only to highlight their properties relevant to data communications. The wireless media will be covered separately in a subsequent section, as their

characteristics are widely different from guided media. In optical fibers, information signal consists of light rays and waves. Certain properties about the reflection and refraction of light rays are used to trap light in such cables. We will discuss optical fibers in the next section in order to place them together with cable types.

4.3. Cables in data communications

If we look at the catalog of a cable manufacturer for data communications, we can expect a large number of products. A more careful study, however, will reveal that most of these cables belong to either twisted pair copper or co-axial categories. The vendors are limited in their choice of manufacturing due to standardization of data cables. It turns out that most of the cabling needs for telecommunications are taken care by the telephone cables or its enhanced versions. For networks that transmit broadband data signals, such as television broadcasting networks, or high-volume long distance trunks, we will find some kind of co-axial cable as the choice for a

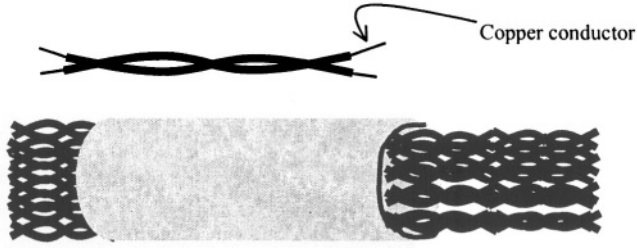


Figure 4-4. UTP pair and bundle

transmission medium. Since the 80's, developments in optical fiber communications have led to the replacement of many co-axial cable systems by optical fiber. In modern metropolitan and local area networks, optical fiber is becoming the choice of medium in more and more cases.

4.3.1. Twisted Pair Copper Cables

A twisted pair copper cable consists of two copper wires twisted together to carry communications signal efficiently. The twisting is done to minimize the electromagnetic interference (EMI). EMI occurs due to the ability of electrical signals to create magnetic signals and vice versa. This ability is responsible for the propagation of electromagnetic (EM) waves and much of the electrical engineering (e.g., generation of electrical power and

operation of electrical motors). It also has the undesirable effect that an EM signal is prone to interference by another EM signal crossing its path or just passing near by. Since many pairs of cable are packaged together, there is the possibility of EMI among signals of all the pairs. Twisting reverses the effect of induction by external EM signal. This helps combat the interference.

As mentioned above, a large bundle of pairs are packaged together in one protective sheath. Each pair can be shielded with the help of a plastic jacket around it. This potentially gives rise to two types of twisted pair cables, the unshielded twisted pair (UTP) and the shielded twisted pair (STP). Telephone cable is an example of UTP. It is supposed to be working in a low-interference environment requiring relatively less bandwidth. However, when bandwidth requirements or interference increases, then either STP or an enhanced category of twisted pair is used. The cable thickness varies from 0.4 to 0.9 mm. It is mostly used in Local Area Networks (LANs) and telephone subscriber loops. It has also been used for long distance trunk systems (up to 4 Mbps). Figure 4-4 shows twists in a UTP.

The American National Standards Institute (ANSI) has defined five categories of UTP that vary in terms of average twist length. These categories, numbered 1 through 5, are part of four cable types that are specified as TIA 568-A cabling system The other three types are: Shielded Twisted Pair (STP), Multimode Optical Fiber and 50-Ohm Co-axial cable. For local area networks, UTP has long dominated the scene and is expected to continue doing so. Category 5 UTP is used for high speed LANs, such as 100 Mbps Ethernet and 155 Mbps B-ISDN PHY connections. Category 3 that can support data rates up to 16 Mbps is most commonly used in 10 Mbps Ethernet and Token Ring (4 or 16 Mbps). Category 3 and category 5 use a copper conductor with a wire gauge of 22-24 but differ in twist lengths. In general, the shorter the twist length, the better will be the EMI rejection. The cost and bandwidth increase with the decreasing twist length. By using balanced transmissions, additive noise can be eliminated because a noise pulse will affect both the lines (positive and negative) equally. Attenuation for twisted pair cable used in telephone loops is about 1 dB/km for voice frequency. The allowable loss for voice signal from telephone to exchange is within 6 dB, restricting the maximum loop length to about 6 km. For digital point-to-point connections, data rates of up to a few Mbps are achievable.

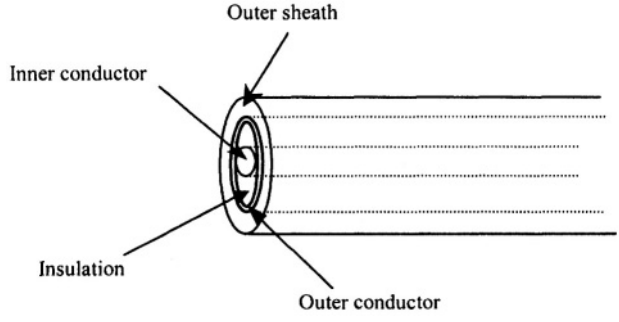


Figure 4-5. Co-axial cable.

4.3.2. Co-axial Cable

The co-axial cable consists of a hollow outer cylindrical conductor, and an inner, wire conductor. It is well known that most of the energy in a thick conductor propagates close to the skin of the conductor. In other words, the core of the conductor is not used efficiently in radio transmission. Thanks to the cabling technology, it is possible to manufacture cables with hollowness. This allows inserting another, thinner, conductor for completing the return path of signal voltage.

Coaxial cable has been the mainstay of long distance telephone and cable television for a long time. It has the capacity of a large number of voice signals making it the cable of choice for inter-exchange trunks. It has also been quite popular in Local Area Networks (LAN)s. It is specified in some local area standards, such as, IEEE803.2 (10-Base-2 and 10-Base-5). Co-ax is still the cable for cable TV and runs to most of the modern homes these days. It is much less susceptible to interference and crosstalk than twisted pair. It provides much higher data rates, suitable for transmission to longer distances. Many cable TV operators offer broadband Internet access through the installed coax. Coaxial cables have slowly been phased out of LANs and replaced by UTP that provides all the needed bandwidth with much less cost. Even though coaxial cables are sturdy, their appearance is usually noticeable and interferes with the decor of the room in which they are installed.

4.3.3. Optical Fiber Cable (OFC)

The OFC is a very thin glass or plastic cable consisting of three concentric cylinders: the innermost core, the outer-most jacket and the cladding in-between. Typical diameters vary from a few μm to about a 100 μm. Figure 4-6 shows one of the several types of optical fiber cable.

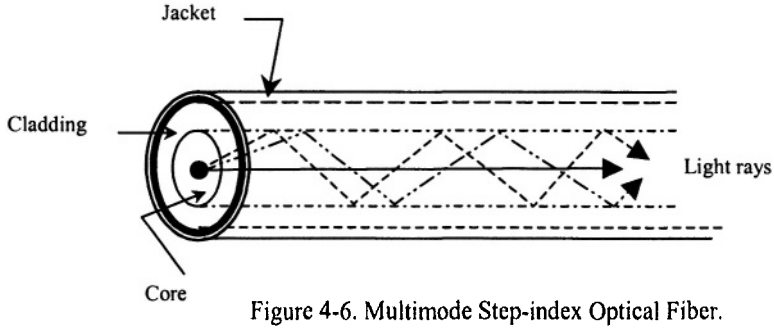


Figure 4-6. Multimode Step-index Optical Fiber.

The jacket is made of plastic or some other insulating material to protect the core and cladding. Signal travels in the core. The optical properties of the cladding are different from those of the core. Eve since The OFC has taken the telecommunications market by storm. Much of the coaxial cable in long distance telephone networks in advanced countries has already been replaced by OFC. Cables have been laid down through the oceans among many countries in the world. Due to bandwidth in access of 1 GHz, OFC is suitable for broadband and long distance communications needs. However, the transmitter and receiver technologies are still dominated by electronic components. Consequently, the real benefit of OFC to the end-user will be pouring in when the electronic devices with comparable speeds are commercially available.

4.4. The Wireless Media

Air and space provide a useful and rather popular medium of communications. The transmission characteristics of air/space make it possible to divide the feasible spectrum into many ‘frequency windows’. Bandwidth is allocated by the regulating agencies, such as Federal Communications Commission (FCC) in USA for various applications. The allocation of a frequency window or windows depends upon factors, such as the data rate needed by the applications using the medium, types of interference at a particular frequency, geographic scope of the applications

(indoor, outdoor LoS, outdoor non-LoS, mobile, satellite) and the number of users.

A large number of different bandwidth spectrums have been allocated including remote-controlled toys, home electronics, wireless phones, mobile phones, point-to-point fixed microwave communications, broadband wireless data networks and communications via satellite. The use of many of these bandwidths requires licensing from the regulating agencies. There are some communications band, however, that do not require licensing. These are reserved for industry application and research. One of such bands is the ISM band. ISM stands for Industrial, Scientific and Medical band. It is to be used for research and development in these areas. It has already resulted in a number of products in communications networking, mainly for wireless local area networks. The ISM band around 2.4 GHz is allocated in many countries and is paving the way for standards in WLANs. In fact, the IEEE WLAN standard has been specified for the ISM band.

4.4.1. Characteristics

Air/space results in a number of impairments. These impairments become more significant when the user is mobile. Path loss is one such impairment that results from natural spreading and attenuation of radio wave in the air. The loss is a function of frequency and distance. It increases with frequency and increases to the square of distances or even worse. It is significant especially for microwave frequencies that range from 2 to 40 GHz. Interference from other sources, too, affects transmission as all users use the same air. This is especially the case for non-licensed bands. Multipath is still another impairment responsible for signal deterioration in wireless communication. This phenomenon relates to the receiver being able to receive multiple replicas of the same signal. The multiple copies are a direct result of the signal being reflected from obstructions greater than the wavelength of the radio wave.

4.4.2. Examples of Wireless Bands

1. Bands around 800 and 900 MHz have been allocated to a variety of applications for subscription based and government services. The first two generations of cellular phones in USA and the GSM in Europe use these bands.
2. 1900 MHz is used in the 2^{nd} and 3^{rd} generation wireless phones, also called PCS (personal communications services).
3. 2.4 GHz, the ISM band is being used for Wireless LANs.
4. 4/6 and 12/14 GHz bands are used for geo-synchronous communications satellites.
5. 28 GHz has recently been allocated to broadband Internet access.

4.5. Physical Layer Protocol Example: EIA-232-F

EIA-232 is perhaps the most widely used Physical layer standard. Proposed by the Electronics Industry Association (EIA) the standard was originally called RS-232. It has also been called as serial port because bit-by-bit transmission of data. The latest revision is called EIA-232F. This standard was specified before computer networking itself. Its main use was considered to transfer data between a mainframe computer and a terminal through a telephone modem (called voice-grade modem). Figure 4-7 shows such a configuration.

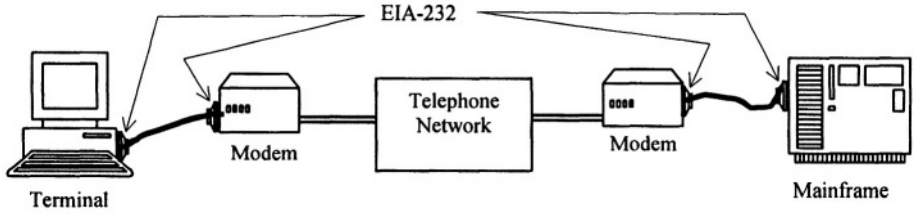


Figure 4-7. The typical EIA-232 usage in the 1960's and 1970's. The mainframe computer could be replaced by another terminal or data processing equipment.

Other assumptions implicit in the specifications were:

1. Low data transfer rate of the order of a few hundreds bps to a few thousands bps. Even the latest revisions are suitable for less than 20 kbps operation. For very short cable, the data rates could be further increased.
2. Short length cables (less than 5 meters) between the modem and the terminal (or the mainframe).
3. Negative bipolar signaling emulating the current-based logic in the earlier (1950s) Teletype machines.

The standard brought with it some terminology that was later followed by some wide area networks. Here are some of the terms.

DTE (Data terminal equipment) is the mainframe computer or a terminal connected to the mainframe or to another terminal. A more general definition of a DTE would be the source and/or sink of data. This could be a personal computer or any other terminal type.

DCE (Data circuit-terminating equipment) is the modem device. If a modem were not used, then DCE would be any device that provides the user network interface (UNI).

Due to cooperation among various international standardization organizations, popular standards specified by one agency are usually adopted by others without any modifications. EIA-232 is such an example. The mechanical characteristics are adopted by ISO in the standard ISO 2110, the electrical, functional and procedural characteristics are specified in ITU standards V.28, V.24 and V.24 respectively. Following is a brief account of the specifications of EIA-232. The description given below is not all-encompassing in any sense and is meant to convey a general understanding of the standard.

4.5.1. Mechanical Characteristics

EIA 232 defines a 25-pin connector referred to as DB-25. Although 22 of the 25 pins have been allocated specific functions, it is rarely the case that all the 22 pins are used during a connection. In fact, some companies have defined their own connectors that vary slightly from one another. Prominent among these has been the IBM PC serial port connector using 9 pins (DB-9 connector). Figure 4-8 gives an account of pin definitions for DB25 and DB-9 on the DTE side. Figure 4-8 is simply a functional diagram and is not drawn according to the true scale or proportions of the actual standard.

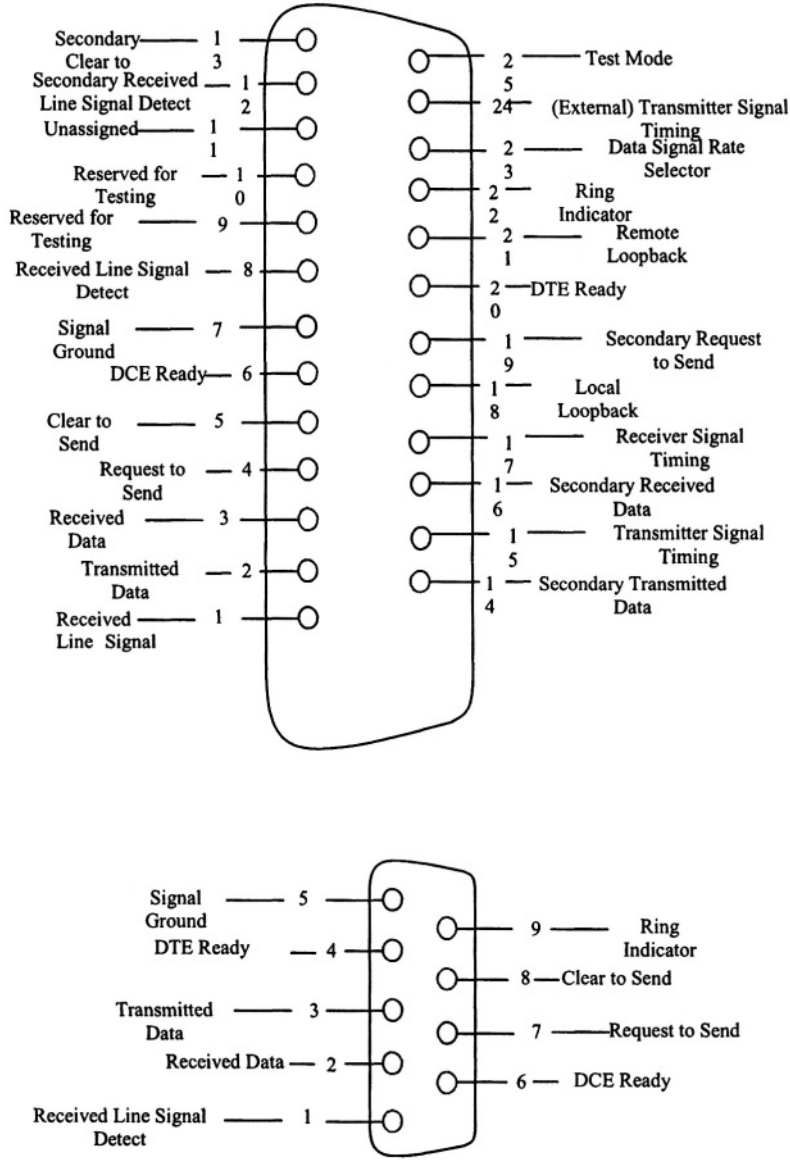
Looking from the DCE side, some of the pin functions will be reversed. Table 4-1 shows this reversal.

Table 4-1 Complementary functions of the DTE and DCE sides

Pin number	DTE Function	Equivalent DCE Function
2	Transmitted Data	Received Data
3	Received Data	Transmitted Data
4	Request to Send	Clear to Send
5	Clear to Send	Request to Send
14	Secondary Transmitted Data	Secondary Received Data
16	Secondary Received Data	Secondary Transmitted Data
19	Secondary Request to Send	Secondary Clear to Send
13	Secondary Clear to Send	Secondary Request to Send

The connector on the DTE side has a male connector while the one on the DCE side uses a female connector.

Figure 4-8. Pin designations for EIA-232 connectors from the DTE side.



4.5.2. Electrical Characteristics

The standard specifies the use of digital signaling (bipolar mark and space) with a voltage below -3 volts representing binary 1 and a voltage above +3 volts represents binary 0. The peak voltage levels are restricted within ±25 volts. For control signals, the circuit state of logical '1' is used to indicate an inactive condition while a transition to logical '0' indicates an active condition. For data signals, the transition of signal pulse on the *signal element timing circuit* is used as transition to the next data element on the *data circuits.* The tolerances of timing vary according to the data rate, from less than 1 ms for very low data rates data rates to less than 5 μsec for data rates up to 8 kbps.

4.5.3. Functional Characteristics

The functional characteristics of EIA-232 follow easily from the pin definitions. Each pin has an associated circuit that performs a specific task. Over the years there have been improvements and additions over the original RS-232. There are still some pins that can be used to expand the functions in future. We will discuss various operation modes under the 'Procedures' in the next section. Here, we simply categorize the functions into the following: Call Setup and Handshaking, Data Transfer, Timing, Control, Testing. Table 4-2 shows the circuits under each category.

Table 4-2 Signal categories of EIA-232

Signal Type	Circuits	Pin Numbers	General Function
Data Transfer	Received Data, Transmitted Data, Secondary Received Data, Secondary Transmitted Data, Signal Ground,	3, 2, 14, 16 and 7	Transmit and receive data with respect to the ground voltage level
Timing	Transmitter Signal Element Timing, Receiver Signal Element Timing, (External) Transmitter Signal Element Timing	15, 17, 24	Provide internal and external clocking information
Control	Ring Indicator, Data Signal Rate Selector, Test Mode, DTE Ready, DCE Ready	22, 23, 25, 20, 6	Control various modes of operation
Testing	Local Loopback, Remote Loopback, Pins 9 and 10	18, 21, 9, 10	Loopback and future testing
Call Setup and Handshaking	Request to Send, Clear to Send, Secondary Request to Send, Secondary Clear to Send Received Line Signal Detector, Secondary Received Line Signal Detector, Ring Indicator	4, 5, 19, 13, 8, 12	Setup call and allow handshaking between communicating DTEs via the DCEs.

4.5.4. Procedural Characteristics

The standard specifies functions of each pin as well as the sequence of their use for different calling modes and phases. These could be the call setup, data transfer using primary or secondary circuits, full-duplex, half-duplex and simplex transfer, synchronous or asynchronous modes using internal or external timing, and test modes, for local and remote loopback. There is a way of utilizing EIA-232 interface without the intervening modems to connect two DTEs directly. In this section, we will take some examples of the procedural characteristics of the EIA-232 standard. Let's start with the definitions of some terms. Suppose DTE A and is connected to a network via DCE A through the EIA-232 interface. Then, by 'direction' of data transfer, we mean either from the DTE A to DCE A or from DCE A to DTE A.

Duplexity of data transfer pertains to the direction of data flow.

Full-duplex data transfer takes place in both directions simultaneously.

Half-duplex data transfer takes place in both directions with turns.

Simplex data transfer takes place in one direction only. Two simplex circuits used to transfer data in opposite directions make up one full-duplex circuit.

Synchronization (Timing) refers to the extraction of periodic timing information from a signal and aligning the data signal transmission and detection to this timing information. All digital communication requires synchronization in order for the receiver to interpret the data signal in the same way as the transmitter. Typically, a separate signal is used just for this purpose. This signal is called **clock.** The clock consists of an alternating pulse stream. Clock could be either generated locally by transmitter and receiver, or by either one, or by a source external to both. The maximum data rate is equal to the number of clock pulses per second. Lower data rates are possible by defining data pulse equal in duration to more than one clock pulses (or *ticks).* In some systems, the clock information is imbedded inside the data signals. We have already seen an example of this in Manchester codes. Following are some examples of procedural characteristics of the EIA 232-F. we assume that both the DTE and DCE have the DTE/DCE Ready circuits activated wherever applicable.

4.5.4.1. Call setup for full-duplex connection

When DTE has data to send to its DCE, it activates the Request to Send line. DCE, if ready to receive data, activates Clear to Send. On detecting this signal, the DTE can start sending data on Transmitted Data. If a DTE has data to send to another DTE over a telephone network then the first data item is the phone number of the remote DTE. This number is routed to the remote DCE via the PSTN. On receiving this signal, the remote DCE activates the Ring Indicator signal to announce an incoming call. If the remote DTE replies

with a DTE Ready signal, then the remote modem sends a signal to the call-originating end. This signal appears on the Received Line Signal Detect. Then the originating modem sends a carrier signal to the remote modem. The remote modem, on receiving this carrier signal activates its own Received Line Signal Detect. This completes the *handshake*. At this point, both DTEs are free to exchange data.

4.5.4.2. Call setup for half-duplex connection

When the originating DTE request is for a half-duplex connection with another DTE, it inhibits the Received Data circuit at the time of call request. The rest is the same. During data exchange in a half-duplex mode, if the receiving (remote) DTE or originating DCE has to send some data, such as flow or error control information, it cannot be done by sending signal over Received Data circuit. In such a situation, the secondary data circuits are used. The secondary circuits are usually very low speed circuits for feedback-like information.

4.5.4.3. Loopback Testing

This feature is provided to help locate faults in the network connections in case of communications breakage. If a modem supports the loopback test, sometimes it is possible to locate the fault. Two circuits are provided in the EIA-232 F standard for loopback testing: one for each, local loopback and remote loopback. The main function of the both is the same, that is, to loopback the received data by internally connecting the receiver to the transmitter of the DCE. In the remote loopback function, this is done only at the remote DCE; while in the local loopback, such is the case in the local DCE alone. Figure 4-9 shows the network connections for the remote loopback test setup.

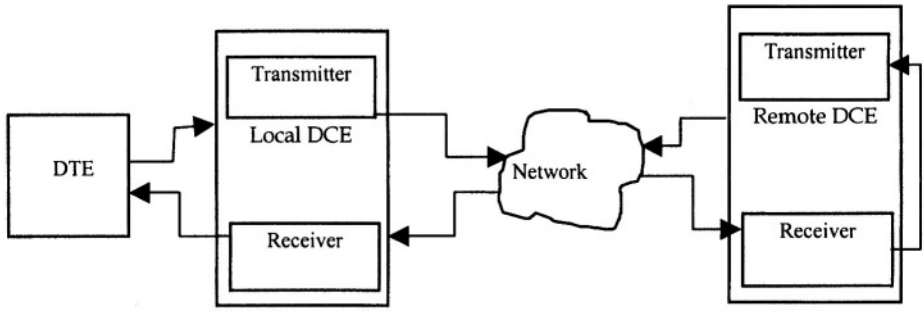


Figure 4-9. Remote loopback test setup. For local loopback testing, the network and the remote DCE are not needed and the local DCE has its Receiver output connected to its Transmitter input.

4.5.4.4. The NULL Modem

The NULL modem is the term used to describe the direct interconnection of two DTEs via EIA-232 connectors without the DCEs on either side. DCEs may be needed only when the communicating DTEs are at a distance greater than the maximum cable length allowed. When the two DTEs are in close proximity from each other, the EIA-232 can be used to connect them directly. Since the DTE side of EIA-232 uses a male connector, a box with both sides having female connector could emulate a NULL modem. Sometimes, such a box is called a *switch box.* Alternatively, the complementary circuits from the two EIA-232 can be wired directly without actually having a 'box' in between. Figure 4-10 below shows the interconnection of two DTE side EIA-232 devices via a NULL modem.

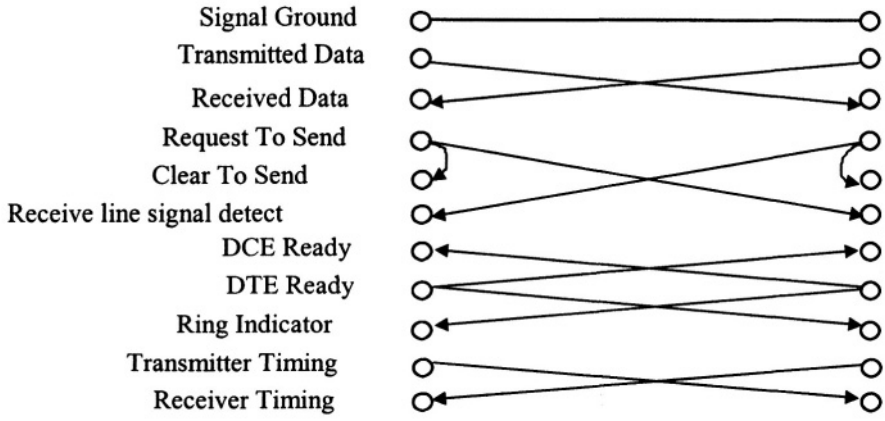


Figure 4-10. NULL MODEM interconnection.

In this figure, the Transmitted Data of one DTE is connected with the Receive Data of the other DTE and vice versa. By doing so, the DTE that is the receiver of data receives it on the same circuit that it would receive if it were connected to a DCE. One interesting connection is of the Request to Send circuit. This is circuit is used by DTE in a DTE/DCE interface to check if the DCE is ready for communication at this time. The DCE replies by activating Clear to Send. On sensing the active condition on Clear to Send circuit, the DTE can proceed with data transmission. In the case of a Null Modem, there is no DCE. However, by connecting Request to Send circuit to own Clear to Send circuit, a DTE simulates a response from DTE as soon as it activates Request to Send. Note that Request to Send is also connected to the other DTE's Receive line signal detect. By doing this, the remote DTE receives the same signal without a DCE as it would if it were connected to a DCE. In this way, the two DCE functions are emulated just by complementary interconnection of the EIA-232 DTE-side interfaces.

4.5.5. PHY for IEEE Wireless Local Area Network

With the above discussion on a popular but *older* PHY interface, we will move on to another example of a *later* physical layer protocol. The next example is from the IEEE802.11 standard for wireless local area networks (WLAN). In this protocol, the transmission characteristics play a greater role than the EIA-232. The WLANs are LANs without wires. Such networks provide the flexibilities of mobility and topology. Coupled with an ease of installation, WLAN are taking over the local network market rather swiftly. The PHY for a wireless data network does not require many functions of a

fixed network PHY. For example, there is no connector. In some cases, such as infrastructure networks, one of the network components, called an access point, may be connected to fixed network. In such cases, the connector is specified by the fixed network PHY. For example, if the access point is connected to an Ethernet, then it will have a socket for the RJ45 connector for UTP, a BNC terminal for this coaxial Ethernet, or an attachment unit interface (AUI) for the thick Ethernet. More than one of such socket types may be provided by many manufacturers of access points. Even though we have already made use of some terminology of WLANs without a definition or description, such as the access point, we will take care of this issue next.

4.5.6. WLAN Types

There are two types of WLANs, the infrastructure WLANs and the independent WLANs. In an infrastructure WLAN, the wireless terminals communicate via a central point called the *access point.* The access point is

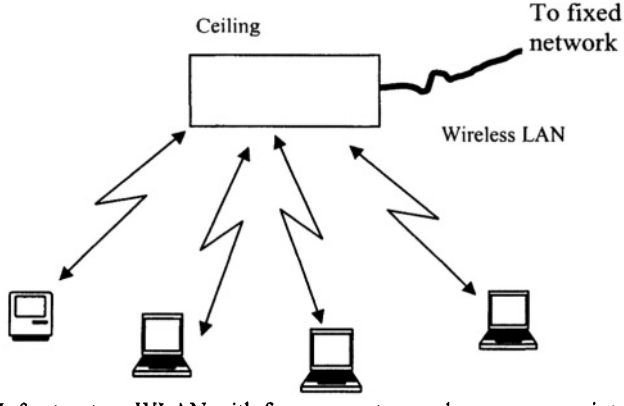


Figure 4-11. Infrastructure WLAN with four computers and one access point.

connected to a wired network. Figure 4-11 shows an example layout for an infrastructure WLAN.

The infrastructure WLANs are easy to expand by using either more than one access points, or simply by using extension points. An *extension point* is just like an access point except that it is not connected to the wired network.

A second mechanism of configuring WLANs is by having all wireless terminals with the capability of direct communications with another terminals. This eliminates the need of an access point. Such networks are called independent or ad hoc networks. *Peer networking* is another term used for

such a layout. An example of a layout of an ad hoc network is shown in Figure 4-12.

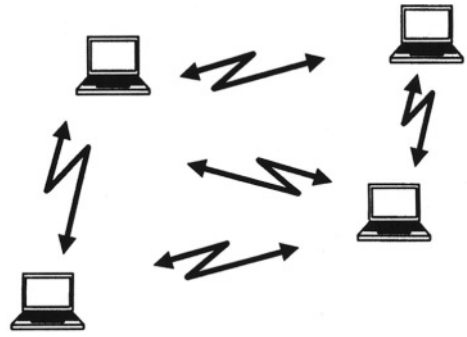


Figure 4-12. Ad hoc WLAN connecting four terminals.

In addition to the two types of layouts, there are three different mechanisms used as communication media in WLANs. Two PHYs use electromagnetic (EM) radiation and the third uses infrared light. The most popularly used EM spectrum for WLANs is license-free and has been allocated for research in industrial, scientific and medical areas. Because of this reason, it is called the ISM band (ISM = Industrial, Scientific and Medical). Various parts of the EM spectrum have been allocated for ISM research in parts of world. The most popular is the one around 2.4 GHz. It's usually is referred to as the 2.4 GHz band. There are no specifications for the physical shape of the antennas and there is no connector. Consequently, the PHY specifications consist of radiation power limitations and a set of services (functions) along with procedures. One part of services is reserved for physical medium dependent (PMD) part of the PHY and another for convergence to MAC layer, called physical layer convergence procedures (PLCP). The IEEE802.11 architecture also specifies a third part of PHY functions for the layer management. We will not consider the layer management functions and will focus only on the remaining two.

4.5.7. Frequency Hopping Spread Spectrum (FH-SS) for 2.4 GHz Specification

Spread spectrum (SS) is a mechanism of sharing channel bandwidth for broadcast type access mechanisms (called multiple access). However, for the IEEE WLAN, it is not used as a multiple access mechanism. Instead, it is used as an interference combating mechanism. This is necessary because the network bandwidth is unlicensed and could be used by any device in an

environment. Let B_S be the signal bandwidth and B_C be the channel bandwidth. Then, in order to use spread spectrum techniques, we must have $B_C >> B_S$. In SS techniques, the signal bandwidth is spread to close to B_C while keeping the total signal energy constant. By doing so, the signal energy carried per bandwidth is reduced a great deal. Consequently, a signal with very low power density (power per Hz) is used for transmission. This is shown in Figure 4-13.The spreading is done in such a way that only intended receiver should be able to detect the signal.

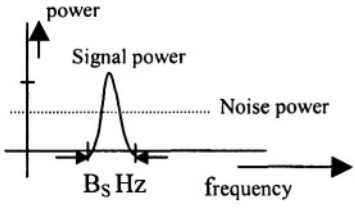


Figure 4-13(a). Signal power spectrum before spreading.

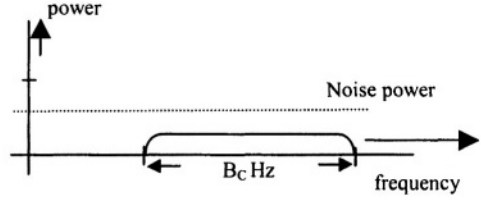


Figure 4-13(b). Signal power spectrum after spreading.

In frequency hopping, this is done by dividing the wide channel bandwidth B_C into many narrow channels, each with a bandwidth close to B_S. The signal is transmitted using one of the narrowband channels for a short duration after which this channel is replaced by another narrowband channel, and so on. Thus, by changing the narrowband channel in a pattern known only to the intended transceiver (transmitter/receiver) pair, a frequency-hopping pattern is implemented. This hopping is called fast frequency hopping if the channel hopping speed is faster than the bit rate. Contrarily, if more than one bit are transmitted per narrowband channel, then we call it as slow frequency hopping spread spectrum (SS-SFH). The IEEE WLAN specifies a slow frequency hopping mechanism.

4.5.7.1. PLCP for frequency hopping

The physical layer convergence procedures define how the MAC layer PDU will be mapped into a physical layer packet suitable for the physical medium. This is done by defining a PLCP PDU that carried the MAC PDU as part of it. Figure 4-14 shows the PLCP PDU specified for this type of PHY.

PLCP preamble	PLCP header	MAC PDU modified to minimize DC bias and randomize MAC data

Figure 4-14. The PLCP PDU as defined in the frequency hopping spread spectrum PHY of the IEEE WLAN.

Each part of the PLCP PDU provides a set of functions. Table 4-3 shows a list of functions.

Table 4-3. Functions provided by various fields of PLCP PDU

PLCP-PDU Field	Functions
Preamble	1. Synchronization with the received packet timing. 2. Select an antenna if diversity is used. 3. Reach a steady state frequency offset correction
Header	1. The length of MAC PDU in number of octets. 2. Indicate the bit rate of the PLCP 3. Header error check using CRC
Modified MAC PDU	1. MAC data scrambling (called data whitening). 2. DC bias removal.

4.5.7.2. The PMD for Frequency Hopping Spread Spectrum

The physical medium dependent (PMD) part specifies the various parameters of the frequency hopping system, the data rates, modulation, and receiver parameters. Table 4-4 shows a list of functions specified under each category. The list is not all-inclusive.

Table 4-4. Functions of Physical Medium Dependent (PMD) layer of FH-SS

Function category	Functions specified
Frequency hopping system	1. Usable channel bandwidth 2. Number of frequency channels. 3. Operating frequency range. 4. Hop patterns 5. Hop rate
Data rate	1. Data rates 2. Symbol rate 3. Tolerances
Modulation	1. Modulation type 2. Representation of logical information in modulation signals 3. Modulation tolerances
Receiver parameters	1. Input signal range. 2. Receiver sensitivity. 3. Tolerances. 4. Radiation parameters. 5. Operating temperature range.

The exact specifications in the table have been excluded intentionally to avoid getting out of scope.

4.5.8. Direct Sequence Spread Spectrum (DS-SS) for 2.4 GHz Specification

Direct sequence (DS) is another technique used to spread the signal bandwidth over the whole channel bandwidth. In this technique, each signal bit is divided (in duration) into *smaller bits.* The smaller bits a called chip. The effect of dividing a bit into many chips is that the resulting signal needs correspondingly higher bandwidth. Each communicating pair of stations uses a different, unique, chip pattern. This chip pattern is called *pseudo-noise code* (PN-code). The PN-code is needed to detect a DS-SS signal by the receiver. This allows for many stations to transmit and receive signals simultaneously. For IEEE WLAN, however, that is not the objective. The objective is to make the receiver detect its signal even if there is interference in the same bandwidth. Figure 4-15 shows a pictorial of DS-SS mechanism.

Just like the FH-SS PHY, the specifications in this type of PHY provide functions and procedures to implement a convergence from MAC to PHY using a PLCP PDU, and specifications for PMD to specify DS-SS parameters, data rates and receiver functions. Due to similarity in the type of functions specified, we will not repeat a list of functions for DS-SS PHY.

Caution must be taken in real life, however. Various PHYs specified in IEEE WLAN standard are not inter-operable. Therefore, care must be taken in purchasing WLAN equipment that must interoperate. A terminal equipped with FH-SS can function with another terminal with the same PHY. The same applies to wireless terminals using DS-SS. Consequently; the conformance to IEEE 802.11 PHY is not enough for two wireless LAN terminals to

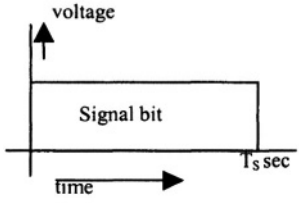


Figure 4-15(a). Signal bit before spreading. T_S = bit duration

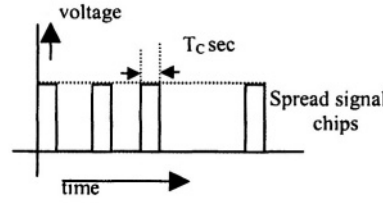


Figure 4-15(b). Signal bit after spreading. T_C= chip duration

interoperate. They must be using the same PHY protocol.

4.5.9. Infrared PHY for IEEE WLAN

The infrared (IR) spectrum of light is located just below the visible light. The wavelength range for IR is 850 to 950×10^{-9} meters (or 850 to 950 nm, n stands for nano). Many natural and man-made processes emit light in this frequency range. IR has been successfully used for remote control that makes use of a light emitting diode (LED) and a detector. Due to extremely high frequency, sufficient bandwidth is available for high-speed communications at this range. In fact, the bandwidth is so enormous that there is no need of bandwidth efficient modulation. Instead, simple on-off mechanism of an LED would make up that transmitter.

Generally, IR devices use line-of-sight (LoS) mode of communications. IEEE WLAN standard proposes a different mechanism, called the diffused IR. In diffused infrared, the infrared radiation is spread in a general (operating) area by reflecting it from a course surface (*diffusion mechanism*). In this way, terminals using IR can move around in the 'lighted' area and do not have to maintain a LoS.

The PLCP for IR have a similar frame structure as the other two types. The difference is in the function parameters and procedures. For example, the preamble of IR-PLCP provides the synchronization and so do

the PLCPs of FH-SS and DS-SS. However, the exact format is different for all three as shown in Table 4-5.

Table 4-5. Variation in Sync field for different PHY types in IEEE WLAN

PHY type	Length of Synchronization field
FH-SS	80
DS-SS	144
IR	57-73 slots (slot = 250 nanoseconds)

It is the PMD where even the functions specified could be different from one PHY to another. Accordingly, many functions specified at the IR PHY PMD relate to optical nature of the IR devices.

4.6. The Integrated Services Digital Network (ISDN) PHY

In the above two examples of PHY, we saw the mechanical characteristics playing an important role in the former and almost no role in the second example. Also, there is a frame structure specified for the IEEE WLAN PHY, while no such thing for the EIA-232F. It may be noted that there were some redundancy in the mechanical specification of EIA-232 that led to the 9-pin connector. The redundancy was also in the form of associating a circuit to every pin. In reality it is possible to use pin combinations to define and implement additional functions. By combining binary logic, n pins can have up to 2^n functions. An example of a use of such capability is the ISDN connector defined in ISO8877 for twisted pair. In this PHY specification, the transmitting and receiving circuits implement some control functions as well. Total of 8 pins have more functions than EIA-232-F or DB-25.

The broadband ISDN has very complex Physical layer because of high-speed communications and a variety of networks that could be interconnected using B-ISDN. Just like the IEEE WLAN, it has a planar protocol architecture with user, control and management planes for each layer. Figure 4-16 shows the user plane functions for PHY.

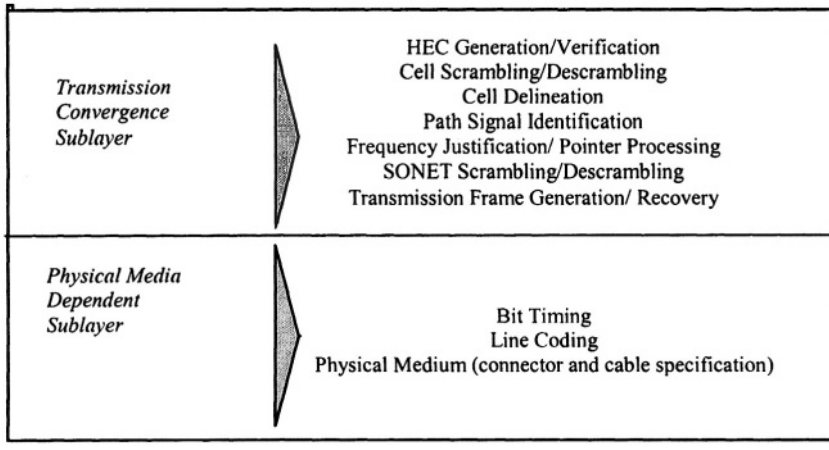


Figure 4-16. B-ISDN PHY functions.

Here's a glossary of the terms used in Figure 4-16.

HEC (Header Error Checksum) is a mechanism of generation of parity check field for error detection in the header.

Cell scrambling is a mechanism of randomizing the data in a cell (layer 2 PDU in B-ISDN, called ATM)

Cell delineation is the procedure of identifying the cell boundaries. HEC field is used for this purpose.

Path signal identification relates to the identification of certain synchronization function in the PHY frame.

Frequency justification/ Pointer processing related to a field called *pointer* in the PHY frame used to load/locate higher layer cells within the PHY frame.

SONET scrambling is the scrambling at the PHY layer transport mechanism called SONET (Synchronous Optical Network).

As in IEEE WLAN, the separation between the medium dependent and convergence functions allows to be defined convergence functions for various physical media. Consequently, there is a diverse base of standards for B-ISDN PHY from the optical to unshielded twisted pair (UTP) cables.

4.7. Review Questions

1: Define the *Channel bandwidth?*
2: List the following in the increasing order of proximity to the electromagnetic interference? Air, co-axial cable, optical fiber cable, twisted pair cable.
3: In a null MODEM why the Request To Send (RTS) circuit of a DTE is connected to its own Clear to Send (CTS)?
4: Give one example of (i) DTE, (ii) DCE
5: Is it possible to define more than 2^n characters using n bits? Explain.
6: What are pros and cons of installing Unicode on all computers?
7: What, in your view, is the optimum solution for using the Unicode then?
8: Why is the bandwidth of infrared higher than the other two WLAN PHYs?
9: We have discussed two types of WLAN configurations defined in IEEE standard: infrastructure and ad hoc. Give an example of a situation in which each one is preferable?
10: There are three types of PHYs defined in IEEE WLAN standard. Do they interoperate with one another?
11: What is the disadvantage of lack of interoperability of PHYs in Q10?

5. Data Link Control Layer Functions and Procedures

Right above the physical layer, the data link control layer adds logical functions to the data exchanged between two PHYs. DLC protocols provide functions and procedures to implement a point-to-point logical connection for reliable exchange of information. In a network path consisting of many links between two communicating nodes, problem on any link could slow down or even halt all communications. A link problem could occur in many ways. Each PDU may require proper addressing mechanism understood by the next receiving station on the link. Addressing could take many forms, such as, a station-specific address, a group-specific address, or even an address for links. Then, there is a need to have a way of identifying the beginning and end of a PDU so that the received information is interpreted in the same way as it was meant to be. Each link has a specific amount of bandwidth restraining the maximum signal bandwidth. In digital communications, bandwidth translates into bit rate. Thus, each link has a maximum allowable bit rate, the link *capacity.* If the bit rate of information coming to a node temporarily increases beyond the outgoing link capacity then data can be stored in buffers. However, if this happens a little too often, then the buffers can overflow and data may be lost. Signals may get lost or be deteriorated due to the channel impairments discussed in earlier chapter. This may result in receiver interpreting a signal with wrong logical equivalent (e.g., a '1' as a '0') or miss the signal altogether.

In summary, when information is exchanged between two nodes across a link, packetization and transmission are not sufficient. The information has to be reliable and exchanged with the maximum exchange rate for a given link capacity and buffer sizes. This makes the job of DLC layer quite complex. However, the functions required by a typical DLC can be easily spotted out. This makes DLC layer one of the easiest to understand for a beginner in data communications. This is a layer that provides functions and procedures for an efficient and reliable exchange of PDUs between two nodes connected directly without an intervening node.

5.1. Data Link Layer Functions

This section is devoted to the functions provided by a typical data link layer protocol. Actual layer 2 protocols may have less or more functions than these, depending on factors such as physical layer reliability, higher layer protocols or even the application needs. A highly reliable PHY will ensure the delivery of signals with integrity reducing error-control duties of DLC. A higher layer (e.g., Network layer) might control the flow of PDUs to DLC so that the buffers rarely or never overflow. Lastly, an application may not require a very high accuracy from the transmitted data, such as in human

D-PDU

Several terms have been used in literature to describe a DLC protocol data unit (D-PDU). Packet is the most general term. Other than that, it's called frame, block, data unit etc. In some protocols, such as ATM, it's called a cell. We will keep the tradition of not being consistent and use various terms alternatively!

speech transmission. Following are the general functions provided by data link layer.

5.1.1. Synchronization

Since the information at the DLC layer is treated as blocks of bits, it is important for the transmitter and the receiver to identify the beginning and end of data blocks. Only then, the two DLC's could interpret each other correctly. The process of locating and identifying the beginning (and ending) of data (block) is called synchronization. Synchronization helps identify the DLC frame by defining its beginning and/or ending bit patterns. Such transmission is called as *synchronous transmission.* Another way of providing synchronization is by transmission of data character by character. Character codes used in this case are the ones defined at another layer to represent character set in bit strings. This character-by-character transmission is called *asynchronous transmission.* We will look into their detail in a later section.

5.1.2. Addressing Modes

In a network, a station could be directly connected to a number of other stations. There could be a separate link between each pair of stations, or

many stations could be sharing a link such as in a LAN. In addition, each station could be identifiable as a destination, or one of a group of destination stations. Sometimes, link layer identification is required for each station. The DLC protocols must provide a way to identify a recipient or a group of recipients of a PDU.

5.1.3. Connection setup and termination

Before data is transmitted by the transmitting station, the receiving station may require to be aware of the incoming data. This is necessary because the receiver of data has to allocate network, processor and memory resources to the received data. Connection setup allows the sending station to inform the receiving station of its wish to send data. The receiving station could accept, reject or delay the request. It could also negotiate the terms of data exchange. A successful exchange of messages to setup a connection is sometimes called *handshaking*. Also, once the sender of data has completed sending data, it must inform the receiver so that the allocated resources may be released and be used by other purpose. This requires the function of connection termination.

5.1.4. Error Control

Error control is defined as the capability of detecting and recovering from errors in the received data. In data transmission, detection and recovery are usually two different processes. In *error detection* mechanisms, the DLC layer at the sending end processes the data so that if there are any errors introduced during transmission they could be detected by processing the data again at the receiving end. In *error recovery* mechanisms, the data unit (or frame, or packet) that was in error is requested again or corrected by the receiver.

5.1.5. Flow Control

Flow control is defined as (a set of) procedures implemented to stop the transmitter from overwhelming the receiver with data. The receiver may be overwhelmed with data if it does not have enough memory to store the received data. Another reason for the need for flow control could be that the receiver of data may have to relay data on a next link that might be slower than the one on which it receives data.

5.1.6. Link Control and Testing

Each DLC layer should have functions that could be invoked in order to supervise and test link condition as a part of logical circuit maintenance.

5.1.7. Multiplexing

The OSI reference model defines one layer above and/or below each layer. However, there may exist many different protocols on each layer. Some of these protocols on the same layer may want to set up a communication session simultaneously. A layer could provide this service if it has the multiplexing function. This function could exist at any layer. Such is the case with many other functions. However, multiplexing requires technologies that have grown independent of data networking. Due to its importance and complexity, we will devote a whole chapter to this topic. Therefore, we will not discuss it further in this chapter.

In the following, we will look at the procedures used to provide some of the above functions.

5.2. Synchronization

The DLC layer protocols have the job to deliver N-PDU to the network layer of a directly connected station. This service is provided through the following steps.

(a) A DLC protocol encapsulates the network layer PDU with its own header/trailer. This results in a D-PDU for transmission to the peer DLC layer.

(b) It then requests the PHY to transmit the D-PDU to the DLC on the other side of the link. The receiving DLC performs some procedures on the received D-PDU and then delivers it to the network layer above it. In order to perform the DLC functions properly, it needs to know how to interpret the received data. If it misses even a single bit of the received data, the whole packet could be interpreted incorrectly. Synchronization is therefore one of the first functions executed by the receiving DLC. The transmission of a D-PDU is sometimes classified as one of the two types with respect to synchronization mechanism, the synchronous transmission. The other type, asynchronous transmission, is mainly a physical layer mechanism. However, the two types of transmissions are do not imply certain protocol layers. We will first discuss synchronous transmission and then asynchronous transmission to complete the discussion.

5.2.1. Synchronous Transmission

Synchronous transmission is defined as transmission of data in identifiable groups or blocks. A block of data, also called a frame, consists of a number of characters or bits and has a well-defined beginning and end. The synchronization function helps receiver know the beginning and end pattern of the frame. Usually, a certain bit pattern is reserved for signaling the receiver about the beginning and/or ending of the block. Data does not have to

be stored in characters for this purpose. Neither a fixed block length is necessary. Sometimes, the synchronization bit pattern is called a *Flag.* Other than the data and Flag, some more fields are added to the block that implement protocol functions of addressing, flow control, error control and link control procedures. These *control fields* vary from protocol to protocol. A typical frame is shown in Figure 5-1.

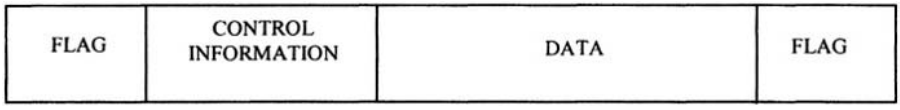


Figure 5-1. A typical location of Flag in D-PDU for synchronization

It would be advisable to use as Flag a pattern that does not occur in any other place inside a PDU. However, this cannot be guaranteed since data consists of a random sequence of '1's and '0's. If, for example, the Flag pattern occurs in the field marked as 'Data' in Figure 5-1, the receiver is likely to assume that this is the end of D-PDU. To avoid this situation, data transparency techniques are required. One technique, called *bit stuffing,* is used to modify the 'Data' field such that the Flag does not occur in it. Another technique would be to have a constant length data field.

5.2.1.1. Bit Stuffing

The objective of bit stuffing is to prevent the occurrence of the Flag pattern in PDU fields where it is not supposed to occur. Since the protocol information consists of fixed fields, one may argue that bit stuffing is not needed in the protocol information field. In practice however, bit stuffing may be done all over the PDU excluding the Flag because of its simplicity as compared to a selective bit stuffing. If the Flag pattern occurs (or is due to occur) at an unwanted place, it is altered in a known way. The following example can be used to illustrate the concept. Suppose a string of 8 alternating '1's and '0's is used as a Flag (10101010). Bit stuffing can be done by inserting a '1' whenever the following 7-bit pattern occurs in data field 1010101. If this is agreed by the transmitter and the receiver, then the bit stuffing consists of the following two-step mechanism.

(a) At the sending end, the bit stuffing mechanism monitors the PDU data field. Whenever 1010101 occurs, it adds a '1' to make it 10101011. This is done regardless of the actual value of the eightth bit in data.

(b) The bit stuffing mechanism at the receiver also monitors data continuously. Whenever the pattern 10101011 occurs, the last '1' is removed from it. In this way, the layer 3 data is delivered without any alteration.

Many physical layer protocols also allow or require transmission of data block by block. Synchronization is needed for such PHY protocols as well. Many PHY protocols, however, treat a character as the transmission block. The synchronous transmission of a single character is not practical due to its small size. The phenomenon of placing and extracting the timing information on a character-by-character basis is called 'asynchronous transmission'.

Next, we will look at the asynchronous transmission.

5.2.2. Asynchronous Transmission

In asynchronous transmission, synchronization is achieved one character at a time. Therefore, the receiver has to know the beginning and end of each character for correct interpretation of the received bit stream. Just like we need a mechanism to specify the beginning and end of a frame in synchronous transmission, we need to define the beginning and end of a character in asynchronous transmission. Characters are transmitted as bit strings defined in one of the several character sets. Some of these have been discussed in Chapter 3.

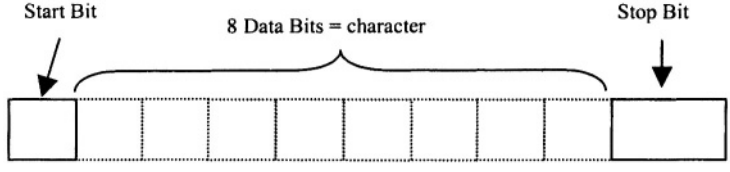


Figure 5-2. Character string for asynchronous transmission.

An example is shown in Figure 5-2.

In this figure, one pulse (or bit) is employed to signal the start of the character, called the *start bit* and a longer bit is used as a *stop bit.* In asynchronous transmission, when there is no signal on the line, a series of pulses is transmitted with polarity opposite to that of the start pulse, so that the receiver can clearly identify the beginning of a transmission. The stop bit can be of 1, 1.5 or even 2 bit duration. Systems that use asynchronous transmission always send data as characters whether the actual data is stored as characters or not. This is a limitation of asynchronous transmission especially when the received data is not being read (or played back) as soon

as it is received. There is too much overhead (in the above example if the stop bit were one bit long, then the overhead is 2/10 which is 20%).

The biggest drawback, however, of the asynchronous transmission is its incapability to being retransmitted in case a character is received in error. If there is error in a character it is not possible for the receiver to tell the transmitter which character was in error. If error propagates throughout a long string of characters then the communication must be started from the

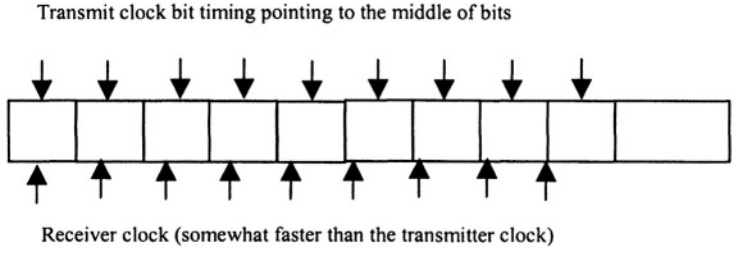


Figure 5-3. Possible scenario of transmit and receive clocks running out of sync.

beginning. This can happen if the bit timing clocks of the receiver and transmitter are not synchronized. Figure 5-3 shows such a scenario.

In this figure, the transmit clock is pointing at the middle of each bit. The receiver, if in complete synchronization with the transmit clock, would also point to the middle of each bit. Suppose that the value of the bit in the middle determines its binary type ('0' or '1'). It is shown that if the transmitter and receiver clocks are out of synchronization by a small fraction of the bit time, the character decoded may be in error after a few passages of pulses. If there is no stop bit used and another character follows this one, then the error propagates and all the remaining characters will be received in error. The figure also shows that a small amount of slip between the two clocks can be tolerated as long as there is a longer stop bit and the two clocks (Transmit and Receive) can be synchronized at the beginning of each character.

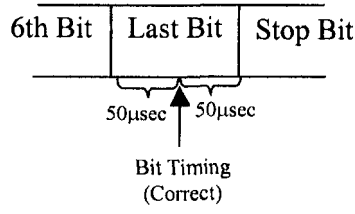
Example 5-1: Consider asynchronous transmission in the form of 7-bit characters with one start bit and 1.5 stop bits. For a 10 kbps link, let the sampling instants at the receiver be in the middle of each bit. What is the maximum tolerance of the alignment between the transmitter and receiver clocks for correct reception of character on this line?

Solution: The bit duration = 1/10,000 = 100μsec

Assume that the transmitter and receiver clocks are aligned at the beginning of the start bit. Assume from this point onward, the discrepancy between the transmitter and receiver clocks is Δ sec per bit duration.
The discrepancy of the two clocks at the middle of the start bit = Δ/2 sec
The discrepancy of the two clocks at the middle of the 1st character bit = Δ + Δ/2 sec
The discrepancy of the two clocks at the middle of the 7th character bit = 7Δ + Δ/2 sec
As long as this total discrepancy is less than half bit time (50 μsec), there is no error.
Therefore the maximum value of tolerance Δ_{max} satisfy the following equation:

$15\Delta_{max}/2 = 50\ \mu sec$
$\Delta_{max} = 20/3\ \mu sec = 6.67\ \mu sec$

See Figure below:



The above example can be used to derive a general result as follows. For a link with bit rate of *R* bps, the pulse duration is *1/R* sec. Therefore, a synchronization error occurs when the receiver clock is *1/2R* off the transmitter clock in decoding *n* pulses. Here, *n* is the number of bits in a character plus a start pulse. Assuming that the two clocks coincide at the beginning of a character with *Δ* seconds drift per pulse duration between the two clocks, the receiver will be pointing to $\pm[\Delta/2+(n-1)\Delta]$ off the center of last bit in the character. So, for correct detection,

$$|[\Delta/2+(n-1)\Delta]| < 1/2R$$

Which gives after some manipulation,

$$\Delta < \left| \frac{1}{(2n-1)R} \right|$$

Or denoting by Δ_{max} the tolerance in clock drift, we have the relation:

$$\Delta_{max} = \left| \frac{1}{(2n-1)R} \right|$$

Example 5-2: Find the synchronization overhead for transmission of a 100 Kbit file using 1010-bit DLC frames consisting of 10-bit synchronization flag at the frame beginning (no flag at frame end)?
Solution: There are 10 bits of synchronization information for every 1000 bits of data, giving an overhead of 10/1000 = 1%.
Therefore, for a file of size 100 Kbit, the total overhead is 1000 bits.

Example 5-3: What would be the synchronization overhead for the same file if it is transmitted using asynchronous transmission mechanism of Example 5-1?
Solution: For each 7 bits of data, the number of overhead bits = 2.5 (one start bit and 1.5 stop bits). In other words, the overhead = 2.5/7 = 35.7%. = 100,000 × .3571 = 35710 bits, approximately.

5.3. Connection Setup and Termination

The DLC layer protocols may be required to setup a connection before data transfer can occur. The connection setup phase usually begins with one station sending a special packet requesting a connection. The receiver of connection request replies with positive or negative response. Each type of response is sent in the form of a packet type. If the response is in the negative, the responding station may include a reason. Alternatively, the reason may be obvious from the response packet type. If the response is positive, the responding station may negotiate the data transfer parameters such as window parameters for flow control and error control mechanisms. Sometimes this mechanism is called *handshaking*. After the connection is setup, data transfer phase starts. At the end of data transfer, one or both stations can initiate connection termination.

There are potentially two ways of making a negative response: by sending a negative acknowledgement to the setup request, and by not responding. If the receiver chooses not to respond, then a connection setup timer at the sending station could be used to interpret the response.

5.4. Addressing

Since a computer can be directly attached to a large number of computers or terminals, there is a need to discern among different computers even on a point-to-point logical connection. Direct connection does not have to be direct physical connection; it could be a logical direct connection such as in Local Area Networks. In a LAN using Bus or Ring topology, all computers are connected to the same cable, as shown in Figure 5-4. In this figure five computers share a cable in a LAN. The LAN protocol allows them to be treated as if they are all connected directly to one another. Thus, the left most computer is directly connected to rightmost computer and there can be a DLC layer between the two.

Therefore, the meaning of 'directly connected' could be misinterpreted sometimes. Following is one way to describe being directly connected. *Two computers on a network are directly connected if they can exchanged data (packets) such that no intervening station alters the DL-PDU.* Note that an intervening station can physically process data. An example of such processing is a repeater whose job is to deal with individual bits and regenerate them on a long cable (Physical layer function). Sometime, a device called bridge is used to connect and isolate two LANs. Bridge reads the destination addresses of all the packets on both LANs in order to decide which LAN should receive the packet. A bridge therefore *does* read the logical information even though it does not alter it. We can say that a bridge facilitates a DLC link between two computers on different LANs. Sometime it is said that the bridge exists on DLC layer[6].

[6] If the bridge performs flow control and error control procedures then it would be implementing DLC otherwise it is only a facilitaor of DLC. Somrtimes, a bridge is used to connect dissimilar DLCs. In is an emulator of DLCs.

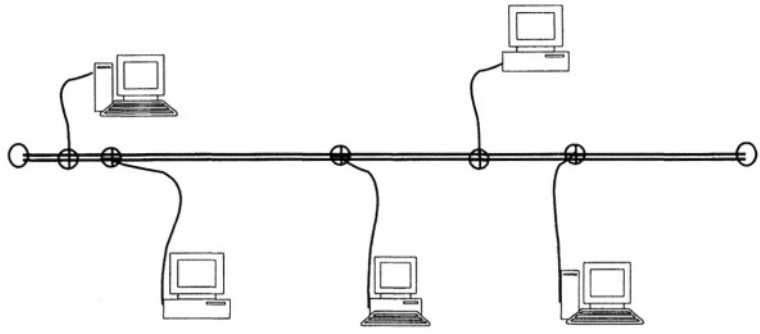


Figure 5-4. LAN connection with Bus topology

The connection at the DLC layer is only in a logical sense, so are the addresses. A DLC protocol communicates with another DLC protocol through DL-SAPs. This gives the possibility of having more than one type of DLC protocols in a computer each with a different SAP. In either case, each DLC layer protocol should have typically three addresses for the following purposes.

Unicast Address to be used to receive a frame addressed *exclusively* to a particular DLC protocol in a computer.

Multicast Address to be used to receive a frame addressed *to a specified group* of DLC protocols on one or more machines. One DLC protocol instance may be a member of more than one multicast groups in general.

Broadcast Address to be used to receive frames addressed to *all* DLC protocols, on one machine or all machines.

Addressing is implemented by allocating an address field in the header of a protocol data unit. The number of bits used to indicate address also places an upper limit on the number of DLC addresses. For example, if a protocol uses 8 bit addresses, the total number of addresses is $2^8 = 256$. In practice these addresses (or *address space*) are divided into the three categories defined above, the unicast, multicast and broadcast. Besides, some addresses are kept for network testing purpose. The real number of useful address space is much below the maximum. Figure 5-5 demonstrates the difference between unicast, multicast and broadcast addressing.

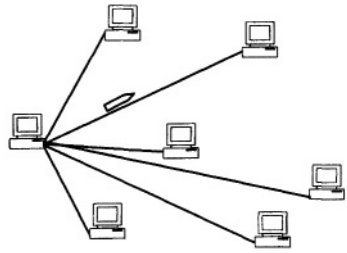


Figure 5-5(a). Unicast addressing

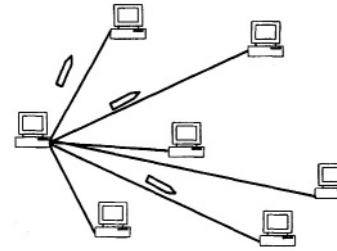


Figure 5-5(b). Multicast addressing

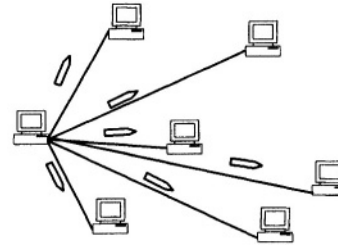


Figure 5-5(c). Broadcast addressing

5.5. Error Control

Transmission impairments are random and affect bits at random locations. The way we characterize a link with respect to its impairment effect is by the probability of error, called variously *link error probability (lep), bit error probability, bit error rate (BER),* or *frame error rate (FER).* For example, if the BER for a link is 0.1, then, on the average, one out of every 10 bits could be expected in error. This does not say which bit number could be in error, or whether there is always an error for each of the 10 bit strings. Protocols for error control are designed based on these probabilities. The frame error rate (FER) is the probability that one or more bits in a frame are in error. If the frame size is *N* then the following expression can be derived for FER in terms of BER.

FER = 1 bit in error and (*N*-1) bits in tact *OR* 2 bits in error and (*N*-2) bits intact *OR* 3 in(*N*-3)and so on until all *N* bits in error.

The probability that a bit is in error = *BER*

The probability that a bit is intact = *1 - BER*

The probability that k bits are in error = $(BER)^k$
The probability that k bits are intact = $(1- BER)^k$
The probability that k out of $N > k$ bits are in error =

$${}^{N}C_k(BER)^k(1-BER)^{N-k}$$

${}^{N}C_k$ (read as N *combinations* k) is the number of ways to have k combinations out of N bits. It is given by $N!/[\ (N-k)!.k!\ ]$

Then

$$FER = {}^{N}C_1\ (BER)(1-BER)^{N-1} + {}^{N}C_2\ (BER)^2(1-BER)^{N-2} + {}^{N}C_3\ (BER)^3(1-BER)^{N-3} + ... + {}^{N}C_0\ (BER)^N$$

We can write the expression for *FER* in the following compact notation.

$$FER = \sum_{k=1}^{N} ({}^{N}C_k)(BER)^k(1-BER)^{N-k} \qquad ...5\text{-}1$$

${}^{N}C_k$ (read as N combination k) = $N!/[k!(N-k)!]$.

An equivalent, simpler formula for *FER* can be derived as follows. The probability that a frame of N bits has no errors is $(1 - BER)^N$. Hence, the probability that a frame of N bits has errors = $FER = 1 - (1 - BER)^N$.

$$FER = 1 - (1 - BER)^N \qquad ...5\text{-}2$$

Also see Equation (5-3) below.

Example 5-4: Find the FER for a frame that consists of 1000 bits. The link has BER of (i) 10^{-4}, (ii) 10^{-3} and (iii) 10^{-2}.
Solution: Using formula 5-2, we get the following values for the FER.
(i) for $BER = 10^{-4}$, $FER = 1 - (1 - 10^{-4})^{1000} = ?$
We use logarithmic method to calculate $(1 - 10^{-4})^{1000}$ as follows:
Let $x = (1 - 10^{-4})^{1000}$
Then $\text{Log}_{10}(x) = \text{Log}_{10}(1 - 10^{-4})^{1000} = 1000\ \text{Log}_{10}(1 - 10^{-4}) = -0.04343$
$x = 10^{\text{Log}10(x)} = 10^{-0.04343} \approx 0.90$
FER = 0.1
(ii) for $BER = 10^{-3}$, $FER = 1 - (1 - 10^{-3})^{1000} \approx 0.63$
(iii) for $BER = 10^{-2}$, $FER = 1 - (1 - 10^{-2})^{1000} \approx 1$
Observations: In the first example of $BER = 10^{-4}$: $FER \approx N \times BER$, while in the second case, this is not true as $N \times BER = 1$ while FER = 0.63. In fact, in the third case, when $BER = 10^{-2}$, $N \times BER = 10$, which cannot be an expression for probability. When BER is so small that $1/N >> BER$, then formula 5-2 can be reduced to the following simple formula:

$$FER = N \times BER \qquad ...5\text{-}3$$

Fortunately, such is the case in most practical situations. Equation (5-3) also explains the meaning of BER being cumulative.

Another interesting observation from this example is about the frame length. It shows that the maximum practical frame length N should be selected such that N << 1/BER otherwise there will be a very high FER. Therefore, for noisy channels with relatively higher values of BER, short frames are preferable. In a way, this justifies the old use of asynchronous transmission. The channels were not as reliable in those days and character-by-character transmission assured a lower FER while character was the frame.

If an error detection mechanism is incorporated then the probability of undetectable error (*PUE)* is of interest. In the absence of an error detection mechanism, *PUE = FER.* This is because without error detection measure the *FER* is also the probability of having a frame with one or more undetectable errors. In the following, we will discuss some measures that can be taken to improve (reduce) such probability.

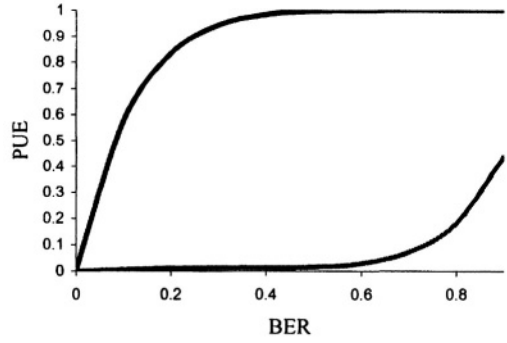
5.5.1. Parity bit

The use of an extra bit with a character is a simple and widely used error detection mechanism. This extra bit is called as the parity bit. It is added to a string of bits (such as a character) in order to make the total number of 1 's always even or odd. It does not have to be the number of 1 's to be considered. Considering the number of 0's will have the same effect. For example, by adding an odd parity bit, we add the knowledge to each character that the number of 1's is an odd number. In this way if the receiver receives a character with an even number of 1's in it, it would know that there is an error somewhere. The error could be undetectable by the receiver if there is an even number of bits in error because that will not change the parity. The improvement can be measured by finding the probability of undetected error (*PUE)* with and without the parity bit. Consider a character set with *N* bits per character. The probability of one or more bits in error is given by $FER = 1-(1 - BER)^N$. However, *PUE,* the probability of an undetected error is much less and is the probability that an even number of bits are in error, that is, $PUE = \sum_{k=2,4,6}^{N} {}^{N}C_k\,(BER)^k(1\text{-}BER)^{N-k}$. In this expression ${}^{N}C_k$ (read as *N* combination *k*) is given by $\frac{N!}{k!.(N-k)!}$.

What does this graph show?

The graph shows the reduction in the probability of an undetected error(PUE) as a function of the bit error rate (BER) for an 8-bit character. Without the use of a parity bit, the PUE is the same as FER.

Figure 5-6. PUE with (lower curve) and without (upper curve) parity bit



The plot in Figure 5-6 shows a comparison of the *PUE,* the lower curve for the character using one parity bit with 7 data bits and the upper curve for the same character without the parity bit. It is obvious from this figure that addition of a parity bit greatly improves the channel behavior.

The obvious flaw in using a single parity bit is that it can't help in detecting an even number of errors because an even number of errors in a character string does not alter the parity. For example, if odd parity of '1's were used to code the 7-bit ASCII character 1001011, the eight-bit code, including the parity bit would be 10010111. Then, if 2, 4 or 6 bits are in error, the total number of '1' remains odd. This results in the receiver decoding it as another character. In order to have the capability of detecting all odd and some even number of errors, a second dimension of parity may be added. A parity character could be generated for a block of characters. Such a character would be called a parity character or byte (if it consists of 8 bits). This is mechanism is referred to as block error check.

5.5.2. Block Error Check

Suppose that a block of characters is coded in the following way. We add a parity bit at the end of each character code. In addition to that, we add a whole byte at the end of a block of characters. An example will illustrate the point.

Example 5-5
Consider the ASCII codes for the set of characters K, J, I, S and T. Using even parity, we have the following codes.

K	1 0 0 1 0 1 1 **0**
J	1 0 0 1 0 1 0 **1**
I	1 0 0 1 0 0 1 **1**
S	1 0 1 0 0 1 0 **1**
T	1 0 1 0 1 0 0 **1**
---	---
$\underline{\pi}$	**1 0 0 1 1 1 0 0**

$\underline{\pi}$ is the parity byte. Each bit of the parity byte is generated as a parity bit for the same location bits of the character block. In the above example, the first bit of $\underline{\pi}$ is '1' because we are using an even parity of '1'. By making it a ' 1', we have an even number of the first bits of all the characters (including the first bit of $\underline{\pi}$). Transmitter sends K, J, I, S, T and $\underline{\pi}$.

The receiver, on receiving all the characters, checks each character horizontally as well as vertically for the even parity. If an odd number of bits are in error in any character, the error will be found by checking the parity bit of the same character. If an even number of bits of a character are in error, the corresponding bits in the parity byte will be in error. However, if an even number of bits are in error in an even number of characters at the same locations, the error will not be detected. This is, however, very unlikely and most of the errors will be detected by this mechanism.

5.5.3. The Cyclic Redundancy Check (CRC)

CRC is an extension of the parity block concept. In this mechanism, a block of parity bits is attached with a block of data bits. The data bits are not treated as characters and the parity block is generated such that it can help receiver detect a large number of error combinations. There is a complex theory and generation mechanism for the parity block. CRC is by far the most popular mechanism of detecting errors in data blocks in synchronous transmission. In theory it can be equally applied to asynchronous transmission.

To understand the CRC mechanism, consider the following variables definitions.

Data block = D (*ofvariable length*)
Number of bits in D = k
CRC polynomial = C (*a known bit pattern with a fixed number of bits*)
Number of bits in C = n (*a constant*)
Parity block = π
Number of bits in π = n - 1 (*one less the number of bits in C*)
Transmitted data = T (*including* D *and the parity block* π *concatenated*)
Number of bits in T = k + n - 1 = Number of message bits + Number of bits in π

In this mechanism, a data block (D) of a variable length is transmitted after appending a parity block (π) to the right side of it such that the resulting data to be transmitted (T) is divisible by a known bit pattern (C).

5.5.3.1. Parity block (π) generation

The CRC polynomial is a known n-bit long string. It is supposed to exactly divide the transmitted message. One simple way to do this would be to first making up an extended form of D by attaching (n - 1) '0's to its right. This will result in a data block with the same length as the would-be transmitted message, but not exactly divisible by C. The next step would be to divide this longer data block by C and get (n-1)-bits long remainder. As a last step, this remainder would be added to the block. Following is a step-by-step procedure to generate the transmitted message. For the next few examples, we will use the following values to explain each step. D = 110111; C = 101 giving k = 6 and n = 3.

Step 1: This equals to shifting D sequence (n-1) places to the left by placing zeros on the (n-1) right places of D.
This is equivalent to multiplying D by 2^{n-1}.
Example 5-6: $D \times 2^2 = 11011100$
Step 2: We divide $D \times 2^{n-1}$ by C and set the remainder = π or
$D \otimes 2^{n-1} = q \otimes C \oplus \pi$ -This is the definition of division called Euclidean theorem.

Example 5-7: $11011100 = 110001 \otimes 101 \oplus 01$ (See the modulo-2 division below)

```
101 ) 11011100 ( 111001
      101
      ---
       111
       101
       ---
        101
        101
        ---
         100
         101
         ---
          01 = π
```

These two steps complete the generation of the parity block.

Note: It would be better to write $\pi = 01$ instead of 1 to ascertain its length being n-1, where n = 3 in this case.

Step 3: We add π (which is n-1 bits), to $D \otimes 2^{n-1}$ to form the sum

$D \otimes 2^{n-1} \oplus \pi = 11011100 + 01 = 11011110 = q \otimes C$ = the message block to be transmitted

Step 4: We transmit $T = D \otimes 2^{n-1} \oplus \pi$, which is divisible by C as seen from step 3.

Example 5-8: With $D \otimes 2^2 = 11011100$ and $\pi = 01$; $T = 11011100 \oplus 01 = 11011101$

Note: The circles around the addition and multiplication signs are used as a reminder that these operations are modulo-2 operations.

5.5.3.2. Error Detection Procedure

Due to channel impairments, some bits in the received message may be in error. If R is the received block, then R = T when transmitted block T has no errors. However, if one or more bits of T arrived in error, then $R \neq T$. Instead R = T + E, where E is the channel error block.

Example 5-9: Let's suppose E is the bit string that shows errors in received data. E has the same number of bits as the transmitted data. We represent

error with '1' in a location where there is an error and '0' in locations where there are no errors. Then,
For no error in transmitted data T:
E =00000000
and
$R = T \oplus E = T$.
With the leftmost bit in error E = 10000000 and
$R = T \oplus E = 11011101 \oplus 10000000 = 01011101 \neq T$
Similarly, with the error in right most bit, we can say E = 00000001 and,
$R = T \oplus E = 11011101 \oplus 00000001 = 11011100 \neq T$

Since error results in the received data block not being equal to the transmitted data block, there is a very high probability that the received data block R is not divisible by C. With that in mind, we will continue the procedure at the receiver as step 5.
Step 5: At the receiver, we divide $R = T \oplus E$ by C, as seen in step 3.
If it divides exactly, we assume that there is no error. If it has a non-zero remainder then errors have been added to T by channel.

Example 5-10

User data block D	=	1010001101 (10 bits)
CRC Pattern	C	= 110101 (6 bits)
Parity block	π	= to be calculated (5 bits)

Step 1: $D \otimes 2^5 = 1010001101\underline{00000}$; appending n-1 zeros to the message
Step 2: $D \otimes 2^5 = 1101010110 \otimes C \oplus 01110 \Rightarrow \pi = 01110$; n-1 bit remainder of division

```
                                    1101010110
110101 ) 1010001101000000 (
         110101
         ------
          111011
          110101
          ------
           111010
           110101
           ------
            111110
            110101
            ------
             101100
             110101
             ------
              110010
              110101
              ------
               01110  = π
```

Step 3: $T = D \otimes 2^5 \oplus \pi = 101000110101110$; Appending π to the message on the right side

Step 4: T = 101000110101110 is the transmitted block.

Step 5: With no errors, the R = T = 101000110101110

R / C = 11010101 (no remainder)

Or $R = 11010101 \otimes 110101 \oplus 0$ {remainder is zero}

Suppose that there is error in the bit number three from right giving
E = 000000000000100

In this case $R = T \oplus E =$ 101000110101110 $\oplus$ 000000000000100 = 101000110101010

When the receiver divides R by C, it gets a non-zero remainder, as shown below:

```
110101 ) 101000110101010 (   1101010110
         110101
         ------
          111011
          110101
          ------
           111010
           110101
           ------
            111110
            110101
            ------
            101110
            110101
            ------
             110111
             110101
             ------
                100   = Remainder
```

From a non-zero remainder, the receiver deduces that the received data block has errors. The receiver discards this data block. In most practical situations, the transmitter keeps a copy of every data blocks so that it can be retransmitted if errors were detected.

5.5.3.3. Polynomial representation of binary numbers

While dealing with the arithmetic operations of binary numbers, we make a few observations in this section. First, these operations do not have the concept of a carry over. This makes them a little different from the *binary arithmetic operations.* In binary arithmetic, one would expect the carry over just like the decimal algebra. The operations used in the discussion of CRC are from a class of operations called modulo algebra. In modulo-q arithmetic, only the numbers with the base of q are used. Binary numbers have a base of 2. This makes the CRC operations modulo-2 operations. The modulo-2 addition (or subtraction) is just like an exclusive-OR operation. While performing modulo-2 division or multiplication, the subtraction or additions are replaced by exclusive-OR operations.

The use of modulo-2 operations makes it very simple to extend the algebraic part to polynomial form. This provides a mathematical structure to the calculations. A modulo-2 polynomial representation of a bit string consists of sum of terms like a_iX^i. The coefficients $\{a_i\}$ can have a value of '0' or '1', which is the value of a bit in position 'i' in the string. The power of X^i has no significance except that it denotes the position of a bit in a binary string. For

example, a binary string 101 has the polynomial value of $a_0X^0 + a_1X^1 + a_2X^2$ with $a_0 = 1$, $a_1 = 0$, and $a_2 = 1$. In other words: $101 = 1 + X^2$.

Representation of binary numbers in polynomial forms simplifies algebraic operations. It has very simple rules for addition (addition and subtraction are same). For example $aX^k + aX^k = 0$, which is equivalent to $1 \oplus 1$ at position k in two binary strings.

Example 5-11: Let's solve Example 5-4 with all binary numbers and operations in polynomial form.

User data block D = 1010001101 (10 bits)

Interpreting data from the right bit as the least significant bit, we can write:

$D(X) = 1 + X^2 + X^3 + X^7 + X^9$

CRC Pattern C = 110101 (6 bits)

$C(X) = 1 + X^2 + X^4 + X^5$

Parity block π = to be calculated (5 bits)

Step 1: $D \otimes 2^5 = 1010001101\underline{00000}$

$D(X) \times X^5 = X^5 + X^7 + X^8 + X^{12} + X^{14}$

Step 2: $D \otimes 2^5 = 1101010110 \otimes C \oplus 01110 \Rightarrow \pi = 01110$

$D(X) \times X^5 = [X + X^2 + X^4 + X^6 + X^8 + X^9] \times C(X) + [X + X^2 + X^3] \Rightarrow \pi(X) = X + X^2 + X^3$

Step 3: $T = D \otimes 2^5 \oplus \pi = 1010001101101110$

$T(X) = D(X).X^5 + \pi(X) = [X^5 + X^7 + X^8 + X^{12} + X^{14}] + [X + X^2 + X^3]$

Step 4: T = 1010001101101110 is the transmitted block.

$T(X) = X + X^2 + X^3 + X^5 + X^7 + X^8 + X^{12} + X^{14}$ is the transmitted polynomial

Step 5: With no errors, the R = T = 1010001101101110

$R(X) = X + X^2 + X^3 + X^5 + X^7 + X^8 + X^{12} + X^{14}$

R / C = 11010101 (no remainder)

$R(X)/C(X) = 1 + X^2 + X^4 + X^6 + X^7$

Suppose that there is error in the bit number three from right giving E = 0000000000000100, $E(X) = X^2$. In this case R = T $\oplus$ E = 1010001101011110 $\oplus$ 0000000000000100 = 1010001101010l0, or R(X) = T(X) + E(X);

$R(X) = [X + X^2 + X^3 + X^5 + X^7 + X^8 + X^{12} + X^{14}] + X^2$

$R(X) = X + X^3 + X^5 + X^7 + X^8 + X^{12} + X^{14}$

When the receiver divides R(X) by C(X), it gets a non-zero remainder, as shown below:

$$
\begin{array}{l}
1 + X + X^3 + X^4 \;\overline{)\; X + X^3 + X^5 + X^7 + X^{11} + X^{12} + X^{14}} \quad X + X^3 + X^5 + X^6 + X^9 \\
\qquad\qquad\qquad\quad X^9 + X^{10} + X^{12} \\
\qquad\qquad\qquad\quad \underline{\qquad\qquad\qquad} \\
\qquad\qquad\qquad\quad X^7 + X^9 + X^{10} + X^{11} \\
\qquad\qquad\qquad\quad X^6 + X^7 + X^9 + X^{11} \\
\qquad\qquad\qquad\quad \underline{\qquad\qquad\qquad} \\
\qquad\qquad\qquad\quad X^3 + X^5 + X^6 + X^{10} \\
\qquad\qquad\qquad\quad X^5 + X^6 + X^8 + X^{10} \\
\qquad\qquad\qquad\quad \underline{\qquad\qquad\qquad} \\
\qquad\qquad\qquad\quad X + X^3 + X^8 \\
\qquad\qquad\qquad\quad X^3 + X^4 + X^6 + \\
\qquad\qquad\qquad\quad \underline{\qquad\qquad\qquad} \\
\qquad\qquad\qquad\quad X + X^4 + X^6 \\
\qquad\qquad\qquad\quad X + X^2 + X^4 + \\
\qquad\qquad\qquad\quad \underline{\qquad\qquad\qquad} \\
\qquad\qquad\qquad\quad X^2 \quad = \\
\qquad\qquad\qquad\quad \underline{\qquad\qquad\qquad}
\end{array}
$$

The remainder X^2 if represented in binary form is 100, which is the same as in Example 5-4.

5.5.3.4. Implementation of CRC

CRC procedure can be implemented in software as well as hardware. In either case, it consists of the (same) above steps. A choice between hardware and software depends, among other things, on the required speed of calculations. A software implementation will perform all operations as part of a program stored in processor memory. Its speed is much slower than hardware implementation. In hardware implementations, a division circuit is used that consists of logic gates, registers and some ADDER circuits. The gates open and close to start/stop the process. The registers store the bit strings. The ADDERS define the polynomial *C(X)*. Figures 5-7 (a) and (b) show an example of a logic circuit that implements division by a general

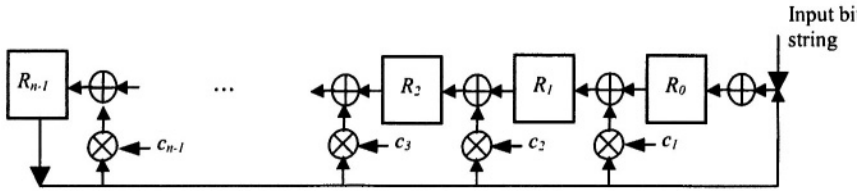


Figure 5-7(a). Divide by $c_0 + c_1 X + c_2X^2 + c_3X^3 + ,.., c_nX^n$ circuit.

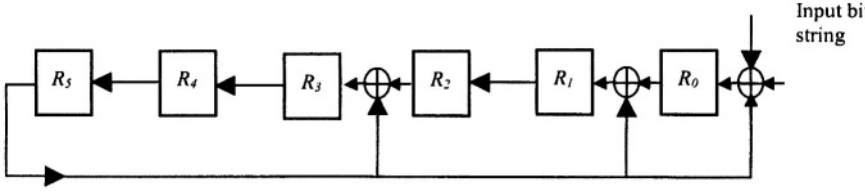


Figure 5-7(b). Divide by $1 + X + X^3 + X^6$ circuit

polynomial $c_0 + c_1 X + c_2X^2 + c_3X^3 + ,.., c_nX^n$. Values of n and $\{c_i\}$ define a particular $C(X)$. Figure 5-7(b) shows division by $1 + X + X^3 + X^4 + X^7$.

The division circuit works in the following way. Data is input at the point labeled as "Input bit string" starting with the leftmost bit. The number of registers is one less than the number of bits in $C(X)$. By the time all data has been pushed to R_0, the division is complete due to the existence of adders and multipliers. The multipliers are replaced by open circuit or straight lines pointing towards the adder corresponding to the c_i's being equal to '0' or '1'. A multiplication with binary '0' corresponds to no line while a multiplication with a binary '1' corresponds to a straight line, indicating a closed connection.

This circuit does not give the quotient of division; it only gives the remainder. When the division is complete the contents of registers $\{R_i\}$ constitute the remainder of the division.

Here's how it can be used at the transmitter. All the bits of D are passed through the circuit while they are being transmitted. At the end of transmission, the contents of the register are the remainder bit string π. At this point, the *n-1* bits register contents are transmitted starting from the leftmost register right after D. This completes transmission of T.

The receiver process is even simpler. The received data block R is passed through the circuit by inserting bits in R_0 and left-shifting one by one. When all bits have been passed through the circuit, the contents of the registers contain the remainder. If they are all '0', then the data block is assumed to be without any errors, otherwise there were some errors in R.

5.5.3.5. How to Decide *C(X)*

The transmitter and the receiver must use the same polynomial for CRC generation and error detection. The *C(X)* must be chosen to maximize the error detection properties. Many standardization agencies have recommended polynomials that would be suitable to be used as *C(X).* Some of these are given below.

CRC-12 => $X^{12}+X^{11}+X^3+X^2+X+1$ = 1100000001111
CRC-16 => $X^{16}+X^{15}+X^2+1$ = 11000000000000101
CRC-CCITT => $X^{16}+X^{12}+X^5+1$ = 10001000000100001
CRC-32 =>

$$X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$$
$$=100000100110000010001110110110111$$

Among these CRC-CCITT has been recommended by ITU and is perhaps the most commonly used CRC polynomial. CRC-32 is a proposal of the IEEE and is considered suitable for LANs.

5.5.3.6. Error Detection Power of CRC

The CRC polynomials are chosen with care. Linear algebra provides us with tools to test the properties of polynomials. Using such tools, the power of error detection can be maximized. Some of the characteristics of the often used polynomials are such that:

(i) all single bit errors can be detected,

(ii) all double errors can be detected as long as *C(X)* has a factor with at least three terms,

(iii) a large number of other error patterns, depending on the way C(*X*) is chosen,

(iii) any odd number of errors as long C*(X)* contains a factor *(X+1),*

(iv) any burst error for which the length of the burst is less than that of the remainder. A burst error is defined as one in which consecutive bits are in error. In other words, following is the error polynomial *E(X)* for a burst error of length *k* starting at bit-position *l*.

$$E(X) = X^{l-1} + X^{l} +,..., X^{l+k-2} + X^{l+k-1}$$

(v) most of the burst errors larger in length than the remainder will be detected as well.

5.5.3.7. Error Recovery Mechanisms

The detection of errors is part I of the error control mechanisms. The recovery process starts after a data block is found to contain errors. The error recovery methods in computer networks are many times closely related to the flow control mechanisms. Because of this, we will postpone a discussion on error recovery mechanisms until after flow control, which we discuss next.

5.6. Flow Control

Flow control mechanisms are necessary in order to stop the transmitter from overwhelming the receiver with data. In general, this could be achieved by letting the receiver control all data flow from the sender. Consequently, the most popular flow control mechanisms allow the receiver to send credit to the sender in terms of how much data could be transmitted. The credit exists at the receiver as what is called the window size. This window size is updated as the receiver processes the received data.

Flow control might be needed due to many reasons. First and foremost is the link capacity. If a link is shared by many transmitter-receiver pairs, the total amount of data on the link may exceed its capacity at some time. Second reason could be the unavailability of sufficient memory resources at the receiver station. It is possible that a link is not congested and there is enough memory available to process/store the received data and still there is congestion. Congestion is a condition in which packets queued to be processed buildup a line above a certain threshold. This queue could simply be because of the receiving station having to forward each packet on a slower link. Flow control could be required and implemented at all layers. When implemented on DLC layer, it results in regulation of data flow across a single link. The simplest implementation of such scheme is when the window size is one packet. In such cases, it is also called Stop-and-Wait (SnW) flow control.

5.6.1. Stop-and-Wait (SnW) Flow Control

In SnW the maximum credit that a sender can get is one packet. After transmitting one packet, the sender of data waits for the next credit. Once the

receiver receives and processes the packet, it will send a small packet acknowledging the reception of the sent packet. This small packet is called as 'Acknowledgement Packet' or simply an ACK.
Figure 5-8 illustrates a timing diagram of such flow control.

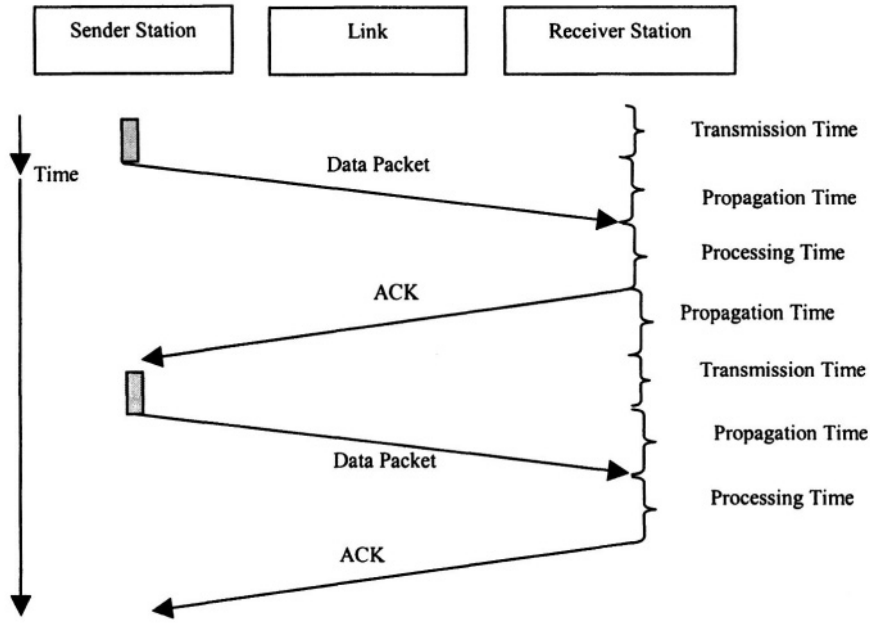


Figure 5-8. Stop-and-Wait mechanism of flow control.

Here are the definitions of various types of time delays in Figure 5-8.
Transmission time: The transmission time is due to a finite capacity of a link. If a PDU has *L* bits and the link capacity is *C* bps, then the transmission time of the PDU is *L/C* seconds.
Propagation time: The propagation time is due to a finite velocity of signal propagation. If the signal is transmitted as electromagnetic wave or optical ray, the speed of propagation is the speed of light. In general, if the transmitter and receiver are *d* meters apart and *c* is the speed of propagation of the signal wave, then the propagation time is *d/c*.
Processing time: The processing time is due to the finite processing capacity of the receiving station. It depends on a number of factors, including but not limited to, the processor type, queue size and protocol type (procedures) used after receiving a data packet.

It may be noted in Figure 5-8 that the transmission time for ACK is considered to be zero. This is because the ACK packet is assumed to be very short *(L << C)* resulting in a negligible value of *L/C*. Similar reason could be used for zero processing time for the ACK.

The main positive feature of SnW is its simplicity and ease of implementation. However, it may be possible in many situations that even though the receiver can process many packets in a certain amount of time but it can't get them together due to the limit of maximum one packet. In this case, the link remains unnecessarily idle. Hence, the link utilization for SnW is generally low. One way to improve the link efficiency is to allow more than one packets to be transmitted before a credit can be expected from the receiver. The receiver could either send credit with the reception of each packet or first receive a number of packets and then send credit for all of them. This mechanism is called sliding windows (SW) flow control and is discussed next. Sometimes terms like *window flow control* or *window-based flow control* is used to describe both, the stop-and-wait and sliding widows (SW) flow control mechanisms. This is because SnW is a special case of SW with the window size set at one.

5.6.2. The Sliding-windows (SW) Flow Control Mechanism

SW is the generalization of SnW to more than one packet. In this mechanism, a receiver allows the sender to send up to a certain maximum number of packets without getting further credit or ACK. This maximum number of packets allowed to be transmitted without receiving an ACK is said to be the maximum window size. Each ACK usually allows expanding the number of packets to the maximum window size.

Consider the following example. Let the receiver specify a maximum window size of 4. Assuming that the packets could be sequence numbered from 0 through 7 (three bit sequence number) a maximum of 4 packets are allowed to be transmitted without getting further credit. Now, suppose that packet numbers 0, 1, 2 and 3 have been transmitted and the receiver has not acknowledged them. After the processing of the received packets is complete, the receiver sends an ACK for all the four packets. Then, on receiving the ACK packet, the transmitter is permitted to send packet numbers 4, 5, 6 and 7.

A more complex case would be when the receiver might send ACK anytime instead of waiting for the entire window to be exhausted. To implement this, the ACKs should be sequence numbered just like packets. Customarily, an ACK*j* is interpreted to mean that the packets numbered up to (*j*-1) have been received and number *j* and above are expected. Thus, if *K* is the maximum window size and *N* is the maximum sequence number, then on receiving ACK*j*, the window will *slide* forward to sequence numbers *j*, *j*+1, ..., *(j*+*K*-1) in a round robin fashion (..., *N*-1, *N*, 0, 1, ...). In other words, the sequence number forms a circular scale numbered 0 through *N*. Circular

because after *N,* the next sequence number is 0, as shown in Figure 5-9. In this Figure, we assume that $N = 7$, and $K = 3$, The 'current' window contains the following sequence numbers in this order, {7, 0, 1}. Once a sequence number is used, the window shrinks towards the direction shown as 'Consumption'. Alternatively, when credit is allocated, the window expands in the direction shown by the arrow labeled as 'Expansion'. If this is a transmit window then the sequence number of the next outgoing packet will be decimal 7 (or binary 111). If this is the receive window then the sequence number of the packer arriving next should be decimal 7 (or binary 111).

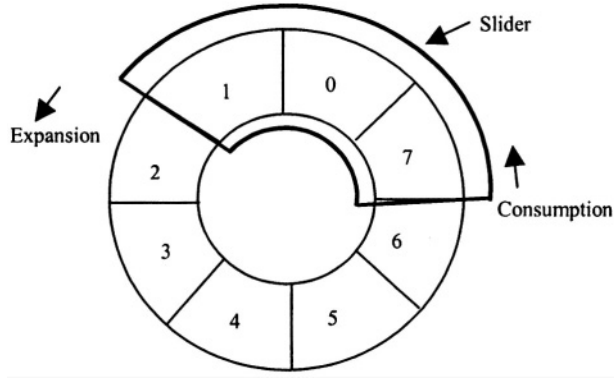


Figure 5-9. The modulo-8 sequence numbering with a window size of 3.

Now, in this DLC connection the transmitter and receiver pair agrees to have a window size of 4 on a link. Let the protocol allow for a 3-bit sequence numbers (000, through 111). Both the stations maintain the window sequence numbers. The window at the transmit side is called the *transmit window,* and it consists of the sequence numbers for the outgoing packets. The window at the receive side is called the *receive window* and it consists of the sequence numbers expected on the incoming packets. Initially, both windows are open to send (receive) four packets sequence numbered {0,1,2,3}. Let the transmitter send packet #0. By doing so, it shrinks its window to size three (numbers 1, 2, and 3) while the receive window is still size four. Next, let the transmitter send another packet (number 1). By doing so, it closes its window further down to a two packets while the receive window is still open to receive all the four packets.

Next, suppose that the receiver receives packet#0. After doing so, it closes its receive window to three packets (#1, 2 and 3) while the transmit window is open for a two packets. Let the receiver receive the second packet (#1), By doing so, the receive window goes down to two as well. At this time, the sender sends packet#2. By doing so, it shrinks its window size to one packet. The receiver processes packet numbers 0 and 1 and sends ACK2 asking for packet number 2 (which is in transit incidentally). After sending ACK2, the receiver *expands* its window to size four and expects packets numbers 2, 3, 4 and 5. On receiving ACK2, the sender (who has already sent packet#2) expands its window size to include packet number 5. Thus, the sender window size is three #(3, 4, 5) while the receiver size is four #(2, 3, 4, 5). Some steps of this operation are shown in Figure 5-10 below.

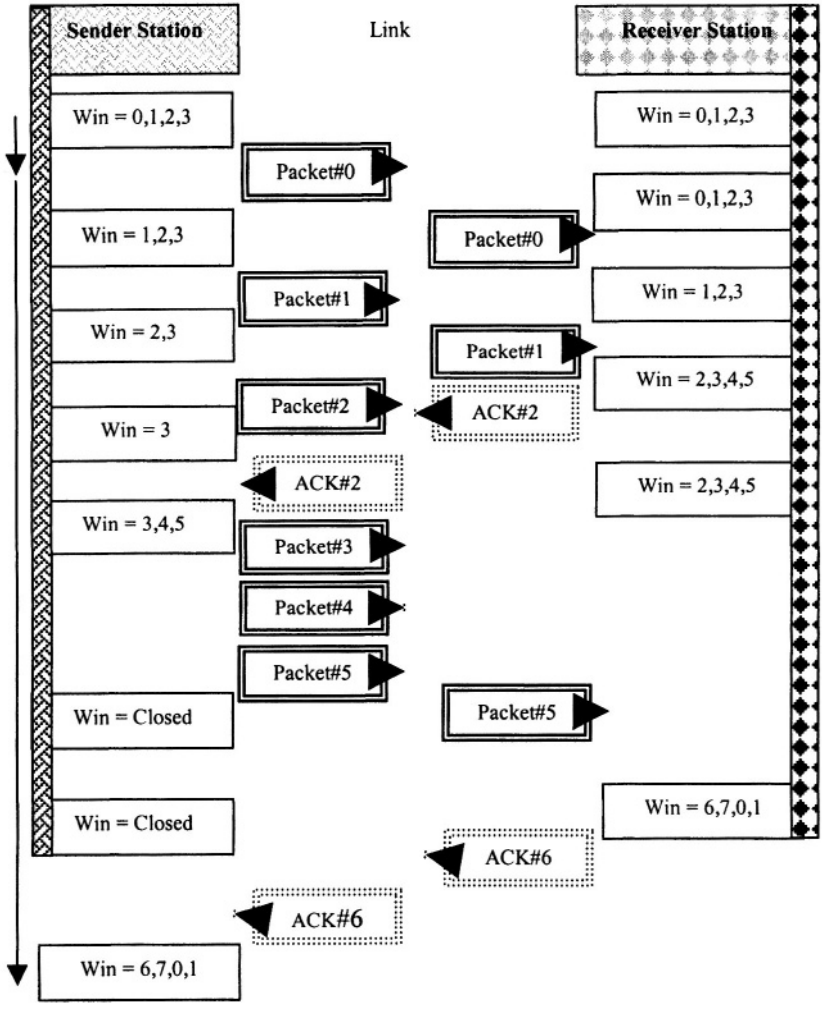


Figure 5-10. Window flow control with window size of 4 and three bit sequence numbers.

5.6.3. Link Utilization of Window Flow Control Mechanisms

The link utilization of such mechanisms depends upon packet size and the channel propagation time. If the window size is chosen such that the channel remains 'filled' for most of the time one could get fairly high link utilization. To get an appreciation of the actual numbers, let us first take the example of SnW.

Let T_{trans} be the transmission time of the packet and T_{prop} be the propagation time. Assume that the transmission time of the ACK and the packet processing times are negligible. Defining link utilization as the fraction of time of channel use for which actual data packets are in transit, we see the following sequence of events.

1. Data packet transmitted takes T_{trans} seconds
2. Packet propagates takes T_{prop} seconds
3. ACK propagates back to sender takes T_{prop} seconds

$$\text{Link Utilization } \rho = \frac{T_{trans}}{T_{trans} + T_{prop} + T_{prop}} = \frac{1}{1 + 2T_{prop} / T_{trans}}$$

By letting $T_{prop} / T_{trans} = T_p$, we get

$$\rho = \frac{1}{1 + 2T_p} \qquad \text{...5-4}$$

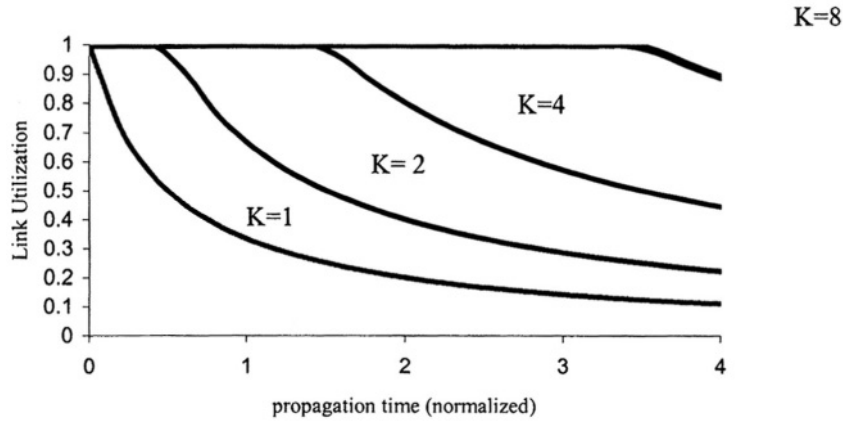
Where T_p is propagation time in units of number of transmission times.
From the expression of ρ, it is clear that the utilization can approach 100% for very close stations with negligible T_p. However, if the propagation time is close to transmission time ($T_p = 1$), then ρ drops to 33%.

By similar reasoning, it can be argued that if K is the window size, then the corresponding link utilization for sliding windows mechanism is proportional to K. That is, if ρ_K is the link utilization of SW mechanism with a window size of K, then,

$$\rho_K = K.\rho = \frac{K}{1 + 2T_p} \qquad \text{...5-5}$$

However, since the utilization cannot be more than 100%, if $K \geq (1 + 2T_p)$, then $\rho_K = 1$. Evidently, by using a window size $K > 1$, the link utilization can be improved K folds.

Figure 5-11. Link Utilization for K=1,2,4,8



The actual improvement in link utilization due to K could be less than the theoretical value. As the window size exceeds unity, the window maintenance, processing and updating becomes a complex function. Besides, recovery of lost packets might require the sender to keep copies of many packets until they have been acknowledged. This is discussed in detail in section 6 on error recovery mechanisms.

5.6.4. Full-duplex Communications Using Window Flow Control

When two stations exchange data packets in full-duplex mode, both stations act as transmitter as well as receiver of data. For flow control purposes, we can consider such a system as consisting of two transmit-receive pairs, each like the one shown in Figure 5-10. Consequently, each station has to create, monitor and update two windows, one for transmitting and the other for receiving of data. In full-duplex mode, often transmission of acknowledgements (ACKs) can be bypassed by using the outgoing data packets to include acknowledgement messages for the received data packets. This mechanism, called *piggybacking* of ACKs is explained in the context of error recovery.

5.7. Flow Control Based Error Recovery Mechanisms

Error recovery mechanisms in data communications are closely related to the flow control mechanisms described above. It is possible to have channel codes that would correct errors at the receiver, needing no particular type of flow control. Such codes are called *error-correcting codes.* Use of error correcting codes is called *forward error correction (FEC).* Such approach is not very popular in data communications due to large overhead required for this purpose. However, when data is to be interpreted in real time, such as voice communications, then the only error control mechanisms that can be used are of FEC type. Luckily, voice quality is not deteriorated much by frame error rates (FER) as high as 1%. In data applications, an FER of 10^{-5} or below is desired. In fact, the high-speed networks are designed for bit error rates of as low as 10^{-10}.

Error control for data communications is added over the flow control mechanism. This is done by discarding a packet with errors and initiating a recovery process. Since packets have sequence numbers, it is possible to request the transmitter to retransmit the packet with errors. The term used to describe this procedure is *backward error correcting* (BEC) mechanism or *automatic repeat request* (ARQ). The SnW flow control can be used for SnW ARQ while the SW flow control can yield either a Go-Back-N or selective reject ARQ.

5.7.1. Stop-and-Wait ARQ

In this mechanism of ARQ, the sending station keeps a copy of every packet transmitted. After transmission it waits for an ACK for each packet before sending the next packet. At the same time, it also initiates a timer. If an ACK is received before the timer expiry, the stored copy of the transmitted packet is discarded, next packet transmitted, and the timer re-initiated. In case that the receiver does not receive the transmitted packet (lost packet) or receives it with errors, it will not send an ACK. The timer at the sending station, in this case, will go off after the expiry of time-off period. On expiring the timer, the transmitter sends the copy of the packet again. This process continues until a successful transmission has occurred. To avoid *duplication,* alternative ACKs and packets may be numbered as ACK1 and ACK0 (or with any other numbers). Thus an ACK0 will acknowledge packet with sequence number 1 and vice versa. In the event of a lost ACK, the transmitter will retransmit the same packet on timer expiry. From its sequence number, the receiver will know that this was a duplicated packet.

It is also possible to use a negative acknowledgement (NAK) to notify the transmitter if a packet arrives in error. Thus, there are potentially two mechanisms used for error recover in this case: time-out with ACK and ACK-NAK pair

5.7.2. Go-Back-N ARQ

In go-back-N ARQ mechanism of error control, a sliding windows (SW) mechanism provides flow control of data between the transmitter and receiver. The transmitter sends packets as allowed by the current window size. If there are no errors in the packets, then the normal flow control operation continues as discussed above. In the event of an error, the receiver discards the packet and does not increment its receive window pointer. Therefore, when it receives the packet with the next sequence number, it may send a negative acknowledgement (NAK). A NAK is usually implemented by sending the ACK packet asking for the discarded packet. Thus, if the transmitter is already expecting an ACK for this packet, it will know that the packet in question was never received (or had errors). It will then *go-back* by resetting its window position at the discarded packet number and restart transmission of this packet and all subsequent ones even if they had been already sent.

An example could perhaps illustrate this mechanism better. In Figure 5-12, packet number 1 arrives in error, and is thus discarded. When packet#2 arrives, the receiver expects packet #1, and therefore it sends an ACK for packet #1. This is like saying, 'I don't need packet #2 which has just arrived, send me Packet #1'. When the ACK#1 arrives at the transmitter it has already transmitted packet #3 and has incremented transmit window accordingly. However, on receiving ACK1, the transmitter resets its window to sequence numbers 1, 2, 3, 4 and starts transmission from packet number 1 onward. A timer may be used in addition to the ACKs as well to assist the sender in estimating whether the last packet was rejected or still in transit.

5.7.2.1. Full-duplex operation

The explanation of the full-duplex flow control mechanisms applies equally to the ARQ mechanisms. Each direction of communication can be treated as an independent transmitter-receiver pair across the data link. However, there is the possibility of eliminating the use of separate packets for acknowledgement or negative acknowledgements. This is discussed under *piggybacking* later.

5.7.2.2. Piggybacking

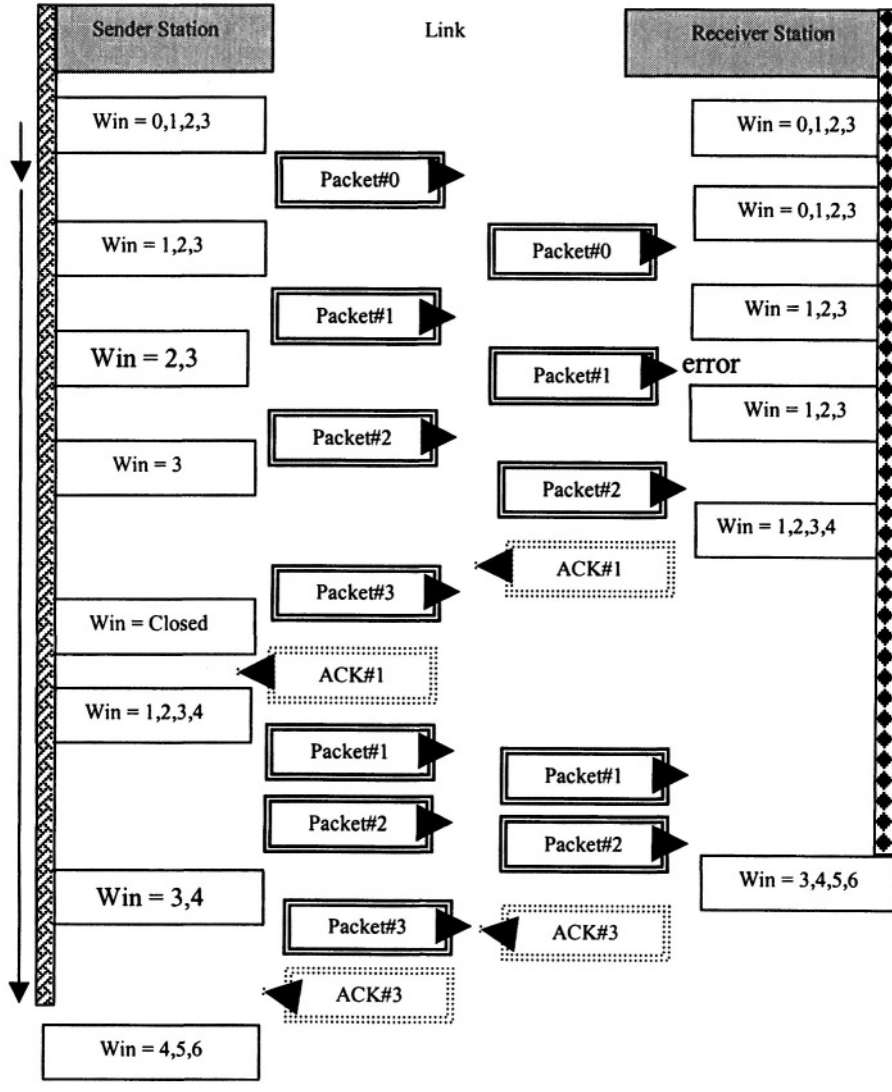
In full-duplex data transfer, ACKs may not be sent separately. Instead, they are piggybacked as part of data packets or frames. So, two stations exchanging frames will send data frames and include the sequence number of the next frame number they expect to receive. A protocol implementing piggybacking would include two sequence number fields in the protocol header, the sequence number of the data packet being transmitted and the sequence number of the data packet expected from the other station.

5.7.3. Selective Reject ARQ

In go-back-N ARQ mechanism, when a NAK for packet *#k* is received, all packets starting with *#k* up to the window size may have to be retransmitted. Some or all of the packets subsequent to *#k* might have already been transmitted. But, the receiver does not acknowledge any packets other than the one it expects. This unnecessarily consumes the link capacity. If a packet subsequent to *#k* could be received without errors, then why not to accept it and ask the retransmission of only the packet in error. Some protocols provide for a selective reject mechanism in which retransmission is sought for only the packet that was in error instead of a whole block of packets. This could be accomplished by sending a NAK as soon as a packet is received in error or out of sequence. While the receiver waits for the retransmission of the discarded packet, it keeps receiving the next in sequence and stores them in a memory area temporarily. Once the packet in error is successfully received, all the stored packets could be marked as 'received' and processed by the receiver. This results in a slightly complex mechanism with an obvious improvement in link utilization.

If link utilization is the performance measure of the ARQ schemes, then the selective reject tops them all. The performance of go-back-N ARQ is much better than stop-and-wait ARQ. The tradeoff is in having more complex implementation. Complexity includes handling the window management functions in addition to the memory required to store all the packets until the transmitter has received an ACK.

Figure 5-12. A transmitter/receiver pair using go-back-N ARQ



5.7.4. Maximum Window Size

One interesting parameter of the window flow and error control mechanisms is the maximum window size that the communicating stations may agree on. For a sequence number field of n bits, the packets can be numbered from 0 through 2^n-1. It appears that one can have a window size of

2^n including all packet sequence numbers in each window. However, the same numbers are used from one transmission window to the next. There is the possibility of an error or duplicity if we use a window of size 2^n. For example, if the receiver ACKs a whole 'windowful' of packets and the ACK gets lost, the transmitter may conclude that all the packets have been lost. It will retransmit the same window starting with packet number 0 after the timer expiry. The receiver, assuming that the ACK has been received successfully, may conclude that the current packet number 0 is the next expected packet. In order of avoid a problem like this, the maximum window size should not be 2^n. For selective reject ARQ, a window size of $2^n - 1$ is possible. For Go-Back-N, the rule is to have the sum of the send and receive windows not larger than 2^n.

5.8. Link Control and Testing

In the description of functions and procedures provided by DLC layer, it is assumed that all links are functioning properly and ultimately all data gets through. This, however, may not be the case in reality. Protocol functions and procedures are required to avoid locking into a situation where the transmitter keeps sending a packet forever without the receiver responding. These are called link control and testing procedures. These procedures could be implemented with the help of special frames exchanged between the two stations or procedures to stop transmission after a certain number of failures. We will look into them when we discuss examples of DLC protocols.

5.9. Review Questions

1. In the OSI layering structure, synchronization occurs at both physical layer and data link layer. Can you briefly state the similarity and difference of these two levels of synchronization?

2. Calculate the probability of frame error for a common transmission environment: if we have a packet of size 10,000 bits, the bit error rate (BER) is 1/100,000 (assume the BER for all the bits are constant and independent). What is the probability of the whole packet arrives without any errors? What if the error probability is 1/10,000,000,000 each bit? If just considering error control, what packet size is more desirable, small or large?

3. There are two aspects for error control techniques: error detection and correction. Summarize each and state under what circumstances are they commonly used and not likely to be used?

4. Parity byte method is actually applying parity bit in two dimensions. For parity bit method, it can't detect all the even bits errors (such as 2 bits error, 4 bits error,...). But for parity byte, the 2-D parity byte method, we are able to detect all 1 bit, 2 bits and 3 bits error and most 4 bit error. Demonstrate how with an example?

5. Many of the error recovery algorithms we discussed, Stop-and-wait, Go-back-N ARQ and Selective-reject ARQ can be thought as variations of ARQ when chosen different window sizes (sender and receiver side). Can you give window sizes for each one of them?

6. Summarize the basic functions of DLC layer? Imagine an optical communication channel where error probability is extremely low. Does every one of the above functions remain a must? What about in a more error prone communication channel like in wireless communication system?

This page intentionally left blank

6. Data Link Control Layer Protocol Examples

In Chapter 5, we discussed the functions to be provided by a data link control layer. Also discussed were some of the procedures and mechanisms used to implement those functions. We learnt that synchronization could be implemented on a block-by-block basis (synchronous transmission) or character-by-character basis (asynchronous transmission). Different address requirements for each station were discussed. The bulk of the discussion was on flow and error control mechanisms. We continue with the discussion of DLC layer in this chapter taking examples of some data link control layer standards. The OSI seven-layer model has a protocol defined at the DLC layer called HDLC (High-level Data Link Control) protocol. We will start out with HDLC and discuss its functions and how they are performed. Following HDLC, we will look at the layer 2 protocol for broadband ISDN, the Asynchronous Transfer Mode (ATM). We will close the DLC examples with a discussion on the medium access control (MAC) sublayer of the IEEE WLAN.

The HDLC is defined as International Standardization Organization's recommendations ISO3309 and ISO4335. It has captured the imagination of much of the world of data communications. It provides a variety of options that the two stations may be able to negotiate before data exchange. The ATM protocol is a natural evolution of the availability of high-speed physical layer in the form of optical network standard, called SONET (Synchronous Optical Network) in the USA and SDH (Synchronous Digital Hierarchy) by ITU. It is well known that constant size and smaller packets would result in relatively less queueing delay. Using these and other results, the networking community all across the world came up with the concept of using a DLC layer above SONET/SDH that would consist of short (53 octet) packets, called cells. In discussion on WLAN, we restrict our focus on the medium access control sublayer. The wireless LANs, besides their unique applicability, provides an interesting case of a medium access control (MAC) sublayer.

6.1. HDLC (High-level Data Link Control) Protocol

The HDLC is designed to provide services to the network layer of an OSI network. It provides functions for a moderate speed data link connection (close to high-speed MODEM) over a moderately reliable link. A derivative of a proprietary standard (SDLC of IBM), HDLC has set a data link layer framework in which many layer 2 protocols have been specified. All communication between the two DLC layers occurs through exchange of HDLC frames. HDLC defines three frame types to be exchanged between three station types using any of the three configurations.

6.2. HDLC Frame Types

The three frames specified in HDLC are
(i) Information or I-Frame that carries user information,
(ii) Supervisory or S-Frame that carries information regarding the I-frames and connection and
(iii) Un-numbered or U-Frame that carries the information about connection and link status.
Figure 6-1 shows a generic HDLC frame.

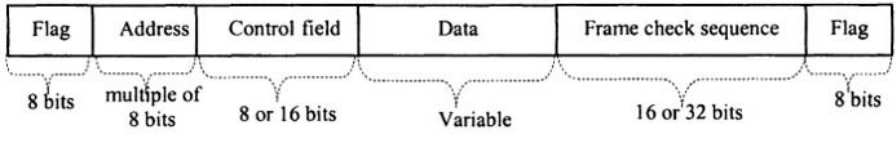


Figure 6-1. Generic HDLC frame structure

The three types of frames provide the necessary functions discussed under 'DLC functions and procedures' in Chapter 5. These include synchronization, connection setup and termination, addressing, flow control and error control. These protocol functions are provided through protocol fields of Figure 6-1. Synchronization and addressing are required for all frames making it an essential part of all frame types. Additionally, a 2 bytes frame check sequence (FCS) is specified for CRC based error detection. The remaining parts consist of control field and (in case of I-frame) user data.

The control field defines frame types. It is an 8-bit field with first bit or two defining frame types, one bit (fifth) to tell whether it is a *command*

frame or *response* frame. The definition of the remaining six or seven bits depends on frame type.
Figure 6-2 shows the differences in the control fields for the three frame types.

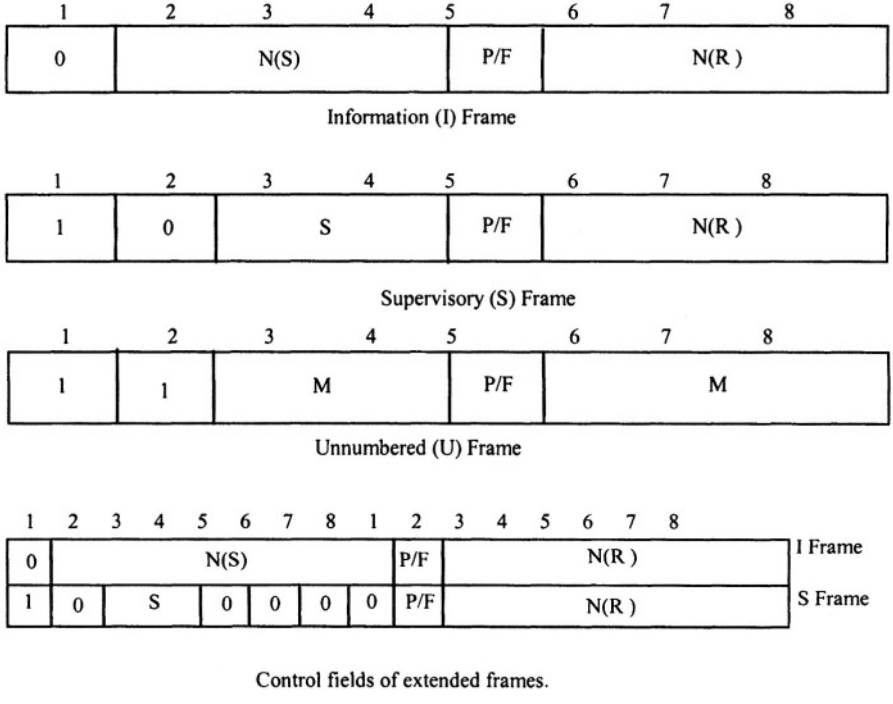


Figure 6-2. HDLC frame types

Here is a description of the fields.

The first bit, if zero, designates the frame to be an I-frame. Otherwise, the first and second bits decide whether it is an S-frame or a U-frame. N(S) is the sequence number, called the send sequence number. This is the sequence number of the frame used by the sender in a window flow or error control mechanism. N(R) is the receive sequence number. This is used in piggybacked ACK mechanism. A frame with N(R) = *j* acknowledges correct reception of all frames until number *j-1*. P/F is called the Poll/Final bit. The terms are borrowed from polling mechanisms in which a commanding station (called Primary Station) polls some attached stations (Secondary Stations) for data. Polling was extensively used with mainframe computers that collected and processed data from a large number of user terminals. In the context of HDLC, the bit is called a Poll bit if the frame contains an enquiry (this makes it a *command* frame). The same bit is called a Final bit if it contains the answer to an enquiry (this makes it a *response* frame).

The 2 S-bits in S-frame define four types of S-frames. These are:

(i). RR (Receiver Ready): To be used as ACK or to initiate temporarily stopped communications. In either case N(R) is the sequence number of the frame expected.

(ii). RNR (Receiver Not Ready). To halt communication temporarily, then to be reopened later (by using RR). It also acts as an ACK, acknowledging all frames sequence numbered up to N(R)-1.

(iii). REJ (Reject). To notify a frame with errors so that the frame and all subsequent frames can be retransmitted, such as in go-back-N ARQ.

(iv). SREJ (Selective Reject). To notify a frame in error so that only that frame can be retransmitted, such as in Selective Reject ARQ.

The I-frame and all the S-frames can be either commands or responses.

The five 'M' bits in U-frame define $2^5 = 32$ possible types of U-frames. Not all of these are specified. Fifteen types of U-frames are defined in Table 6-1 along with their functions and whether each is a command or a response. We will revert to U-frames and their operation at a later stage when we discuss examples of HDLC operation.

Table 6-1. U-frame types in HDLC

C = Command, R = Response

S.No.	U-Frame Type	C/R	Function
1	Set Normal Response Mode/Extended (SNRM/SNRME) {2 frames}	C	Connection request in NRM/E mode
2	Set Asynchronous Response Mode/Extended (SARM/E) {2 frames}	C	Connection request in ARM/E mode
3	Set Asynchronous Balance Mode/Extended (SABM/E) {2 frames }	C	Connection request in ABM/E mode
4	Set Initialization Mode	C	Initialize link control functions
5	Disconnect (DISC)	C	Terminate a connection
6	Unnumbered ACK	R	ACK to set mode commands
7	Disconnected Mode (DM)	R	NAK to set mode commands
8	Request Disconnect (RD)	R	Request for DISC
9	Request Initialization Mode (RIM)	R	Request for SIM
10	Unnumbered Information (UI)	C/R	control info
11	Unnumbered Poll (UP)	C	solicit control info
12	Reset (RSET)	C	reinitialize sequence numbers
13	Exchange Identification (XID)	C/R	for testing
14	Test (TEST)	C/R	loopback type testing
15	Frame Reject (FRMR)	R	unacceptable frame

6.3. HDLC station types

HDLC defines three station types

6.3.1. Primary station

The primary station is the one that controls communications between two stations. This term is borrowed from mainframe computers connected to many terminals. The mainframe acts as a primary station to poll attached terminals for information. The frames sent by primary station are called *commands.*

6.3.2. Secondary station

The secondary station is the one that is controlled by the primary station. Its frames are called *responses.* When secondary station wants to initiate a function, it requests the primary station to perform it.

6.3.3. Combined stations

A combined station may act as primary as well as secondary. Communication between combined stations is usually called *balanced* mode of communication.
Communications between a primary and secondary station is called *unbalanced* mode of communication.

6.4. Operation modes

HDLC defines three (3) types of operation modes. These are: normal response mode (NRM), asynchronous response mode (ARM) and asynchronous balanced mode (ABM).

6.4.1. Normal Response Mode (NRM)

This is an unbalanced mode. The primary station may initiate data transfer to send data or allow the secondary to send data. The secondary cannot send data without prior permission from the primary.

6.4.2. Asynchronous Balanced Mode (ABM)

When both stations are of the combined type, they can use ABM. In this mode any station can initiate or terminate the data transfer. As obvious from its names, this is a balanced mode.

6.4.3. Asynchronous Response Mode (ARM)

In this unbalanced mode of data transfer, the secondary does not need explicit permission from the primary to initiate data transfer. However, the primary is responsible for connection control and error recovery mechanisms.

6.4.4. Extended Modes

The communications modes in which extended frame formats are used are called *extended modes.* All the three modes could be used in extended format, e.g., normal response mode extended (NRM*E*) and asynchronous response mode *extended* (ARM*E*). The characteristics of the extended mode are the same as that of the same non-extended mode. The only difference is that the frames in extended format use 16-bit control fields instead of 8-bit fields. The extended control field allows for the use of 7-bit N(S) and N(R) fields. This in turn has the effect of allowing window size increase due to a possible 2^7 frame sequence numbers.

6.5. The HDLC Frame

There are 5 different fields in the HDLC frame. Control field defines the frame types. The functions provided by other fields are more specific and are discussed in this section.

6.5.1. Flag

Flag is an 8-bit pattern that uniquely identifies the start or end of a frame. The bit-pattern 01111110 reserved for this field. The detection of a flag is an indication of the beginning of a new frame. When a number of frames are sent one after the other, the trailing flag is also the beginning flag for the next frame. There is the possibility of the occurrence of flag pattern in the data field or some other field. If this is allowed then the receiver may take it as the beginning of a new frame. For data transparency b*it stuffing* is used in the following manner. If the 6-bit pattern 011111 occurs anywhere other than the Flag, it is changed by inserting a '0' after the fifth '1'. This will ensure that 01111110 will not occur any where other than the flag. At the receiver, it is assumed that whenever there are five '1's followed by a '0', the '0' has been added externally. So, the receiver will remove the '0'. In this way, reception of 01111110 guarantees the occurrence of Flag.

6.5.2. Address Field

The address field is used to uniquely identify a secondary station or a group of secondary stations. The default length of the address field is 8 bits. This gives a total number of $2^8 = 256$ possible addresses. However, the first

bit is not used as part of address. Only the remaining 7 bits are used for address. A primary station can address a maximum of $2^7 = 128$ secondary stations unless an extended address format is used.

6.5.2.1. Extended address format

The HDLC address field may be extended in multiples of 8 bits. In this format, the first bit of the first address byte has the binary value '0'. This is an indication to the receiver that this is an extended address. The address continues into the next byte with only seven rightmost bits of the next byte used for the address giving a total of 14 bits. The left most bit is again used for the same purpose, that is, to tell whether this is the last address byte or not. A value of '1' is used for the left most bit of the last address byte. This makes it easy for the receiver to decode the address, be it extended or not. Figure 6-3 demonstrates the use of extended addressing. An address of all '1's is the broadcast address.

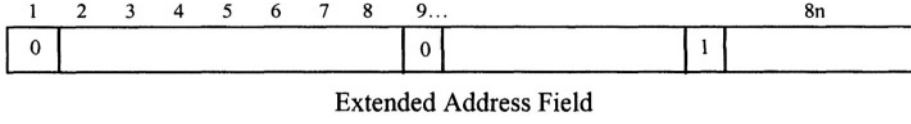


Figure 6-3. Extended address format using n bytes.

6.5.3. Frame Check Sequence (FCS)

The FCS is used for error detection. This field is calculated and appended at the transmitting station by dividing the rest of the frame (except for the Flag) by $1 + X^5 + X^{12} + X^{16} = 1000100000000100001$, called the CRC-CCITT.

The mechanism of calculating FCS is explained in Chapter 5 under error recovery.

6.6. HDLC Protocol Operation

The HDLC is a connection-oriented protocol. Before data frames (I-frames) could be exchanged, the transmitting and receiving stations must agree on the parameters of transmission, such as window size and sequence numbers to be used. In this section, we will look into examples of HDLC operations. These examples will demonstrate the use of (i) U- frames, (ii) I-

frames and (iii) S-frames. The U-frames perform a variety of functions ranging from connection set up, link testing to connection termination (disconnection). The I-frames perform essentially two jobs, information exchange and piggybacked acknowledgements. The S-frames perform jobs such as acknowledging or rejection notification of a packet, sending busy signal. In addition to getting a reject or negative ACK packet, timeout is also used as a way of determining whether a frame was lost.

6.6.1. Selection of Timeout

The timeout mechanism is at the core of any retransmission mechanism. In this mechanism, a timer is set to the maximum waiting time before an acknowledgement is expected for a packet. The duration of timeout depends on how long should it take for the transmitter to get back an ACK if the receiver has received the frame and sent down an ACK. This should account for a round trip of propagation time, transmission times for the I-frame and the ACK packet, and any processing times. Without a timeout, a station may have to wait forever for a response in case a packet gets lost and doesn't make it to the receiving station. When a station sends a packet, it initializes the timeout clock at the maximum value. If the timeout expires, the sender assumes that the packet was not received. It then initiates a retransmission of the same packet. If the timeout occurs again, the retransmission is repeated. If the problem seems to be persistent, a link failure is assumed and link control and recovery procedures may be initiated. Sometime, control is passed to a higher layer in such a situation.

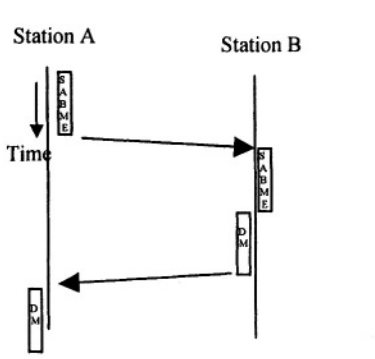
6.6.2. Connection Setup and Termination

U-frames are exchanged under a variety of situations and with many purposes. The very first frames used in order to setup a logical connection between two stations connected via HDLC are U-frames, the set mode and ACK/NAK frames. Here's an example.

Station A wants to setup a connection with station B in order to be able to exchange frames in an asynchronous balanced mode with extended sequence numbers. It will send the U-frame SABME (set asynchronous balanced mode extended). As soon as station A sends SABME frame, it also sets the timer for the timeout period. If the frame is lost or B does not reply, the timeout event occurs and station A may send the frame again. If, however, station B would like to allow the connection, it will send unnumbered ACK (UA) frame back to A. On receiving UA the connection setup is complete and the two stations enter the information exchange phase.

In case of d*isconnection,* either station could originate the process by sending a DISC frame. The receiver of DISC sends UA to acknowledge the

disconnection request. Figure 6-4 shows the exchange of U-frames for this purpose.



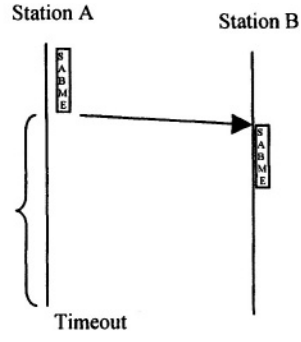


Figure 6-4. Failed connection setup attempt (a) explicit using DM frame, (b) Timeout

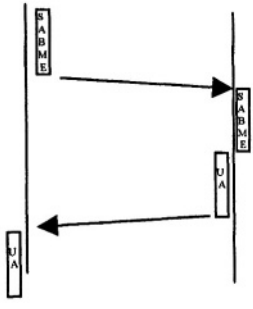


Figure 6-4 (c). Successful connection setup

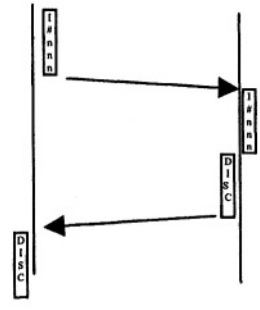


Figure 6-4(d). Disconnection after receiving I-frame #nnn

6.6.3. Data Exchange

Some of the main strengths of HDLC protocol are in its capabilities of multiple duplexity, windows flow and ARQ error control functions. The flow control is provided by RR and RNR frames. These are used together with the N(S) and N(R) sequence numbers. I-frames can also be used for error control in full-duplex connections. Error control function is provided with the help of FCS and sequence numbering. There are four types of S-frames and they are used for different purposes. Here are some examples.

6.6.3.1. Half-duplex Connection

Figure 6-5 is an illustration of two instances of two stations using HDLC protocols in half-duplex mode. It is assumed that the connection has already been setup and stations A and B have negotiated to use a window size of 4 with starting sequence number of 0. A non-extended connection mode restricts the packet sequence numbers from 0 to 7. A go-back-N ARQ is assumed. Station A is the sender of I-frames and station B is the receiver. Both stations maintain a window variable: station A for the send sequence number N(S) of the next outgoing frame and station B for the send sequence number N(S) of the next incoming frame. The window variable at station A is called the send window and the one at station B (receiving station) is called the receive window. At the beginning of the data phase, both variables point at the same number, say 0. In the first case, station A sends I-frame#0 and receives the ACK as RR frame with N(R) = 1. After transmitting frame #0, station A advances the window pointer to #1. Following the reception of RR with N(R) = 1, station A sends out two more I-frames with N(S) = 1 and 2. On receiving the I-frame with N(S) = 2, station B sends out RNR with N(R) = 3. The supervisory frame RNR is acknowledging all I-frames up to #2. It also requests station A to temporarily stop sending I-frames.

In the second instance in Figure 6-5, station A sends I-frame in the same manner. However, this time, I-frame with N(S) = 1 has errors, and is therefore discarded. At this point, the pointer of the receive window of station B keeps pointing to N(S) = 1 for the next I-frame. When an I-frame arrives with N(S) = 2, the station B sends a NAK using RR. The RR frame in this case can be interpreted both as a NAK or ACK. It has N(R) = 1, acknowledging correct reception of I-frame #0 and negatively acknowledging any I-frame with $N(S) \neq 1$. On receiving RR with N(R) = 1, station A concludes that I-frame #1 did not make it to station B. Therefore, it resets its window pointer back to #1 (go-back-N) and starts sending I-frames again.

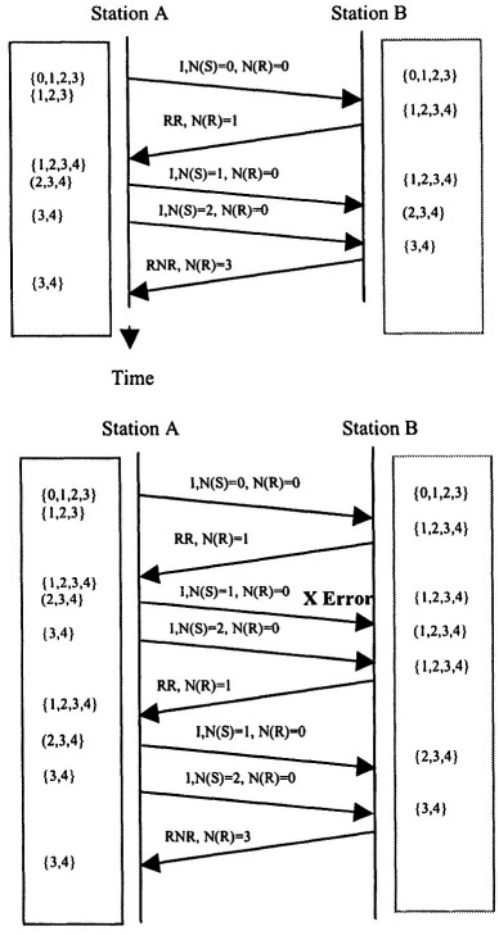
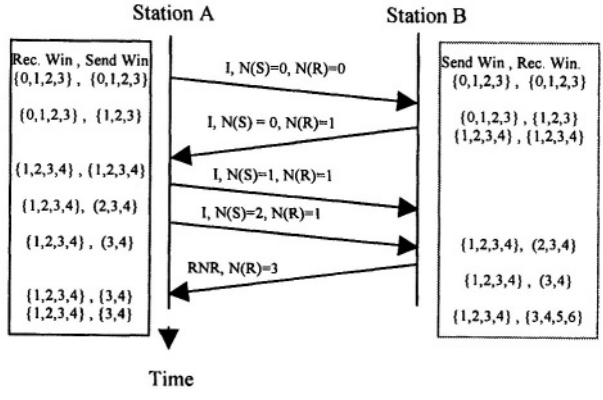


Figure 6-5. Examples of data transfer in half-duplex mode. On the top, station A transmits and station B acknowledges I-frames using RR and RNR using a Window size of 4.
On the bottom, same thing shown except that the first transmission of I-frame#1 is in error and station B sends a negative acknowledgement of I-frame #1 using RR. The values in the magenta and orange boxes show the send and receive window values respectively.

Full-duplex Operation



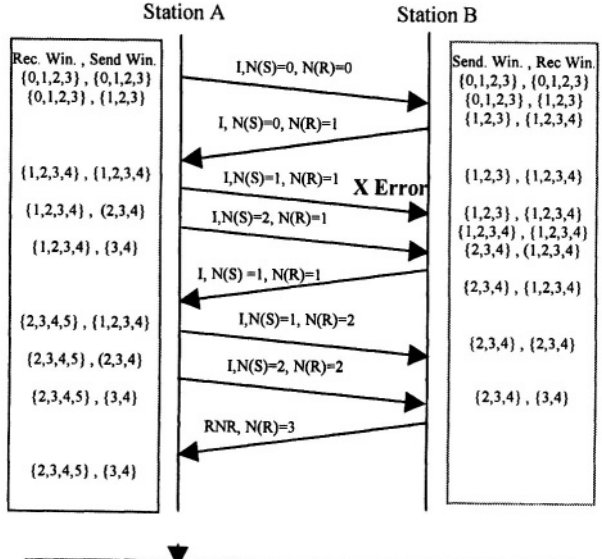


Figure 6-6. Examples of data transfer in full-duplex mode. On the top, station A transmits and station B acknowledges I-frames using RR and RNR using a Window size of 4.
On the bottom, same thing except that the first transmission of I-frame#1 is in error and station B sends a negative acknowledgement of I-frame #1 using RR. The values in the magenta and orange boxes show the send and receive window values respectively.

The full-duplex operation is just like the half-duplex except for a few differences. In this case, in addition to carrying data, an I-frame could also be used to acknowledge a correctly received frame. The I-frame field N(R) = *j* acknowledges all I-frames up to the one with N(S) = *j*-1. It is also a request to the other station to send I-frame *#j*. RR can also be used to acknowledge I-frames in full-duplex mode. Finally, RNR can be used to acknowledge I-frames asking to halt any further transmission of I-frames. It may be noticed from Figure 5-6 that each station has to maintain two windows in full-duplex mode, the send window pointing to N(S) for the next outgoing I-frame, and the receive window pointing to N(S) of the next incoming I-frame.

6.6.3.2. Use of RR and RNR for Busy Condition Notice and Recovery

The use of RNR by station B in Figures 6-5 and 6-6 may indicate that station B was temporarily busy. When B is available for receiving more I-frames, it will send RR to A. If A wants to check on B whether it is available or not, it could send an RR packet with P = 1. If B is busy, it will respond with an RNR with F = 1. When B is not busy, it will send RR with F = 1.

6.6.3.3. Use of REJ and SREJ

REJ and SREJ are negative acknowledgement or NAK packets. They are used to report that the packet number mentioned was not received and instead a higher sequence number may have arrived. REJ is used with go-back-N ARQ, while SREJ with selective reject ARQ. After receiving an REJ frame the transmitter retransmits all packets starting with the sequence number of the packet that was lost or was in error. In the example of stations A and B, suppose that A has transmitted frame numbers 2, 3, 4, 5 and is waiting for an ACK. I-frame number 3 is lost and B gets number 4 after number 2. On receiving packet number 4, the station B will send REJ3 indicating that it is expecting packet number 3 and above. B will not receive frames #4 and above until it has received #3. On receiving REJ3, station A will send packet number 3, 4 and 5 again.

In case of SREJ, only the rejected packet number is to be retransmitted. In the above example, if A sends packet numbers 2, 3, 4 and 5 and if number 3 is lost then B will send SREJ3 on receiving packet number 4. However, it will not discard packets numbered 4 and above even if they arrive before the correct reception of #3. On receiving SREJ3, station A will retransmit packet number 3 only. On receiving packet number 3, station B can acknowledge all packets numbered 2 through 5 and send an ACK for packet number 6.

Out of the three operation modes (NRM, ARM and ABM) the ABM is mostly used in modern networks. This is because it allows any station to initiate or terminate a connection. LAP-B, a subset of HDLC that uses only

the ABM has been specified for X.25, a public packet switched network. LAP-B stands for link access procedures balanced and is also specified for bearer data channels in ISDN. An amended and upgraded version of HDLC called LAP-D (link access procedures for channel D) is specified for control channel (D) of the ISDN. HDLC has dominated the scene for medium speed serial communications links. With the advent of optical networks, channel reliability and data transmission speeds have jumped several orders of magnitude. The next example protocol (ATM) is specified keeping in view optical networks. Due to a lack of mass market for SONET/SDH data rates, ATM has been made available for existing, slower networks as well.

6.7. Asynchronous Transfer Mode (ATM) Protocol

During much part of the 90s, the ATM protocol has perhaps got more attention of researchers and networking industry than any other networking protocol. ATM was conceived as the ultimate solution to the integration of real-time and non-real-time applications on a single network. The developments in ATM took various forms as described in the information box under 'ATM technology'. However, it seems to be destined to being nothing

ATM Technology

Originally conceived as a layer-2 protocol for B-ISDN, ATM has assumed various network forms. This is perhaps because of the existence of VCI/VPI in an ATM cell Due to its switching capability based on VCI/VPI, ATM automatically can be considered as a network layer protocol. Since VCI/VPI also provide a mechanism for sharing a common link. ATM LANs have been realized by connection a few ATM switches in close proximity from one another with each switch having more than one ATM ports. ATM Network consists of high-speed links connected via ATM switches and computers equipped with ATM network adaptor. ATM backbone is an ATM network that carries traffic from many networks. Wireless ATM (WATM) is a wireless data network using ATM backbone to connect basestations. The common factor among all these network and equipment tvpes is the ATM protocol, consisting of 53-octet cells and 5-octet header providing a small, reasonably comprehensive set of functions.

more than an internetworking link layer protocol for Internet backbone.

6.7.1. The ATM Cell

The functions provided by the ATM protocol are easily understood from the five-octet header shown in Figure 6-7. We will discuss each field briefly in the following.

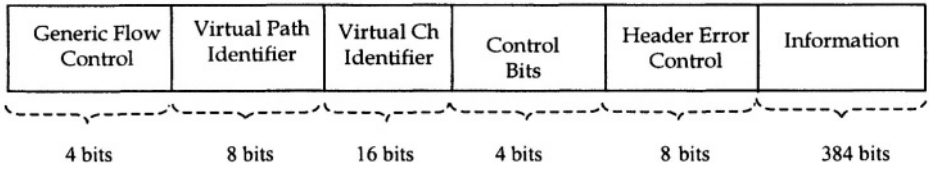


Figure 6-7. ATM cell structure

6.7.1.1. Generic Flow Control (GFC)

This field is present only in cells at the user-network-interface (UNI). Inside the network (usually referred to as the network-network interface or NNI), the first 12 bits (including the 4 bits of generic flow control) constitute the Virtual Path Identifier (VPI). Therefore, the GFC function is restricted between the traffic source and the call admission node. By doing so, the network is expected to have traffic that already meets the flow control limitations. However, the flow control on any link is needed to avoid not only overflow of storage facilities at the receiver, but also to restrict the traffic on a link to within the link capacity. ATM flow control can't be applied in a single unique way. The following example illustrates this problem.

Consider a scenario shown in Figure 6-8. In this Figure, a link *l* receives traffic from 3 incoming links and can direct it to one or more of the 3 outgoing links. The incoming traffic arrives in random patterns. So, the total instantaneous traffic on the link fluctuates. Sometimes, it may even exceed the link capacity. That is why the flow control need may arise. In other situations, the link *l* may be replaced by a switching or routing node. Viewed in this way, flow control is needed so that:

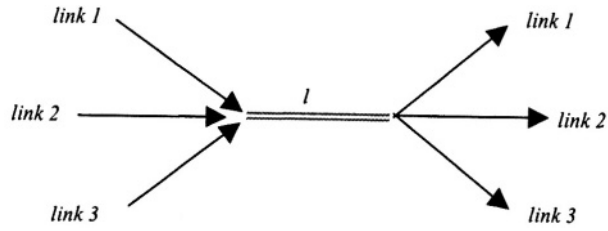


Figure 6-8. Example of a shared link. The link receives traffic from 3 incoming links and sends out to one or more of the three outgoing links. Arrows point to the traffic direction.

1. The buffers in the switching node may not get congested.
2. Processor may not be fast enough to process all the received cells quickly.
3. The outgoing links may not be speedy enough to clear all the buffers.

If the traffic is regulated only between the user-network interface, congestion can still occur inside the network.

If a single flow control mechanism is used then all packets will be treated in the same way inside the network. That means that the network can't differentiate among traffic from a quality of service (QoS) point of view. This is against the very essence of ATM protocol - it is to provide a variety of QoS types. This problem is resolved by defining QoS classes in an ATM network. These classes are allocated network resources based on their requirements. Many QoS classes have been defined by the standardization agencies. Calls belonging to each category have their own admission control and resource allocation procedures. These procedures are defined as part of UNI signaling protocol. An example UNI signaling protocol is the ITU's Q.2931 that is also specified as part of ATM forums specifications. The GFC field itself has not been extensively used for flow and congestion control.

6.7.1.2. Virtual Path/Channel Identifiers (VPI/VCI)

Every ATM cell is stamped with a VPI/VCI pair. These IDs are unique to a call. They help in switching cells and sharing the link. Link sharing is made possible by their use because an ATM switch can easily differentiate among calls with their help. They help in switching because every ATM switch could maintain a table for each incoming link with three columns: one for incoming VPI/VCI, one for outgoing VPI/VCI and one for outgoing link. Consider Figure 6-9.

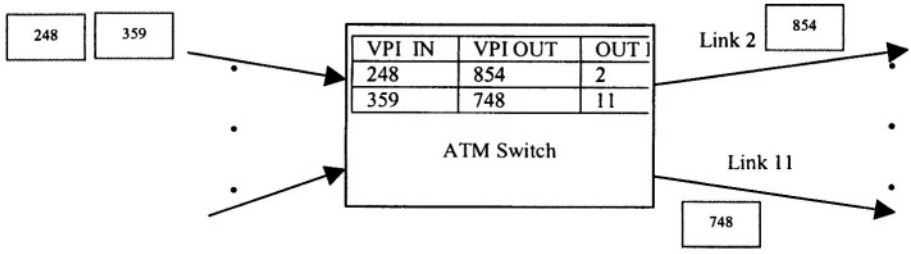


Figure 6-9. Use of VPI for switching.

The figure shows an ATM switch using VPI-based switching. In this case, the value of VCI is not allocated or needed for switching. Since switching and routing is the main function of the network layer, one may wonder whether an ATM switch implements layer 2 or 3. The answer to that question is that the switch performs the layer 3 function during call setup and cell switching while it performs layer 2 functions of error control. In the call setup phase, the source-destination addresses help each switch to find the appropriate path. Based on the path availability, VPI or a pair of VPI/VCI is allocated and passed between adjacent switches. This '*signaling*' mechanism helps each switch maintain a link-by-link path for each call.

Inside the network, a total of 2^{12} virtual paths, each with 2^{16} virtual circuits can be defined for each link. This gives ATM network an extremely powerful multiplexing capability. In earlier ATM switches VPI-based switching has been employed. Once ATM becomes a desktop technology, VCI-based switching may be more in demand.

6.7.1.3. Control Bits

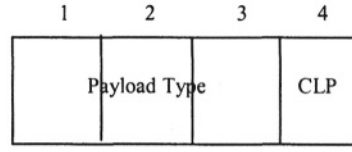


Figure 6-10. The control field.

The 4 control bits provide two functions, namely, payload type (3 bits) and the cell loss probability (1 bit). See Figure 6-10 for organization of the
control bits into payload type (PT) and cell loss probability (CLP). PT helps differentiate among $2^3 = 8$ types of ATM payload. For example, a certain value of PT can be use if it is network control data. Network control data may be used for link control and Management functions. A different value of PT could be used for user data. Potentially, up to 8 types of network and user data can be defined.

The cell loss probability (CLP) bit has two potential uses. One is to define two types of ATM connections: high-priority and low-priority. For example, a CLP = 0 may be used to mean that the ATM cell carrying this value has a common priority. It should not get any special treatment, while a CLP = 1 could be used to mean that the cell has high priority and should be given preference over cells with CLP = 0. The second use of CLP is to mark or tag certain cells for flow control. Here's how this can be achieved.

Cell Tagging: Since ATM is projected to provide telephone-like quality to Internet-like (bursty) traffic, there is a need to monitor traffic into the ATM network. This function is provided at the UNI and is called *policing.* The policing function consists of a mechanism to check whether a source is obeying a traffic contract between the source and the network. This contract is agreed upon during the call setup phase. Due to the random nature of traffic generation, the agreement can't be usually obeyed 100% time. For this purpose, all incoming cells are first stored in a buffer and then let out of the buffer at a speed agreed by the network and the traffic source. This is shown in Figure 6-11 and is variously called Leaky Bucket (LB) or Generic Cell Rate Algorithm (GCRA).

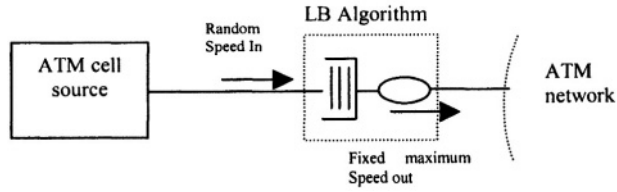


Figure 6-11. The Leaky Bucket algorithm for policing.

In this policing mechanism, the cells that violate the speed limit could be treated in a variety of ways. They could be discarded summarily, or simply tagged. The tagged cells could be discarded later in case of congestion inside the network. The CLP bit may be used for tagging the violators. In this way, it may help the network to relieve congestion.

6.7.1.4. Header Error Control (HEC)

The fifth octet of the header field, called header error checksum (HEC), is a parity check octet. HEC carries the information about the parity of the header fields. It uses CRC to detect and correct errors. A single bit error in the header can be corrected. A double bit error can be detected but not corrected. Another use of HEC is *cell delineation.*

Cell Delineation: The ATM cell does not have synchronization information, such as a flag. Transmission convergence (TC) sublayer of the physical layer detects the ATM cell boundaries. It does so by using the HEC field. The physical layer itself uses larger frames that can carry many ATM cells. These cells have to be delivered to the next ATM layer one-by-one. Before the PHY can deliver cells to the ATM layer, it checks for HEC of many consecutive cells, assuming no errors. This is possible due to two reasons:

1. The remainder after passing the header through a CRC circuit is all-zeros when there is no error.
2. Since all ATM cells have equal length (of 53 octets), once a cell has been delineated, boundaries of subsequent cells are easily determined.

In this way, the HEC field helps the PHY-TC to delineate ATM cells. See Figure 6-12 for detail.

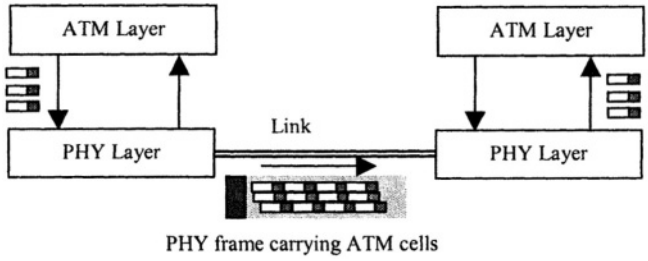


Figure 6-12. The need of cell delineation to deliver ATM cells by the PHY

6.8. ATM Protocol Procedures

The five-octet header of an ATM cell provides a fairly short set of functions. To summarize, the ATM protocol provides two main functions: VPI/VCI and HEC. The PT and CLP can be used to provide additional functions. Originally, ATM was to be the layer 2 over highly reliable optical PHY standards. Reducing the protocol functions was highly desirable to meet the stringent quality of service (QoS) requirements of real-time applications. In any case, many of the necessary functions are provided by the network through a contract negotiation and enforcement. For example, classification of traffic into five service categories allows for reserving network resources according to QoS of a category.

Between the two main functions of the ATM layer, the VPI/VCI allocation procedure is briefly described in the section above.

6.8.1. Virtual circuit and the frame relay protocol

The concept of virtual circuits was first used in the X.25 network architecture at the packet layer procedures (equivalent to layer 3 of the OSI-RM). At layer 2 of the X.25, a subset of HDLC, called LAP-B, is used. LAP-B, like HDLC, has several point-to-point functions. These functions increase the link reliability at the cost of link utilization. Another variation of HDLC, called LAP-D, was recommended to be use for ISDN. An advanced form of LAP-D, called frame relay, eliminates some of the layer 2 functions to define a swift and simple protocol. This protocol rendered the intermediate nodes nothing more than relay stations. The result was a throughput improvement from 56 kbps to 1.54-to-2 Mbps. The success of frame relay owes to the elimination of much of the node-to-node processing. It uses HDLC-like frame structure with the control field replaced by a new ‘frame relay’ field depicted in Figure 6-13.

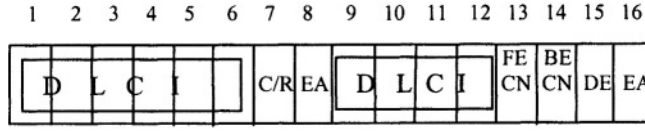


Figure 6-13. DLCI in Frame relay

The data link control identifier (DLCI) in frame relay set the stage for the future of virtual circuit based routing. It consists of the destination address of the frame. The frame relay switching nodes do not have to allocate a permanent physical path for each call. They use a table-look up method described above (see Figure 6-9) and use only virtual numbers to switch and route frames to the destination. In the actual frame relay standard, specified as ANSI T1.618, an extension of address is allowed to another 16 bits. ATM extended the same concept with further reduction of functions, and a small, constant cell size. With reference to Figure 6-13, some other functions provided by frame relay are as follows:

C/R: Command/Response bit.

FECN/BECN: Forward/Backward explicit congestion notification. Together, these two bits eliminate the need of link-by-link flow control. FECN is a single bit used to notify the destination that network is experiencing congestion. On receiving FECN message, the destination may inform the traffic source about the congestion. This can be done using an ACK. with the BECN 'ON'. Alternatively, any intermediate node, on sensing congestion, can activate the BECN bit on a packet going towards the traffic source (backward).

DE: Discard eligibility function that allows the user or network to tag lower priority frames. These frames could be discarded in case of congestion.

EA: Extended address bit allows for the use of address extended another 16 bits.

6.8.2. Error Control

HEC can correct one error in the ATM cell header and detect 2 errors. The detection procedure is CRC using $1+X^2+X^4+X^6$ as the CRC polynomial. The equivalent binary value is 1010101. The information field is not error controlled by the ATM layer. Due to high-speed, HEC is all done in hardware. Since the header is also used in cell delineation, the information field is scrambled in order to prevent the occurrence of HEC field value in the information field.

6.9. Medium Access Control (MAC) Layer for IEEE Wireless LANs

The wireless LAN architecture is similar to a broadcast type fixed LAN. In both cases, there is a direct connection among all stations, requiring only the physical and data link layers. All the stations in the network receive every packet of data transmitted. However, only the intended recipient is expected to actually process the packet and respond to it. This may be accomplished either by establishing a relation of 'trust' among the stations, or by making it physically impossible for an intruder to interject the network traffic. In fixed LANs, it is possible to keep the intruders away rather easily, as the transmission medium can be physically monitored. However, in WLAN such is not the case. The wireless terminal does not have to be used visibly, it could be in a pocket, a bag, or even outside a room. Similarly, power in a fixed LAN is easily provided from the mains. Such is not the case with a wireless terminal. The power source is a part of wireless terminal and, therefore, must be used judiciously. These and many other wireless-specific factors make the wireless MAC quite complex. However, the common functions of the WLAN and fixed LAN MAC are not less complex. For example, in both cases, there is a chance that two or more stations try to broadcast their packets simultaneously, resulting in mixing of packets before arriving anywhere. This mixing due to simultaneous transmission is termed as a packet *collision.* One simple way to avoid such collision is to regulate transmission of packets. This has been successfully done using *polling in* mainframe computing systems. In a typical mainframe computer, a large number of terminals are connected to a single processing system. The computer operating system polls the terminals in some specified order asking if they have any data to send. This avoids the occurrence of any collision.

Another mechanism to prevent collisions is to use a special packet, called a *token,* to be captured before a station can transmit a data packet. In these *token passing* mechanisms, if a token is unavailable, it is assumed that some station is using it. Once the station with token has finished transmission of its data, it releases the token on the network medium. The token is then free for the next specified station. There are many other mechanisms that might prevent collisions. But, all such mechanisms suffer from two potential problems: number one, they utilize too much network bandwidth in regulating the transmission; number two, they require a controlling station. In a mainframe computer, the mainframe provided a good point of control. However, in LANs, where all user terminals are same, or similar, implementing control can easily lead into maintenance situations. Due to these reasons, *controlled access mechanisms* in LANs have not been very popular.

Contrary to the controlled access mechanisms, the *random access,* or *contention-based access* mechanisms are used more often in wireless and fixed LANs. This necessitates a discussion on such medium access techniques before we describe the salient features of the IEEE WLAN MAC sublayer.

6.9.1. Random Access in LANs

As opposed to the controlled access, stations using random access mechanisms do not have to wait for their turn. In the simplest of such mechanisms, a station with a data packet will try to send it right away. This is the definition of *ALOHA,* the first and the most important medium access mechanism in shared medium networks. ALOHA was invented for a wireless network in a university (University of Hawaii) by a Professor (Dr. Abramson Norman). Its simplicity allows trading in complexity for added bandwidth utilization. Naturally, there are more efficient forms of ALOHA, of which *slotted ALOHA* needs special mention.

In slotted-ALOHA, or S-ALOHA, the principle is the same (transmit whenever you have something to transmit), but the timing is made stringent by allowing the exact transmission only at the beginning of a specified time slot. This improves the bandwidth utilization to almost twice over that of the p*ure ALOHA*. However, it still leaves open two or more stations to collision as long as they can have data packets generated during the same slot. An advancement to the ALOHA schemes is to require stations to sense the channel before transmission and see if it is idle. This channel sensing is termed as *carrier sense.* The multiple access mechanisms that employ carrier sense are called as *carrier sense multiple access (CSMA).*

CSMA mechanisms have a variety of possible implementations. The carrier sensing can be used not only for sensing the channel before transmission, but after transmission as well. In fixed network, the signal level can be detected by using simple electronics. Consequently, if two packets collide, the raised signal level is considered to be the result of a collision. Such mechanisms that implement carrier sense as well as collision detection are called *carrier sense multiple access with collision detection (CSMA/CD).* The text box in the following shows example LANs using various random access mechanisms.

LAN Examples

IBM's Token Ring uses a token passing protocol at the MAC sublayer. It is very similar to a standard specified by IEEE committee 802.5. Another IEEE LAN standard using token passing mechanism is the IEEE Token Bus, specified in IEEE 802.4 standard. The first packet radio network used ALOHA and it is still used in many wireless networks for accessing the call setup channels. Examples are the cellular systems, American Mobile Phone System (AMPS) and the Code Division Multiple Access (CDMA) standard in North America. Satellite systems have used either reservation-based protocols, trading in more efficiency in exchange for more complexity, or sometimes S-ALOHA. The most popular LAN standard, that is Ethernet (more correctly, IEEE802.3) uses CSMA/CD.

6.9.2. Collision Avoidance

The IEEE WLAN 802.11 has been specified by keeping in mind the popularity of the Ethernet. Ethernet uses CSMA/CD. It is, however, not practical to implement carrier detect mechanism in a wireless environment. This is primarily due to the rapid reduction in carrier amplitude as a function of distance in the air (*see Figure 4-1, Chapter 4 for example*). Channel sensing is, however, possible. Consequently, the IEEE WLAN MAC specifies the use of CSMA. Instead of collision detection, the standard specifies *collision avoidance*. The channel sensing mechanism is called CSMA/CA or carrier sensing multiple accesses with collision avoidance. Due to this closeness to Ethernet, the IEEE WLAN has also been dubbed as wireless Ethernet. In the following paragraph, we describe the collision avoidance mechanism.

Suppose that a station 'A' has a packet to send. The collision is avoided by making sure that if another station, 'B', is sensing the channel longer than 'A', then 'A' may not transmit before station 'B'. This is easily achieved by requiring 'A' and 'B' (and all other) stations to sense the channel for a specified, fixed, amount of time. Suppose that the required sensing time is 20 msec. In this case, if 'B' started sensing the channel 10 msec, before 'A', then after passing only 10 msec., 'B' will transmit its packet. Since 'A' is still sensing the channel, it will know that the channel is not idle any more and will go into some waiting state before it starts sensing the channel again. Refer to Figure 6-14 for the timing example.

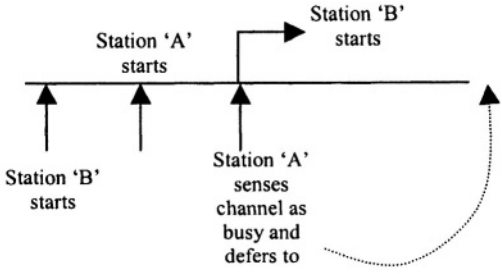


Figure 6-14. Example of collision avoidance by sensing channel for a fixed amount of time.

This waiting time is called the *back-off time*. It is possible that in high load situation, another station senses the channel busy, backs off, and comes back to find it busy again. This may lead into many stations coming back from backoff time together if the backoff time is fixed. To avoid such a situation, the backoff time is randomized to be between two limits (a minimum and a maximum). To further disperse simultaneously generated packets, the maximum limit of backoff time is increased every time there is another backoff. In fact, the backoff time in IEEE standard is doubled with every backoff event for the same packet. This doubling of backoff time with every sensing of busy channel is called *binary exponential backoff*. The IEEE WLAN specifies a CSMA/CA with binary exponential backoff. All stations coordinate in this way to provide MAC sublayer functions. These functions are grouped under the termed 'distributed coordination function (DCF)'.

6.9.3. The Distributed Coordination Function (DCF)

IEEE WLAN standard uses the term DCF to describe the CSMA/CA mechanism and its associated functions. The concept of coordination is extended to include some regulated transmission at the top of DCF. The regulated transmission is provided through the point coordination function (PCF). DCF provides channel time reservation for PCF. The PCF uses this reserved time for delay-bound traffic using polling. The *delay-bound* traffic is one that is sensitive to delay and its variation. An example of delay-bound traffic is the human voice signal. Thus, the DCF has been specified keeping in mind the *multimedia* communications. Multimedia refers to traffic with multiple requirements of delay and frame error rate (FER). An example of

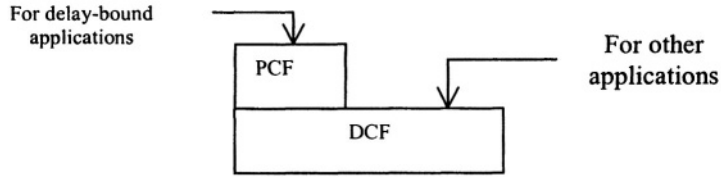


Figure 6-15. Delay-bound applications use DCF through PCF.

multimedia call is a videoconference that uses pictures as well as voice. In its full implementation, the access protocol is as shown in Figure 6-15 below.
In order to implement collision avoidance, the standard specifies interframe spacing (IFS).

6.9.3.1. Interframe Spacing (IFS)

Each packet transmitted using DCF obeys the IFS rules specified in the standard. The amount of IFS may have one of the four values depending on the following factors.
(i) When the packet is the single data packet generated in a station, or first of a series of data packets, then the IFS is some typical value, called DCF/IFS or DIFS.
(ii) When the packet is a short priority packet, acknowledgement of a data packet or warning all stations of a reservation/confirmation of reservation, the IFS for such packets is shorter than DIFS and is accordingly called short IFS or SIFS.
(iii) If a packet is to be transmitted by a delay-bound station during the time reserved for point coordination function (PCF), the IFS is again shorter than DIFS but larger than SIFS. This is called PCF/IFS or simply PIFS.
(iv) If the PHY indicates to the MAC that a frame could not complete transmission or was in error, an extended IFS is used to allow for the recovery of the lost packet. This is basically DIFS with an extended value. It is called EIFS, or extended IFS.
The probability of collision is reduced substantially due to the IFSs.

6.9.4. MAC Frame Structure

Jus like any link layer protocol, many of the functions of IEEE WLAN are provided through the protocol information field of the MAC PDU (MPDU). There are several types of MPDUs defined in the standard. They all have the same general format consisting of a header, data part and trailer. Figure 6-16 shows the MPDU structure.

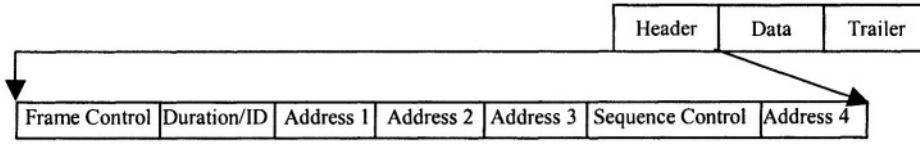


Figure 6-16. PMAC for IEEE WLAN. The trailer consists of 32 bits of frame check sequence using IEEE CRC-32.

Following is the definition of each field of the header and trailer.

Frame Control consists of 16 bits divided into 11 different fields, each providing a different function. These include protocol version, frame types, transmission directions, fragmentation, retransmission, power management and privacy information.

Fragmentation and Re-assembly: When a data packet is too long for a single transmission, MAC layer at the transmitting station may divide it into smaller packets, each called a fragment. The MAC at the receiving station can reassemble the fragments into the original data packet.

Power Management is provided by defining 2 power modes, the Power Save Mode and the Active Mode.

The Address Fields define various addresses, such as the basic service set (BSS) ID and the transmitting and receiving station IDs.

Sequence Control provides the fragmentation and sequence number functions.

Data depends on frame type. Several frame types have been defined as described in the next section.

Trailer consists of the remainder obtained from the modified MPU divided by the IEEE CRC-32 polynomial $1+X+X^2+X^4+X^5+X^7+X^8+X^{10}+X^{11}+X^{12}+X^{16}+X^{22}+X^{23}+X^{26}+X^{32}$. The modification of the MPDU for CRC calculation results in a known, non-zero remainder at the receiver when the received packet is error free and divided by the same polynomial. The non-zero remainder has the advantage of the all-zero remainder in that it has less dc bias and more clocking information for some data encoding schemes.

6.9.5. MAC Frame Types

A large number of MAC frames provide various types of services. These types are enumerated in the following:

(i) The data frame
(ii) Management frame
(iii) ACK frame
(iv) Poll frame
(v) Beacon frame

Etc.

In short, the complexity of the WLAN is enhanced due to the following reasons:

(a) The medium has highly variable characteristics.
(b) Power and bandwidth are at a premium.
(c) It interacts through a fixed network that could be using different protocol architecture.
(d) Mobility adds complexity.
(e) Provision of multimedia adds complexity.

The performance of the early versions of IEEE WLAN has been questionable for voice and data integrated services. However, later versions provide a technology solution to the integration problem. The earlier versions provide bit rates of 1 or 2 Mbps while the scene is set for bit rates in access of 50 Mbps at the writing of this book.

6.10. Review Questions

1: If SREJ can be used in HDLC and is more efficient, then what is the need of REJ?
2: Under what conditions the bit#5 of the HDLC control field is called a (i) poll bit, (ii) a final bit?
3: Give an example circumstance in which an extended window size will be preferable to a non-extended one?
4: What frames (of the I, S an U) can be used (i) User information transmission, (ii) acknowledgement (iii) recover from busy condition
5: Is piggybacking provided in the IEEEWLAN MAC layer?
6: We learnt that in IEEE WLAN MAC, PCF is provided at the top of DCF, then how can there be contention-free communication?
7: If carrier can be sensed in wireless medium, why can't the collision be detected?
8: ATM standard has both error correction and detection, while HDLC does not have error correction. Is there any advantage of having error correction in an ATM network?
9: What are the benefits of (i) small size and (ii) fixed size of ATM cell?
10: What type of switching is more complex: VCI-based or VPI-based?

7. Multiplexing and Carrier Systems

In earlier chapters, we learnt that the channel bandwidth must be greater than or equal to the signal bandwidth for a successful transmission of a data signal. The channel bandwidth sometimes limits the ability of the user equipment to transmit information. Historically, there have been three ways used to achieve higher data rates. One simple way is to use channels with higher bandwidths. The second mechanism is to design modulation and coding mechanisms to use the available bandwidth more efficiently. The bandwidth efficient modulation and coding schemes result in higher data rates per unit bandwidth, thus increasing the data rates achievable in a channel for a given channel bandwidth. This technique has resulted in gradual increase in the data rates of the telephone line MODEM. The third important factor contributing to the increased data rates is the improvements in cable manufacturing. This has helped in many ways and can be considered as a part of the first mechanism. Not only higher bandwidth cables are available these days, but also there are ways to allow higher data rates on the already existing cable types. This is possible due to the developments in manufacturing cables with fewer impurities and ones using special circuitry to undo many channel impairments. The cable technology has influenced telecommunications to a point that even new protocols have been introduced with reduced processing. An example of such protocols is the frame relay technology that minimizes processing at layer 2 in order to achieve higher end-to-end throughputs. Most of the long haul telecommunications transmission systems, however, use channels with much higher bandwidths than a single user signal would need. Many users share each of these high-speed channels. *Multiplexing* is the mechanism used for channel sharing. In this Chapter, we will look at the difference between two types of transmissions once more, the digital and analog transmissions. We will then define multiplexing schemes that can be used with either type of transmission. In the end, we will look into digital multiplexing in greater detail and discuss carrier systems using digital multiplexing. A *carrier system* is the term used to describe the transmission systems typically used for long-haul communications for private and public networks. These systems provide a set of standard bandwidths or data rates from which a user can choose. The equipment is designed according to the carrier system it will use, conforming to the signal and transmission formats.

Multiplexing is mostly discussed with reference to the physical layer of the OSI-RM. The fact is that all layers make use of multiplexing. It is one

of the functions that can be provided at any layer to open more than one simultaneous connections. Our main focus is on multiplexing at the physical layer, but we will also include a section on multiplexing at other layers.

7.1. Analog and Digital Transmissions

A channel with a bandwidth ***B*** Hz would allow a signal of bandwidth ***B*** Hz or smaller to be transmitted through it. Sometimes the signal must be modulated in order to be able to travel to longer distances. The channel bandwidth can be used in two ways. The first method of using the channel bandwidth is by transmitting a signal that may vary continuously with time. This can either be the information signal or the modulated signal. The job of modulation is to transfer the signal frequencies to a higher level where the effect of attenuation is reasonably reduced. The transmission of a signal varying continuously with time is called *analog transmission.*

An alternative way is to code signal in some discrete time pulses by using mechanisms such as PCM and then transmit these pulses. When binary coding is used, the pulses are called binary pulses and each pulse is said to represent one bit. The speed at which the bits can be transmitted is represented in units of bits per second. Such a transmission is called a *digital transmission.* In digital communications, the speed of communications, sometimes called data rate, is closely related with the bandwidth. One key factor that separates many forms of digital transmission schemes is their bandwidth efficiency. *Bandwidth efficiency* η may be defined as the number of bits transmitted per unit bandwidth. For example, if $\eta = 1$, then the data rate at which a signal may be transmitted on a channel is same as the channel bandwidth. For $\eta = 2$, the data rate is twice the bandwidth.

Sometimes, we regard the data rate as the bandwidth of a digital channel even though they are not exactly same. Digital transmission carrier systems are characterized by their data rate and analog carrier systems are characterized by their bandwidth. Generally, the bandwidth of long-haul communication systems is much higher than used by a single user. Therefore information from a large number of users is combined over a single transmission carrier. The combining of signals to transmit over a single channel is called *multiplexing.*

7.1.1. Analog and Digital Multiplexing

The term *multiplexing* refers to the transmission of multiple communication sessions on a single physical path. In other words, the channel bandwidth/data rate gets *divided* among a number of users. In analog multiplexing, it will be the bandwidth that is divided; while in digital multiplexing, it is the channel data rate that is divided. The terms analog or digital are non-standard because of the variety of ways data rate can be

divided, and because of the fact that analog multiplexing does not ascertain that the signal will be transmitted in analog form. The more standard terms used are *frequency division multiplexing* (FDM) for analog signals and *time division multiplexing* (TDM) for digital signals. As we will see from the definitions below, they are not entirely different from each other.

7.1.2. Frequency Division Multiplexing (FDM)

In FDM, the channel bandwidth is divided into many smaller frequency bands and one or more of these bands are allocated to a single user. For example, speech signal needs a bandwidth of about 4KHz. If the channel bandwidth is 48 KHz then it can be broken into 12 voice bands (voice channels), each with a bandwidth of 4 KHz. In this way, 12 voice calls can be multiplexed on a channel with a bandwidth of 48 KHz. Generally, FDM is automatically considered to be using analog transmission, even though that does not have to be the case. FDM is still in heavy use, but is hardly considered for analog transmission in new standards[7]. It is simple in concept and has been very popular in speech and video communications. All public broadcast systems still use FDM. The main 'cons' of using FDM are,

1. If there are N traffic sources multiplexed using FDM, each one of them uses different frequency band. This is evident from Figure 7-1. This results in requiring different electronics for each user. The *bandpass filters* used for separating modulated signals from one another may also introduce distortion at the band-edges of the signals.
2. It is a highly inflexible system in that the equipment is designed for one type of bandwidth division.

[7] Some wireless communications standards specify FDM as a multiplexing of choice with digital transmission used in each FDM channel.

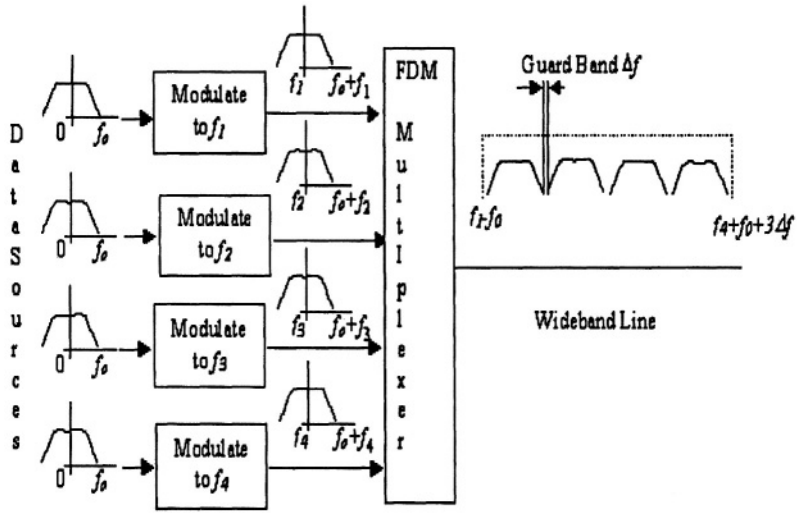


Figure 7-1. Each signal in FDM is modulated with a different carrier frequency.

As seen from the above figure, the signal from each source has similar frequency profile before modulation. The bandwidth of each signal is f_o before modulation. The channel bandwidth is a little more than 8 times the signal bandwidth. Each signal is modulated by employing a different carrier frequency. The carrier frequencies are shown as f_1, f_2, f_3, and f_4 The minimum difference between carrier frequencies of two adjacent signals should be equal to twice the bandwidth of each signal plus a small amount called as *guard band.*

Guard band (denoted by Δf in the figure) is the excessive bandwidth used to separate adjacent signals so that they do not interfere. Interference is possible because of two reasons. First, actual signals are not strictly band-limited. Second, the filters used to separate each band are non-ideal and always allow some crossover between adjacent frequency bands.

7.1.3. Frequency Division Duplexing (FDD)

When two stations are exchanging information, each station is the transmitter and receiver of data. There are two directions in which data is transmitted. Therefore, a channel is needed in both directions. FDD is the term used to imply dividing a frequency band into two so that one band can be used in each direction of communications. Even though FDD is a type of FDM, it is usually considered separately. The term FDM is used to imply division of frequency band in one direction of transmission. Many times. FDD could be used either for FDM systems or TDM systems. It simply provides two simplex channels that could be used for analog or digital communications. Many current digital cellular systems make use of FDD.

could be used either for FDM systems or TDM systems. It simply provides two simplex channels that could be used for analog or digital communications. Many current digital cellular systems make use of FDD. With the bandwidth divided into two, the FDM or TDM may be employed in each of the two frequency bands.

7.1.4. Time Division Multiplexing (TDM)

Each user in TDM uses all of the bandwidth for a specified time. The time of channel use is divided among two or more users. Over a cycle of usage, the effect of TDM is the same as dividing the frequency bandwidth. TDM is more attributed to digital transmission systems than analog. The channel has an aggregate data rate much higher than needed by each user. The user is allowed to transmit at the channel rate for a brief amount of time. After the expiry of allocated time, the next user is allowed to transmit or receive, and so on. However, this arises a host of different scenarios. We discriminate between two types of TDMs to deal with the different ways in which time for channel use could be allocated.

7.1.5. Synchronous TDM

In synchronous TDM, all transmissions from multiplexed users occur at specified time instants. For example, each user is allowed to transmit for a time beginning at a given instant and ending at another instant. If the user does not have data to transmit at the beginning of the specified interval, the channel remains unused.

Synchronous TDM is performed by defining channel as having a certain data rate. A multiplex cycle or frame is then defined as consisting of a certain number of bits to be repeated. This multiplex cycle is further divided into time slots such that the sum of slots is equal to the multiplex frame. Following example may help understand this concept.

Example 7-1: Consider 8-bit PCM voice transmission. Let a digital transmission channel have a data rate of 768 kbps or bits per sec. This channel has 12 times the data rate required by a single voice source. Suppose that a TDM cycle consists of ($12 \times 8 = 96$) bits. Let the cycle be divided into 12 slots, each of 8 bits. Then 12 voice sources, each using 64 kbps 8-PCM can be multiplexed on this line. Note that in 8-bit PCM, 8 bits represent a single sample. Therefore, samples from 12 sources are multiplexed and interleaved on a single channel.

In synchronous TDM, the time slots are allocated to traffic sources without any regard to whether these time slots will be used continuously or not. It is well known that a voice source is active only about 40% of the time.

That means that for 60% of the time there will be no data from each voice source and the slots will remain unused for about 60% of time.

7.1.6. Statistical TDM

Also called asynchronous TDM, in this method of TDM specific slots are not allocated to individual users. Instead, all data sources store their data in a buffer. The multiplexer visits data buffers one by one. If a buffer contains some data to be transmitted, it is transmitted. Otherwise, the multiplexer goes on to the next data buffer. Therefore, the channel bandwidth is not wasted if any buffer has some data to send all the time.

The difference between synchronous and statistical TDM is shown in the Figure 7-2.

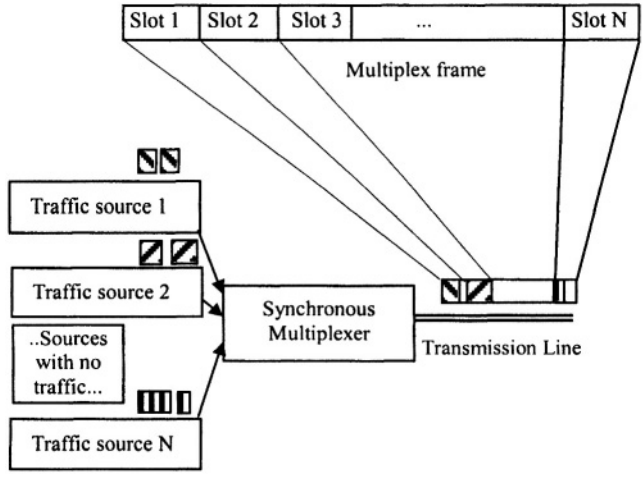


Figure 7-2(a) Synchronous multiplexing. Even if frame is partially filled, the source has to wait for the next frame to send more than one slot of traffic

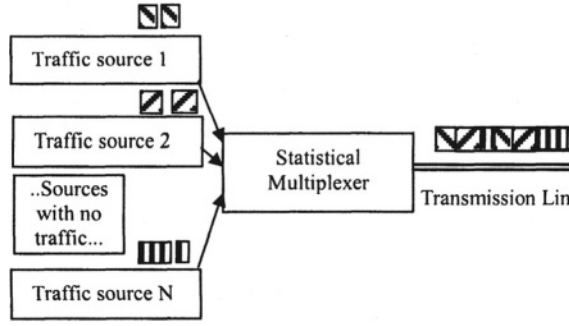


Figure 7-2(b) Statistical multiplexing. If frame is partially filled, the source doesn't has to wait for the next frame to send more than one slot of traffic.

The Figure shows data from N users multiplexed on a single transmission line. A fixed frame length is defined in synchronous TDM in Figure 7-2(a). Each of the N data sources is allocated a specified part of the frame. The Figure shows that during one particular TDM cycle only three users have data to send. The frame carries data from these three users and remains largely empty. Shown in 'white' is the portion of the channel frame for which no one transmits any data, even if some of the sources have data to transmit.

Contrary to this Figure 7-2(b) shows that there is no multiplex cycle in statistical TDM. The multiplexer visits each source for the same duration of time as in synchronous TDM. However, if a source does not have any data to send, the multiplexer does not stay there for the maximum allocated amount of time. Instead, it moves on to the next source of data. In this way, more data is transmitted for the duration of the same frame. In synchronous TDM even if an attached source does not have any data to send, the channel expects that the source will use the allocated slot. However, in statistical TDM, if a source has nothing to send, the multiplexer goes on to next source.

In synchronous TDM, each source can transmit data only in the designated time slot or slots. Consequently, if the multiplexed data is heading towards a switch, the switch knows which slots are being used by each source. The data can be switched according to this information. However, the situation is different for statistical TDM. In this case, any slot could be carrying data from any of the sources. In fact, there is no particular need of dividing the frame into slots. The information about who is the source of a slot has to be included with the data. So, the user data blocks for statistical

TDM consist of a header that tells the switch about the source or destination (or both) of data.

The frame for statistical TDM does not have to be of a fixed length. Depending on factors, such as traffic from other sources, a source can transmit a variable number of bits each time. In this way, the frame for statistical TDM requires pretty much all functions of synchronous communication via frames, namely, flag, address and other frame control information.

7.1.7. Statistical Versus Synchronous TDM

The slot allocation in synchronous TDM is just like allocating a channel in circuit switched network. It does not require addressing and framing information. Therefore, when all sources of traffic use their designated time slots as if the slots were separate channels. When traffic load is high enough to keep the TDM channel busy for most time, the synchronous TDM is more efficient than statistical TDM as it does not have any frame headers. Therefore, statistical TDM is not always better than synchronous TDM or vice versa. In fact, they both have strengths and weaknesses. The main strengths of statistical TDM lie in the following facts:

1. It does not require a strict definition of beginning and ending of a slot/frame. This property makes its implementation simple. This resembles packet switching mechanism.
2. It utilizes the channel capacity more efficiently, especially when individual sources have bursty traffic. Bursty traffic is characterized by repeated patterns of sudden data generation followed by long pauses.
3. It is better suited to traffic sources with varying requirements of channel capacity. Sources requiring higher capacities might send longer frames, and more often than the other slow speed sources. Even though synchronous TDM can provide this capability by assigning multiple time slots, the smallest unit of data in synchronous TDM is the slot itself and there is always a possibility that a slot remains only partially filled, thus wasting the channel capacity.

In addition to the above benefits of using statistical TDM, there are some drawbacks associated with it as well. Some of these are:

1. The data has to be stored in buffers, that requires additional cost of memory.
2. Due to buffer storage, there will be delay distortion introduced. This would deteriorate the quality of real-time data.
3. There is an additional overhead attached to each frame that helps to identify the boundaries, addressee/addresser and other information about data. At higher loads, this header cuts down on the efficiency of the line.

The synchronous TDM has the following benefits over the other.

1. Once slot allocation has occurred, there is a fixed relation between a data source and its slots. This is analogous to circuit allocation in circuit switched network.
2. There is no extra delay distortion, which makes it ideal for voice-like communication.

There are tradeoffs in terms of some disadvantages, such as:

1. In conditions of low traffic volume from all or some data sources, the link utilization is lower than statistical TDM.
2. The multiplexer and traffic sources all have to be synchronized. Usually, the way this is done is by having a central clock providing synchronization to the main multiplexer with clocks from each source synchronized with the main clock. The main clock runs at the line capacity, and is sometimes called as the *master clock.* This is the real cost of synchronous TDM and offsets the memory cost of statistical TDM. Local clocks of individual sources generate data locally. Data is multiplexed according to a predetermined order. The multiplex cycle is divided into frames and slots, and one or more slots are assigned for individual data sources.

7.1.8. The TDM Switch

A TDM switch consists of a multiplexer and demultiplexer pair. The *demultiplexer* performs the inverse operation of a multiplexer. For this purpose, it needs the information about slot allocation. It extracts data back from slots and forwards bits to the assigned recipients. A multiplexer / demultiplexer pair is shown in Figure 7-3.

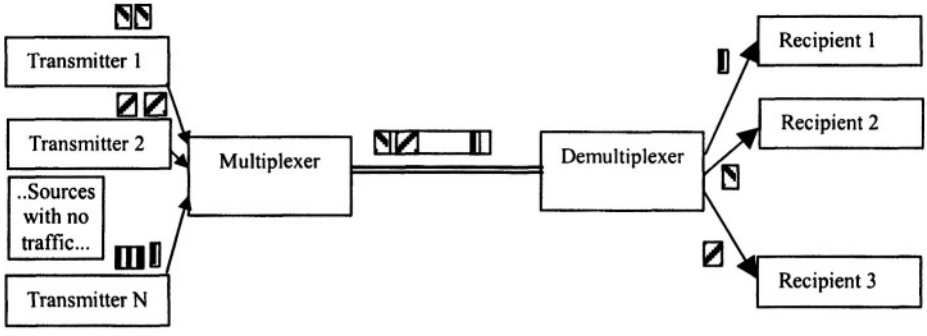


Figure 7-3. Multiplexer and demultiplexer pair. The demultiplexer has the prior knowledge of recipients and switches the appropriate slots traffic according to this knowledge.

The data leaving a demultiplexer in a TDM switch may consist of TDM frames as well. In this case, the frames arriving at the demultiplexer does not have the exact same format as the frames leaving the demultiplexer.

The switch may be getting input from many lines and making new multiplex frames according to the destination of each slot. In addition to a pair of multiplexer/demultiplexer, the TDM switch also has the intelligence of knowing where each slot is headed for. It will then switch the slots

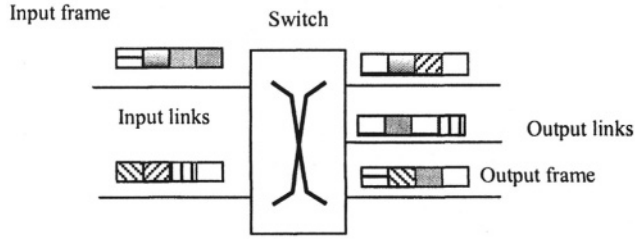


Figure 7-4. A TDM switch performs multiplexing and demultiplexing.

individually into frames on the output links. See for example Figure 7-4.

Thus, a TDM switch performs both multiplexing and demultiplexing. In Figure 7-4, there are two input links and three output links of the TDM switch. Slots from different traffic sources are labeled with different patterns. The switch stores a table that has the detailed information about which slot on which input link is supposed to be switched to which outgoing frame and slot. It demultiplexes the incoming frames and stores the information in the slots to wait for the appropriate output frame and slot. At the output, it performs multiplexing of the stored slots according to the information in a lookup table. Thus, when there is no data for a particular output slot, the slot will be transmitted empty (shown as white) in Figure 7-4.

In statistical TDM, the switch is a router that performs switching based on the source and destination addresses contained within the data block from each user. In general, the statistical TDM switch performs the following tasks.

1. Framing: In order to identify the beginning and end of a user data block.
2. Demultiplexing: In order to be able to switch each slot to the required output link.
3. Storage of slots: In order to wait for the designated output slots on designated outgoing links.
4. Multiplexing: On each output link.

7.1.8.1. Framing

Framing is the term used to describe the process of identification of the beginning and end of a TDM frame. Framing is easily achieved in synchronous TDM by using some extra bits that can be delineated by the

receiving device (switch or demultiplexer). The framing bit on successive frames constitutes a known bit pattern, such as, alternating 1 and 0 string. It is very unlikely that any other bit will have the exact same pattern for a large number of frames. Once such a pattern is established, the transmitting and receiving devices are synchronized to the beginning or end of the frame. The total length of frame also determines the beginning of the next frame. In case of asynchronous TDM, a mechanism such as the use of Flag is employed for block identification from each source.

7.1.8.2. Pulse stuffing

In practical implementations, the sum of data rates of multiplexed sources should be equal to the total line bit rate for synchronization purpose. If the aggregate data rate of the multiplexed traffic is less than the line rate then there will be gaps in the TDM frame. If nothing is transmitted in these idle time slots, the clocking information may be lost. The idle time on TDM channels are filled with *pulse stuffing*.

Let's suppose that C is the capacity of a line and that N traffic sources are multiplexed on this line, each allocated I/N fraction of the capacity. Let R_k be the actual data rate of source *k.* Then, ideally R_k should be equal to C/N for all values of & (1, 2 ... N). If $R_k > C/N$, then some of the data will be lost and if $R_k < C/N$ then some of the TDM slot will be unfilled. Pulse stuffing is used to the fill the partially empty TDM slots when $R_k < C/N$ so that $R_k = C/N$ can be satisfied. Pulse stuffing is performed by inserting extra bits (pulses) at known locations according to a predetermined rule. The demultiplexer of such a frame has the knowledge of pulse stuffing and can easily remove these pulses from the frame to extract and deliver the data bits that were sent by the transmitter. In practice, pulse stuffing is done for known data rates so that the number and positions of pulse stuffing are known by the transmitter and the receiver.

7.2. Digital Carrier Systems

As with the network protocols, standardization of multiplexing is the key to the success of such systems. Transmission systems designed by public and private companies ultimately lead the way to standards. In digital carrier systems, North America (and Japan) has mostly followed standards set long ago by AT&T Laboratories (former Bell Laboratories). The digital signaling (DS) hierarchy of AT&T evolved to conform to the earlier analog carrier systems using FDM. It provides a number of DS system levels, with DS-1 as the building block of others. Other carrier systems of rate higher or lower than DS-1 are derived from this system. More popularly known as the T-1 carrier,

the DS-1 system will be discussed as the example carrier system[8]. Two other carrier systems gotten popular in the eighties and nineties will be briefly touched upon too. These are the high-speed synchronous optical network/synchronous digital hierarchy (SONET/SDH) and the relatively new digital subscribers' line (DSL) systems.

7.3. The DS-1 Carrier System

The DS-1 multiplexing system is part of the T-1 transmission system and is sometimes referred to as T-1 itself. In Europe and many Asian countries, an equivalent system is used, called E-l carrier system. There are many differences between T-1 and E-l. The T-1 was perhaps influenced by the channel structure of an earlier AT&T Bell Laboratories System based on FDM hierarchy. The FDH hierarchy used a 12-channel multiplex group as one of the fundamental unit for marketing. The T-1 uses a multiple of 12 channels, that is 24 digital channels, for 64-kbps PCM standard for digital voice.

The system defines a TDM frame that can be used for digital voice, data or for integrated voice-and-data services. It has especially been designed for 8-bit PCM voice systems. Such systems sample analog voice signal at a rate of 8000 samples per second. Each sample is converted into an 8-bit binary value. Each channel in a DS-1 frame allows for one sample to be transmitted from each voice source. A maximum of 24 voice sources can use one DS-1 frame. This implies the following:

1. Each frame must be repeated as frequently as the sampling rate: 8000 times per second so that voice samples could be carried as they are generated.
2. There are a total of minimum $24 \times 8 = 192$ bits per frame. One more bit is added for framing purpose giving a frame size of 193 bits per 125 μseconds. The 125 μsec is the frame repetition rate, which is 1/8000.

The DS-1 frame schematic is shown in Figure 7-5.

[8] DS-1 is the transmission characterization of the T-1 carrier system.

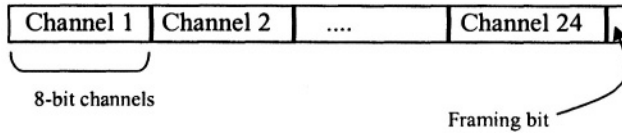


Figure 7-5. The 193-bit, 125 μsec DS-1 frame.

7.3.1. Total Bit Rate

Since the DS-1 frame had 193 bits and there are 8000 frames per second, the total bit rate = 193 × 8000 = 1.544 Mbps

7.3.2. Signaling Information

There are two types of information carried by a DS-1 frame: the user information and the signaling information. The example of user information is digital voice or data. The signaling information is used by the network to delineate and control user information. In the case of DS-1 system, there may be as many as 24 users using each frame. There is the need for routing and control of this information. This is done by *robbed bit signaling* in voice communication and *common channel signaling* in data communications.

In *robbed bit signaling* one of every 48 bits in a voice channel is stolen by the network to be used for signaling. Recall that each voice channel has 8 bits per frame. Therefore, stealing one bit out of six frames per channel leaves 47 out of the 48 bits used for actual user information. The stolen bit is the least significant bit of a channel of every sixth frame. Thus, user channel contains a signaling channel with a bit rate 1/48 times the bit rate of voice channel. Consequently, the 8-bit PCM does not get a true bit rate of 64 kbps. Instead, it has a bit rate of $(47/48)^{th}$ of 64 kbps. The signaling bit rate per voice channel is 1/48 × 64 kbps. In summary, when DS-1 is used to carry voice only traffic, the 24 channels are used for 8-bit PCM. Signaling information is added to each voice channel by stealing one bit per channel in every sixth frame.

In *common channel signaling,* a channel is stolen from every frame. This is done when DS-1 frame is used to carry data from data terminals. Twenty-three channels are used for data and the 24^{th} channel is dedicated to signaling. This channel carries the signaling information about all the remaining 23 channels. Common channel signaling, by definition, is a signaling mechanism in which a separate channel carries the signaling information about a group or groups of channels. In this case, one DS-1 channel carries the signaling information. Another type of signaling is the in-

channel signaling in which the user and signaling information is carried over the same channel. In addition to common channel signaling, in-channel signaling is used by allocating one bit of every data channel for this purpose. The term *in-channel signaling* implies the use of the same channel for user and signaling information transmission.

In summary, when data is to be carried over DS-1 frame, 23 of the 24 DS-1 channels are used for user data communications. The 24^{th} channel is used for signaling. The main use of this channel is fast recovery from a framing error. In the 23 channels, the user information is carried by using 7 bits per channel, thus giving a data rate of $7 \times 8000 = 56$ kbps. The 8^{th} bit of each data channel is used to define a signaling channel for each data channel. When DS-1 is used to carry combined voice and data information, then all the 24 channels are used. Multiple DS-1 frames can be multiplexed to obtain higher data rate systems, such as DS-2, using 4 DS-1 frames, and DS-3 combining 30 DS-1 frames.

In the E-l carrier system, there are 32 channels, each for 64 kbps data rate, providing a total bit rate of 2.048 Mbps. Thirty of these channels are used for user data (voice or non-voice) and the remaining two for various signaling and control functions. The term T-1 is used to describe the raw bit-rate of 1.544 Mbps using DS-1 format. There is no such differentiation of terms in E-l: the term E-l conveys the meanings for the signal format as well as the raw bit rate.

7.3.3. Problems with T-1/E-1 Systems

Even though T-1/E-1 and their higher hierarchies have been quintessential to most of the voice and data network infrastructures, they are not without their problems. Some of these dealt with the need of demultiplexing the whole carrier to extract only a part of it, and others were due to the need of repeaters at every 3000 ft from the first node and 6000 ft thereafter. Advancements on synchronization, cabling and communication and modulation theory have helped devise new systems. We will discuss two of these, which give T-1 no more than the title of a forerunner. One of these is the SONET/SDH hierarchy that takes care of demultiplexing issue and engulfs in it all the DS-*n* systems as a part of it; The other is the digital subscribers line (DSL) system and its various forms, that not only brings the power of T-1 to the subscriber's premises, but also gets rid of the repeater needs by employing better data encoding, modulation and error control techniques.

7.4. Synchronous Optical Network/ Synchronous Digital Hierarchy

With developments in optical fiber cables, it was obvious in the earlier 80s that bandwidths unimaginable at that time will be a reality shortly. Protocols were needed to handle bit rates achievable from such bandwidths. Towards the end of the decade, Bellcore of USA had the right formula to use optical fiber in the form of a new carrier system called SONET. Later, it was adopted as a standard by ANSI. The ITU-T followed by announcing the same under a new name, SDH. The equipments designed for SONET may not be used for SDH due to some cosmetic variations, but the concept is the same for both.

SONET uses fame-based multiplexing, with enough overheads to have a whole layer for administrating and managing the frame. Figure 7-6 shows the basic SONET frame and how higher-rate frames are formed.

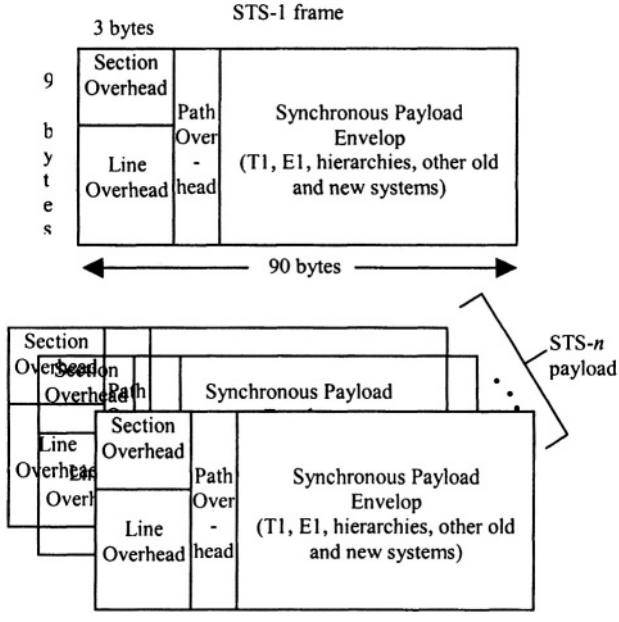


Figure 7-6. A number of STS-1 frames, the basic unit of SONET frame, can be combined to makes higher rate hierarchies, STS-*n*.

Each of the frame field performs a number of functions. Some of these are listed in Table 7-1. Originally conceived for use over optical fiber

cable, SONET-like rates are now possible on copper cables. A frame is transmitted every 125 μs, giving a frame generation rate of 8000 frames per second. This is in league with the voice sampling rate for 4 kHz bandwidth. A frame repetition rate of 8000/s allows for each transport of a channel of T-1 and other 'legacy' carrier systems. The basic frame format for SONET is called STS-1. For STS-1, the specified bit rate is 51.84 Mbps, as can be calculated from Figure 7-6. The bit rate for STS-3 is $3 \times 5.84 = 155.52$ Mbps.

In the SDH terminology, the basic frame is STM-1, which has a rate of STS-3. There are some minor differences in frame formats as well. Both SONET and SDH have the same advantages over T-1/E-1 carrier systems. Some of these are:

1. There is a central synchronization clock used, which is very stable.
2. The use of pointers in the overhead section makes the multiplexing of payload area flexible. The pointers can be used by the recipient to know the exact location of the multiplexed frame, be it DS-1, DS-3, or ATM.
3. The error rates are very low, of the order of one in billion, which makes these systems highly reliable.
4. The SONET network is used together with add-drop multiplexer (ADM). ADMs make it possible for a recipient station to extract or add data from/to payload without carrying the demultiplexing of the entire frame.

Table 7-1 Main functions of SONET header

Frame Field	Main function
Section overhead	1. Framing 2. Error control 3. Administrative alarms
Line overhead	1. Pointer for the payload section 2. Error control 3. Alarms
Path overhead	1. Error control 2. Payload types 3. Equipment types 4. Connection verifiers

The STS-1 signal is in electrical form and uses electrical connectors. Signals for STS-3 and higher are typically in optical form. The connectors for these signals have the designation of oc-*n*, where oc stands for optical carrier and *n* is as in STS-*n*. Optical carrier systems with rates as high as oc-192 (= 9.95328 Gbps) are commercially available. Among the most popular, there are oc-3, oc-12 and oc-48.

7.5. Digital Subscriber's Line (DSL)

One application of the DS-1 systems was in digital loop carrier (DLC) systems. By doing so, a single T-1 carrier could transport signals from 24 voice channels, or 30 for E-1. T-1 systems use AMI line coding that requires the first repeater at a distance of 3000 ft and then a repeater every 6000 ft. The ISDN subscribers' loop, called the DSL changed these numbers substantially. It was designed for the ISDN basic rate access, which consisted of two 64-kbps user data channels, called the B (Bearer) channels and one control channel of 16 kbps, called a D-channel. That gave a total bit rate of 160 kbps for about 18000 ft without a repeater. Further research on improving DSL resulted in the invention of high-speed DSL, or HDSL. HDSL used bandwidth efficient modulation schemes and echo cancellation to provide T-1/E-1 rates for up to 12000 ft. With a sprawling number of users connecting to Internet, another form of DSL, called the asymmetrical DSL, or ADSL has become one

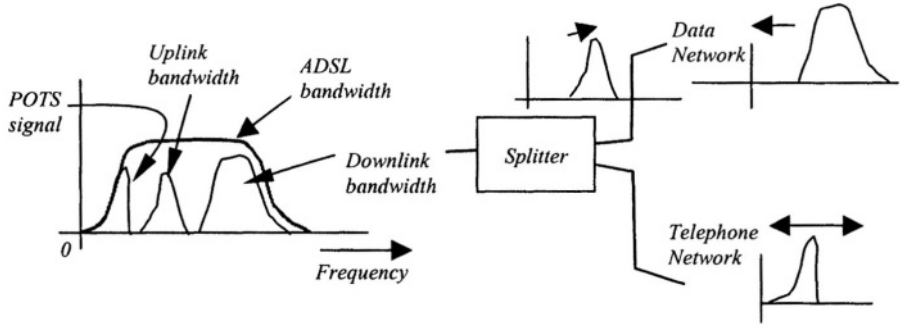


Figure 7-7. ADSL integrates POTS with data uplink and downlink by multiplexing the three signals.

of the biggest sensations of the carrier systems to connect phone and data terminals to PSTN and IP networks simultaneously. ADSL, originally conceived for video on demand (VoD) applications in early 90s, promised to take care of the asymmetric nature of the Internet traffic. In addition to that, it integrates the phone signal by using a multiplexing scheme that is a combination of TDM and FDM.

In a VoD system, there are two directions of communications, one from the user to the service provider, called the uplink and the other in the opposite direction, called the downlink. The bandwidth demand in the two directions are quite different, hence the asymmetric solution. With VoD not succeeding the marketplace, ADSL went into the background until the

discovery of Internet by business and shopping community in early to mid 90s. ADSL fits best for an access network that provides high bandwidth in the downlink and relatively slower data rates in the up link.

7.5.1.8.1. Integration With Telephone

One of the main reasons for ADSL making a splash is its integration with the telephone system (plain old telephone system or POTS). ADSL provides various data rates depending on the distance. Up to 18,000 feet, a rate equal to T-1, is achievable, and for distances shorter than 10,000 ft, rates in excess of 6 Mbps, are possible. E-1 rates bound the ADSL distance is up to about 16000 ft. On the uplink side, rates between 16 to 64 kbps are typical. ADSL makes these rates possible and retains the POTS as well by utilizing about 1 MHz of the bandwidth of copper, and multiplexing the three signals on it. This is shown is Figure 7-7. A component called *splitter* is used to separate the DSL signals from the POTS signals on the uplink and for combing them on the downlink.

ADSL divides the bandwidth into three channels: for two-way voice, uplink data and the third for high-speed downlink data. Voice signal exists roughly between 300 to 3400 Hz. Some guard band is left between the three bands and the ends to avoid cross talk. It can use any network protocol architecture above it, ATM being the most cited in literature. Since ADSL was originally meant for high-quality video, its standards provided very high reliability. However, the cost of adding extra error control processing is in the form of delay. 20 ms of delay is typical, which is highly undesirable. Because of this reason, ADSL, in spite of its big splash has not been regarded a satisfactory solution. However, due to the introduction of optical fiber in the local access networks (from exchange/ISP to home or business), the lengths of copper wires have shortened recently. This has led to the realization of very high-speed DSL, called VDSL that does not entail the problems of ADSL. The VDSL is projected to provide all the capabilities of ADSL, without significant delays. Besides, data rates as high as STS-1 are achievable. It reflects the integration of not only POTS and the Internet, but also the integration of many developments made until today in networking.

7.6. Multiplexing at higher layers

It is only the physical layer that uses link capacity directly. All other layers communicate with their peers through a logical connection. The connection is identifiable by either the unique pair of addresses or by a unique connection identifier. Some times, a layer is capable of opening two or more logical connections simultaneously. This can happen in many ways depending

on factors, such as whether the protocol at a particular layer is connectionless or connection-oriented, how many protocols are there on the layer above. The example in Figure 7-8 is an illustration of some situations.

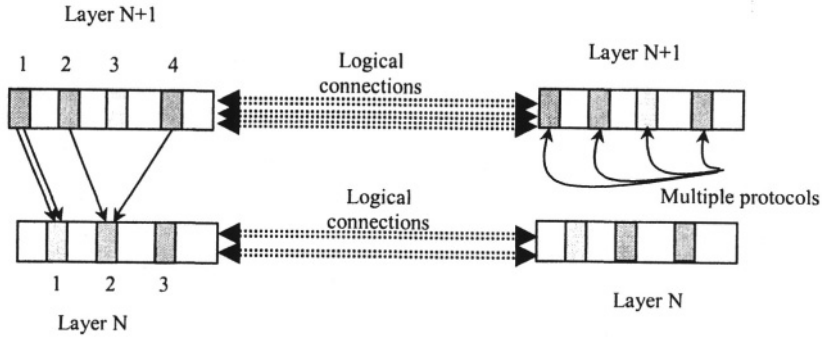


Figure 7-8. Two cases of higher layer multiplexing.

The figure shows two layers in each of the two communicating stations. Layer N is shown to have three different protocols differentiated by three different boxes. Similarly, layer N+1 has four protocols. Both layers are shown to have established multiple logical connections with their peers. Layer N+1 has three logical connections and layer N has two logical connections. Multiplexing is provided in many ways in this figure as explained in the following.

7.6.1. Multiple Protocols Per Layer With Connection-oriented Mode

With the help of logical connections, three protocols are exchanging PDU in layer N+1 and two in layer N. The PDUs among different protocols could be easily discriminated by unique connections IDs. These connection IDs would point to specific protocols in the destination layer, in the destination computer.

An example of this case is the TCP/IP suite of protocols. It has many protocols at the each layer. For example many processes at the applications layer can be communicating simultaneously with processes in another application layer in another machine or machines. The TCP protocol provides multiplexing by allowing multiple TCP (logical) connections to open simultaneously.

7.6.2. Multiple Connections Per Protocol

The figure also shows protocol number 1 in layer N+1 opening two connections together invoking the services of protocol number 1 in layer N. The PDUs for both of these connections from protocol 1 in layer N+1 could be traveling with or without a logical connection. Alternatively, there could be a single logical connection for both. An example of multiple connections from the same protocol without a logical connection is the connectionless user datagram protocol (UDP) protocol sending multiple PDUs (from different applications) through the IP protocol. An example of multiple PDU types from one layer using a single logical connection is a layer 2 point-to-point protocol (PPP) carrying multiple IP datagrams over a single PPP link.

The purpose of Figure 7-8 is to demonstrate that there are a variety of ways in which multiplexing is realized in upper layer. We have seen in Chapter 6 that the ATM cell provides multiplexing by defining a 16 bits virtual channel number (8 bit virtual path identifier VPI and 8 bit virtual circuit identifier VCI). A protocol above ATM can make use of many VCIs, or many VCIs can be used as a single VPI, thus providing multiplexing of many Virtual Circuits in a single Virtual Path. In fact, one way that the relationship between the VPI and VCI has been describes is as shown in

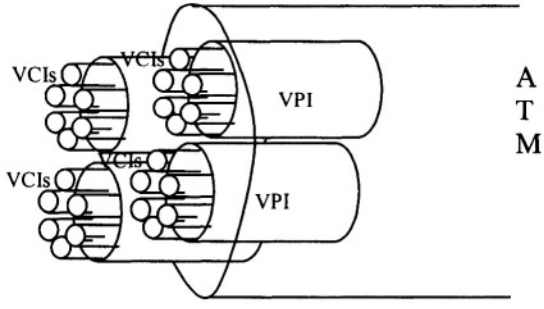


Figure 7-9. ATM protocol can be thought of as a pipe carrying 2^8 VPI pipes inside the network, each of the VPI carrying another 2^8 small pipes, one for each VCI.

Figure 7-9.

In this figure, VPI is the address of a data pipe carrying many thinner pipes, each labeled with a VCI. In turn, ATM is like a *very* thick pipe carrying many virtual paths.

Multiplexing capability at a higher layer protocol is a powerful function and is essential for all future protocols, as we move towards links with hundreds of Gbps capacity and applications that require simultaneous delivery of many packets types, data, voice and video, between two or more stations located at various places in the world.

7.7. Review Questions

1: Can analog communications be used with TDM? Explain.
2: What is the main benefit and problem of using (i) FDM, (ii) TDM?
3: Under what general load conditions synchronous TDM is better than asynchronous TDM? Why?
4: Switching for which of the synchronous or asynchronous TDM is more like (i) virtual circuit, (ii) connectionless?
5: What is statistical multiplexing gain?
6: Why is the design of DS-1 based on a 125μsec frame?
7: As a result of framing, the receiver of DS-1 frames knows the first (or last) bit of the frame. What is the benefit of this?
8: Would you agree if someone claims that DS-1 voice channels don't really provide a true 64 kbps voice channel? Explain.
9: What are some of the problems with T-1/E-1 systems and hierarchies?
10: How are these problems avoided in DSL and SONET?
11: The DSL cable was originally laid down for 4kHz voice signal. How has it become possible to get T-1 rates on it?
12: Why VDSL is less complex than ADSL and much more higher and speed?

8. The Network and Higher Layer Functions

So far in this book, we have addressed the lowest two layers of the OSI-RM. Local area networking is possible using these two layers alone. As long as the transmitter knows in what format the data should be sent, how it should be organized, and the receiver knows what to do with the data, no other layer may be required. If, however, the receiver needs the knowledge of any of these additional functions then higher layer are needed to provide those functions. Consider the example of a multimedia file transfer between two computers using different code representations, say, one machine using IBM's EBCDIC, and the other using ASCII code set.

The sending machine on the LAN has the files stored in a certain format. The user on this machine may use or edit the contents of the file with the help of certain application software package/s. For security purposes, the file data might need to be encrypted while being transmitted. Also, in order to reduce the volume of data to be transported, the file contents are to be sent in compressed form. Further, assume that the received data is played back as it is received. The two applications could setup a successful communication session as long as they both use a common application data packaging. Such a common interface is specified by an application layer protocol. Even if the two computers are using different code sets, data compression and encryption, communications can proceed as long as the two have negotiated this information, and have the capabilities to perform these functions. From the definition of the OSI layers, we know that these functions are to be provided by the presentation layer. The situation becomes more complex as we move away from a single LAN toward an interconnection of many LAN, such as in a campus or business network. It changes further as we move towards an interconnection of networks administrated by different organizations. There could be multiple possibilities of data path, link layer functions at some point on the path may not be as robust as the network user may want, and there could be a variety of ways in which data presentation, formatting and utilization could be different on different machines. Networking is not complete without taking care of all these function. These functions are to be provided by layers above the data link control layer. Even though the main subject of this book is the physical and data link layer, it would leave a big void in understanding the networking concepts without any mention of the higher layers. We devote this chapter solely to that purpose.

8.1. The Network Layer

Going back to Chapter 1, we know that the expertise from at least three main areas of engineering sciences goes into the design of a computer communications network. These are, the fields of the computer, the communications and the networking systems. The most critical layer for a computer scientist is the application layer, as it interfaces with the computer operating system. For communications engineer, it is the physical layer that plays the principal role. The network layer is the most complex and important from a networking engineering point of view. It provides the main networking function, that is, route data from the origination point in the network to the destination point in the same or another network.

The protocols at this layer receive services from data link layer protocols and provide service to the transport layer protocols. The address of a network layer is the network address. Thus, more than one network addresses

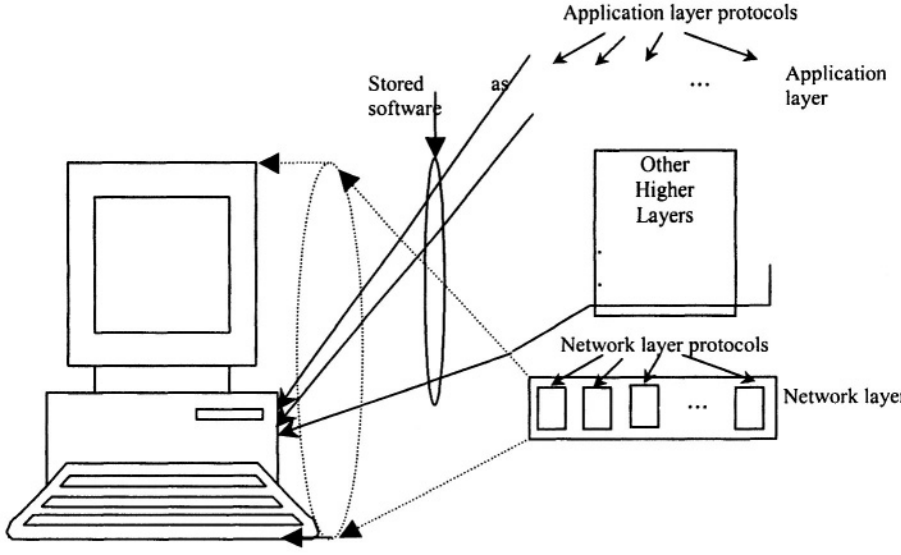


Figure 8-1. Computer is a network node. The network layer is the owner of the computer, shown by dashed lines and circle. Other higher layers are designated by software or hardware modules in the computer.

imply multi-homing; that is, connection to more than one networks. In fact, the term "network node" means the device addressed at this layer. Such is not the case at other layers where the addresses specify modules, or protocols in a layer. This idea is shown in Figure 8-1.

In other words, when we configure our computer for a network, then the network layer address is the computer's address as a network node. The addresses at other layers are the protocol addresses. Many of the protocol functions can indeed be implemented in hardware as well as software. Hardware implementation is possible due to advancements in VLSI that has not only resulted in affordable prices of the hardware modules, but a significant improvement in processing speeds. In view of this, the above figure is only a symbolic reference to emphasize the importance of the network layer in a network.

8.2. Typical Functions of Network layer

Through the use of primitives, a network layer requests PDU-delivery services from the data link layer. In general, there could be many data link layer protocols to choose from. The selection of a particular DLC layer protocol depends on the routing mechanism. The routing mechanism, in turn, depends on many factors, such as, the types of network, the type of connection mode at the network layer and the type of PDU to be delivered. Usually, there is a different sequence of events for connectionless and connection-oriented network layer.

8.2.1. Connectionless Network Layers

Usually, the real benefit of connectionless layer 3 could be achieved by not fixing the packet route. A router can, in this way, use the best route at a given time. These routes are recorded in the router memory as *routing tables.* A router has to update routing table according to the varying network conditions. On receiving a new packet, a general set of functions to follow could consist of the following items:

1. Check the validity of the packet: The packet is called datagram in case of connectionless network protocol. There are two ways in which a datagram may be invalidated (i) if it has been delayed so much that it has become useless, or (ii) an error has been detected in it. To implement (i), a protocol may have the function, such as "Time To Live"(TTL) in the connectionless protocol. TTL could be interpreted as the maximum number of hops allowed, or the maximum time that a datagram is allowed to stay in the network. A network layer on a routing/switching node will decrement TTL if it is greater than zero. If a packet is received with TTL equal to zero, it is discarded. The existence of TTL is very important for dynamically changing routes to avoid loops. IP uses this concept. In case of a packet with an error, the protocol may have a mechanism to correct or recover the packet, or just discard the packet entirely. In case of discarding, another protocol on the same or another higher layer may be responsible for its recovery.

2. Process the packet header: The packet header contains parameters for many functions to be performed on a valid packet. Accordingly, after the validity check, other functions, such as flow control may be performed.
3. Route selection: Each router maintains a routing table that is usually in the form of the best next hop for a given destination address. The next hop could be a host attached directly with the router, or it could be another router in the same network, or a different network. For each different case, a different

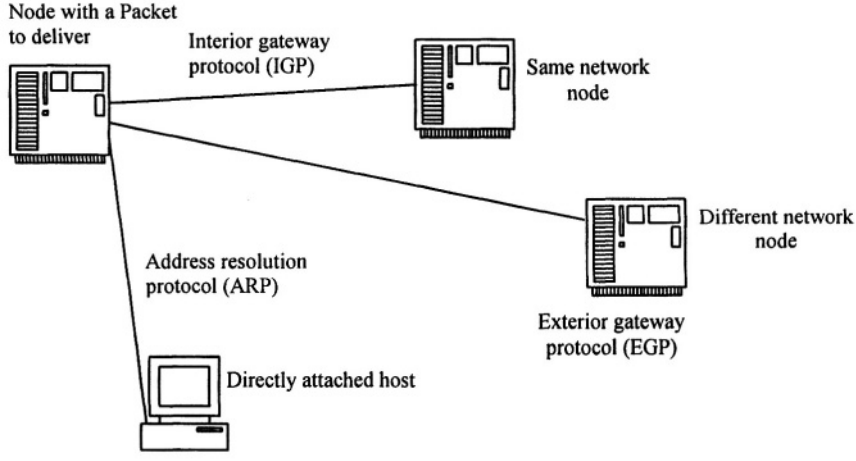


Figure 8-2. Different routing protocols are needed depending on the types of next node.

protocol may be needed. An example of use of different protocols is shown in Figure 8-2.
Figure 8-2 shows three possibilities for the next node, along with three different protocols. In order to route a packet to a computer directly attached with the router, the router needs the lower layer hardware address, such as an Ethernet network interface card address. This helps the router to differentiate among various LANs that a computer may be homing. An example protocol for this is the address resolution protocol (ARP) used in the Internet. A network under a common administration has typically the same routing protocol in all its routing nodes. Such nodes exchange packets under a protocol different from ARP-like. Such a protocol is also called the interior gateway protocol (IGP). Routing among nodes from networks under different administrations could possibly have another set of protocols. The routing in such cases is based on agreements between the owners of networks. A term used for such routing is the *policy-based routing.* An example of protocol use

for this purpose is the *border gateway protocol* (BGP) of the Internet. BGP is a type of exterior gateway protocol (EGP). There are a host of protocols currently used in the wide area networks for rout selection and updating.

All the routing protocols perform two essential tasks: (i) set some metric to weigh the usage capability of each outgoing link, and (ii), to find the next best node based on some best path selection to the destination. The best path selection is made for the metric defined in (i). Example metric are congestion in the outgoing links, capacity of the links, price of the capacity, or even a combination of these. Two algorithms which have been used most extensively for best path selection to treat a network as a graph with links making up the graph edges and the nodes making up vertices. The routing metrics are treated as weights or lengths of the edges. The problem of path selection is then reduced to shortest metric path selection. Such a path is called the shortest path. One of these algorithms is called the Bellman-Ford algorithm and the other is called the Dijkstra's algorithm. The *Bellman-Ford algorithm* finds shortest paths of all lengths (in terms of number of links, or hops) from one vertex (source node) to another (destination node). The *Dijkstra's algorithm* finds a single shortest path from one vertex to another. An example of representing a network as a graph is shown in Figure 8-3. The network, with 4 nodes, has links labeled with their capacities as routing metrics. For this example, we assume that the links are bi-directional providing same capacities in both directions. In practice, it could be either this way, or links in opposite directions could have different capacities.

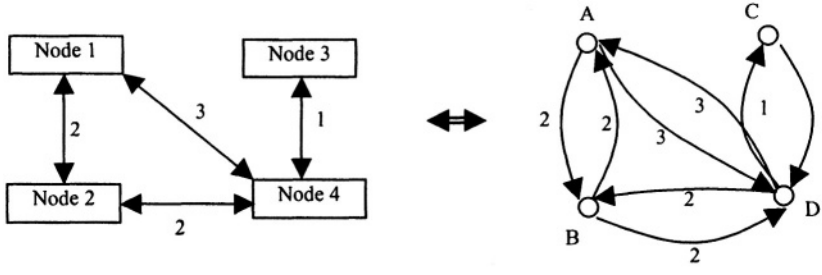


Figure 8-3. A graphical representation of a four- nodes network. Nodes are labeled as 2, 3 and 4 corresponding to vertices A, B, C and D. respectively.

4. Fragmentation: Once a route has been selected, the next job is for the network layer to make sure that the packet size is not too long for the outgoing link. Many protocols specify a maximum length. If the maximum PDU length is smaller than the one at hand, then packet fragmentation may be

done. Fragmentation function is provided through the protocol information that typically stores a flag to signal fragmentation along with a sequence number field that tells the location of a particular fragment in a larger PDU. An algorithm at the sending node converts PDU into fragment. If the parent packet has a unique identifier, it may be copied in each fragment. Such is the case in the IP, for example.

5. Header assembly: Once the packets are ready and have been fragmented (in case it is needed), a new header is required for the PDU. Since some protocol information is likely to change during the above 4 steps, the new header carries the new protocol information for the next node.

6. Packet scheduling and forwarding: After the route has been selected, the

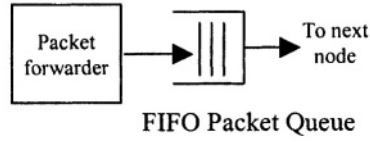


Figure 8-4(a). Packet forwarding without service differentiation.

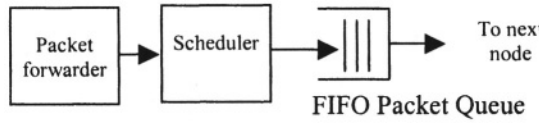


Figure 8-4(b). Packet forwarding with service differentiation

network layer is ready to

transmit the packet or fragment on the selected link. There could be more than one packets for a single output link due to incoming packets from many links. The packets are queued for transmission by the packet forwarding part of the network layer. If all the packets are to be treated equally, then a first-in-first-out (FIFO) queue is formed. If, however, certain packets have to get priority over others depending on their *type of service,* then a program called scheduler will determine the order of transmission. Service differentiation and service integration are two popular terms used to describe the phenomenon of setting up transmission order according to priorities.

With the above sequence of events, a network layer providing connectionless service performs the network layer function of routing N-PDUs based on a route that is selected dynamically. Most of the routers also have the facility of choosing permanent routes, called *static routing.* Also, a number of vendors give choice of scheduling mechanisms as well as route updating protocols.

Best effort delivery is the term used to describe the packet delivery service without a fixed route and guarantee. Lack of guarantee implies the absence of an error control mechanism. Sometimes, an error reporting mechanism is implemented that notifies the sender of the packet that the packet has been lost, was out-of time, or destined for indeterminate address.

8.2.2. Connection-oriented Mode

The connection-oriented mode of packet transmission is very much like circuit switching. In this mode, a virtual circuit is established between the source and destination pair before data transfer occurs. The virtual circuit is typically selected by each node for individual calls so that the network resources can be shared among all node (*distributed control*). The functions described for connectionless network layer are executed for this mode as well. There are differences in terms of their implementation or the time of usage.

Packet validity and header processing (functions 1 an 2) are not different from connectionless mode. Route selection is done at the call setup time. This is not to be confused with static routing. Static routing is applicable to connectionless and connection-oriented modes. In static routing, the route between a source-destination pair is fixed. In the case of connection-oriented routing, the path may be dynamically determined. But once it is determined, the path is fixed only for the duration of call. Every time a source-destination pair wants to set up a call, the route may be selected dynamically, or statically. The route information is stored in the node in the form of a lookup table. The table has entries to match the connection identifier of the incoming packets to an outgoing link (and another connection identifier for the outgoing path). The router performs routing at the call setup time and switching during the data transfer phase. Figure 8-5 shows an example of the table base switching mechanism.

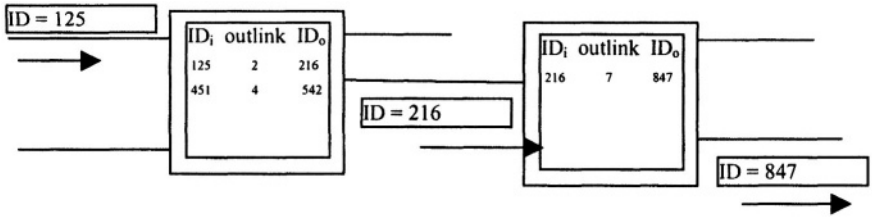


Figure 8-5. Use of lookup tables for virtual circuit switching.

In this figure, tables in each switch have three columns, ID_i, outlink and ID_o. ID_i is the virtual circuit identifier (VCI) of an incoming packet and ID_o is the same for an outgoing packet. All nodes on a route maintain such a table to provide an end-to-end routing path. The figure shows two directly connected nodes.

Fragmentation may be performed in connection-oriented mode just like in the connectionless mode. In the case of a virtual circuit, however, the fragments are guaranteed to be delivered in sequence because all fragments take the same route. VCIs vary from node to node on a path. Therefore, a new header is formed at each node before forwarding a packet. Packet scheduling may occur if there are packets with differentiable resource requirements.

Flow control and error control are easier to provide with connection-oriented transmission due to an in-sequence delivery. Once such is the case, the procedures of validation of the received packets will change. If a packet arrives in error, the recipient node may request a retransmission from the originator of the packet. Similarly, the number of packets to be transmitted may be restricted by the receiving node if flow control mechanism using windows is to be used.

8.3. The End-to-end Layers

The layers above the network are called end-to-end layers, as they provide functions on an end-to-end basis. These functions are provided using the same logic as lower layer, using header with PDUs and procedures at hosts. The job of the network layer is to liberate the higher layers from knowledge of the details of the network infrastructure. These layers process data only in a logical sense.

Layer 4, i.e., the transport layer, provides functions similar to the data link layer. These include reliability functions, such as flow and error control, synchronization, addressing, and in-sequence delivery of PDUs. A connection-oriented transport layer could be implemented at the top of a connection-oriented or connectionless network layer. In such a case, the transport layer will use some sequence numbering and retransmission

mechanism for reliable, in sequence transport of PDUs. Since the functions of transport layer are similar to the data link layer, it may appear that efficient layer 2 protocols may leave transport layer redundant. Such is not the case in

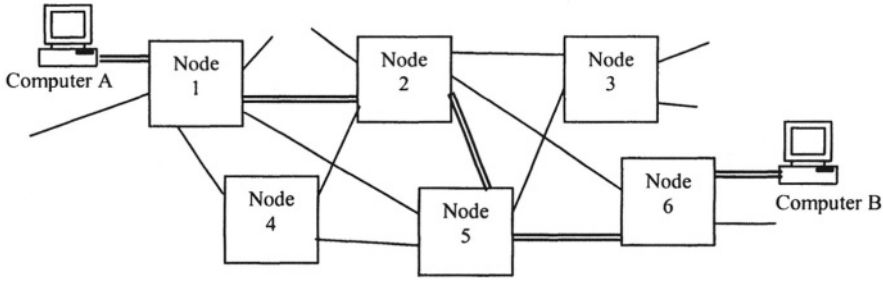


Figure 8-6. Two computers exchanging data over a multi-hop path. The path is shown as double line.

general. Consider Figure 8-6 as an example.

In this figure, two computers, A and B, have an end-to-end data path established. The data path is shown as concatenation of double lines. All packets pass through nodes 1 to 2, to 5 to 6. Every node accepts data from many inputs, for examples, assuming that data moves from left (computer A) to right (computer B), node 2 has three input and 3 output links. Similarly, node 2 has three input and three output links. The load on any output link depends on the routing decisions for the traffic on the input links. It might happen that at some point in time, one or more outgoing links on some nodes may get congested. This will result in a PDU taking too long to reach from A to B. Suppose that congestion is detected first at node 5. If only link layer procedures are used for flow control by node 6, then the influx of data at node 5 will be more than the outflow. This will result in inevitable overflow of buffers at node 5. If, however, the transport layer in computer B sends a message to the transport layer in computer A to slow down, this may ease the flow of packets throughout the path without a backpressure on any node. In fact, as can be imagined from this example, a lack of layer 4 can result in breaking down of the whole network with only one node being congested.

In short, the end-to-end functions of transport and higher layers are just as essential as the network and lower layers. Short description of the functions of session, presentation and application layer are given in Chapter 2. We will close further discussion on this topic and move on to an example of higher layer protocol.

8.4. X.25 Packet Layer Protocol

X.25, an ITU-T standard for public data networks, is the precursor of much of the public data networking. It continues to be used as a medium-speed data network, and most importantly, as an example of studying the network protocol architectures. Among other characteristics of X.25, one learns that it is not always possible to delineate a protocol as corresponding to one particular layer of the OSI-RM. This, we have shown in Figure 8-7, by drawing a possible parallel between the two protocol models, the OSI and the X.25.

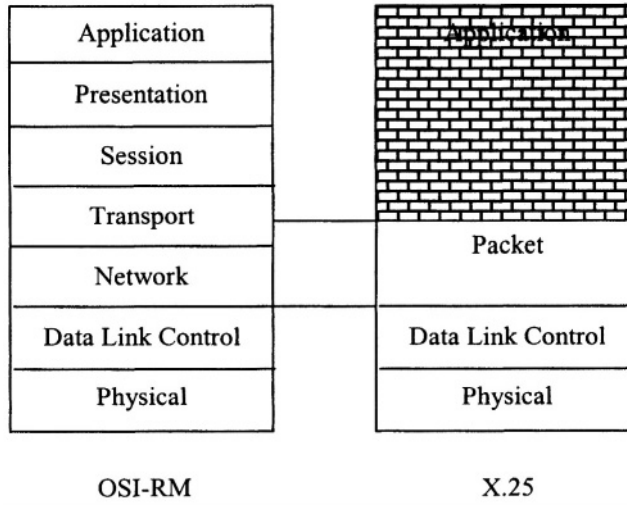


Figure 8-7. A comparison of the X.25 architecture with the OSI reference model.

As seen from Figure 8-7, the lower two layers of the X.25 network match exactly the OSI-RM layers. In fact, the protocols at these layers are specified by ISO as well for the same layers of an OSI network: the DLC of X.25 specifies LAP-B (link access procedures for channel B), a derivative of HDLC. Similarly, EIA-232 is a choice of protocols specified at layer 1. There is another PHY specified for X.25, designated as X.21. X.21, though couldn't get the same popularity as the serial EIA-232, it uses a less complex connector with more complex and efficient logic. One reason for the unpopularity of X.21 has been perhaps the universal acceptance of IBM PC

and its clones. It has serial ports using 25 as well as 9 pin connector EIA-232 interfaces. Another reason is the gradual improvement in the performance of EIA-232 revisions. It is the packet layer of the X.25 that creates the debate sometimes. It is defined under the title of packet layer procedures in ITU-T terminology.

8.4.1. X.25 Packet Types

Figure 8-8 shows the X.25 header format. As seen from the header, there are two types of packets: data packets and control packets. The control packet type field can be used to define $2^7 = 128$ control packets, for call setup, termination, acknowledgements and various facilities provided by X.25, such as fast select (to bypass flow control procedures for a control packets) and rate negotiation.

The packet layer of X.25 provides the following functions:

1. Connection oriented data transfer using 12-bit virtual circuit IDs and a number of control packets: The protocol header has 12 bits of VCI divided into two types, a 4-bit group number and an 8-bit channel number. With the help of twelve bits, a total of 2^{12}=4096 virtual channels are possible for multiplexing.

The virtual channels can be used either on demand, such as in a virtual call, or allocated permanently to some terminals, such as in permanent virtual circuit (PVC). For duplexing purpose, channels are sometimes arbitrarily divided into two groups for incoming and outgoing calls. The X.25 protocol is defined between a data terminal, called data terminal equipment (DTE), and data circuit-termination equipment (DCE). A number of concatenated X.25 links could provide a multihop, end-to-end logical

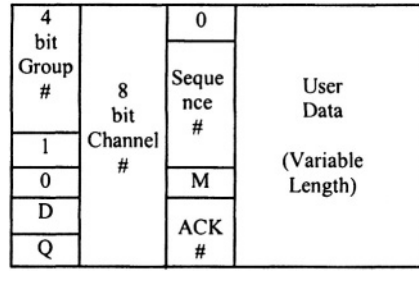


Figure 8-8(a). X.25 Data packet.

connection.

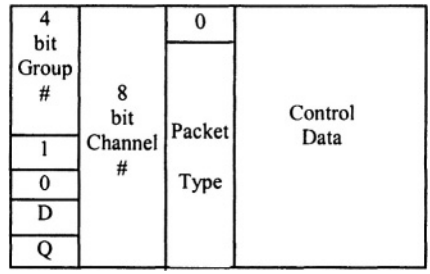


Figure 8-8(b). X.25 Control packet.

2. Flow control by sequence numbering: Just like in HDLC, the sequence numbers for received and sent packets could be used to implement a window flow control mechanism. There is also a provision for extending the sequence number field to 7-bit instead of 3-bit sequence numbers.
3. Error control by CRC: The sequence numbers also help in requesting a retransmission of packets. CRC can be imbedded in data to check the whole packet for errors.
4. Piggybacked ACKs: These are possible due to the send and receive sequence numbers.
5. Fragmentation with the help of two data packet types: A bit (M) is use to fragment data. When a large packet needs to be fragmented, then the M bit of each fragment except the last one is set equal to 1, From the knowledge of M bit, the receiver can figure out that fragmentation has occurred. The last fragment has M=0, from which the receiver knows that this is the last fragment of a fragmented packet. The packet type with M=1 and another bit D=1 (*see next function for an explanation of D bit*) is called an *A packet.* Any packet that is not an *A* packet is called a *B* packet. There are two ways of defining *B* packets, as seen in the next function of the protocol.
6. End to end acknowledgements: One bit (D) is kept to request end-to-end acknowledgements. X.25 protocol defines a *complete packet sequence* by allowing the sender of data packets to request an ACK at the end of such a sequence. Receiver, keeps receiving X.25 packets without acknowledgement until it receives a packet with D=1. A packet with D=1 signals the end of a complete packet sequence.

In case of incoming fragments, a complete packet sequence may consist of a many groups of *A* packets, each group followed by a single *B* packet. If the *B* packet has D=0, the receiver does not send an acknowledgement. It keeps receiving the sequences of zero or more *A* packets

followed by a *B* packet. ACK is transmitted only when a *B* packet with D=1 is received. This is shown in Figure 8-9.

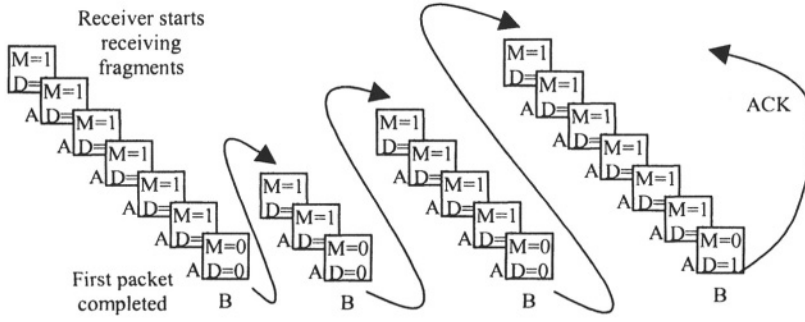


Figure 8-9. Example of a complete packet sequence. An end-to-end ACK is sent on receiving a B-packet with D=0.

8.5. Review Questions

1: Can we have a network operating with multiple links and without the network layer? Explain.
2: Is routing possible without a routing table? Explain?
3: What is multihoming? What is its benefit?
4: How will Internet change if IP becomes connection-oriented?
5: Compare the use of scheduler with connection-oriented mode?
6: What is the Integrated Services architecture and what is its purpose?
7: What is the Differentiated Services architecture and what is its purpose?
8: Is the use of Unicode going to reduce the complexity of the presentation layer?
9: What is the effect of lower layers on the higher layer when lower layers are (i) efficient, (ii) inefficient?
negative effect on the overall network performance because the higher layers exchange information with more propagation delay.
10: X.25 is a standard defined for the user-network-interface. Can it be used for a multihop network? Explain.
11: Is the D-bit function a network layer function? Why.

9. Performance Models for Data Networks

Once a network with the latest technology is fully operational, the question of performance pops up automatically. How could a network be ensured to deliver? Layering helps a lot in making the design systematic and upgradeable. In order for the initial design of a network to be optimum, each layer has to be individually optimized. For this purpose, the protocols on each layer have to be designed with correct assumptions about the functions of other layers. The example of frame relay illustrates this point very well. Frame relay was introduced much later than X.25. By this time, the transmission links had begun to be relatively more reliable, processor technologies are more powerful, and the storage technologies less expensive. Reliable links meant fewer errors, requiring a reduced error control processing. Likewise, cheaper memory and hard-disk resulted in increased storage capability. These factors, coupled with faster processing speeds, resulted in opportunities to design link layer protocols with lesser functions and more data throughput.

Frame relay protocol cuts down much of the layer 2 processing can resulted in throughputs as much as 24 to 30 times of X.25 with the same physical infrastructure. Frame relay was an opportunity seized. ATM took the use of reliable physical layer to the next step by promising integration of various forms of data on a single network. It would accept data from all sources, be it delay-sensitive, loss-sensitive or any combination, in the form of short, fixed sized packets called cells, and use frame relay like switching to provide reliable end-to-end services. ATM's performance combined with the ease of access interface of the Internet is showing promise for future of the Internet. However, it had brought home the necessity of performance based network design. This is so because the future networks are expected to perform with known target of delay, loss and jitter. If we can't design networks with given performance, we may as well not be able to get from them what we want. This has made the study of performance a critical issue. We devote this entire chapter to performance models. The objective is to allow a student of introductory level gauge the performance issue and its complexity.

9.1. The Network Performance

Every bit of information sent to a communications network is a job assigned to it. The network is expected to deliver the information at another point such that it is reliable, readable and useful. Reliability prevents errors, readability makes it intelligible, and usefulness pays. The following example may help illustrate the point. If a stockbrokerage company allows its subscribers to get the latest stock market information online and buy shares live, it would need the underlying network to be: Reliable, so that the information can be trusted; Readable, so that the user can get the correct meaning from it; In time, so that it may not be too late to use the information. Reliability, readability and timeliness are the three dimensions of performance. In the above example, the job assigned to the data network is to convey the stock information with certain metrics for these three dimensions. When a data packet travels from origination point through the layers of network, to the end user, it passes through many processes and obstacles. If a large number of data packets are in transit together, they may have to be stored in a storage facility, such as a queue of packets. Figure 9-1 shows a simple depiction of such journey with a single switching node between the origination and the destination points.

It is obvious from the figure that, in order for the network to transport data packet to the destination, each link, layer and process has to deliver some performance so that the three dimensions of performance are kept within tolerable ranges. For examples, suppose a packet requires a reliability of 99%, readability of 95% and delay less than 0.5 seconds. This requires that the accumulated packets in error should not be more than 1 in 100, the intelligibility of information contained in the packet should be 95 out of 100 and the total transmission, propagation and waiting time in queues should be less than 0.5 seconds. Of the three, only two can be objectively measured, the reliability and the delay. Intelligibility is highly subjective and usually only the end-user can determine if there is an acceptable level of intelligibility or not. Luckily, for numerical data, intelligibility is easily related to reliability and delay. The exact definition of performance may be layer dependent still in most cases it may be regarded as the pair (**r**, **d**), where **r** stands for reliability and **d** stands for delay. Let's use this pair to interpret the meaning of performance at various layers of the OSI-RM in the sections to follow.

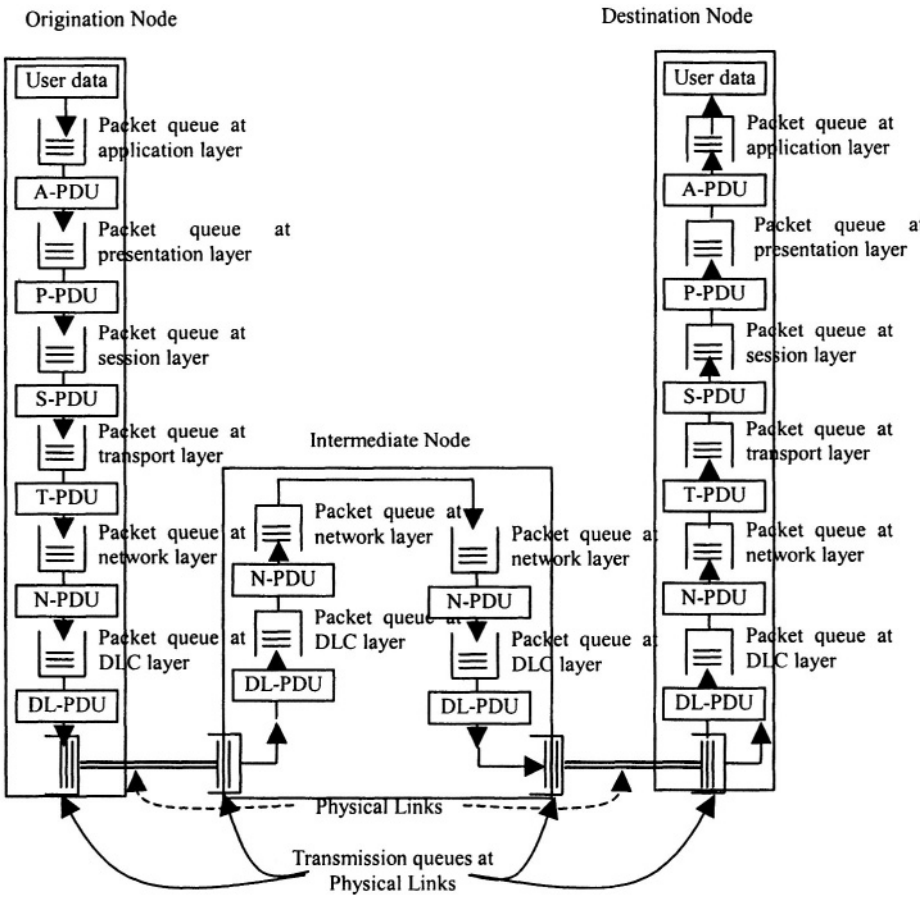


Figure 9-1. In a network with seven-layer architecture and one intermediate node between end-users, a packet could be easily queued and processed as many as 20 times before it reaches the destination.

9.2. Performance of the Physical Layer Protocols

Physical layer consists of the following components:

1. A medium that could be a cable, air, water, or space.
2. Connector/s providing a mechanical interface between the computer and the medium.
3. Specifications for information signal, such as, electrical, optical, etc.
4. Frame format if the PHY is using one.

The PHY performance is linked to the capacity of the transmission medium. Capacity is defined as the maximum number of bits that can be transmitted over a link in one second. Lower capacities take longer times for

transmission. When information signal travels over the medium, distortions and impairments may cause the signal to deteriorate beyond correct recognition. This distortion is increases with the increasing data rates on a certain medium. We measure distortion and impairments in terms of the bit error rate for a given data rate. Hence, a link could have a BER p with a link capacity C. The BER translates into reliability and the capacity into delay. In other words, a simple performance model (**r**, **d**) for a physical layer is given by (p, C).

9.2.1. Performance Improvement at PHY

Suppose that a physical layer protocol provides a performance metric of $(p, C) = (10^{-3}, 10^{4})$. This means that information can be transmitted at a rate of 10 kbps with a BER of 10^{-3}. The BER reflects the channel errors and detector inaccuracies.

9.2.1.1. Channel Errors

As discussed in Chapter 3, the physical properties of medium and external interference cause impairments that lead to signal distortion. The channel is like a filter and has a certain frequency bandwidth profile (called channel response). During transmission, a voltage signal could lose its shape or amplitude, or even both. Figure 9-2 shows an example of such distortion.

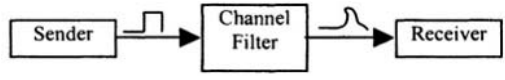


Figure 9-2. Effect of channel filter on data pulse.

In this figure, the effect of channel or medium is that the shape of the data pulse is distorted. This is due to thermal agitation of the electrons making up the material, external interference and other characteristics, such as reflection and echoes of the signal. To combat signal impairments, many preventive measures can be taken. One of them is to make a statistical model of the channel as a filter and find out what will be the best way to correctly recognize the incoming data pulses at the receiver. This is part of modulation theory in communications systems design. Different modulation schemes have been designed for channel conditions and certain applications of data. They vary in performance and complexity. For example, BPSK is better than ASK in terms of BER, but is more complex in implementation. In addition to the modulation schemes, error-correcting codes are also used for correcting the errors in data reception. These codes are implemented at a logical level inside the physical layer. They take a block of data and do some processing to

add parity bits to the blocks. These data plus parity bits are transmitted one by one and received by the receiver. The receiver decodes the data by using inverse operations and is capable of correcting some of the errors. Sometimes, these codes are labeled by their correcting capability. A code designated as (*n, k, t*) takes an information block of *k* bits, adds *n-k* parity bits to it and is capable of correcting *t* errors at random locations. In channels with *burst errors,* scrambling techniques are used to randomize errors. *Burst errors* are errors in concatenated form. *Scrambling* is done at the transmitter end to change the actual order of bits. *De-scrambling* is performed at the receiver to put back the bits in actual order. If a burst error occurs in scrambled data, it becomes random after de-scrambling and can then be applied to error correcting codes.

9.2.1.2. Receiver Accuracy

It is possible that the channel may not deteriorate the signal too much but the receiver still makes an error in recognizing it. We saw an example of such error in asynchronous communications. If the receiver and transmitter need to match the beginning and end of a data pulse, then a discrepancy in matching the clocks may result in errors. The situation becomes more critical when digital passband modulation is used. In this case, synchronization has to be maintained at several levels. The receiver generates the carrier signal locally that must match in frequency to the carrier signal used by the transmitter. In case of phase modulation, the phase of the locally generated signal must also match. This is accomplished by special circuits used at the receiver called *phase locked loops* (PLL). The job of a PLL is to lock the phase and the frequency of the locally generated carrier signal to the received carrier. Usually, errors in phase are more critical than errors in frequency. The reason for this is that the signal consists of sinusoidal waveforms. The values of these waveforms are a function of angle. An error in angle (phase) will result in an error of power measurement. Figure 9-3 shows the effect of an error in phase for a cosine signal.

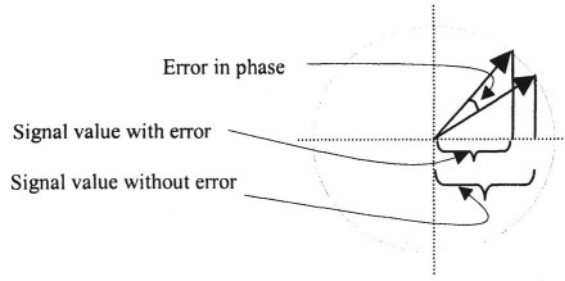


Figure 9-3. Effect of phase on amplitude

As seen from the figure, a small change in phase results in a significant change in the amplitude. The signal power is proportional to the square of the amplitude, thus further magnifying the effect of error.

It should be obvious by now that performance improvement at the PHY is an inflexible and complex task. It can be done only at the time of layer design. Some factors can still be controlled. Two of them are the medium resistance and capacitance. Resistance results in signal attenuation and can be dealt with by using the minimum practical length of cable. Capacitance results in reduction of effective signal power. In early telephone systems, the existence of capacitance was responsible for the reduced span of telephone network until inductive loading was discovered. The inductive loading resulted in expanding the length of subscriber loops.

9.3. Data Link Layer Performance

The job of data link layer is more straightforward as compared to the physical layer. It is entrusted with exchanging blocks of data with another DLC layer directly across the physical link. It does so by setting up a logical channel first. Following the channel setup, it uses many frame types for exchanging data, acknowledgements and control information. The data exchanged between two DLC protocols consists of DLC header and the user data (user data being the network layer PDU in general). Therefore, if it uses a physical link with capacity *C,* all that capacity is not used for user data transfer. Part of it goes as overhead in the following categories:

1. Call setup overhead. It is the time that it takes to setup the logical connection. If the call setup time is t_s seconds, then this overhead is equivalent to consuming $C \times t_s$ bits for call setup.
2. Information frame overhead: It is the part of the DLC information frame that consists of protocol control field, such as synchronization, addressing,

sequencing, error control field etc. If a frame of length L has x overhead bits, this is equivalent to consuming $x/(L-x)$ fraction of capacity for this overhead, or $xC/(L-x)$ bits per second.

3. Procedural overhead: This is the overhead that results from either exchanging frames that carry no user data, or waiting to process a frame while the link is idle. Such can be the case in flow control and error control procedures. In the following, we discuss these procedures one by one.

9.3.1. Flow Control Procedures

Flow control procedures depend on the receiver's demand or permission for data packet transmissions. The receiver sends acknowledgement messages permitting or preventing further transmission. The fastest that a receiver can receive data packets is the link capacity. Every overhead processing of packets or exchanging of acknowledgements reduces this speed. Consequently, flow control always results in slowing down the data exchange rate. The amount of slowing down is usually measured in terms of link utilization discussed in Chapter 5. For example, for Stop-n-Wait, the link utilization is $1/(1+2T_p)$ where T_p is the propagation time in units of transmission times. Window flow control results in an improvement of link utilization that can approach very close to 100%, at least in theory. In practice, window flow control may raise another source of overhead, that is, the queuing of multiple packets before they are processed. This problem is diagrammatically shown in Figure 9-4.

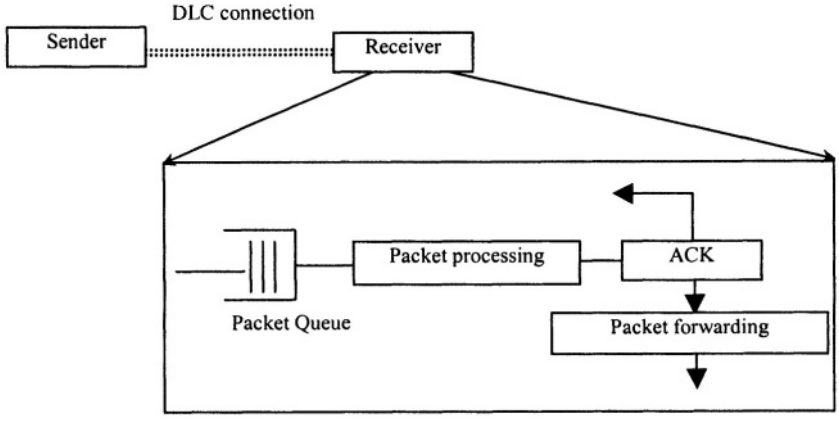


Figure 9-4. Window flow control may result in packets queueing for processing that results in extra delay.

The figure shows a sender and receiver engaged in window flow control with the receiver allowing receiving multiple packets before it sends an ACK. It may happen that, at a certain point, the processor is busy while a packet arrives. If this happens, the packet may be stored in a queuing facility to be retrieved and processed later. The queuing of packets results in extra waiting that may worsen the delay requirements of the data transmission. This is quite likely as the processor is performing many tasks together, two of which are shown, namely, packet forwarding and ACK generation and management.

Analysis of queues is an essential tool for performance measurements of networks. A typical queueing system has the following components:

(i) An arrival process describing the statistics of packet arrival.

(ii) Packet length that could be statistical or deterministic.

(iii) Queueing discipline that relates to which packet from the queue should be given next service.

(iv) Maximum queue size that pertains to the maximum tolerable delay, packet overflow conditions and storage requirements for the queue.

(v) Number of output channels, also called servers. In many queueing situations, more than one output links are available for packet processing.

(vi) Population size that could be finite, or infinite.

In queueing literature a simplified notation is commonly used that includes all or most of the above components. This notation is of the form *A/B/n/K/N,* where *A* denotes the arrival process, *B,* the service process (proportional to packet length and number of channels), *n,* the number of channels, *K,* the maximum queue size an *N,* the population size.

Example 9-1

M/D/1/5/5 denotes a queue that has Markovian arrival statistics (the inter-arrival times having the probability density function of negative exponential), constant (deterministic) packet lengths, single server, five as the maximum queue size allowed, and also a population of five. Obviously, this can be a model for a window flow control with a window size of 5.

We learn from the queueing theory that the effective capacity of a link reduces substantially as a result of queuing. In fact, due to queuing, a link can be replaced by two links, one with the same capacity as that of the original link, and the other with a capacity depending on the queue size. This is shown in Figure 9-5.

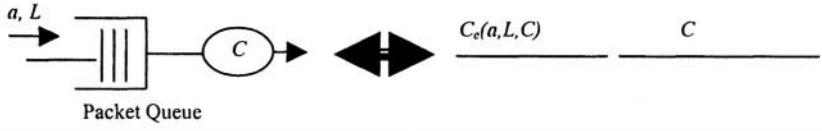


Figure 9-5. A queue is just like having an extra line in series with the original link. The extra line has a capacity C_e that is a function of packet arrival rate (a), average packet length (L), and the original link capacity.

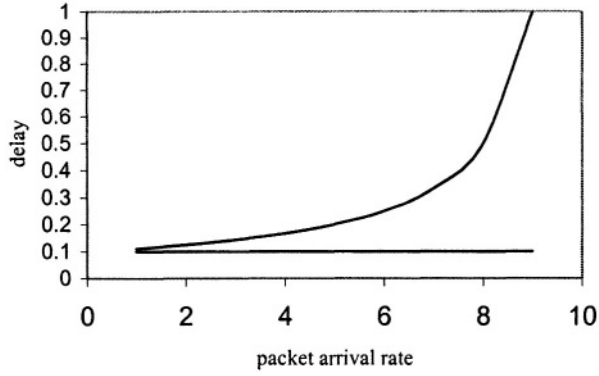
The figure shows that due to the queue, the total transmission time taken by a packet is increased from L/C to $(L/C + L/C_e)$. The new quantity C_e is a function of arrival statistics, packet length statistics and C. The second quantity, that is, L/C_e is the waiting delay and is of the statistical nature as well.

Example 9-2

It is a well known fact that if the packets arrive with a Poisson statistics, the packet lengths have a negative exponential distribution, there is a single serve, infinite maximum size and first-come-first-serve queueing discipline, then the average of the total transmission time T is given as: $T = (L/C + L/C_e) = 1/(C/L - a)$.

Figure 9-6 shows the plots of the transmission time with or without queue for $L = 1000$ bits, $C = 10{,}000$ bits and a varying between 1 to 9 packets per seconds. The lower, straight line shows a fixed transmission delay of 0.1 seconds for all values of arrival rates. The upper monotonically increasing graph line is the transmission delay due to queuing.

Figure 9-6. Effect of queuing on delay



9.3.2. Error Control Procedures

In Chapter 5, we learnt that there are potentially two error control mechanisms, forward error control (FEC) and backward error control (BEC). In FEC, error-correcting codes are used that have the capability of correcting a specified number of errors. In BEC, the receiver has the capability of only detecting the errors, but not correcting them. The transmitter keeps a copy of the unacknowledged packets. If a retransmission request is received from the receiver, the copy is retransmitted. The stored copy of the packet is discarded on receipt of an acknowledgement.

There are two types of performance related issues in error control. First issue to study is the effect of error control mechanism versus no error control mechanism. Second important issue is the overhead due to error control. In the study of error control procedures (Chapter 5), we have demonstrated how an error control procedure as simple as adding a single parity bit improves that chances of a packet with errors to be detected. In fact, in coding theory the term *coding gain* is used to compare the benefits of various codes. The coding gain is related to the amount of signal power saved due to avoiding errors as a result of coding. In this chapter, we concentrate on cost of error control. Each added bit for error control is an overhead bit and utilizes a part of the link capacity. Each retransmission also utilizes the link capacity.

9.3.2.1. Performance Models for FEC and BEC

The effect of FEC could be easily modeled as the overhead due to parity bits. The processing overhead is minimal unless the codes are implemented in software. Such is usually not the case these days. This is due

to the availability of inexpensive logic circuits, all thanks to the developments in VLSI. The coding delay in this case is equal to the length of the hardware register. However, the coding overhead is quite significant and codes of rate $^1/_3$ and ½ are common. A code of rate ½ has 50% overhead bits.

The coding overhead is much less significant in case of BEC. In fact, we saw that the CRC field is fixed at two to four bytes. This is not much for a typical frame length of one thousand bytes. However, due to retransmission, the link is used more than once for the same packet. That is the source of overhead.

It can be shown easily that if q is the probability that a packet may need retransmission, then the packet arrivals at the sending station are increased inversely proportional to $1-q$. The queue model in Figure 9-7 shows how.

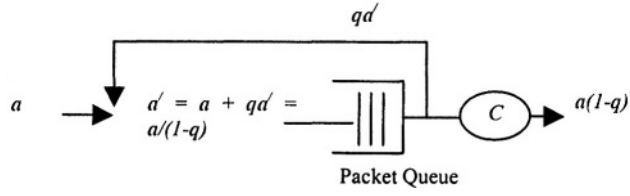


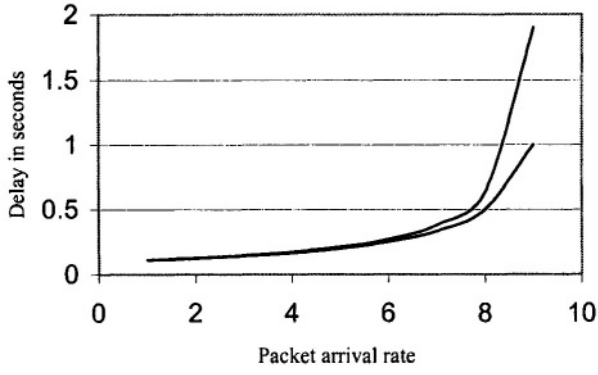
Figure 9-7. The effect of BEC on queueing model of flow control.

Thus, the effect of BEC is to increase an actual packet arrival rate from a packets per second to $a/(1-q)$ packets per second. This will have an automatic effect on delay being increased. There will result in drop in the link utilization as well. The above is a very simplified model of error control. An exact model will be much more complex and harder to solve.

Example 9-3

If we plot the same graph as in Figure 9-6 with $q = 0.05$, we will get a curve with increased delay. This is shown in Figure 9-8 below. The new value of T is
$1/[C/L - a/(1-q)]$. The delay gets worse as a or q increases.

Figure 9-8. Effect of BEC. Upper curve shows delay with q = 0.05. Lower curve shows delay without errors



9.4. Performance of the MAC Sublayer

The MAC sublayer is responsible for two main functions: channel access and multiple accesses. The channel access function decides which station will transmit next while the multiple access function decides for how long, or at what speed the transmission is allowed. Table 9-1 shows how these functions are implemented in various categories of MAC protocols.

In contention based MAC protocols, the amount of traffic load offered to the medium is higher than the successfully transmitted traffic. This is due to packet collisions. The performance of MAC protocols has, therefore, two factors to it (load, throughput). Load (ρ) is defined as the product of the packet generation rate on the medium (a) and average packet length in time (L/C); and the throughput (S) is defined as the fraction of the load successfully carried. The performance of the MAC protocols is defined as maximum throughput as a function of load $S^*(\rho)$. For example, as long as the packet intergeneration times and lengths have both negative exponential distributions, the maximum throughput for ALOHA is known to be about 18%. This happens when the load is 50%. For loads above 50%, the throughput starts getting below 18%. It is almost doubled for slotted ALOHA and goes into the 40% for CSMA based systems.

Table 9-1. Channel access and multiple access in MAC protocol categories

MAC protocol category	Channel access mechanism	Multiple access mechanism
Pure ALOHA	contention	maximum packet size
Slotted ALOHA	contention	slot size
Token passing	token	maximum time per address
TDMA/FDMA/CDMA	contention or reservation	speed controlled by resource allocation
CSMA	contention with caution	maximum packet size
Polling	in turn	maximum data size
Reservation (ALOHA etc.)	contention	maximum time limit, or data size

Most of the actual MAC protocols are much harder to analyze because of their complexity. Wireless LANs have the most complex MAC protocols and are close to impossible to analyze due to many factors that keep varying with time. For example, one of the phenomena observed in wireless LANs is the *hidden terminal effect* (HTE). In HTE, two wireless terminals can't see each other, even though they can both access and be accessed by the access point. In such cases, assumptions are usually made to get an approximate account of the throughput.

It can be safely claimed that many functions at the DLC layer can be tuned to provide optimum performance. The performance parameters can be set according to a specific environment at the time of system configuration. Alternatively, two stations can negotiate the optimum parameters during call setup phase. Both approaches are used in practice. It is hard to put a unique value set for (**r**, **d**) on a DLC protocol.

9.5. Performance of the network and higher layers

Among the higher layers, the network layer is the most crucial from a performance point of view. This layer performs a variety of functions that help a data packet find its way across a network or Internet to the destination. Even in state-of-the-art technology, this layer is the bottleneck of performance merely due to two reasons.

1. It is highly complex and deals with a large number of issues, related to addressing, forwarding, queue management and signaling.
2. There are a variety of ways of implementing each network layer protocol.

Consequently, the performance of a network is a strong function of implementation. It is the position of the author that many current network layer protocols are not thoroughly defined leaving performance 'holes'. This is perhaps due to the complexity of this layer, as said above.

9.5.1. Connectionless and Connection-oriented Protocols

As shown in Figure 9-9, a network node consists of a switching fabric and a number of incoming and outgoing links. At each incoming link, the data link layer protocol queues an arriving packet in the network protocol queue. Once a packet is in the network layer queue, it may start getting processed by the network layer. The processing depends, among other functions, on the orientation of connection. In a connectionless protocol, such as IP, there is a lot more processing than a connection-oriented case.

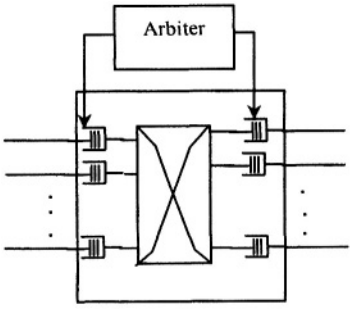


Figure 9-9. A switching node with multiple inputs and outputs.

In connection-oriented protocols, the call setup phase consists of control packets traversing the network path in both directions. Each node on the way allocates some resource to the source of this packet and makes an entry in a routing table. Due to the statistical nature of the traffic, the sum of the allocated resources is sometimes permitted to be higher than the processing and transmission capacities. Temporary hiccups in performance may occur at times when the data is arriving faster than the manageable amount. Queuing mechanisms may provide some temporary relief from such situations. At the end of data transmission, the traffic source signals to the network to release the reserved resources for other users. In this way, the following delays may occur during transmission of data in a connection-oriented mode.

1. Signaling additive for call setup and release.

2. Temporary queueing of packets when a link packet is busy.
3. Propagation delay.
4. Some processing delay while the routing table is being consulted just before forwarding the packet.
5. Flow and error control delays only if implemented.
6. Since there are many input queues, it is possible that additional mechanisms are added to the switch for arbitration of the output links. The arbitration process adds some delay as well.

For a given set of packet arrival and length statistics, the performance of such a node depends on the cumulative effect of all the above factors.

There are no call setup and release phases in a connectionless mode of data communications. Each packet is treated like a new call. The network nodes maintain a table derived from communicating either with the immediate neighboring nodes or depending on an agreement between the owners of nodes. In some cases, the route can be specified in the packet itself. When the packets from different sources do not need to be differentiated in terms of performance requirements, this seems to be a simple procedure consisting of receiving and forwarding datagrams. Routing of such datagrams consists of reading their network address (destination address) and then forwarding them on an outgoing link according to the routing table. Since it is not possible to regulate traffic in this case, a common queue is the only way to make sure that the datagrams get serviced according to a fair mechanism. Figure 9-10 shows a possible performance model for such routing.

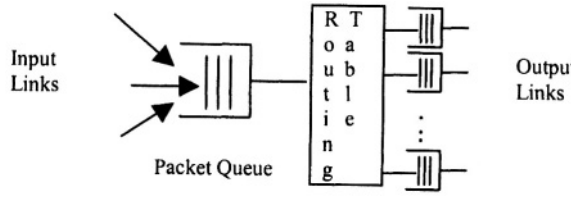


Figure 9-10. Performance model for connectionless routing

If the protocol is making use of some flow and error control mechanisms, they will add to the delay. The connectionless protocols become exceedingly complex if packets from various source-destination pairs need to be differentiated in terms of quality of service (QoS). Then a QoS-based routing mechanism is necessary, which we discuss next.

9.5.2. QoS Differentiation in Connectionless Protocols

In some connectionless protocols, none of the quantities in (**r**, **d**) are promised by the network layer. The Internet Protocol (IP) is an example of such a protocol. It is suitable for data traffic that does not have too stringent requirement for delay. A higher layer protocol could use retransmissions to assure reliability in such networks. In applications that place certain bounds on end-to-end packet delay, these protocols can't perform unless a major change is incorporated in their functions, procedures or both. The Internet Society, in its endeavors to establish a multimedia infrastructure has approved several changes to the IP based networks. Some of them relate to the suggestions of new service architectures using resource accounting and traffic control. Two of these architectures, the Integrated Services and Differentiated Services have captured most attention recently. At the heart of both of these is queue management at the routing nodes *(see Figure 9-11).* Since all the packets in the network do not have the same QoS requirements, a single queueing discipline applied to all can't guarantee performance. There have been several breakthrough proposals in service mechanisms of such a queue and some of the options are described in the next sections. These are: (i) priority queueing, (ii) fair queueing and (iii) custom queueing.

9.5.2.1. Priority Queueing

In this queueing mechanism, traffic is categorized according to some priority. A separate queue is maintained for each priority type. In other words, each queue is assigned a priority number, say, 1 through *P*. The queuing number 1 has the packets with the highest priority, while the queue number *P* stores traffic with the lowest priority. A packet arriving at the network layer is expected to contain the information of its priority level. Each priority queue has a first-come-first-serve discipline.

There are several complex issues relating to this queueing system. Of these, the most challenging issue is the allocation of priorities. QoS requirements can't always be translated into simple queue mechanisms. For some applications delay (**d**) is important, while for others loss and buffer overflow (**r**) are more important. Also, in a connectionless network this scheme can be misused in the absence of fairness enforcement. Another complicating factor is that static priority is not fair under all circumstances. With the passage of time, a packet's priority level may change. This necessitates re-queueing the packet, thus changing the service discipline, or rotating priority numbers among different queues. Such systems are extremely hard to analyze, leaving their success a rather game of chance. However, queues with static priorities are fairly simple to analyze, as one queue is completely serviced before the next priority level is taken for service. All in all, it could provide a good model for private networks.

9.5.2.2. Fair Queueing

In fair queueing mechanisms, packets are classified in accordance with the network resource allocated to them. This requires the use of some signaling protocol to allocate resources at each intermediate node. If there are K traffic classes, with class k packets allocated ϕ_k as the minimum fraction of the link capacity, a scheduling algorithm can be designed in order to meet the minimum guarantee. In many of these algorithms, the resource fraction actually allocated is different denoted by $g_k \geq \phi_k$. This is possible as long as Σ_k $\phi_k \leq 1$ or traffic from all the active sources is not present for sometime.

Analyses of fair queueing systems have been carried out for many scheduling algorithms. These analyses provide the conditions under which certain delay bounds can be met. In general, a fair queueing system is very complex to analyze exactly. However, since it is assumed that there is a traffic agreement between the traffic source and the network, this assumption makes the analysis quite workable. One advantage that fair queueing has over priority queuing is its ability to guarantee a specific amount of network resource. Due to this reason, it is incorporated in many new products for multimedia network deployment. However, there is no easy way to design a scheduling algorithm that would meet a given set of requirements of QoS for the multimedia traffic.

9.5.2.3. Custom Queueing

This is a queuing discipline halfway between the above two. In custom queuing, network resources are allocated to queues according to some priority. Consequently, the packets in a queue do not have absolute priority over packets in another queue; the queues are simply served at different rates. This is more like dividing network into many service categories according to their resource needs. There is no delay-bound guarantees, as are in fair queueing. Simple to manage and analyze it may seem that custom queuing is less efficient due to the unpredictability of traffic requirements. Consequently, it may be suitable only to enterprise networking where resources could be allocated at will by the owner.

In addition to the queueing discipline, factors such as resource negotiation, traffic contract enforcement and metering make QoS based routing a performance nightmare. Yet, it is the only way to provide real-time multimedia in a connectionless network until the emergence of technology solutions.

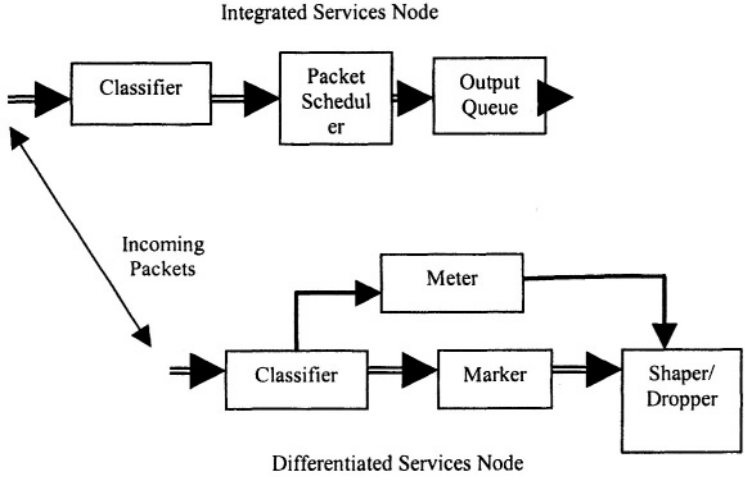


Figure 9-11: The forwarding path of an IP router providing Integrated and Differentiated Services.

In conclusion, network layer is even more complex to predict the performance parameters (**r**, **d**) than the data link control layer.

9.5.3. Performance of End-to-end Protocols

Layers 4 and above are sometimes called the end-to-end layers, or higher layers. The protocol information of these layers is not processed or changed inside the network. Other than that, there are many functions common between these and the lower layers, such as flow and error control, synchronization, addressing etc.

From a performance point of view, a packet belonging to higher layers queues up on the three layers of all the intermediate nodes. An end-to-end queuing model using only one queue at the intermediate nodes is shown in Figure 9-12.

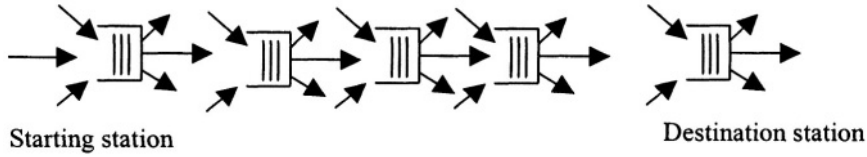


Figure 9-12. Queueing model of an end-to-end protocol.

As seen from Figure 9-12, each intermediate queue receives packets from many incoming links and routes them in a similar manner. This makes the queues dependent on one another. Due to this traffic correlation at various queues, simple queueing analyses do not hold sometimes. However, in a network with a high traffic volume, it has been observed that the traffic characteristics do not change from one point in the network to another. For such network, it is possible to treat each node in isolation and determine the end-to-end performance results from individual nodes. The volume of research in this area has recently increased and very soon we may have very reasonable and realistic traffic models to help us analyze and design efficient end-to-end protocols.

9.6. System Simulation for Performance Prediction

The performance analysis becomes intractable for most of the realistic performance models. This is due to the highly stochastic nature of traffic generation and its volume. Even if we could perfectly model the traffic generation and volume, the interaction among layers, protocols, and functions would result in performance models that could not be handled by currently available analytical tools. This problem has made system simulation a necessity for performance prediction of data networks.

9.6.1. What is Simulation?

In simulating a network, protocol, or a function of a protocol, we use computer software or some hardware emulation to mimic the actual operation conditions. Since the late 80s, the use of computer simulation has grown thanks to three main factors:

1. The ever-increasing power of the microprocessor.
2. The dropping prices of memory.
3. The availability of software packages with ready-made simulation environments.

One of the main challenges in simulation is to imitate the randomness in packet generation times and packet lengths. If the statistical properties of

these events are known, then a random number generator is used to simulate the traffic behavior.

9.6.1.1. What is a Random Number?

The events that have probabilistic outcomes may be described by a function called the random variable. A random variable is a function of outcomes and is usually assigned numerical values corresponding to each outcome. For example, if we define X as a random variable to describe the outcome of a coin toss, we can arbitrarily assign a number, say 1, when the outcome is Heads, and another number, say 0, when the outcome is Tails. In a fair coin with equal probability (that is, ½) for getting Heads or Tails in a toss, we can also have the mathematical formulation for X as follows:

$$\text{Probability}\{X = 1\} = ½ \text{ and } \text{Probability}\{X = 0\} = ½$$

Of course, the probabilities could be different, and also X could be defined to have either discrete or continuous values. It is customary to define a random variable for probabilistic events and then infer the properties of event from the probability description of the random variables. There are many well-known probability distributions of random variables: for example, the normal distribution for a continuous random variable and Poisson distribution for a discrete random variable. It is possible to represent a random variable from one type of distribution as a function of another random variable with a different distribution.

9.6.1.2. The Uniform Random Variable

One type of random variable of particular interest is the uniform random variable. A uniform random variable is the one that has values in a continuous range with the probability of a sub-range being independent of the location of the sub-range. For example, if X is uniformly distributed between 0 and 1, then it can assume any value in that interval. Besides, the probability that X is between, say, 0.1 and 0.3 is the same as that it is between 0.6 and 0.8. In real life, a uniform random variable represents absolute uncertainty of having a particular outcome among the whole range of outcomes.

Since the actual number of values in any range is infinite, a computer can't simulate a continuous random variable. It can however generate a random variable that can take a fairly large number of values in a range. The computer, however, does not generate a variable; it only generates a number that could be different every time it is generated. In other words, a computer *simulates* a random variable by *generating* a random number. Many programming languages have random number generator as a keyword. It can be used again and again and can imitate a random event. By generating an appropriate number of random number types, we can model most of random

behavior in a network. Usually, we start with a uniformly distributed random variable and then express other types of random variable as a function of uniform random variable.

9.6.1.3. What is a Pseudorandom Number?

A true random number would be one that is impossible to predict by any means except in probabilistic sense. However, if we analyze the computer program that generates the random number, many times we can tell the exact sequence of the numbers to be generated. This is because all programs use a special number to randomize the generation process. If this number, called seed, has a well-defined structure then the resulting random number can be predicted exactly. Even though, the number generated thus will have the same properties as a true random variable, it is not truly random. It is called pseudorandom number.

9.6.2. Designing a Simulation Program Versus Using a Package

From the above discussion on generating a random number, it should be clear that writing one's own simulation is a complex job. It requires the exact knowledge of the problem to be simulated, as well as a thorough knowledge of how computer languages work. Finally, it requires a very high level of stochastic processes knowledge.

Many times, the network behavior needs to be understood without actually understanding the theoretical models explaining the behavior. In this situation, it helps to use some software package and using graphical user interface, simulate the network behavior. Experts in networking, communications and computer fields design these packages. They are very reliable and usually designed for most typical systems. There is a negative side to using these packages as well. First of all, by designing one's own simulation, one really gets to understand events in depth. Using graphical blocks and clicking on the 'run' button cannot compensate for this. Secondly, by using a software package, one has to stick with the assumptions made in there. Thirdly, if a better software package becomes available at a later time, one may not be able to use a simulation made from one package to another. Due to this, many simulation packages come with programming interfaces so that the user can flexibly add material to the ready-made models. Examples of such packages are MIL3's Opnet™ and free shareware network simulator (NS).

9.7. Performance of Wireless and Mobile Networks

The main difference between the wireless and fixed networks is not the cable or lack of it, but the mobility related design issues. Not only do these

issues change the network architecture, but they also require a whole new approach to the network design. Call control procedures take the bulk of system resources and require network-wide capacity planning instead of a link-wide or node-wide planning in a fixed network. Due to these differences, the road to packet based wireless networks has been long and bumpy. Finally, in wireless standards together termed as IMT-2000, we see a promising integration of real-time and non-real-time services. In the following, we will describe some of the performance related models of the wireless and mobile networks.

9.7.1. The Wireless Network Channel

When we shift focus from fixed networks to wireless ones, the first issue to consider is the effect of channel on signal properties. In cable networks, the signal energy remains confined to within the cable and shielding provides protection against already manageable external interference. In fact, some cables, like optical fiber, do not have external interference and the signal attenuation is dependent mostly on the manufacturing quality. The wireless signal travels through scattering, reflections and refraction. Besides, in an environment where wireless mobile network would be in high demand, there is typically a lot of mobility. This makes the mobile channel highly variable and statistical in nature. Accordingly, we characterize the wireless channel by these qualities.

9.7.1.1. Propagation Loss

The attenuation in wireless channels is proportional to some power (n) of the distance (d). For free space, $n = 2$. For other wireless environments, such as indoor and downtown areas, n varies typically between 3 and 5. A large number of studies in different environments have resulted in a large number of propagation models. However, not only these models are non-deterministic, but are not applicable for a general network design to cover indoor, outdoor, city and suburbs. Since some propagation models are necessary in order to decide the antenna power and coverage, the standardization organizations have recommendations about propagation loss. A wireless network designed for one type of environment may not perform as good in another environment.

9.7.1.2. Interference

Even though most of the wireless bands are licensed, there are still numerous sources of interference, both natural and manmade. In some cases, such as wireless LANs, the frequency band is usually not licensed. This gives anyone the liberty to use the same frequency at will. In addition to these factors, the wireless signal is easy to be interfered by enemy. Consequently,

the physical layer for wireless network has to have a lot more robustness to meet the reliability and readability targets. Special techniques have been designed to combat interference in a high-noise and compromise environment. These techniques, commonly known as *spread spectrum,* have been briefly described in the PHY example of IEEE WLAN in Chapter 4.

9.7.1.3. Frequency Selectiveness

Under the performance models for PHY, it has been said earlier in this chapter that the medium acts like a filter. A filter is a device that has selective response. A frequency filter is a device whose response is different for different frequency ranges. Usually cables have a smooth response over a certain range of frequencies and gradually attenuating response outside this band. This makes it easy to define a bandwidth for a cable. The wireless channel, however, has highly frequency selective response. Figure 9-13 shows an imaginary cable and wireless band. Due to the frequency selective behavior of a wireless channel, signal distortion is more prominent for some frequency components than the others. This makes it highly difficult to use a repeater or amplifier to boost the signal power. The repeaters or amplifiers boost a certain range of band equally, thus pronouncing those parts of signals that have been much less attenuated. Instead of amplification, equalization is the usual recourse to deal with frequency selective channel behavior. An *equalizer* understands the channel behavior and then mimics its inverse so that only those parts of the signal spectrum get boosted that are attenuated more.

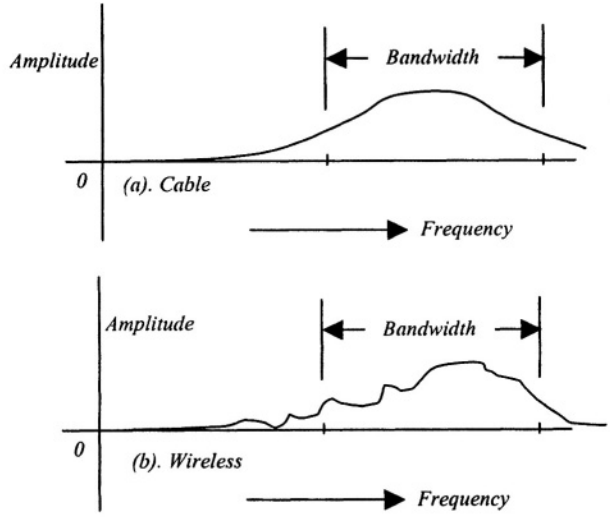


Figure 9-13. An imaginary channel frequency profile of equivalent (a) cable and (b) wireless channels.

In order to avoid the use of equalizer, we must know the range of frequencies over which the channel response remains fairly smooth. Such a bandwidth is called the coherence bandwidth of the channel. If $\boldsymbol{B}$ is the coherence bandwidth of a channel, it can allow digital signal consisting of pulses approximately $1/\boldsymbol{B}$ seconds in width. Thus, the coherence bandwidth upper limits the data rate without using special circuits to equalize the frequency selective response. A signal that is within the coherence bandwidth will be attenuated in such a way that all of its frequency components will be attenuated equally. Such a fading of signal is called *flat fading* and it can be cured with the help of repeaters or amplifiers. Alternatively, if a signal bandwidth is larger than the coherence bandwidth, it may fade only in selective part of the signal spectrum. Such a fading is called frequency selective fading. As said earlier, the frequency selective fading is harder to cure and equalization is used for this purpose to invert the channel effect.

9.7.1.4. Time Selectiveness

Not only is a wireless signal subject to the frequency selective behavior of the channel, but also its time selectiveness. The result of time-selectiveness is a variation of characteristics as a function of time. The signal could fade for a long time or quickly for many short durations. If a signal fades away for a long time, we call this slow fading. If the signal fades rapidly, then this behavior is called as fast fading. One measure of the

rapidness of fading is the bit duration in digital transmission. Accordingly, a channel is slow fading as long as fading duration is longer than the bit duration and it is fast fading if the fading duration is shorter than the bit duration. Just like the frequency-selective fading, the fast fading is hard to invert and time equalization is needed for this purpose. It can be avoided by restricting the signal duration within the fading time, called *coherence time.* If $\boldsymbol{T}$ is the coherence time, during which fading remains flat, then its bandwidth should be restricted within $1/\boldsymbol{T}$ Hz.

9.7.1.5. Multipath

Multipath is the phenomenon responsible for some of the channel characteristics discussed above. It relates to the signal taking multiple paths to the receiver. Figure 9-14 shows a simple example due to reflection.

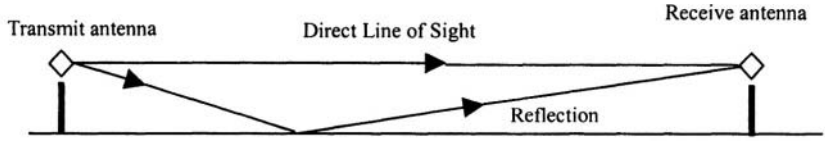


Figure 9-14. Multipath phenomenon due to reflection.

Due to different distance traveled by each multipath signal, there are many replicas of the signal arriving at the receiver with some time difference. Sometimes, this spreading of the receiving time is called the *delay spread.* The coherence bandwidth is inversely related to the delay spread. The spreading pattern changes due to the mobility of the transmitter, receiver or the obstacles around them. This mobility also induces *Doppler's effect*, which is the spread in the frequency spectrum of the received signal. The time coherence is inversely related to the Doppler's bandwidth.

9.7.1.6. Diversity

Even though the multipath appears to be only a destructive phenomenon, it has also been used constructively. In one type of receiver, called the rake receiver, the signals from all paths are combined constructively to form a stronger received signal. This phenomenon of using multipath to combine the signal components is sometimes called as *path diversity.*

Diversity is a general mechanism used extensively in wireless communication systems to enhance the signal and treat the channel effects. In

frequency diversity, more than one carrier frequencies are used to transmit the same information signal - to combat frequency-selective behavior. Accordingly, if one part of channel spectrum is weak at a time, the signal at another frequency may be stronger. In time diversity, same signal is transmitted more than once, so that if the channel characteristics vary with time, then a strong signal may be detected during at least one of the times that it was sent. In *space diversity* multiple antennas or multiple polarizations of electromagnetic waves are used to diversify the reception.

In the diversity systems, there are two methods used to combine the received multiple signals. In one method, called maximal ratio combining *(MRC),* all the received components are combined after delay compensation, such as in rake receiver. In switched diversity systems, only the strongest component of the received signal is used.

It may be remarked at this point that in most of the wireless systems, reliability is such an enormous issue that not much attention has been paid to the other metric of performance measure, the delay. This is naturally changing with the packet mode transmission approved for 3rd and higher generations of wireless networks.

9.7.2. Resource Management in Wireless Networks

A wide area wireless mobile network generally consists of a mobile switching center (MSC) connecting many basestation controller systems (BSC). Each of the BSC is further connected to a number of many basestations (BS). The basestations are transceiver sets with the capability of sustaining many connections simultaneously. The mobile stations (MS)

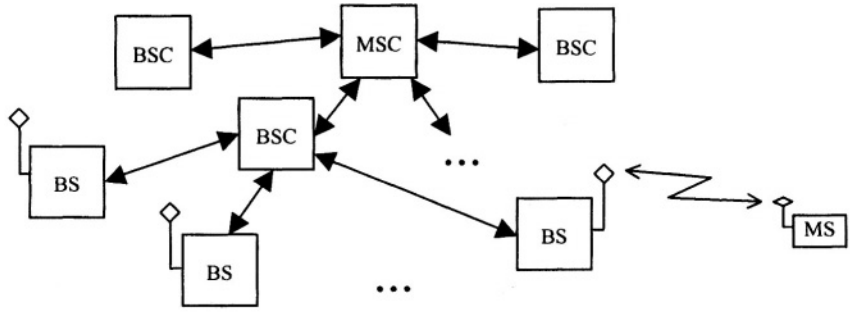


Figure 9-15. A wireless mobile network configuration.

communicate with the basestation only. See Figure 9-15 for an example.

In most of the systems worldwide, the MSC is responsible for allocating network resources, communicating with the telephone network,

storing and processing various databases for registration and authentication, and communicating with other MSCs in the system and out. The wireless resources are divided among the basestations in such a way that the same frequency band can be reused again and again. This organization of the frequency resource is called frequency planning. In *cellular* networks, the frequency planning is done by dividing a geographical area into small service areas, each called a cell. Usually, the frequency spectrum is divided into equal bands so that in a group of cells, each cell can use a different frequency band. Such a group is called a *cell cluster.* Each cell in a cluster uses different frequency band from other cells. Thus, the users in a single cluster do not interfere with one another. Adjacent clusters are designed so that the cells using same frequency bands are at the farthest possible distance. Cells using the same frequency bands are called co-channel cells. Figure 9-16 shows a cellular system layout with two clusters, each having 7 cells in it. The system frequency spectrum is divided into seven bands designated as $f_1, f_2, f_3, f_4, f_5, f_6$ and f_7. The figure shows hexagonal cell geometry. In practice, cells with any other shape could be used as long as they cover a whole area. The hexagon has some geometric properties that give them advantage over other shapes. The center of each cell contains one BSC and one or more BS's, depending upon whether the antennas used are omnidirectional or directional.

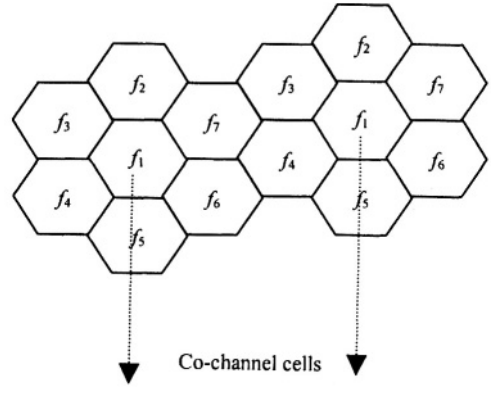


Figure 9-16. An example of cell geometry using 7 co-channel cells in a cluster.

Frequency planning, together with antenna number and type are the subject of cell planning. The cluster geometry and distances between co-channel cells are studied under network planning.

The cellular network planning is done keeping many factors in view that impact the performance. For example, some important issues are:

1. What is the cell radius? This impacts design issues such as coverage and mobility handling. Mobility handling is discussed in a section to follow.
2. What is the distance between co-channels? The mobile stations and antennas in co-channels are the source of interference for each other. It is well known that the interference characteristics of the above layout are a function of the ratio of cell radius and co-channel distance.
3. How is the frequency resource to be used in each cell? There are several options such as:
(a) Frequency division multiple access (FDMA) in which the frequency band is divided into many smaller band, each sufficient for a single call. These smaller bands are called the frequency channels.
(b) Time division multiple access (TDMA) in which all the system bandwidth is allocated to a single transmitter/receiver pair for a fixed, brief period of time, called a TDMA channel.
(c) Code division multiple access (CDMA) in which spread spectrum communication is used for multiple access, All transmitters use the whole bandwidth but different spreading code.

The performance of a mobile system depends on frequency planning and cell planning. The general trend in mobile system engineering is to move away from FDMA to TDMA and CDMA. The third generation systems are predominantly CDMA-based. There have been various claims about the capacity of a CDMA system being much larger than the other two, however, there are no real systems to prove that point. The popularity of CDMA follows from several other benefits, such as easy cell planning for multimedia. The performance of CDMA systems is admittedly less rigorously known than the other two types. TDMA and FDMA have fixed performance depending on the number of frequency/time channels and the grade of service (GoS). The grade of service (GoS) is defined as the probability that a call will not get system resources due to all channels being already occupied. For public phone network, PSTN, a value of 2% is recommended by regulating agencies. Cellular phone companies generally target a GoS of 1%.

9.7.3. Mobility Management in Mobile Networks

The real strength of wireless networks is in providing a user the freedom to move around. The challenge of designing such a network is in allowing mobility with efficiency. Mobility management deals with registration of legitimate users, their roaming record and transferring a call from one basestation to another without interruption when the user moves closer to another basestation. The definition of being close to a basestation is not in the real physical sense, it is rather in the *radio* sense: meaning the closest basestation is the one from which the user gets the strongest signal.

9.7.3.1. Handoff

The process of a mobile terminal changing the basestation is called a *handoff*. When a user moves from one basestation to another within the same system, it is called the intra-system handoff. Intersystem handoff occurs when a user moves to a different system. The handoff process is performance sensitive in many ways:

1. It requires extra bandwidth resource. The reason for this is that if a handoffed user is treated just like a new call in the new cell, there is a probability (GoS) of its being blocked. Even though there is an equal probability of being blocked when the user originated the call, it is more annoying to be cutoff while in communication. If extra resources are allocate to handoffed calls the annoyance of being dropped could be reduced. However, the system performance is worse from the bandwidth utilization point of view. The situation becomes even more constrained when a user is handoffed to another system. The MSC of the other system may not have the same call control mechanisms as the service provider of the current system. In such cases, companies usually purchase bandwidth on a volume or call basis and charge the user extra money for roaming.

9.7.3.2. Registration

Registration of users is necessary for many reasons: firstly, in order to prevent unauthorized use of the system, and secondly, to keep track of the user for routing the incoming calls. Usually two databases are used by MSC for this purpose: the home location register (HLR) that stores the user service profile and the current location of the subscribers to this MSC and a visitor location register (VLR) that stores the identification of users roaming in this cluster. VLR also informs the HLR of a roamer so that an incoming call can be directed to this system.

All of the above factors impact the performance of a wireless mobile system. Current wireless data systems have either less mobility than cellular systems (e.g. IEEE WLAN), or no mobility (e.g., broadband wireless Internet access systems). But, due to the medium, their performance analysis is still much more difficult than the fixed networks.

9.8. Review Questions

1: If the performance of layers in a layered model can be ordered into a hierarchy, such that there is one layer that performs the best and another that performs the worst, then which layer is the determining factor for network performance, the best or the worst?
2: Efficient modulation and channel coding have been regarded as two methods to enhance the performance of PHY. Can the two be combined together to give more efficiency?
3: Give example of parameter tuning of some function at layer two?
4: What is the benefit of choosing a constant size packet, such as in ATM, over variable size, such as in HDLC?
5: What is the difference between switching and routing? Which one is expected to perform better in terms of delay?
6: Flow control is performed on DLC and transport layers. At which layer is it easier to analyze flow control and why?
7: List one benefit and one disadvantage of writing your own simulation program?
8: Can the computer-generated lotto be cheated?
9: It is stated that in CDMA all users use all the bandwidth, does this cause co-channel interference?
10: What is the maximum number of users possible in a system with N channels for (i) FDMA, (ii) TDMA and (iii) CDMA?
11: What is the frequency reuse factor in cellular systems?

References

- Abramson, N. and Kuo, F., *Computer Communication Networks*. Englewood Cliffs, NJ: Prentice Hall, 1973.
- Agraval, G. P., *Fiber-Optic Communication Systems*, *3rd Edition*, John Wiley and Sons, 2002.
- Agrawal, P. and Sreenan, C., "Get Wireless: A Mobile Technology Spectrum", IT Pro, August 1999.
- Ahuja, V., "Routing and flow control in systems network architecture", IBM Syst. J. 18, 2 (1979), 293-314.
- Al-Dhahir, N., Diggavi, S. N., "Maximum throughput loss of noisy ISI channels due to narrow-band interference", IEEE-Communications-Letters. vol.5, no.6; June 2001; p.233-5.
- Aslett, M., "Slim line [network computer]", Unix-&-NT-News. no. 129; Dec. 1999-Jan. 2000; p.34-5. 2000
- Bertsekas, D., and Gallager, R., *Data Networks*. Englewood Cliffs, NJ: Prentice Hall, 1992.
- Black, U., *Data Link Protocols* Englewood Cliffs, NJ: Prentice Hall, 1993.
- Bradner, S. O. and Mankin, A., *IPng Internet Protocol Next Generation.* IPng Series. Reading, MA: Addison-Wesley Longman, 1996.
- Comer, D., and Stevens, D., *Internetworking with TCP/IP*, *Volume 2: Design Implementation*, *and Internals*. Upper Saddle River, NJ: Prentice Hall, 1999.
- Bennett, D.B. and Sohn, S. M., "Layered Networks, {A} New Class of Multistage Interconnection Networks", Proceedings of the 1991 International Conference on Parallel Processing, VOL I, Architecture, 1991, CRC Press
- Davidson, J., Peters, J., Gracely, B., and Peters J., *Voice over IP Fundamentals,* Cisco Press, 2000.
- De Prycker, M., *Asynchronous Transfer Mode Solution for Broadband ISDN.* 3rd Edition. Upper Saddle River, NJ: Prentice Hall, 1995.
- DiMarzio, J. F., *Network Architecture and Design: A Field Guidefor IT Consultants,* Indianapolis, IN; Sams; Hempel Hemstead, Prentice-Hall 2001.

- Drewes, C., Aicher, W. and Hausner, J., "The wireless art and the wired force of subscriber access", IEEE-Communications-Magazine. vol.39, no.5; May 2001; p.118-24.
- Eskafi, F. and Zandonadi, M., "A network layer for intelligent vehicle highway systems", Proceedings of the Applied Telecommunications Symposium (ATS'99). 1999 Advanced Simulation Technologies Conference. SCS, San Diego, CA, USA; 1999; viii+271 pp.p.157-62.
- Even and Litman, "Layered Cross Product--{A} Technique to Construct Interconnection Networks", NETWORKS: Networks: An International Journal, VOL29, 1997.
- Feinler, J. Jacobsen, J. and Stahl, M., *DDN Protocol Handbook Volume 2, DAPRA Internet Protocols*: DDN Network Information Center, SRI International, Menlo Park, CA: December 1985.
- Forouzan, B. A., Coombs, C. A. and Fegan, S. C., *Data Communications and Networking,* McGraw-Hill Higher Education, 2000.
- Forouzan, B.,A., and Fegan, S.C., *TCP/IP Protocol Suite,* McGraw-Hill Higher Education, 1999.
- Freeman, R. L., *Practical Data Communications, 2nd Edition,* John Wiley and Sons, 2001.
- Freeman, R. L., *Telecommunications Transmission Handbook.* New York: Wiley, 1998.
- Gibson, J. D. (*Ed), Multimedia communications: directions and innovations,* Academic Press, San Diego CA, 2001.
- Goralski, W., *Introduction to ATM Networking.* New York: McGraw-Hill, 1995.
- Greeen, P. E., "An All-Optical Computer Network: Lessons Learned", IEEE Network, March, 1992.
- Halsall, F., *Data Communications, Computer Networks and Open Systems, 3rd Edition,* Workigham England, Reading MA, Addison-Wesley Publishing Co. 1992.
- Harrison, P. G., "An analytic model for flow control schemes in communication network nodes" IEEE Trans. Commun. COM-32, 9 (Sept. 1984), 1013-1019.
- Haykins, S., *Communication Systems.* New York. Wiley, 1995.
- Held, G., *Understanding Data Communications.* 6th Edition, Indiana: New Riders, 1999.
- Heldman, R. K., *Layered Architectures for a Computer Network,* McGraw-Hill, 1992.
- Henckel, L. and Kuthan, J., "The network computer for an open services market", IFIP TC-6 Eighth International Conference on

High Performance Networking (HPN'98). Kluwer Academic Publishers, Norwell, MA, USA; 1998; xii+699 pp. p.497-507.

- Hershey, P. C. and Silio, C. B. Jr., "Time transformed machine for high speed computer network performance measurement", Globecom '00 - IEEE. Global Telecommunications Conference. Conference Record (Cat. No.00CH37137). IEEE, Piscataway, NJ, USA; 2000; 3 vol. xlvi+1898 pp. p.684-9 vol.1.
- Hessel, C., "Using retransmission information in a black side data link layer", MILCOM 2000 Proceedings. 21st Century Military Communications. Architectures and Technologies for Information Superiority (Cat. No.00CH37155). IEEE, Piscataway, NJ, USA; 2000; 2 vol. xxvii+1238 pp. p. 1083-8 vol.2.
- Hong, S. H. and Ko, S. J., "A simulation study on the performance analysis of the data link layer of IEC/ISA fieldbus", Simulation. vol.76, no.2; Feb. 2001; p.109-18.
- Horak, R., Miller, M.A., and Newton, H., *Communications Systems and Networks,* New York, John Wiley and Sons, 2000.
- Jain, R.,"A timeout-based congestion control scheme for window flow-controlled networks" IEEE J. Sel. Areas Comraun. SAC-4, 7 (Oct. 1986), 1162-1167.
- Kleinrock, L. *Queueing Systems-Vol, 1 Theory.* New York: Wiley, 1975.
- Kleinrock, L., "On flow control in computer networks", In Proceedings of the International Conference on Communications (June 1978), pp. 27.2.1-27.2.5.
- Kurose, J. F . and Ross, K. W ., *Computer Networking: A Top-down Approach Featuring the Internet,* New York, Addison-Wesley, 2002.
- L. W. Brinn, "Computer Networks", ACM SIGCSE Bulletin , Proceedings of the sixteenth SIGCSE technical symposium on Computer science education March 1985, ACM Press New York, NY, USA , Pages: 135-139 Series-Proceeding-Article Volume 17 Issue 1.
- LaMaire, R., Krishna, A. and Bhagwat, P. *Wireless LANs and Mobile Networking: Standards and Future Directions.* IEEE Communications Magazine, Vol. 34, No. 8, August 1996.
- Laquey, T. L., *User's Directory of Computer Networks.* Bedford, Ma.: Digital Press, July 1989.
- Lee, W. C. Y., *Mobile Cellular Communications Systems.* New York: McGraw-Hill, 1995.
- Lee, W.C.Y., *Mobile Communications Engineering.* 2nd edition. New York: McGraw-Hill, 1998.

- Mankin, A. and Ramakrishnan, K.K., *Gateway Congestion Control Survey.* RFC 1254. August 1991.
- Marino, P., Dominguez, M. A., Poza, F. and Nogueira, J., "Field bus data link layer development by formal description languages", Proceedings of International Workshop on Soft Computing in Industry '99 (IWSCI'99). Muroran Inst, of Technol, Muroran, Japan; 1999; ix+510 pp. p.431-6.
- Mori, H. and Sakamura, K., "A new network layer protocol with routing address and tables auto-configuration mechanism", Proceedings 20th IEEE International Conference on Distributed Computing Systems. IEEE Comput. Soc, Los Alamitos, CA, USA; 2000; xvi+708 pp. p.84-96.
- Moshos, G., *Data Communications: Principles and Problems.* New York: West Publishing Co., 1989.
- Oguchi, N., Chen, Y. M., Ogawa, J., Tsuruoka, T. Taniguchi, T. and Nojima, S., "address resolution protocol in network layer", Proceedings 23rd Annual Conference on Local Computer Networks. LCN'98 (Cat. No.98TB 100260). IEEE Comput. Soc, Los Alamitos, CA, USA; 1998; xii+400 pp.
- Pandey, A. K., "Network Management System", Telecommunications. Vol.50, no.2; March-April 2000; p.11-18.
- Park, S. C., Lee S. W., Song, Y. J., Cho, D. H. and Dhong, Y. B. "Performance improvements of data-link layer protocol in WATM", 2000 IEEE International Conference on Communications. ICC 2000. Global Convergence Through Communications. Conference Record. IEEE, Piscataway, NJ, USA; 2000; 3 vol. xxxii+1814 pp. p.1090-4 vol.2.
- Parsa, C. and Garcia-Luna-Aceves, J. J., "Improving TCP Congestion Control over Internets with Heterogeneous Transmission Media", Proceedings of the Seventh Annual International Conference on Network Protocols
- Perlman, R., *Interconnections: Bridges and Routers,* Addison-Wesley, Reading, MA: 1992.
- Piscitello, D. M., Chapin, A. L., Piscitello, D. and Chapin, B., *Open Systems Networking (TCP/IP and OSI),* Addison-Wesley Publishing Co., 1993.
- Prasad, N. and Prasad, A. (Eds), *WLAN Systems and Wireless IP for Next Generation Communications,* Artech House, 2002.
- Proakis, J. G., *Digital Communications.* 2nd Edition, New York: McGraw-Hill, 1989.
- Reed, K., *Data Network Handbook: An Interactive Guide to Network Architecture and Operations, 3rd Edition,* Thomson Learning, 1995.

- Robertazzi, T.G., *Computer Networks and Systems, 3rd Edition,* Springer Verlag, 2000.
- Russel, T., *Telecommunications Protocols,* New York, McGraw Hill, 1997.
- Schwartz, M. *Computer-Communication Network Design and Analysis.* Englewood Cliffs, NJ: Prentice Hall, 1977.
- Simpson, H. R., Matra BAe Dynamics, "Layered Architecture(s) : Principles and Practice in Concurrent and Distributed Systems", Proceedings of the 1997 Workshop on Engineering of Computer-Based Systems, Institute of Electrical and Electronics Engineers 1997.
- Sriskanthan, N., Lim, S. C. and Subramanian, K. R., "PIM-ASCII100 digital data encoding scheme for short distance wireless infrared transmission", ISCE '97. Proceedings of 1997 IEEE International Symposium on Consumer Electronics (Cat. No.97TH8348). IEEE, New York, NY, USA; 1997; xxii+312 pp. p.145-8.
- Stallings, W., *High Speed Networks: TCP/IP and ATM Design Principles.* Upper Saddle River, NJ: Prentice Hall, 1998.
- Stallings, W., *Local and Metropolitan Area Networks, 6th Edition.* Upper Saddle River, NJ: Prentice Hall, 2000.
- Stallings, W., *Data and Computer Communications, 6th Edition,* Upper Saddle River, NJ: Prentice Hall, 1999.
- Stallings, W., *Handbook of Computer Communications Standards: The Open System Interconnection, Facsimile Edition.* Upper Saddle River, NJ: Prentice Hall, 1998.
- Stephen R. Kimbleton, G. and Michael Schneider, "Computer Communication Networks: Approaches, Objectives, and Performance Considerations", ACM Press New York, NY, USA, Pages: 129–173, ISSN:0360-0300, Periodical-Issue-Article, 1975.
- Stevens, W. R., *The Protocols (TCP/IP Illustrated, Volume I),* Reading MA, Addison-Wesley Publishing Co. 1994.
- Streenstrup, M., *Mobile Communications.* Guest Editorial, IEEE Network, March/April 1994, Vol. 8, No. 2.
- Suganuma, T., Kinoshita, T. and Shiratori, N., "Flexible Network Layer in dynamic networking architecture", Proceedings Seventh International Conference on Parallel and Distributed Systems: Workshops. IEEE Comput. Soc, Los Alamitos, CA, USA; 2000; xiii+563 pp. p.473-8.
- Sux, W., Grillo, D., "Flow control in local-area networks of interconnected token rings" IEEE Trans. Commun. COM-33, 10 (Oct. 1985), 1058-1066.

- Tenunbaum, A. S., *Computer Networks, 4th Edition,* Prentice-Hall PTR, 2002.
- The Wireless LAN Alliance. *Introduction to Wireless LANs.* 1996
- Thorp, N.M. and Ross D., *X.25 made easy,* New York, Prentice-Hall, 1992.
- Thottan, M. and Ji, C., "Fault prediction at the network layer using intelligent agents", Integrated Network Management VI. Distributed Management for the Networked Millennium. Proceedings of the Sixth IFIP/IEEE International Symposium on Integrated Network Management. (Cat. No.99EX302). IEEE, Piscataway, NJ, USA; 1999; xxvi+958 pp. p.745-59.
- Van-Deventer, M.O., Ramos, J., Blain, L. Grant, R. and Skjoldstrup, B., "Node functionalities and architectures for the optical network layer, results from EURESCOM P615", Broadband Access Networks. NOC '97. IOS Press, Amsterdam, Netherlands; 1997; viii+272 pp. p.34-41.
- Wecker, S., "Computer Network Architectures", Computer VOL. 12 Pages 58-72, SEP, 1979.
- Weiss, J. and Schremp, D., *Putting Data on a Diet.* IEEE Spectrum, August 1993.
- Wetteroth, D., *OSI Reference Model for Telecommunications,* McGraw-Hill Professional Publishing, 2001.
- Zong, P. and Filip, M., "Real propagation event simulation of selective repeat data link layer protocol for VSAT FCM system", Electronics-Letters. vol.36, no. 17; 17 Aug. 2000; p. 1492-4.

Index