



IT111 - Introduction to Information Technology

Bachelor of Science in Information Technology

LECTURE 10 COMPUTER NETWORKS AND INTERNET SOFTWARE ENGINEERING

Prof. Lamiaa Fattouh Ibrahim

Everyone Smart University

Faculty of Informatics

Spring 2023



Objectives

- ☐ Describe local and wide area networks (LANs and Wans).
- ☐ Distinguish an internet from the Internet.
- ☐ Describe the TCP/IP protocol suite as the network model in the internet.
- ☐ Define the layers in the TCP/IP protocol suite and their relationship.

Introduction

A network is a combination of **hardware** and **software** that sends data from one location to another.

The hardware consists of the physical equipment that carries signals from one point in the network to another.

The software consists of instructions that make the services that we expect from a network possible.

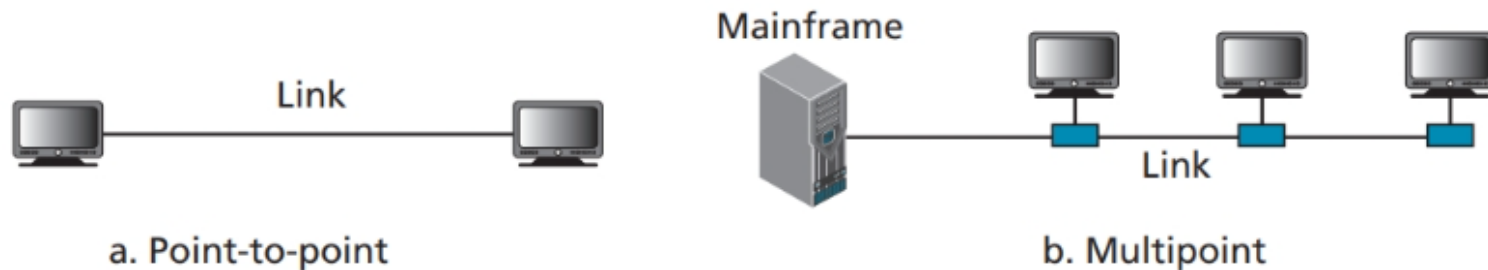
6-1 OVERVIEW

6.1.1 Networks

- A **network** is the interconnection of devices capable of communication.
- A device can be a **host** (such as a computer, desktop, laptop, workstation, mobile phone, or security system) and can be a **connecting device** such as a **router** connecting the network to other networks, a switch which connects devices together, a modem (modulator-demodulator) that changes the form of data, etc.
- These devices in a network are connected using wired or wireless transmission media.
- When connecting two computers, we have created a network, although very small.

Type of Connection

Figure 6.1 *Types of connections: point-to-point and multipoint*



A **point-to-point** connection provides a dedicated link between two devices, and the entire capacity of the link is reserved for transmission between these two devices.

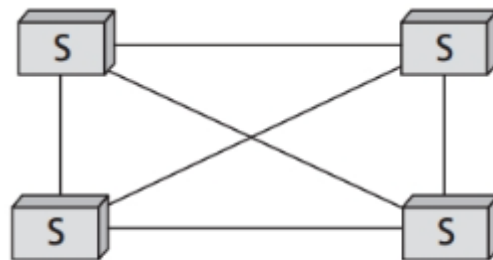
A **multipoint connection** (also called multidrop connection) is one in which more than two specific devices share a single link (Figure 6.1.b). In a multipoint environment, the capacity of the channel is shared, either spatially or temporally.

Physical topology

The term physical topology refers to the way in which a network is laid out physically. Two or more devices connect to a link; one or more links form a topology. The topology of a network is the geometric representation of the relationship of all the links and devices (usually called nodes) to one another. There are four basic topologies possible: mesh, star, bus and ring

Mesh Topology

In a mesh topology, every device has a dedicated point-to-point link to every other device.



a. Mesh

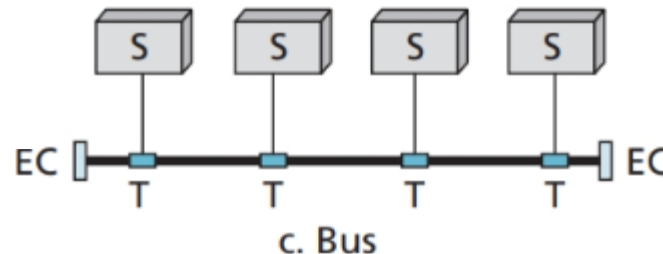
S: Station

Bus Topology

A bus topology uses a multipoint link. One long cable, called the bus, acts as a backbone to link all the devices in a network. Nodes are connected to the bus cable by drop lines and taps (connectors).

T: Tab

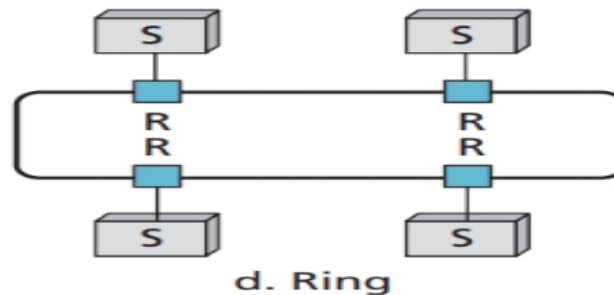
EC: End Cable



Ring Topology

In a ring topology, each device has a dedicated point-to-point connection with only the two devices on either side of it. A signal is passed along the ring in one direction, from device to device, until it reaches its destination. Each device in the ring incorporates a repeater. When a device receives a signal intended for another device, its repeater regenerates the bits and passes them along.

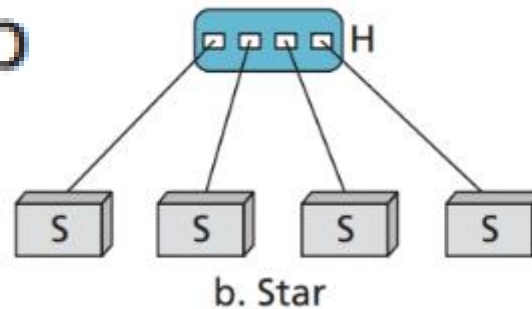
R: Repeater



Star Topology

In a star topology, each device has a dedicated point-to-point link only to a central controller, usually called a hub.

H: Hub



Categories of networks

Today networks can be divided into three broad categories:

- 1) **Local-area networks (LANs):** A LAN normally covers a small area, while a MAN covers a small city or town.
- 2) **Wide-area networks (WANs):** a WAN can be worldwide
- 3) **Metropolitan area networks (MANs):** a MAN covers a small city or town.

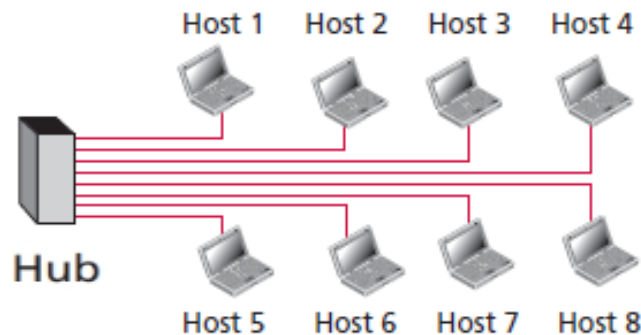
Local area network

A **local area network (LAN)** is privately owned and connects hosts in a single office, building, or campus.

LANs can be as simple as two PCs and a printer in someone's home office, or it can extend throughout a company and include audio and video devices.

Each host in a LAN has an identifier, an address, that uniquely defines them. A packet sent by a host to another carries both the source host's and the destination host's addresses.

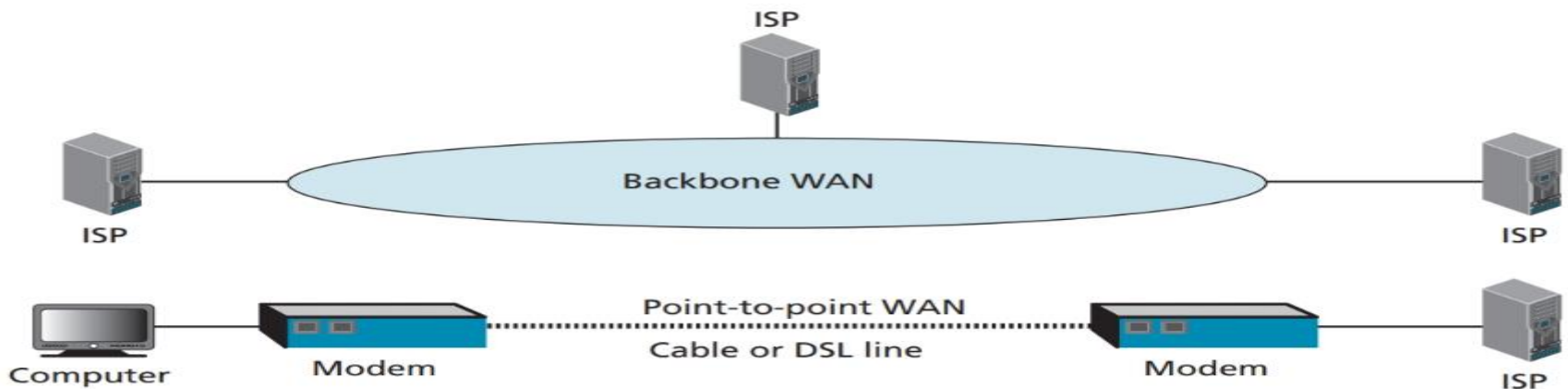
Figure 6.1 *Example of a LAN*



Wide area network

A **wide area network (WAN)** is an interconnection of devices capable of communication.

Figure 6.4 A point-to-point WAN and a backbone WAN



Metropolitan Area Networks

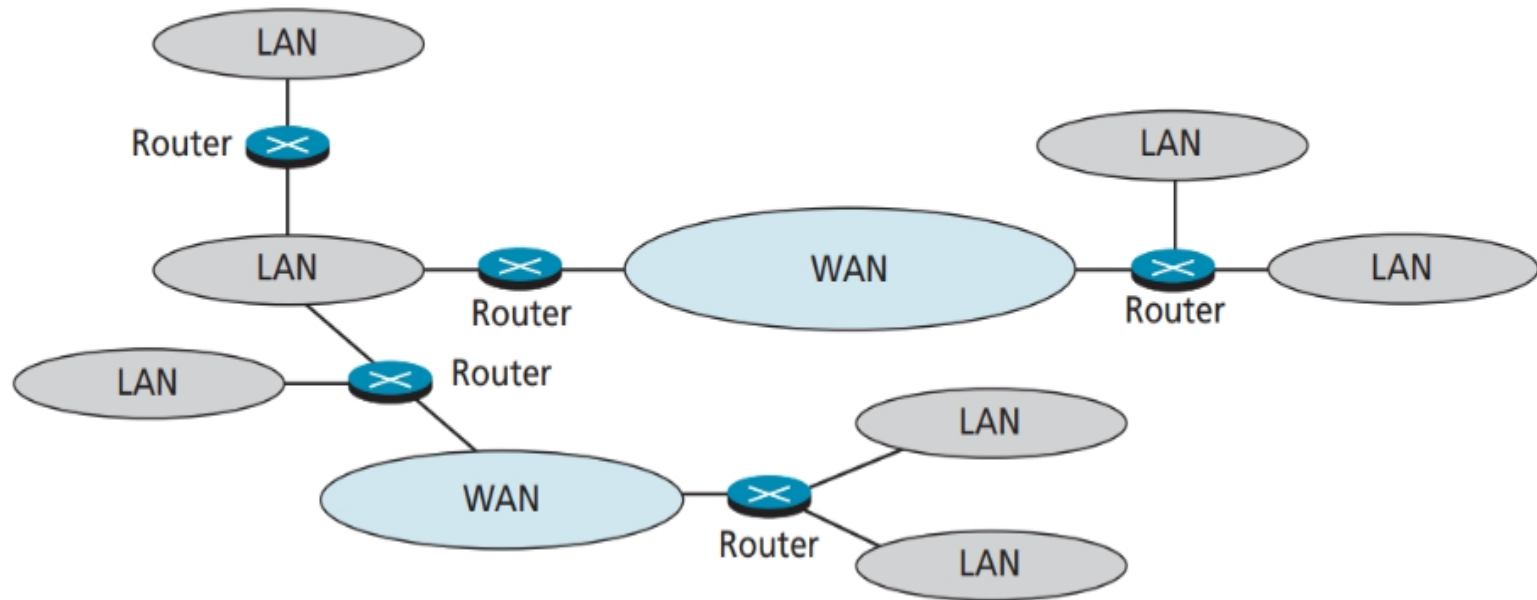
A metropolitan area network (MAN) is a network with a size between a LAN and a WAN. It normally covers the area inside a town or a city. It is designed for customers who need high-speed connectivity, normally to the internet and has endpoints spread over a city or part of city. A good example of a MAN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

An internet

Today, it is very rare to see a network in isolation as networks are connected to one another. When two or more networks are connected, they become an internetwork, or an internet (lowercase 'i').

An example of an internet. In other words, a network is a group of connected communicating devices such as computers and printers while an internet is two or more networks that can communicate with each other.

Figure 6.5 *An internet made of WANs, LANs, and routers*



Seven LANs and two WANs.

6.1.2 The Internet

An internet is two or more networks that can communicate with each other. The most notable is **the Internet** which is made of thousands of networks.

It is difficult to give an accurate representation of the internet, because it is continually changing—new networks are being added and networks of defunct organizations removed. Today most end users who want an internet connection use the services of internet service providers (ISPs).

An ISP is an organization with one or more servers (high-capacity computers) that are connected to the internet through high-speed links. Individual internet users or small companies can be connected to the servers of a local ISP by establishing a service contract and paying a fee.

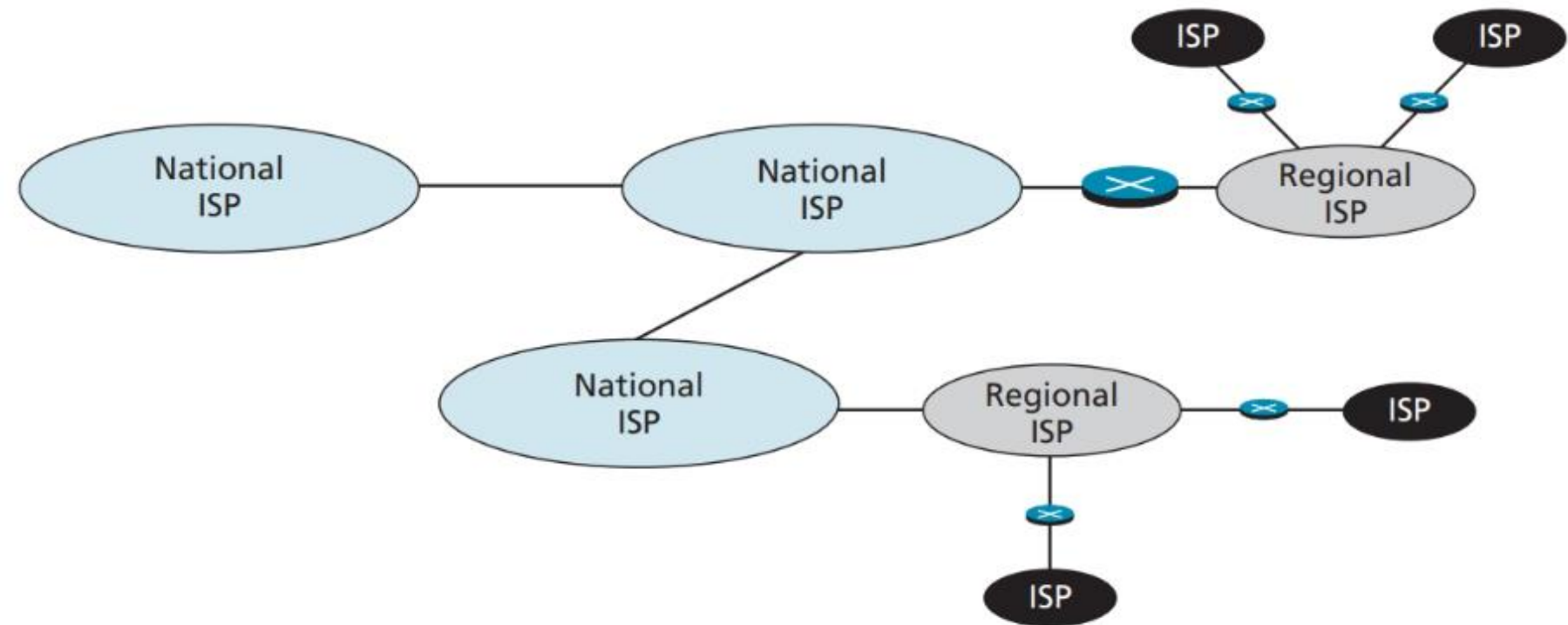
10. 18

There are:

- international service providers,
- national service providers,
- regional service providers and
- local service providers.

The internet today is run by private companies,
not the government.

Figure 6.6 Hierarchical organization of the internet



6.1.3 Hardware and software

For communication to happen both hardware and software are required.

This is similar to a complex computation in which both a computer and a program are needed.

These combinations are coordinated with each other using **protocol layering**.

6.2 TCP/IP PROTOCOL SUITE

Compare this to a service provided by a computer network. For example, the task of sending an email from one place in the world to another can be broken down into several tasks, each performed by a layer. Each layer uses the services of a lower layer. At the lowest layer, a signal or a set of signals, is sent from the source computer to the destination computer.

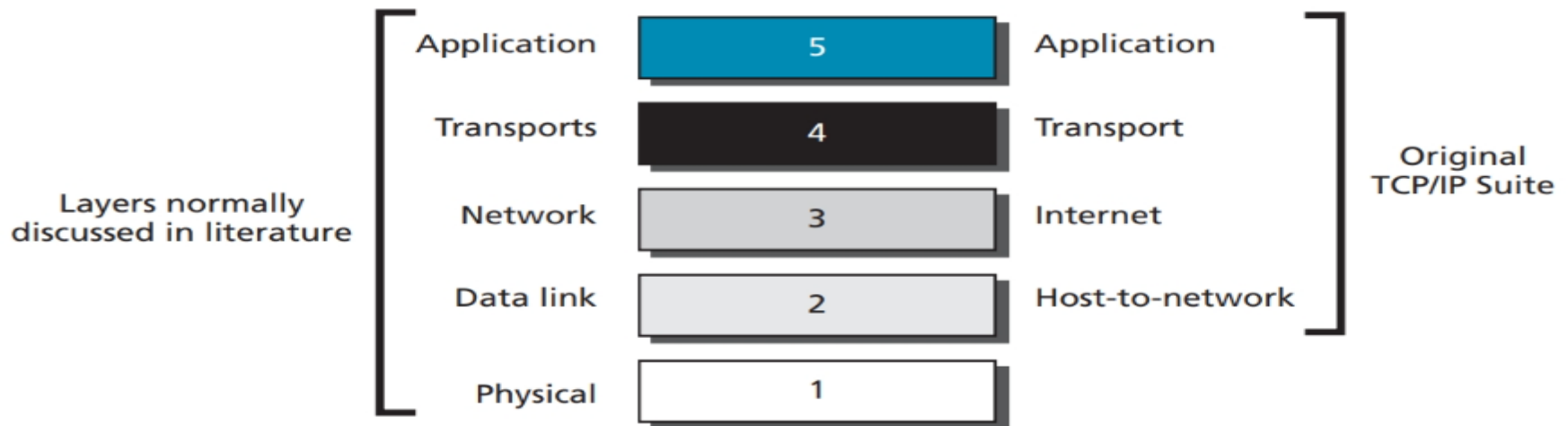
6.2 TCP/IP PROTOCOL SUITE

To divide the services required to perform a task, the internet has created a set of rules called protocols. These allow different local and wide area networks, using different technologies, to be connected together and carry a message from one point to another. The set, or suite, of protocols that controls the internet today is referred to as the TCP/IP protocol suite. The abbreviations (TCP and IP) will become clear as we explain different protocols.

6.2 TCP/IP PROTOCOL SUITE

The original TCP/IP protocol suite was defined as having four layers: host-to-network (or link), internet (network), transport and application. However, the TCP/IP protocol suite today is normally considered as a five-layer model.

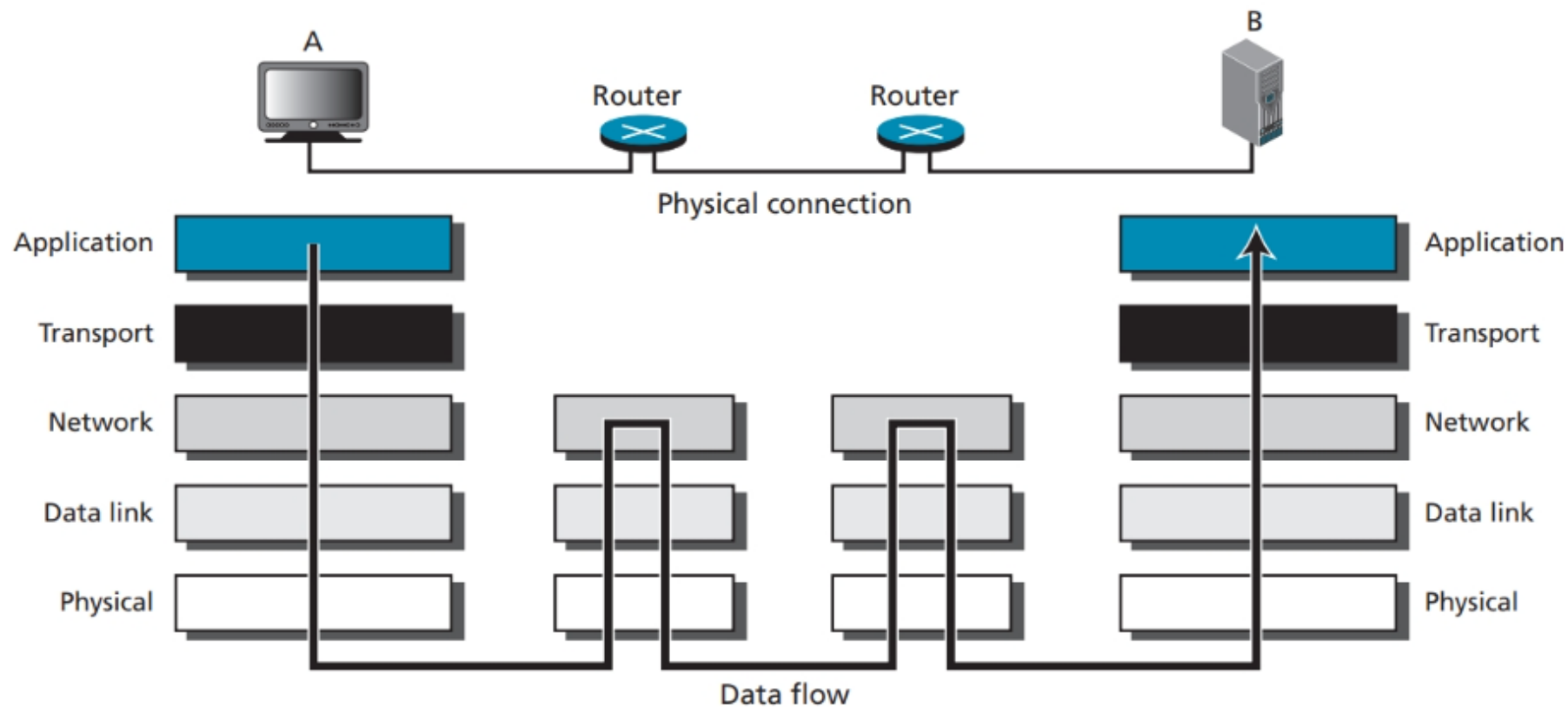
Figure 6.7 The TCP/IP protocol suite



6.2 TCP/IP PROTOCOL SUITE

Figure 6.8 shows the layers involved when a message is sent from device A to device B. As the message travels from A to B, it may pass through many routers. Routers use only the first three layers. Within a single machine, each layer calls on the services of the layer immediately below it. Layer 3, for example, uses the services provided by layer 2 and provides services for layer 4.

Figure 6.8 *The interaction between layers in the TCP/IP protocol suite*



6.3 LAYERS

This section briefly describes the function of each layer in the TCP/IP protocol suite. We show how a message travels through the different layers until it reaches the physical layer and is sent by the transmission media.

6.3 LAYERS

Modularity in this case means Independent layers. A layer (**module**) can be defined as a black box with inputs and outputs, without concern about how inputs are changed to outputs.

Note: If two machines provide the same outputs when given the same inputs, they can replace each other.

Advantages:

- Separation of the services from the implementation. A layer needs to be able to receive a set of services from the lower layer and to give the services to the upper layer.
- Communication does not always use only two end systems; there are intermediate systems that need only some layers, but not all layers. If we did not use protocol layering, we would have to make each intermediate system as complex as the end systems, which makes the whole system more expensive.

6.3.1 Application layer

The application layer enables a user, whether human or software, to access the network. It provides support for services such as electronic mail, remote file access and transfer, browsing the world wide web, and so on. The application layer is the only layer seen by most users of the internet.

The application layer is responsible for providing services to the user.

Layer 7: Application layer

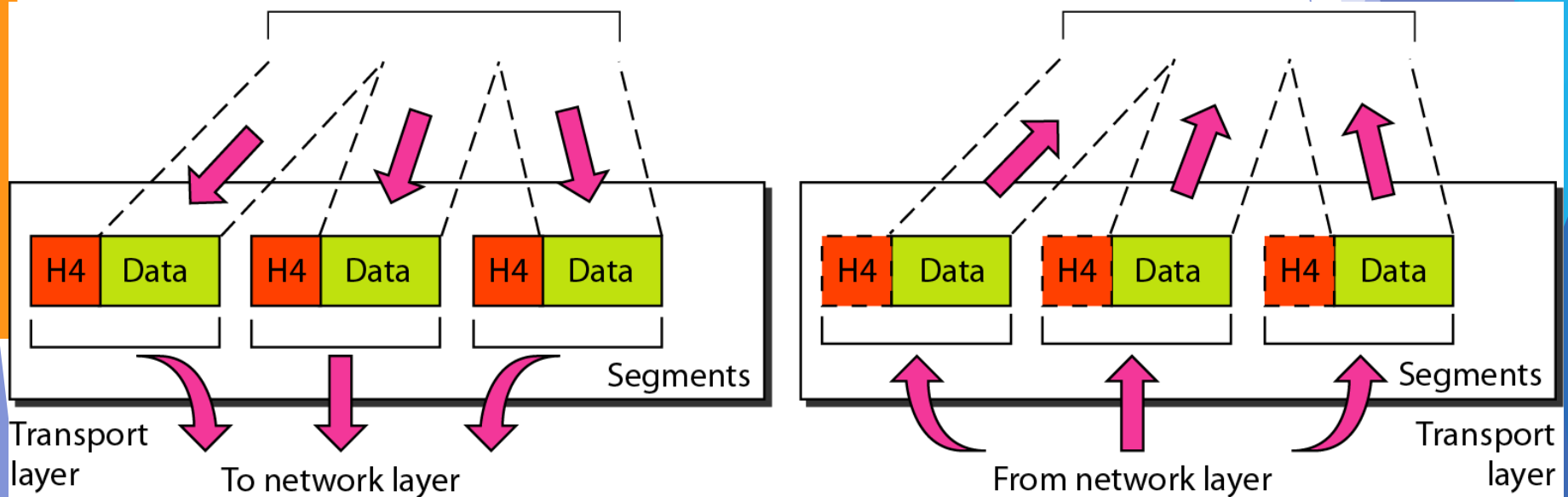
The application layer is responsible for providing services to the user.

Services including:-

- **Network Virtual Terminal (TELNET)**
- **File transfer, access and management (FTP)**
- **Mail services (SMTP)**
- **Directory services (HTTP)**

Layer 4: Transport layer

- Responsible for process-to-process delivery of the entire message.
- A process is an application program running on a host.
- It treats each packet as a separate message.



Layer 4: Transport layer

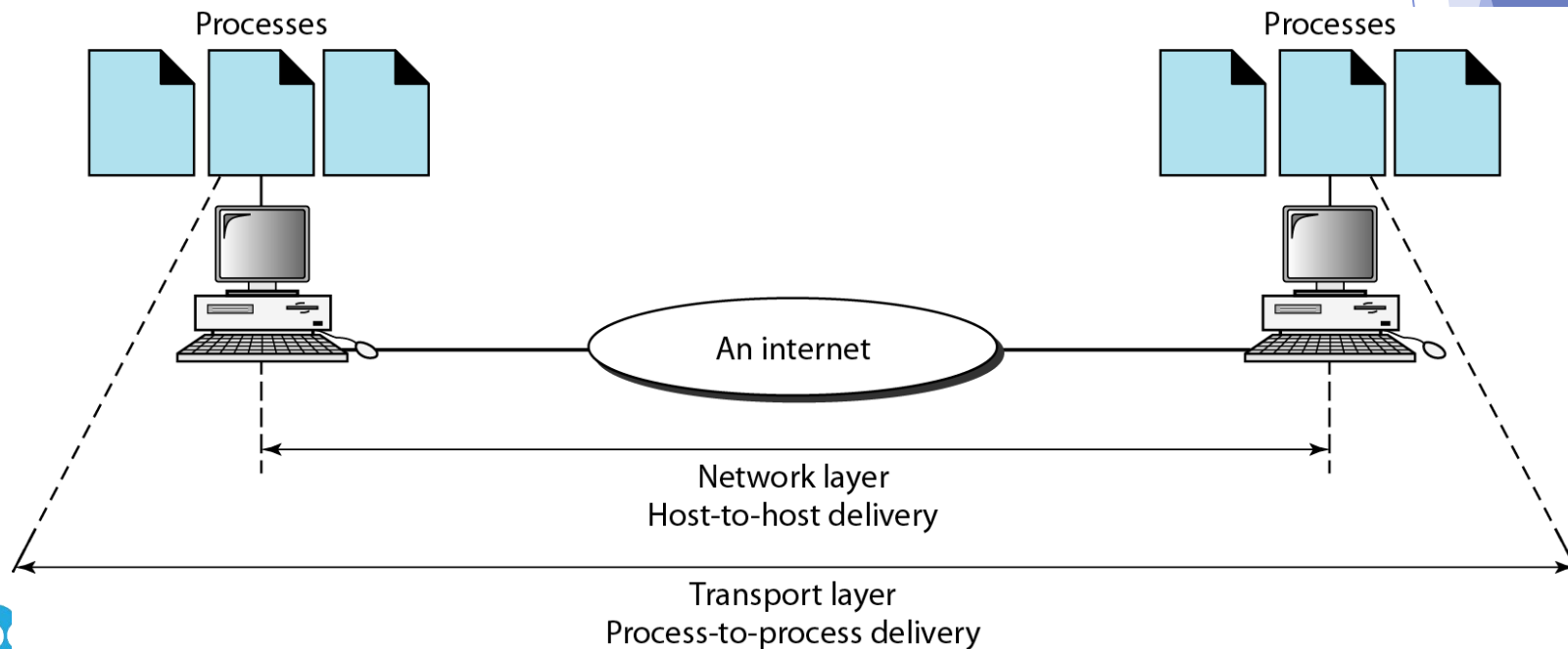
The transport layer is responsible for the delivery of the “whole” message from one device to another.

It also responsible for:

- **service point addressing** : deliver the entire message to the right process using Port number as a header address
- **segmentation and reassembly** : each segment contains a sequence number to help in the reassembly.
- **connection control** : connection/connectionless (TCP/UDP)
- **flow control**
- **Error control**

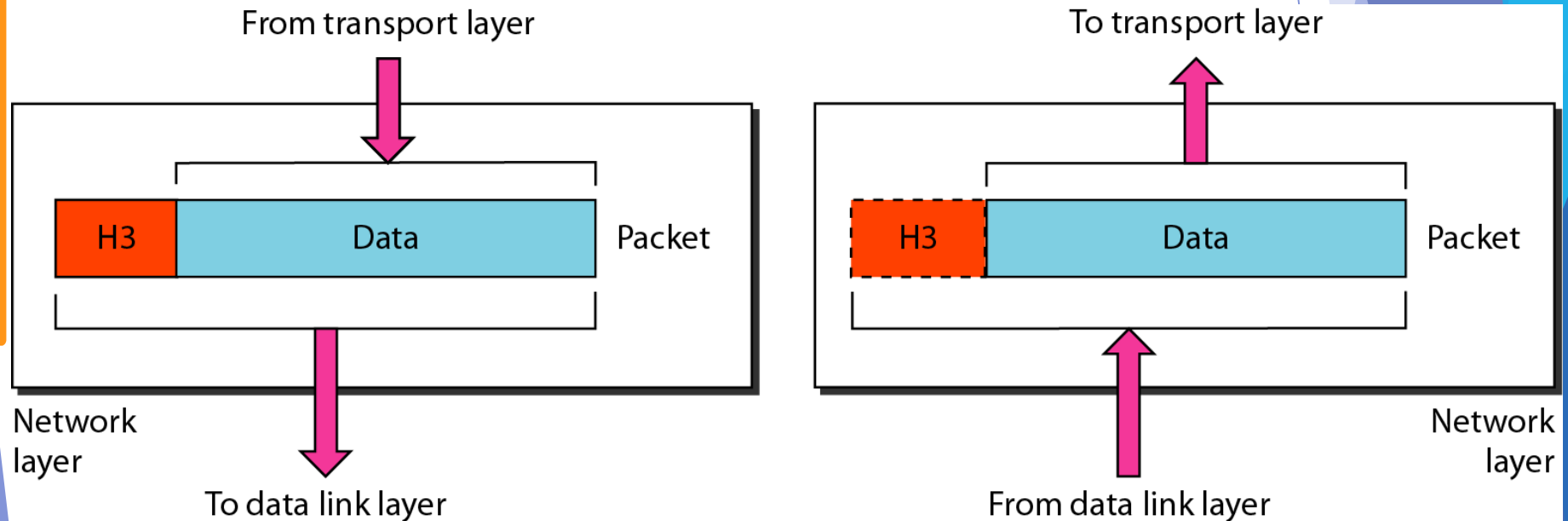
Layer 4: Transport layer

Reliable process-to-process delivery of a message



Layer 3: Network layer

- Responsible for the source-to-destination delivery of a packet, across multiple networks.
- If two systems are connected to the same link (network), there is no need for a network layer.



10.34

Layer 3: Network layer

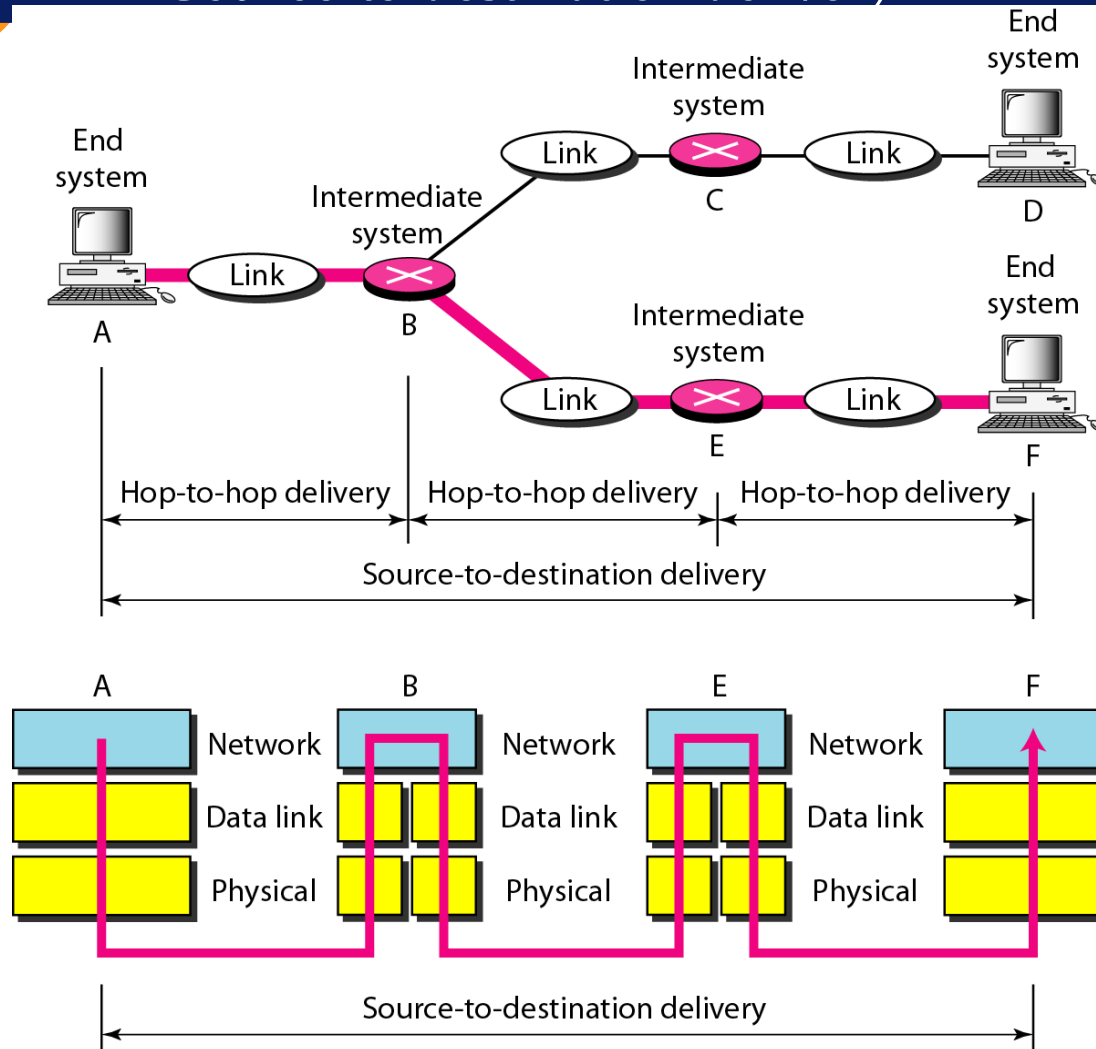
The network layer is responsible for the delivery of individual packets from the source host to the destination host.

It also responsible for:

- **Logical addressing (IP)** for destination outside the network
- **Routing** : making decision based on the destination

Layer 3: Network layer

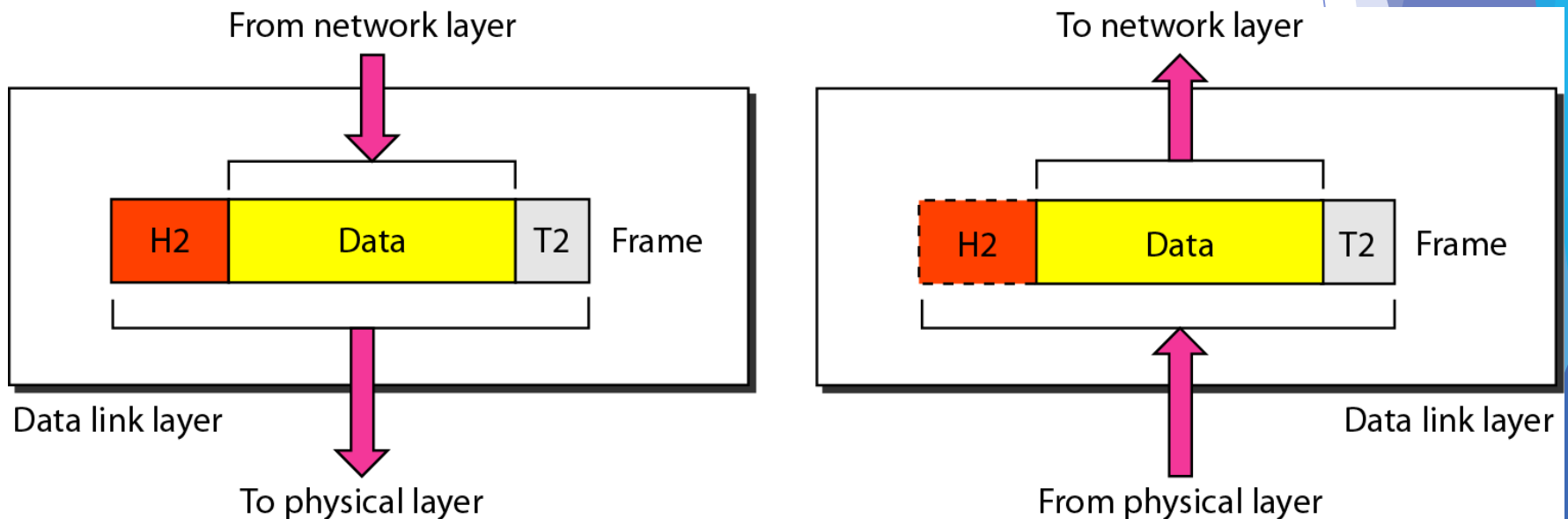
Source-to-destination delivery



10. 36

Layer 2: Data link layer

- Transforms the physical layer to a reliable link.
- It make the physical layer appear error-free to the upper layer.



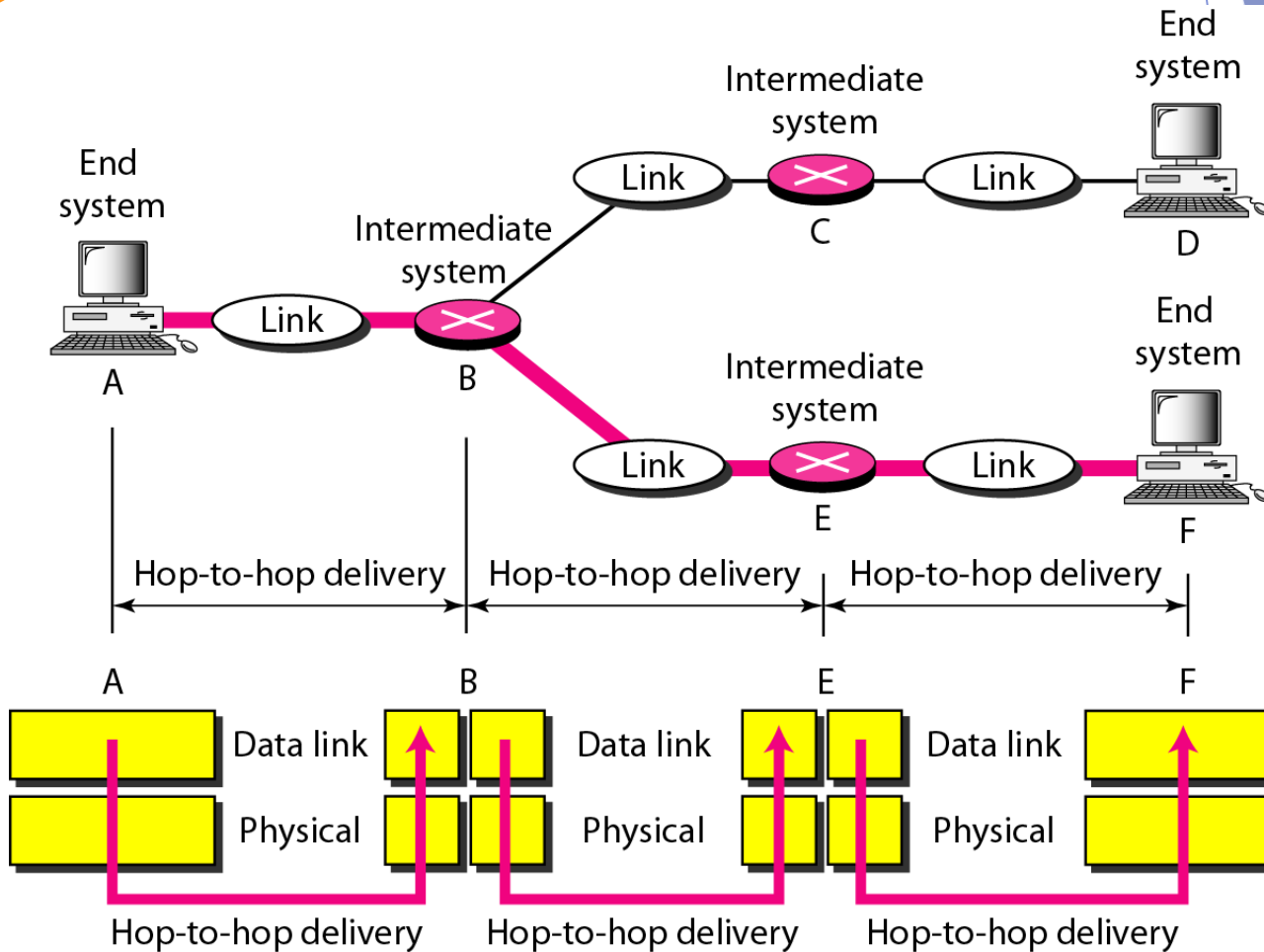
Layer 2: Data link layer

The data link layer is responsible for moving frames from one hop (node) to the next.

It also responsible for

- **Framing**
- **Physical addressing** for devices on the same network (MAC)
- **Flow control** to avoid receiver overwhelming of data rate
- **Error Control- trailer** detect/retransmit lost/damaged frames
- **Access Control** to determine which device has control over the link

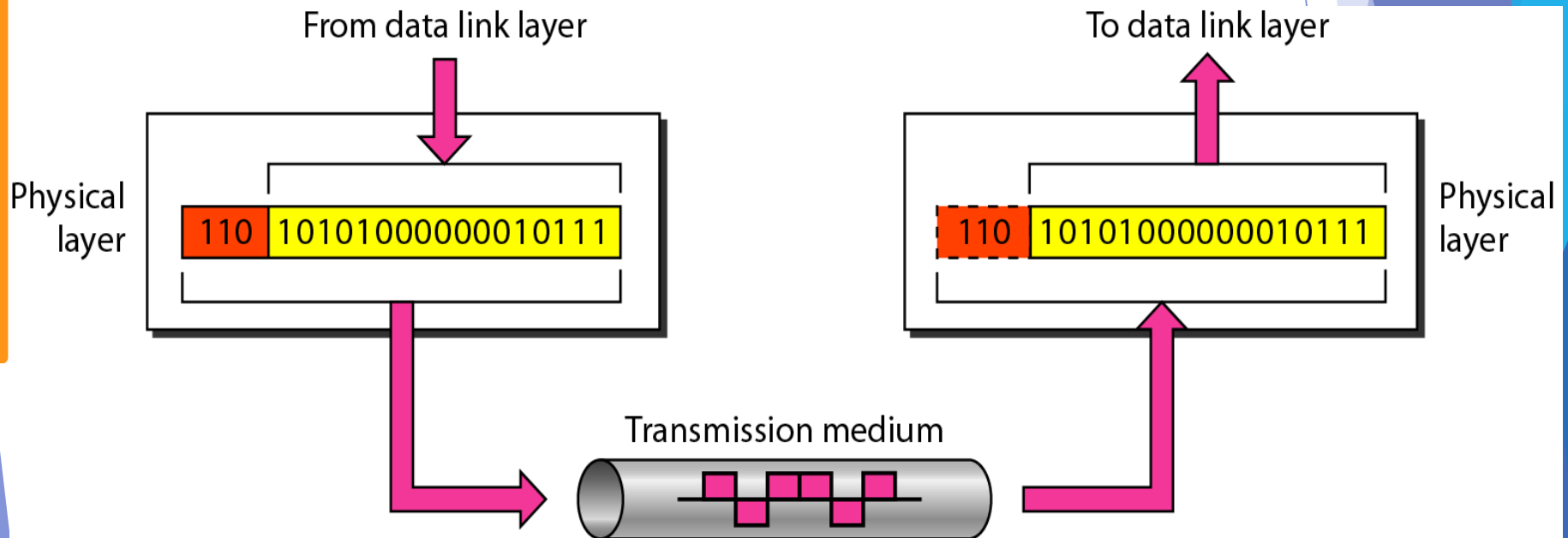
Layer 2: Data link layer Hop-to-hop delivery



0.39

Layer 1: Physical layer

- Coordinate the functions required to carry a bit stream over a physical medium.
- It also defines the procedures and functions that physical devices and interfaces have to perform for transmission to occur.



10.40

Layer 1: Physical layer

The physical layer is responsible for movements of individual bits from one hop (node) to the next.

It also responsible for:

- **Physical characteristics of interfaces and medium**
- **Representation of bits** (how 0,1 changes to signals)
- **Data rate** (num of bits sent each second)
- **Synchronization of bits** (using the same bit rate, same clock)
- **Line configuration** (type of connection: P2P, multipoint)
- **Physical Topology** (Bus/Star/Mesh/Ring)
- **Transmission mode** (simplex/Duplex)

CHAPTER 10

Software Engineering



10. 42

Objectives

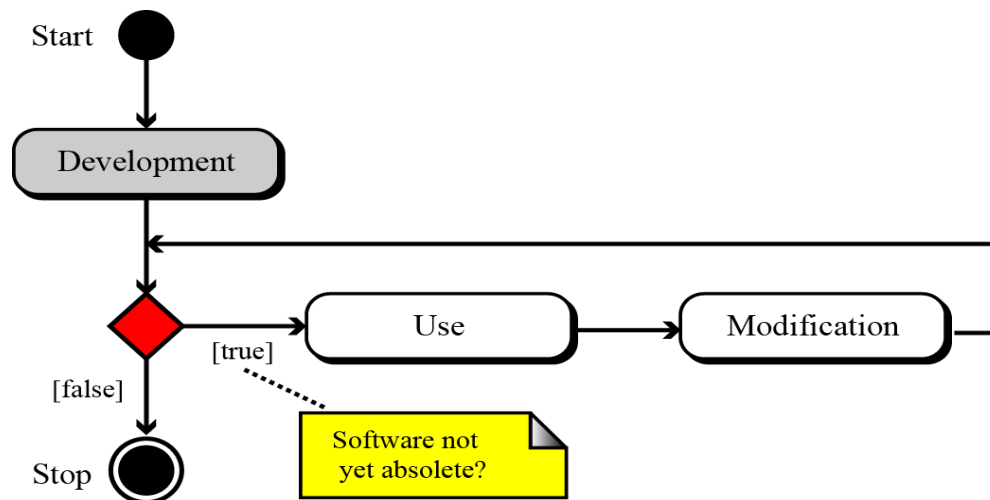
After studying this chapter, the student should be able to:

- ☐ Understand the concept of the software life cycle in software engineering.
- ☐ Describe two major types of development process, the waterfall and incremental models.
- ☐ Understand the analysis phase and describe two separate approaches in the analysis phase: procedure-oriented analysis and object-oriented analysis.
- ☐ Understand the design phase and describe two separate approaches in the design phase: procedure-oriented design and object-oriented design.

10-1 THE SOFTWARE LIFECYCLE

A fundamental concept in **software engineering** is the ***software lifecycle***. Software, like many other products, goes through a cycle of repeating phases (Figure 10.1).

Figure 10.1 The software lifecycle



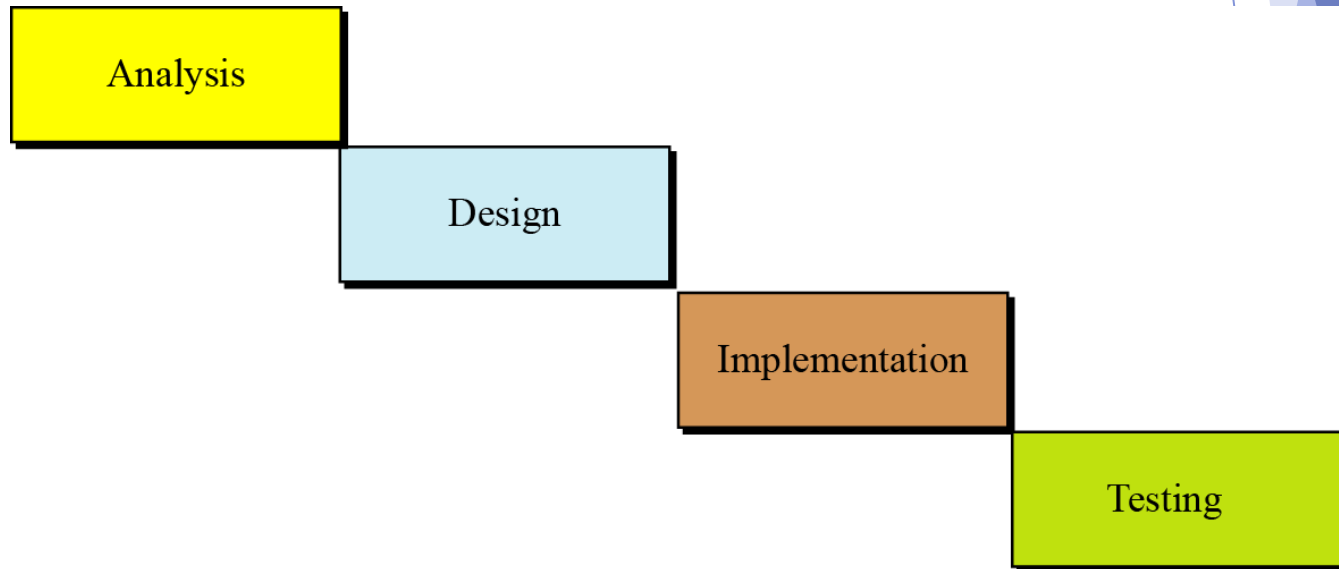
10.1.1 Development process models

Although software engineering involves all three processes in Figure 10.1, in this chapter we discuss only the development process, which is shown outside the cycle in Figure 10.1. The development process in the software lifecycle involves four phases: **analysis**, **design**, **implementation**, and **testing**. There are several models for the development process. We discuss the two most common here: the **waterfall model** and the **incremental model**.

The waterfall model

In this model, the development process flows in only one direction. This means that a phase cannot be started until the previous phase is completed.

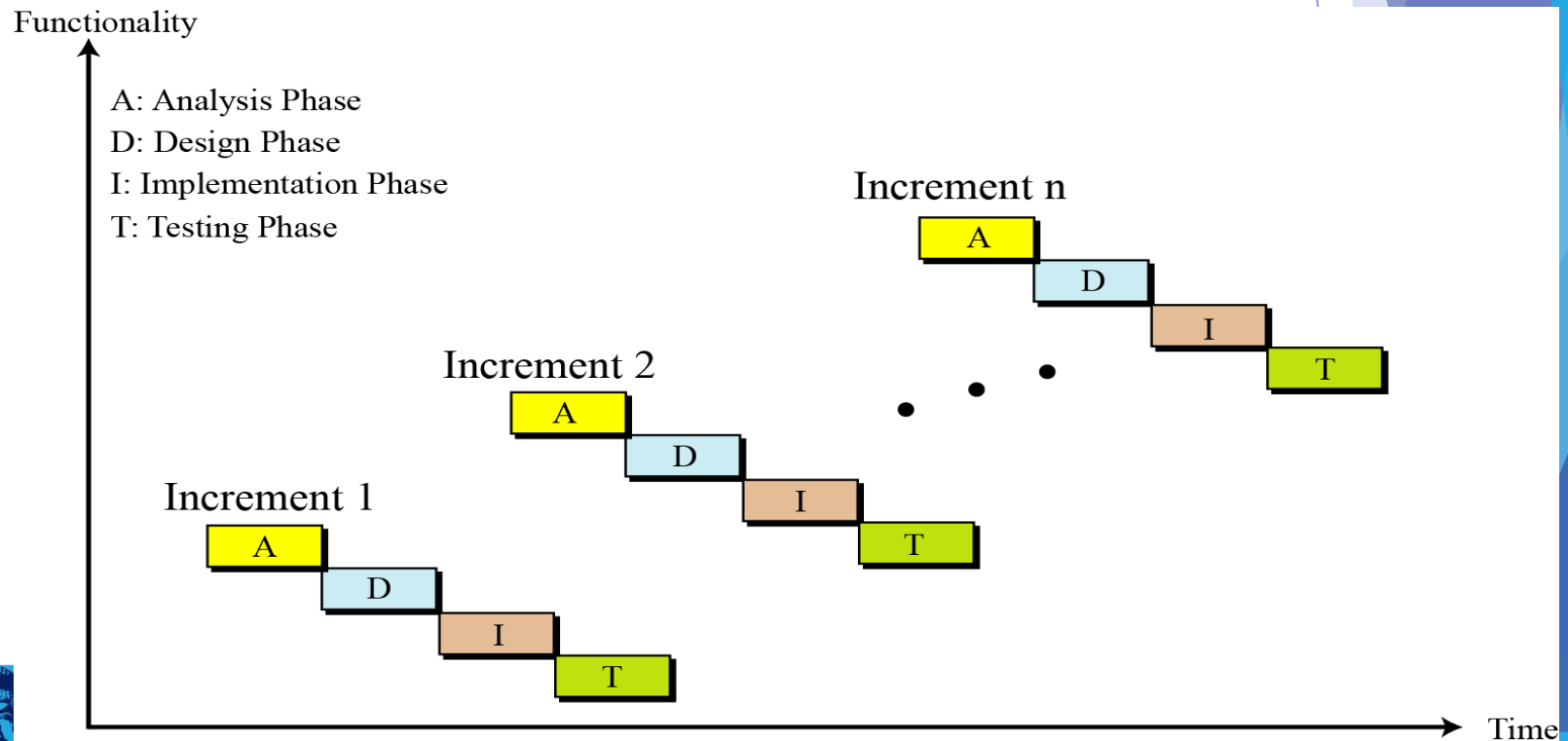
Figure 10.2 The waterfall model



The incremental model

In the incremental model, software is developed in a series of steps.

Figure 10.3 The incremental model



10-2 ANALYSIS PHASE

The development process starts with the **analysis phase**. This phase results in a specification document that shows **what the software will do without specifying how it will be done**. The analysis phase can use two separate approaches, depending on whether the implementation phase is done using a procedural programming language or an object-oriented language. We briefly discuss both in this section.

10.2.1 Procedure-oriented analysis

Procedure-oriented analysis—also called structured analysis or classical analysis—is the analysis process used if the system implementation phase will use a procedural language. The specification in this case may use several modeling tools.

10.2.2 Object-oriented analysis

Object-oriented analysis is the analysis process used if the implementation uses an object-oriented language. The specification document in this case may use several tools.

10-3 DESIGN PHASE

The design phase defines how the system will accomplish what was defined in the analysis phase. In the design phase, all components of the system are defined.

10.3.1 Procedure-oriented design

In **procedure-oriented design** we have both procedures and data to design. We discuss a category of design methods that concentrate on procedures. In procedure-oriented design, the whole system is divided into a set of procedures or modules.

Modularity

Modularity means breaking a large project into smaller parts that can be understood and handled easily. In other words, modularity means dividing a large task into small tasks that can communicate with each other.

10.3.2 Object-oriented design

a class is made of a set of variables (attributes) and a set of methods. The object-oriented design phase lists details of these attributes and methods.

10-4 IMPLEMENTATION PHASE

In the waterfall model, after the design phase is completed, the implementation phase can start. In this phase the programmers write the code for the modules in procedure-oriented design, or write the program units to implement classes in object-oriented design. There are several issues we need to mention in each case.

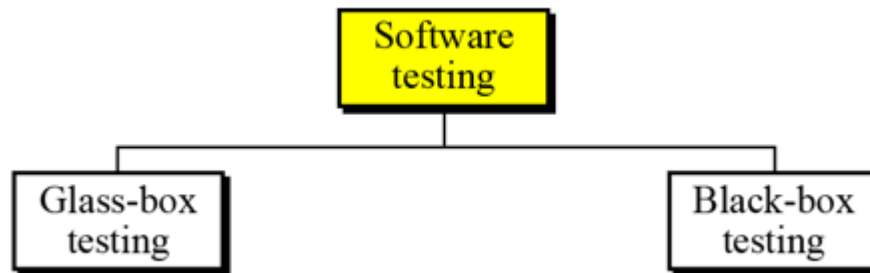
Choice of language

In a procedure-oriented development, the project team needs to choose a language or a set of languages from among the procedural languages discussed in Chapter 10. Although some languages like C++ are considered to be both a procedural and an object-oriented language—normally an implementation uses a purely procedural language such as C. In the object-oriented case, both C++ and Java are common.

10-5 TESTING PHASE

The goal of the testing phase is to find errors, which means that a good testing strategy is the one that finds most errors. There are two types of testing: **glass-box** and **black-box** (Figure 10.11).

Figure 10.11 Software testing



10.5.1 Glass-box testing

Glass-box testing (or **white-box testing**) is based on knowing the internal structure of the software. The testing goal is to check to determine whether all components of the software do what they are designed for. Glass-box testing assumes that the tester knows everything about the software. In this case, the software is like a glass box in which everything inside the box is visible. Glass-box testing is done by the software engineer or a dedicated team.

10.5.2 Black-box testing

Black box testing gets its name from the concept of testing software without knowing what is inside it and without knowing how it works. In other words, the software is like a black box into which the tester cannot see. Black-box testing tests the functionality of the software in terms of what the software is supposed to accomplish, such as its inputs and outputs. Several methods are used in black-box testing, discussed below.

10-6 DOCUMENTATION

For software to be used properly and maintained efficiently, documentation is needed. Usually, three separate sets of documentation are prepared for software:

- **User documentation:** user guide
- **System documentation:** defines the software It self
- **Technical documentation:** describe the installation and Service documentation, which defines how the system should be maintained and updated if necessary

Thank You