Computer networks (CMPC-208)

Chapter no1.

Introduction to network and protocols architecture.

We begin our study with a simple model of communications, illustrated by the block diagram in Figure l.
The fundamental purpose of a communications system is the exchange of data between two parties. Figure l.lb presents one particular example, which is the communication
between a workstation and a server over a public telephone network.
Another example is the exchange of voice signals between two telephones over the same network. The key elements of the model are

Source. This device generates the data to be transmitted; examples are telephones and personal computers.

Transmitter. Usually, the data generated by a source system are not transmitted directly in the form in which they were generated. Rather, a transmitter transforms and encodes the information in such a way as to produce electromagnetic signals that can be transmitted across some sort of transmission system.
For example, a modem takes a digital bit stream from an attached device such as a personal computer and transforms that bit stream into an analog signal that can be handled by the telephone network.

Transmission System. This can be a single transmission line or a complex network connecting source and destination.

Receiver. The receiver accepts the signal from the transmission system and converts it into a form that can be handled by the destination device. For example, a modem will accept an analog signal coming from a network or transmission line and convert it into a digital bit stream.

Destination. Takes the incoming data from the receiver.
The first item, **transmission system utilization,** refers to the need to make efficient use of transmission facilities that are typically shared among a number of communicating
devices. Various techniques (referred to as multiplexing) are used to allocate the total capacity of a transmission medium among a number of users.
Congestion control techniques may be required to assure that the system is not

overwhelmed by excessive demand for transmission services.

In order to communicate, a device must **interface** with the transmission system. All the forms of communication discussed in this book depend, at bottom, on the use of electromagnetic signals propagated over a transmission medium. Thus, once an interface is established, **signal generation** is required for communication. The properties of the signal, such as form and intensity, must be such that they are (1) capable of being propagated through the transmission system, and (2) interpretable as data at the receiver.

Not only must the signals be generated to conform to the requirements of the transmission system and receiver, but there must be some form of **synchronization** between transmitter and receiver. The receiver must be able to determine when a signal begins to arrive and when it ends. It must also know the duration of each signal element.

Beyond the basic matter of deciding on the nature and timing of signals, there are a variety of requirements for communication between two parties that might be collected under the term **exchange management.** If data are to be exchanged in both directions over a period of time, the two parties must cooperate. For example, for two parties to engage in a telephone conversation, one party must dial the number of the other, causing signals to be generated that result in the ringing of the called phone. The called party completes a connection by lifting the receiver. For data processing devices, more will be needed than simply establishing a connection; certain conventions must be decided upon. These conventions may include whether both devices may transmit simultaneously or must take turns, the amount of data to be sent at one time, the format of the data, and what to do if certain contingencies, such as an error, arise.

The next two items might have been included under exchange management, but they are important enough to list separately. In all communications systems, there is a potential for error; transmitted signals are distorted to some extent before reaching their destination. **Error detection and correction** are required in circumstances where errors cannot be tolerated; this is usually the case with data process ing systems. For example, in transferring a file from one computer to another, it is simply not acceptable for the contents of the file to be accidentally altered. **Flow control** is required to assure that the source does not overwhelm the destination by sending data faster than they can be processed and absorbed.

Next, we mention the related but distinct concepts of **addressing** and **routing.** When a transmission facility is shared by more than two devices, a source system

must somehow indicate the identity of the intended destination. The transmission system must assure that the destination system, and only that system, receives the data. Further, the transmission system may itself be a network through which various paths may be taken. A specific route through this network must be chosen.
Recovery is a concept distinct from that of error correction. Recovery techniques are needed in situations in which an information exchange, such as a data base transaction or file transfer, is interrupted due to a fault somewhere in the system. The objective is either to be able to resume activity at the point of interruption or at least to restore the state of the systems involved to the condition prior to the beginning of the exchange.
Message formatting has to do with an agreement between two parties as to the form of the data to be exchanged or transmitted. For example, both sides must use the same binary code for characters.
Frequently, it is important to provide some measure of **security** in a data communications
system. The sender of data may wish to be assured that only the intended party actually receives the data; and the receiver of data may wish to be assured that the received data have not been altered in transit and that the data have actually come from the purported sender.
Finally, a data communications facility is a complex system that cannot create or run itself. **Network management** capabilities are needed to configure the system, monitor its status, react to failures and overloads, and plan intelligently for future growth.
Thus we have gone from the simple idea of data communication between source and destination to a rather formidable list of data communications tasks. In this book, we further elaborate this list of tasks to describe and encompass the entire set of activities that can be classified under data and computer communications.

Data communication and network.

In its simplest form, data communication takes place between two devices that are directly connected by some form of point-to-point transmission medium. Often, however, it is impractical for two devices to be directly, point-to-point connected. This is so for one (or both) of the following contingencies:
The devices are very far apart. It would be inordinately expensive, for example, to string a dedicated link between two devices thousands of miles apart.
There is a set of devices, each of which may require a link to many of the others at various times. Examples are all of the telephones in the world and all of the terminals and computers owned by a single organization. Except

for the case of a very few devices, it is impractical to provide a dedicated wire between each pair of devices. The solution to this problem is to attach each device to a communications network.

Figure 1.3 relates this area to the communications model of Figure l.la and also suggests the two major categories into which communications networks are traditionally classified: wide-area networks (WANs) and local-area networks (LANs).

The distinction between the two, both in terms of technology and application, has become somewhat blurred in recent years, but it remains a useful way of organizing the discussion.

Wide-Area Networks.

Wide-area networks have been traditionally been considered to be those that cover a large geographical area, require the crossing of public right-of-ways, and rely at least in part on circuits provided by a common carrier. Typically, a WAN consists of a number of interconnected switching nodes. A transmission from any one device is routed through these internal nodes to the specified destination device. These nodes (including the boundary nodes) are not concerned with the content of the data; rather, their purpose is to provide a switching facility that will move the data from node to node until they reach their destination.

Traditionally, WANs have been implemented using one of two technologies: circuit switching and packet switching. More recently, frame relay and ATM networks have assumed major roles.

Circuit Switching

In a circuit-switched network, a dedicated communications path is established between two stations through the nodes of the network. That path is a connected sequence of physical links between nodes. On each link, a logical channel is dedicated to the connection. Data generated by the source station are transmitted along the dedicated path as rapidly as possible. At each node, incoming data are routed or switched to the appropriate outgoing channel without delay. The most common example of circuit switching is the telephone network.

Packet Switching

A quite different approach is used in a packet-switched network. In this case, it is not necessary to dedicate transmission capacity along a path through the network. Rather, data are sent out in a sequence of small chunks, called packets. Each packet is passed through the network from node to node along some path leading from

source to destination. At each node, the entire packet is received, stored briefly, and then transmitted to the next node. Packet-switched networks are commonly used for terminal-to-computer and computer-to-computer communications.

Local Area Networks

As with wide-area networks, a local-area network is a communications network that interconnects a variety of devices and provides a means for information exchange among those devices. There are several key distinctions between LANs and WANs:

1. The scope of the LAN is small, typically a single building or a cluster of buildings. This difference in geographic scope leads to different technical solutions, as we shall see.

2. It is usually the case that the LAN is owned by the same organization that owns the attached devices. For WANs, this is less often the case, or at least a significant fraction of the network assets are not owned. This has two implications. First, care must be taken in the choice of LAN, as there may be a substantial capital investment (compared to dial-up or leased charges for widearea networks) for both purchase and maintenance. Second, the network management responsibility for a local network falls solely on the user.

3. The internal data rates of LANs are typically much greater than those of widearea networks.

PROTOCOLS AND PROTOCOL ARCHITECTURE

When computers, terminals, and/or other data processing devices exchange data, the scope of concern is much broader than the concerns we have discussed in Sections 1.2 and 1.3. Consider, for example, the transfer of a file between two computers. There must be a data path between the two computers, either directly or via a communication network. But more is needed. Typical tasks to be performed are.

1. The source system must either activate the direct data communication path or inform the communication network of the identity of the desired destination system.

2. The source system must ascertain that the destination system is prepared to receive data.

3. The file transfer application on the source system must ascertain that the file management program on the destination system is prepared to accept and store the file for this particular user.

4. If the file formats used on the two systems are incompatible, one or the other

system must perform a format translation function.
In discussing computer communications and computer networks, two concepts are paramount:

- Protocols
- Computer-communications architecture, or protocol architecture

A Three-Layer Model

In very general terms, communications can be said to involve three agents: applications,
computers, and networks. One example of an application is a file transfer operation. These applications execute on computers that can often support multiple simultaneous applications. Computers are connected to networks, and the data to be exchanged are transferred by the network from one computer to another. Thus, the transfer of data from one application to another involves first getting the data to the computer in which the application resides and then getting it to the intended application within the computer.
With these concepts in mind, it appears natural to organize the communication task into three relatively independent layers:
Network access layer
Transport layer
Application layer

The TCP/IP Protocol Architecture

Two protocol architectures have served as the basis for the development of interoperable
communications standards: the TCPIIP protocol suite and the OSI reference model. TCPIIP is the most widely used interoperable architecture, and OSI has become the standard model for classifying communications functions. In the remainder of this section, we provide a brief overview of the two architectures; the topic is explored more fully in Chapter 15.
TCPIIP is a result of protocol research and development conducted on the experimental packet-switched network, ARPANET, funded by the Defense Advanced
Research Projects Agency (DARPA), and is generally referred to as the TCPIIP protocol suite. This protocol suite consists of a large collection of protocols that have been issued as Internet standards by the Internet Architecture Board l (IAB). l

There is no official TCPIIP protocol model as there is in the case of OSI. However, based on the protocol standards that have been developed, we can organize the communication task for TCPIIP into five relatively independent layers:

1-Application layer
2-Host-to-host, or transport layer
3-Internet layer
4-Network access layer
5-Physical layer

The **physical layer** covers the physical interface between a data transmission device (e.g., workstation, computer) and a transmission medium or network. This layer is concerned with specifying the characteristics of the transmission medium, the nature of the signals, the data rate, and related matters.

The **network access layer** is concerned with the exchange of data between an end system and the network to which it is attached. The sending computer must provide the network with the address of the destination computer, so that the network may route the data to the appropriate destination. The sending computer may wish to invoke certain services, such as priority, that might be provided by the network.
The specific software used at this layer depends on the type of network to be used; different standards have been developed for circuit-switching, packet-switching
(e.g., X.25), local area networks (e.g., Ethernet), and others. Thus, it makes sense to separate those functions having to do with network access into a separate layer. By doing this, the remainder of the communications software, above the network
access layer, need not be concerned about the specifics of the network to be used. The same higher-layer software should function properly regardless of the particular network to which the computer is attached.

Regardless of the nature of the **applications** that are exchanging data, there is usually a requirement that data be exchanged reliably. That is, we would like to be assured that all of the data arrive at the destination application and that the data arrive in the same order in which they were sent. As we shall see, the mechanisms for providing reliability are essentially independent of the nature of the application.

Thus, it makes sense to collect those mechanisms in a common layer shared by all applications; this is referred to as the **host-to-host layer,** or **transport layer.**

The transmission control protocol (TCP) is the most commonly-used protocol to provide this functionality.

Finally, the **application layer** contains the logic needed to support the various user applications. For each different type of application, such as file transfer, a separate
module is needed that is peculiar to that application.
Figure 1.9 shows how the TCPIIP protocols are implemented in end systems and relates this description to the communications model of Figure l.la. Note that the physical and network access layers provide interaction between the end system and the network, whereas the transport and application layers are what is known as end-to-end protocols; they support interaction between two end systems. The internet
layer has the flavor of both. At this layer, the end system communicates routing information to the network but also must provide some common functions between the two end systems.

The OSI Model

The open systems interconnection (OSI) model was developed by the International Organization for Standardization (ISO) as a model for a computer communications architecture and as a framework for developing protocol standards. It consists of seven layers:

1-Application
2-Pesentation
3-Session
4-Transport
5-Network
6-Data Link
7-Physical

the OSI model and provides a brief definition of the functions
performed at each layer. The intent of the OSI model is that protocols be developed to perform the functions of each layer.
The designers of OSI assumed that this model and the protocols developed within this model would come to dominate computer communications, eventually replacing proprietary protocol implementations and rival multivendor models such as TCPIIP. This has not happened. Although many useful protocols have been developed in the context of OSI, the overall seven-layer model has not flourished.

Instead, it is the TCPIIP architecture that has come to dominate. Thus, our emphasis in this book will be on TCPIIP.

Application Provides access to the OSI environment for users and also provides distributed information services.
Presentation Provides independence to the application processes from differences in data representation (syntax).
Session Provides the control structure for communication between applications; establishes, manages, and terminates connections (sessions) between cooperating applications.
Transport Provides reliable, transparent transfer of data between end points; provides end-to-end error recovery and flow control.
Network Provides upper layers with independence from the data transmission and switching technologies used to connect systems; responsible for establishing, maintaining, and terminating connections.
Data Link Provides for the reliable transfer of information across the physical link; sends blocks of data (frames) with the necessary synchronization, error control, and flow control.
Physical Concerned with transmission of unstructured bit stream over physical medium; deals with the mechanical, electrical, functional, and procedural characteristics to access the physical medium.

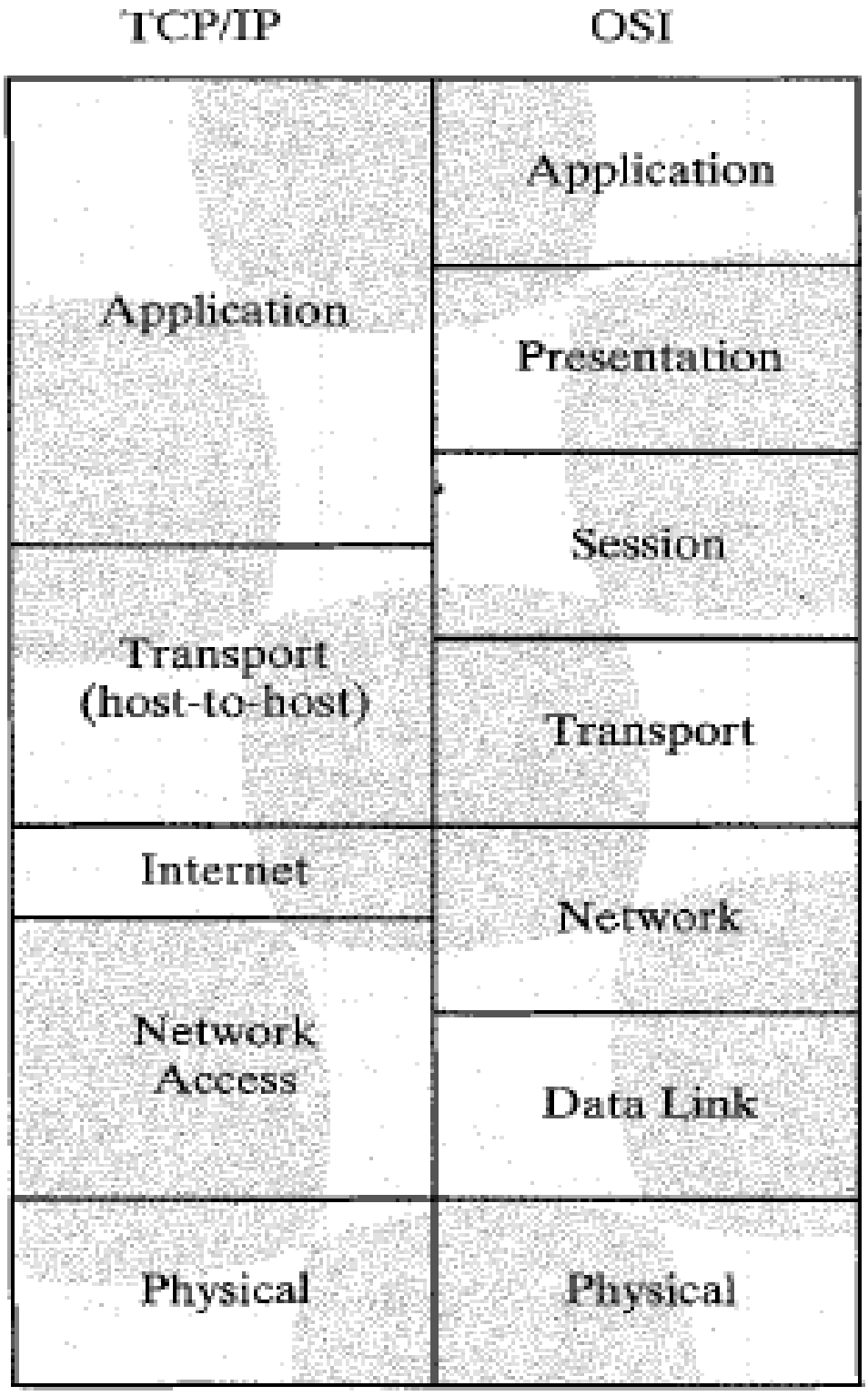

FIGURE 1.11 Protocol architectures.