Computer Networks Introduction

جامعة الانبار

قسم الهندسة الكهربائية

مادة شبكات الحاسوب – 2021

م.م. براء سعيد العبيدي

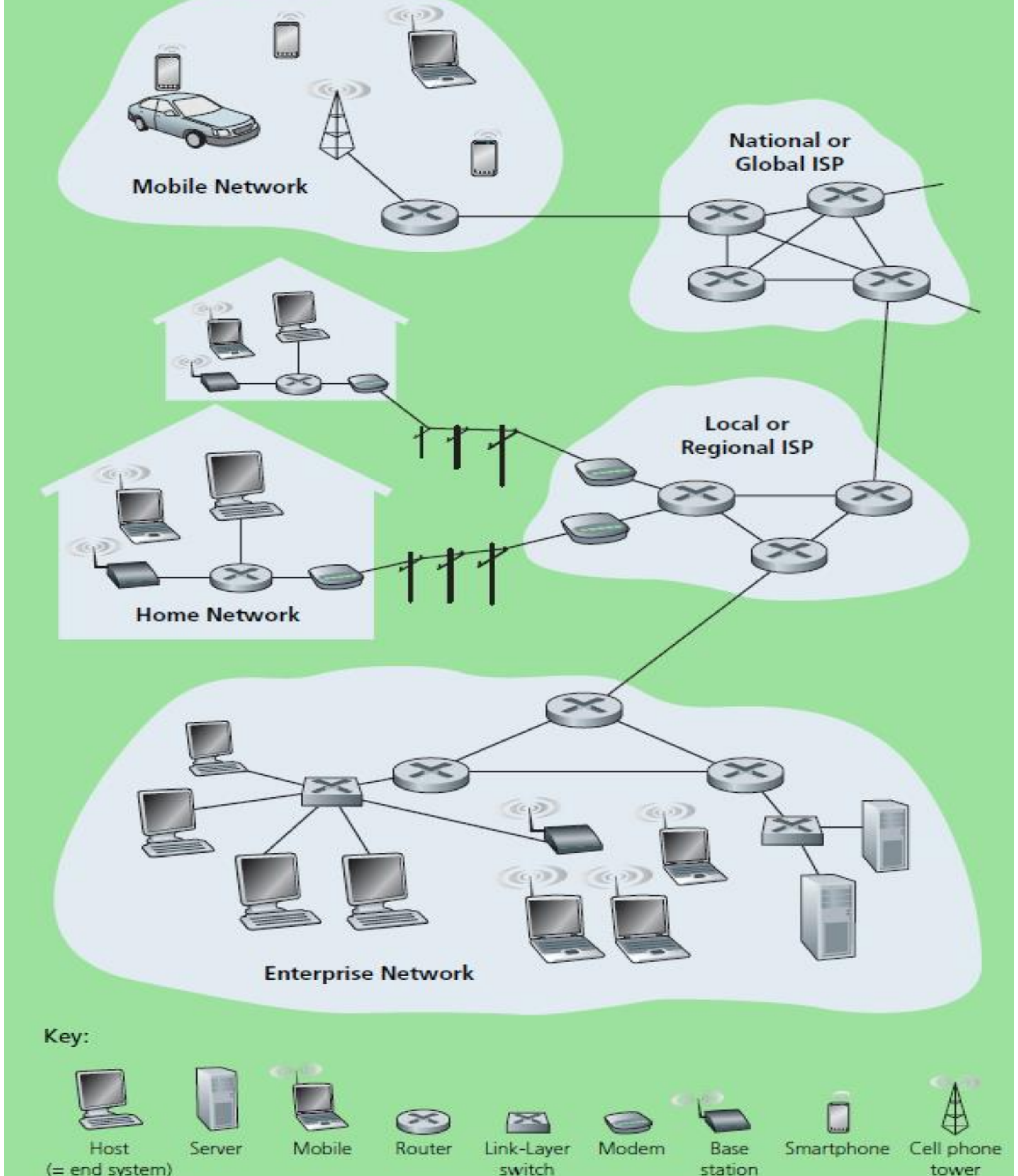
What is a Computer Network?

- A computer network is a group of computers that use a set of common communication protocols over digital interconnections for the purpose of sharing resources located on or provided by the network nodes. (Wikipedia)
- There are other devices that are attached to the network, other than computers; such as smart phones, tablets, TVs, gaming consoles, Linux workstations, servers, printers, sensors, security systems, home appliance, etc.
- These devices are called hosts or end systems.
- The hosts or end systems are connected through communication links, routers and switches.
- When one end system has data to send to another end system, the sending end system divides the data and adds header bytes to each chunk. The resulting packages of information, known as packets, are then sent through the network to the destination end system, where they are reassembled into the original data.

What Is the Internet?

- The Internet is a computer network that interconnects hundreds of millions of computing devices throughout the world.
- Internet Service Providers (ISPs): an entity (e.g., a company or an organization) that provides access to the Internet for end users.
- This includes residential ISPs such as local cable or telephone companies; corporate ISPs; university ISPs; and ISPs that provide WiFi access in airports, hotels, coffee shops, and other public places.
- Each ISP is in itself a network of packet switches and communication links.
- Each ISP network is managed independently, runs the IP protocol and conforms to certain naming and address conventions.

- Some pieces of the Internet.



Network Protocols

- A **protocol** defines the format and the order of messages exchanged between two or more communicating entities, as well as the actions taken on the transmission and/or receipt of a message or other event.

شبكات الحاسبات

قسم الهندسة الكهربائية
المرحلة الثالثة
م.م. بـراء سعيـد العبيدي

Application Layer

- Network applications are the reason for existence of a computer network—if we couldn’t conceive of any useful applications, there wouldn’t be any need for networking infrastructure and protocols to support them.
- Internet applications include:
  - World Wide Web, encompassing Web surfing, search, and electronic commerce, instant messaging and P2P file sharing,
  - Voice over IP and video conferencing such as Skype, Facetime, and Google Hangouts.
  - User generated video such as YouTube and movies on demand such as Netflix.
  - Multiplayer online games such as Second Life and World of Warcraft, PubG.
  - Social networking applications—such as Facebook, Instagram, Twitter, and WeChat.
  - Mobile apps

Network Application Architectures

I. Client-server architecture:

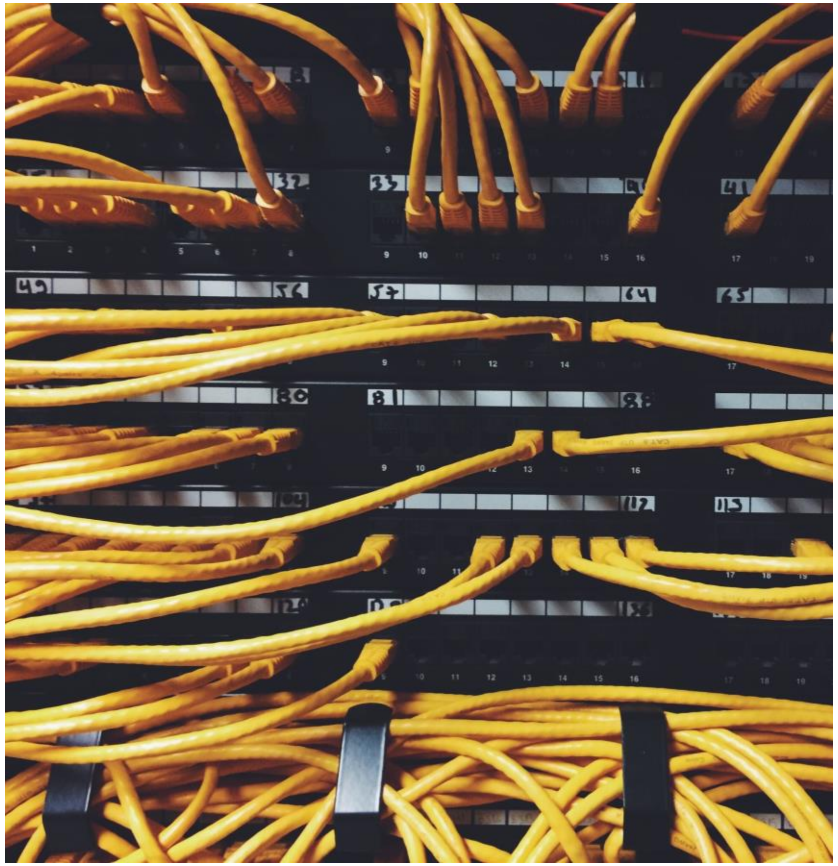
- there is an always-on host, called the server, which services requests from many other hosts, called clients.
- a single-server host is incapable of keeping up with all the requests from clients. For example, a popular social-networking site can quickly become overwhelmed if it has only one server handling all of its requests. For this reason, a data center, housing a large number of hosts, is often used to create a powerful virtual server.

II. Peer-to-Peer (P2P) architecture:

- there is minimal (or no) reliance on dedicated servers in data centers. Instead, the application exploits direct communication between pairs of intermittently connected hosts, called peers.

The Interface Between the Process and the Computer Network

- A network application consists of pairs of processes that send messages to each other over a network.
- Any message sent from one process to another must go through the underlying network.
- A process sends messages into, and receives messages from, the network through a software interface called a **socket**.



Application Layer Protocols

Application	Application-Layer Protocols
Electronic mail	SMTP (Simple Mail Transfer Protocol)
Web	HTTP (Hyper Text Transfer Protocol)
File transfer	FTP (File Transfer Protocol)
Streaming multimedia	HTTP
Internet telephony	Skype

HTTP: Hyper Text Transfer Protocol

The Web's application-layer protocol, is at the heart of the Web.

HTTP is implemented in two programs: a client program and a server program.

The client program and server program, executing on different end systems, talk to each other by exchanging HTTP messages.

HTTP is a **stateless protocol.**

Web (http) uses the client-server application architecture

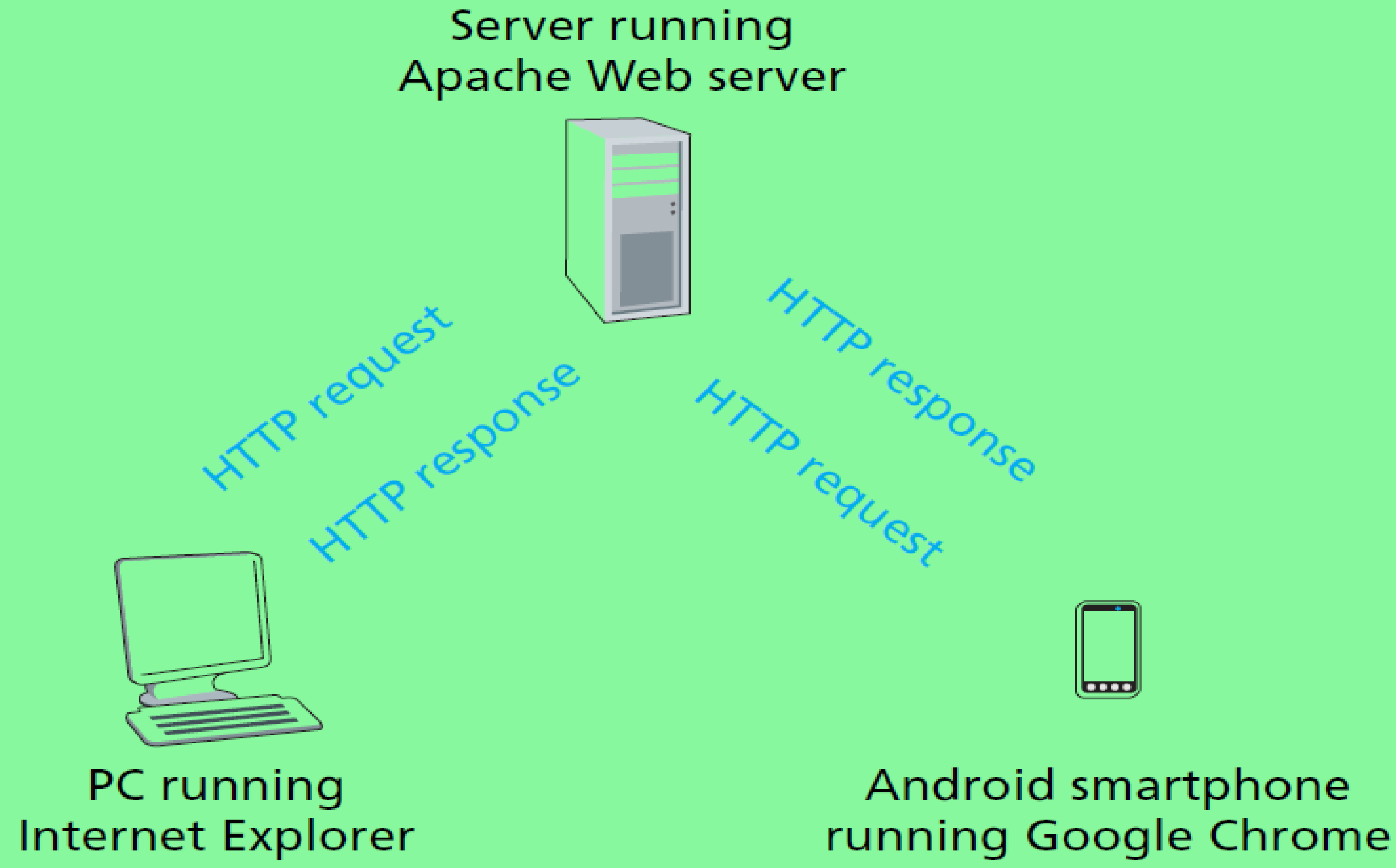


Figure 2.6 ♦ HTTP request-response behavior

Non-Persistent and Persistent Connections

A Web page consists of objects. An object is simply a file—such as an HTML file, a JPEG image, a Java applet, or a video clip—that is addressable by a single URL.

If a Web page contains HTML text and five JPEG images, then the Web page has six objects: the base HTML file plus the five images.

Non-persistent connections: each request/response pair be sent over a *separate* connection.

persistent connections: all the requests and their corresponding responses be sent over the same connection.

An Example of non-persistent connections

Suppose the page consists of a one base HTML file and 10 JPEG images, and that all 11 of these objects reside on the same server (total 11 objects). The URL (Uniform Resource Locator) is:

`https://www.uoanbar.edu.iq/EngineeringCollege.index`

1. The HTTP client process initiates a connection to the server `www.somuoanbar.edu` on port number 80, which is the default port number for HTTP. Associated with the connection, there will be a socket at the client and a socket at the server.
2. The HTTP client sends an HTTP request message to the server via its socket that contains the path /EngineeringCollege.
3. The HTTP server process receives the request message via its socket, retrieves the object from its storage (RAM or hard disk), encapsulates the object in an HTTP response message, and sends the response message to the client via its socket.
4. The HTTP server process closes the TCP connection.
5. The HTTP client receives the response message.
6. The first four steps are then repeated for each of the referenced JPEG objects.

Round-Trip Time (RTT)

- RTT is the time it takes for a small packet to travel from client to server and then back to the client.
- Example:

A user clicks on a URL (link) requesting an HTML file.

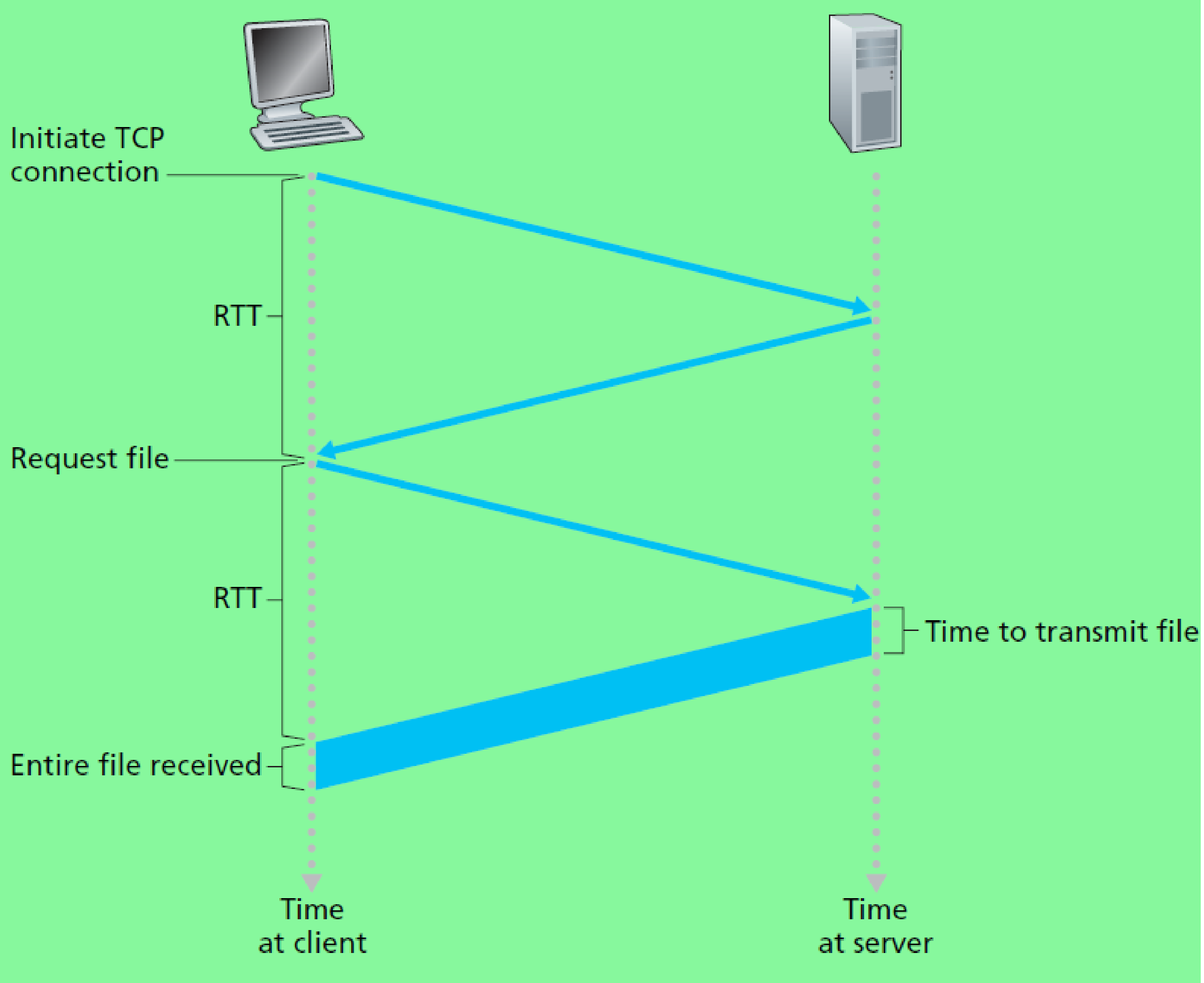


Figure 2.7 ♦ Back-of-the-envelope calculation for the time needed to request and receive an HTML file

HTTP Message Format

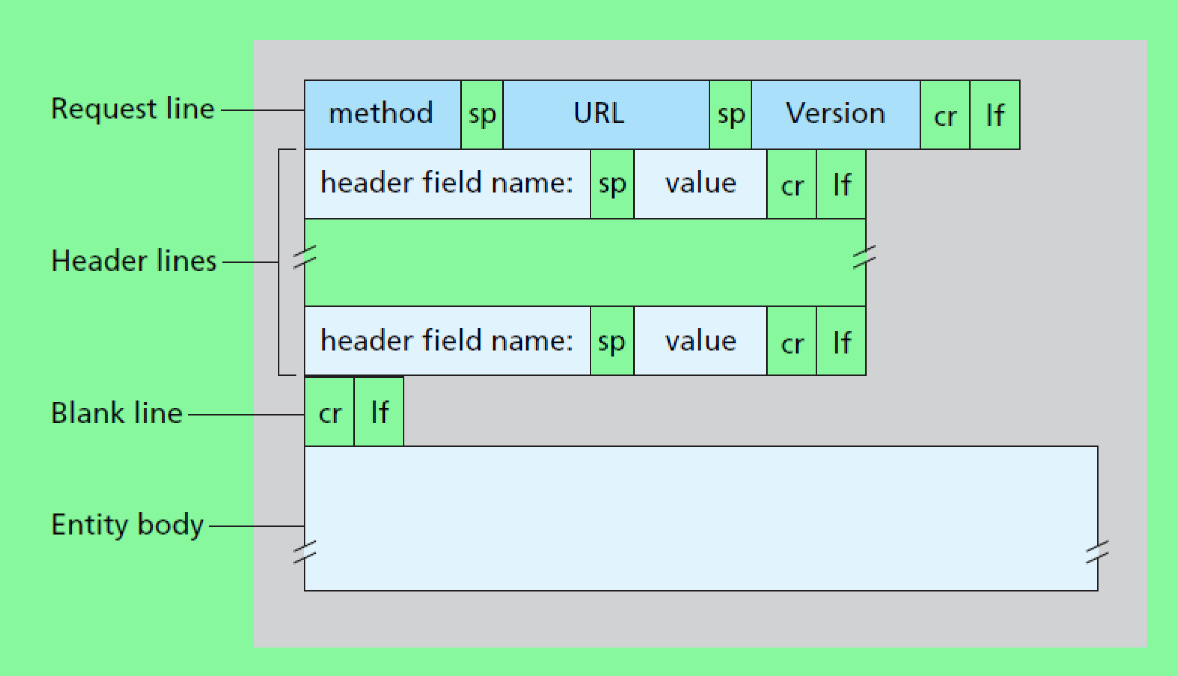


Figure 2.8 ♦ General format of an HTTP request message

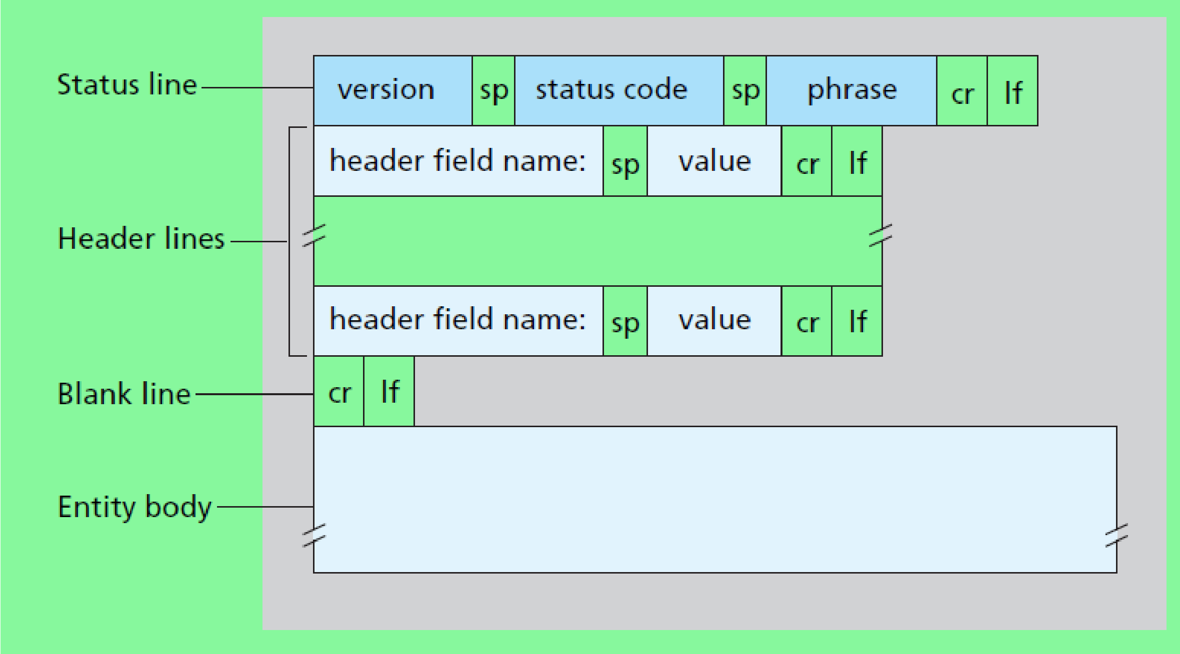
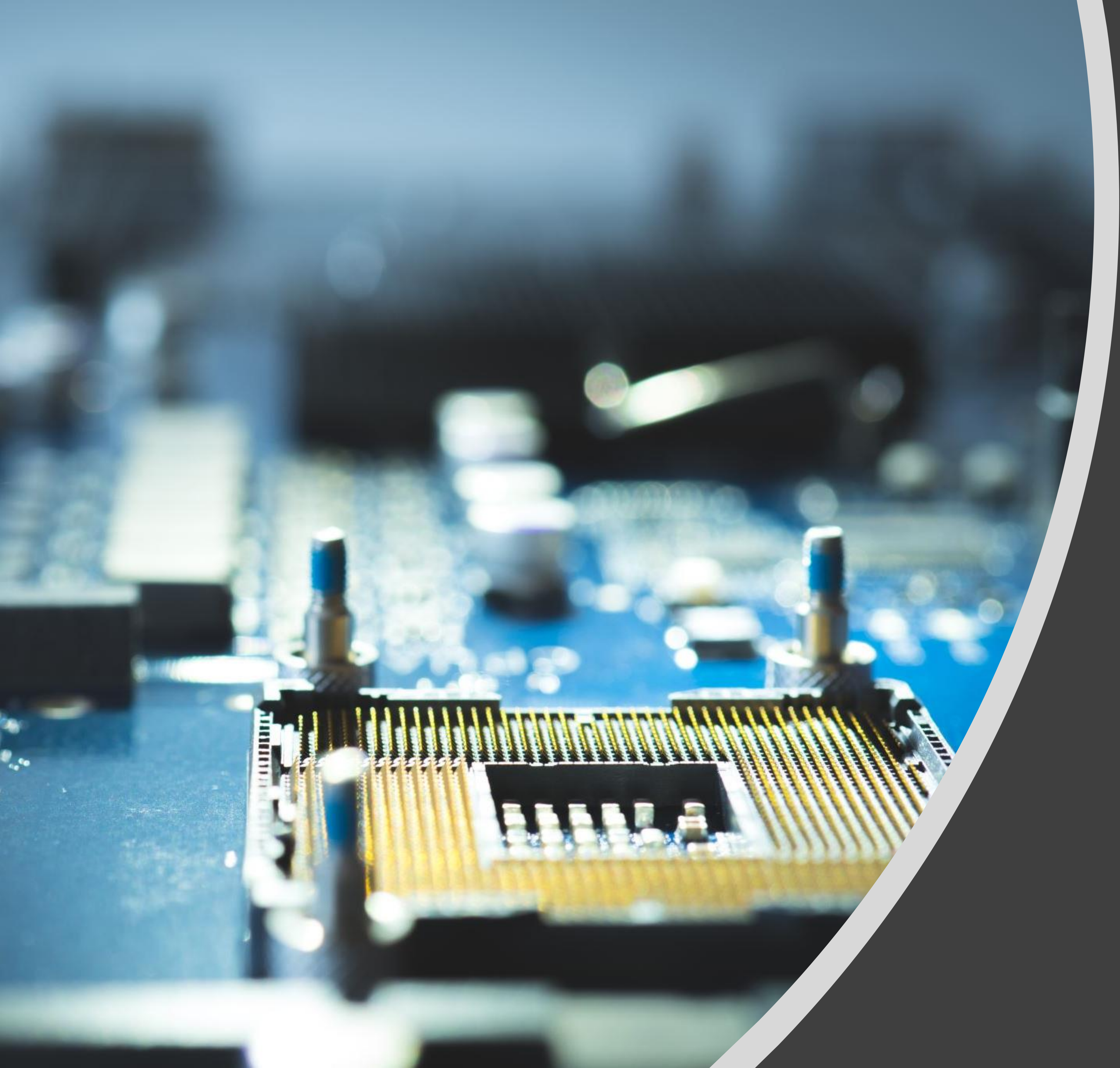


Figure 2.9 ♦ General format of an HTTP response message



Computer Networks

Transport Layer

جامعة الانبار

قسم الهندسة الكهربائية

مادة شبكات الحاسوب – 2021

م.م. براء سعيد العبيدي

An overview of the Transport Layer

- A transport-layer protocol provides for **logical communication** between application processes running on different hosts. L*ogical communication* means that from an application's perspective, it is as if the hosts running the processes were directly connected; in reality, the hosts may be on opposite sides of the planet, connected via numerous routers and a wide range of link types.
- Data units are called **segments**.
- On the sending side, the transport layer converts the application-layer messages it receives from a sending application process into transport-layer segments.
- This is done by breaking the application messages into smaller chunks and adding a transport-layer header to each chunk to create the transport-layer segment.
- The transport layer then passes the segment to the network layer at the sending end system, where the segment is encapsulated within a network-layer packet (a datagram) and sent to the destination.
- On the receiving side, the network layer extracts the transport-layer segment from the datagram and passes the segment up to the transport layer. The transport layer then processes the received segment, making the data in the segment available to the receiving application.

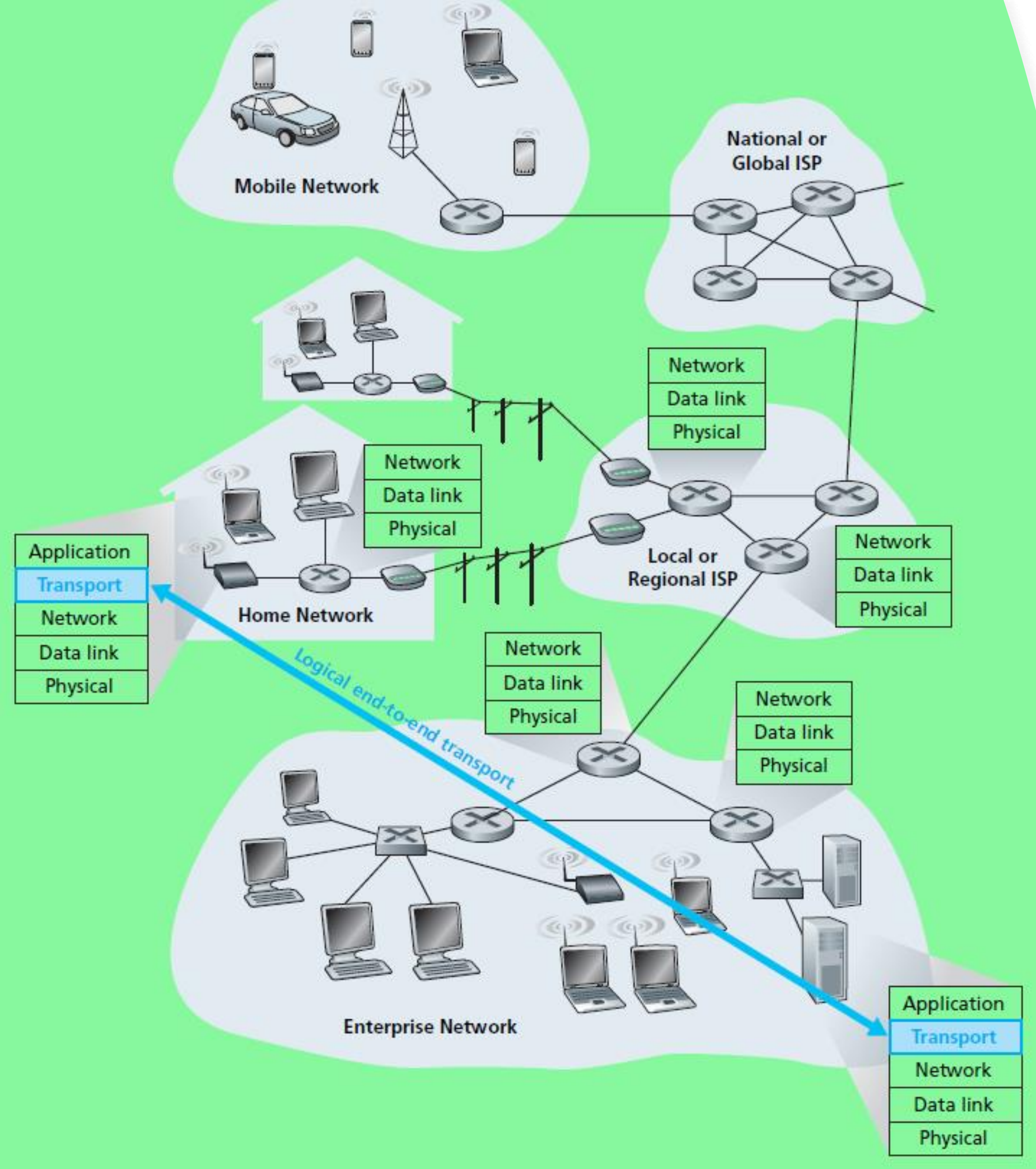


Figure 3.1 ♦ The transport layer provides logical rather than physical communication between application processes

Logical End-to-end Transport

- A transport-layer protocol provides logical communication between *processes* running on different hosts (computers, mobile devices, etc.).

Transport Layer Protocols

TCP (Transmission Control Protocol):

- provides a reliable, connection-oriented service to the invoking application.

UDP (User Datagram Protocol):

- provides an unreliable, connectionless service to the invoking application.

Multiplexing and Demultiplexing

- **Multiplexing :** The job of gathering data chunks at the source host from different sockets, encapsulating each data chunk with header information (that will later be used in demultiplexing) to create segments, and passing the segments to the network layer.
- **Demultiplexing :** At the destination host, the transport layer receives segments from the network layer just below. The transport layer has the responsibility of delivering the data in these segments to the appropriate application process running in the host.
- **How multiplexing and demultiplexing are done?**
- Each segment have special fields that indicate the socket to which the segment is to be delivered. These special fields are the **source port number field** and the **destination port number field**.
- Each port number is a 16-bit number, ranging from 0 to 65535.
- The port numbers ranging from 0 to 1023 are called **well-known port numbers** and are restricted, which means that they .are reserved for use by well-known application protocols such as HTTP.
- Each socket in the host is assigned a port number, and when a segment arrives at the host, the transport layer examines the destination port number in the segment and directs the segment to the corresponding socket. The segment's data then passes through the socket into the attached process.

Segment Format

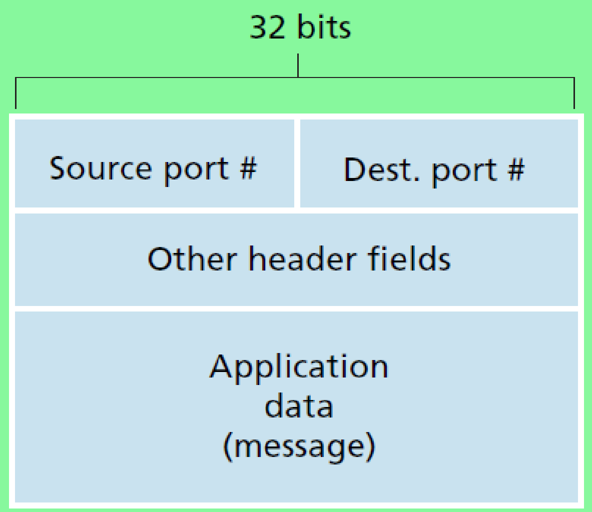
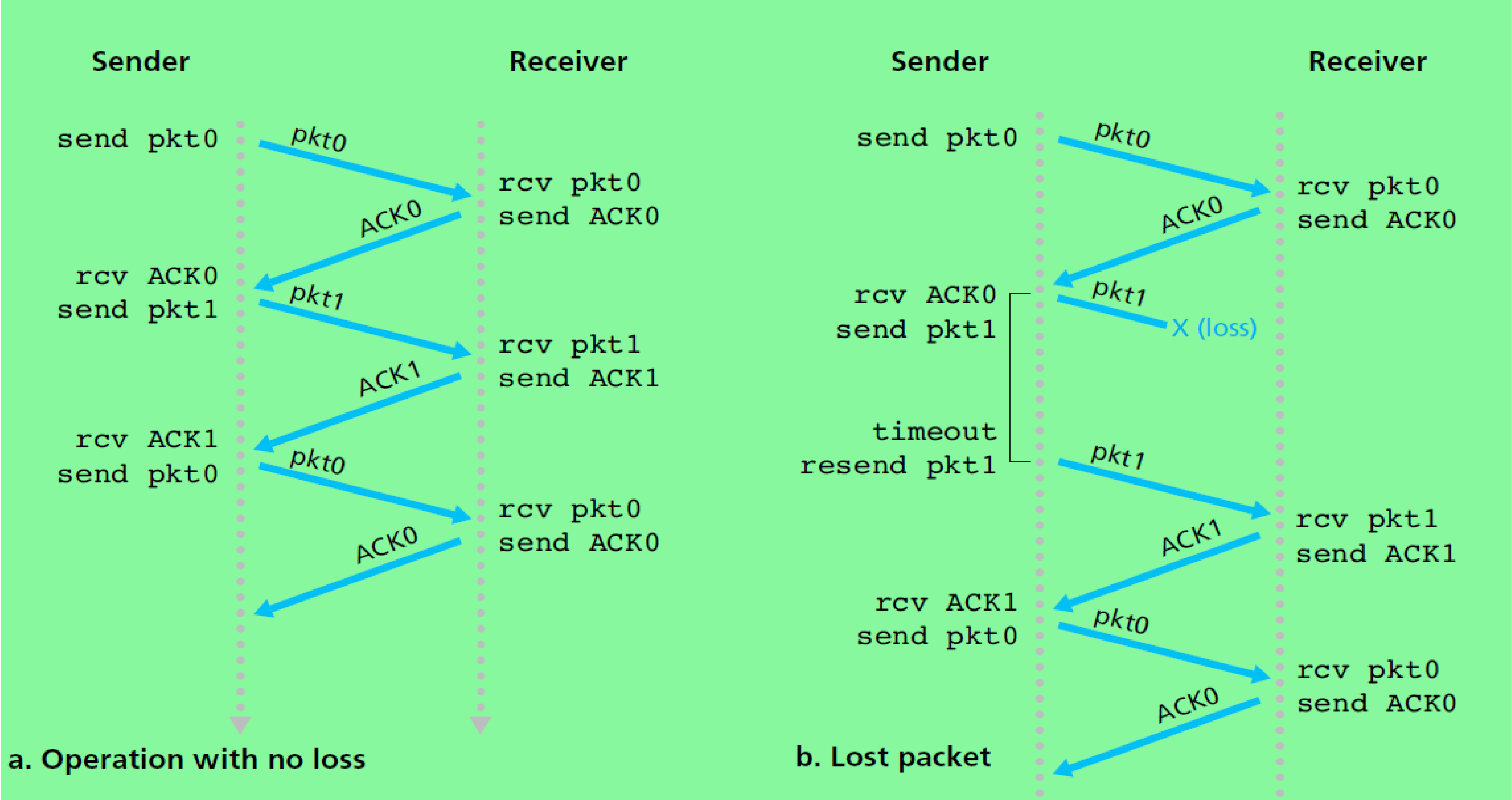


Figure 3.3 ♦ Source and destination port-number fields in a transport-layer segment

- Source port number and destination port number are used for multiplexing/demultiplexing.

Reliable Data Transfer (rdt)



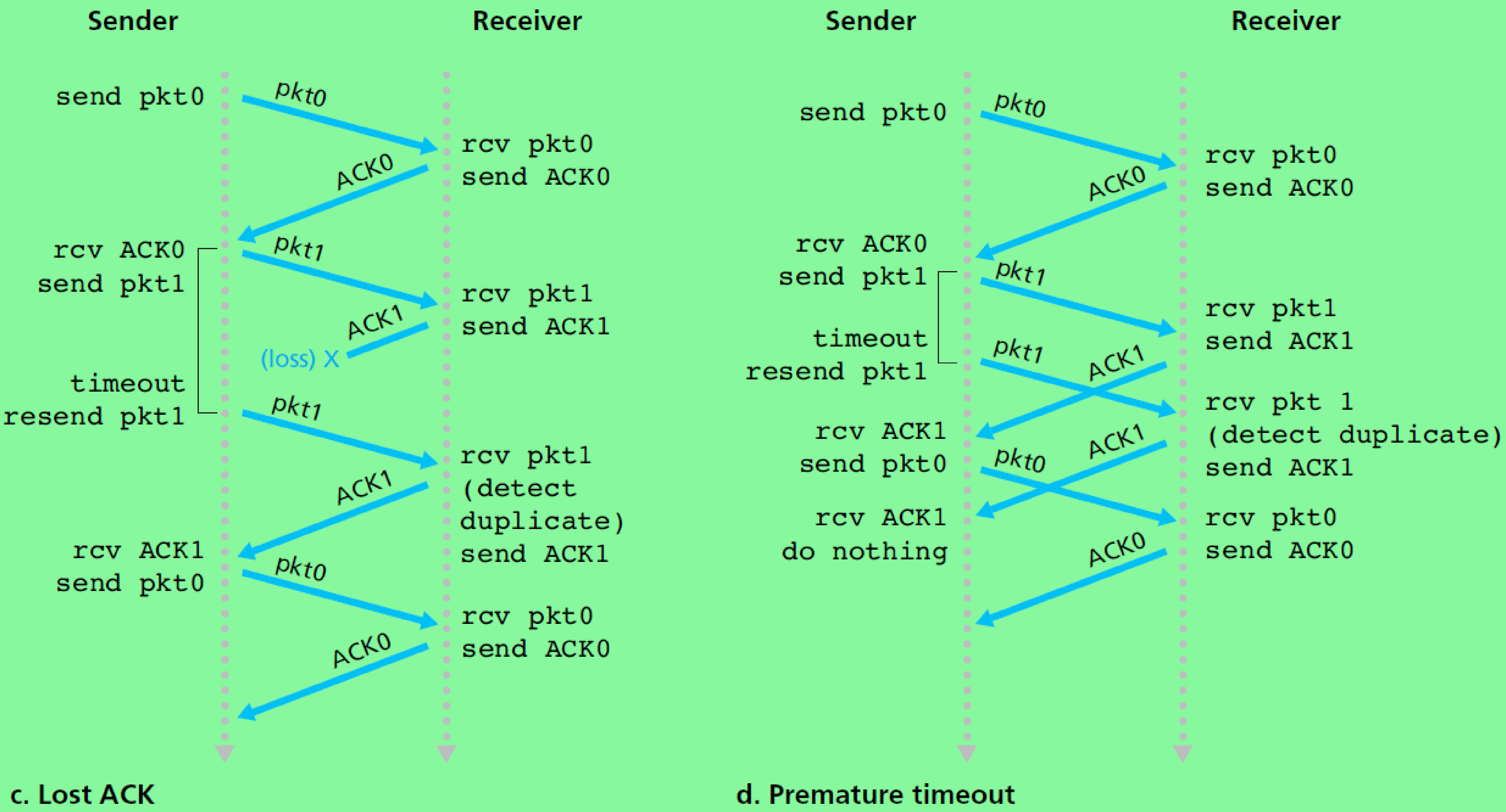


Figure 3.16 ♦ Operation of `rdt3.0`, the alternating-bit protocol

Stop-and Wait Vs. Pipeline Protocols

- Send one segment and wait until an ACK is received is called stop-and-wait.
- Sending a bundle of segments and wait for an ACK for all the sent segments is called pipeline.

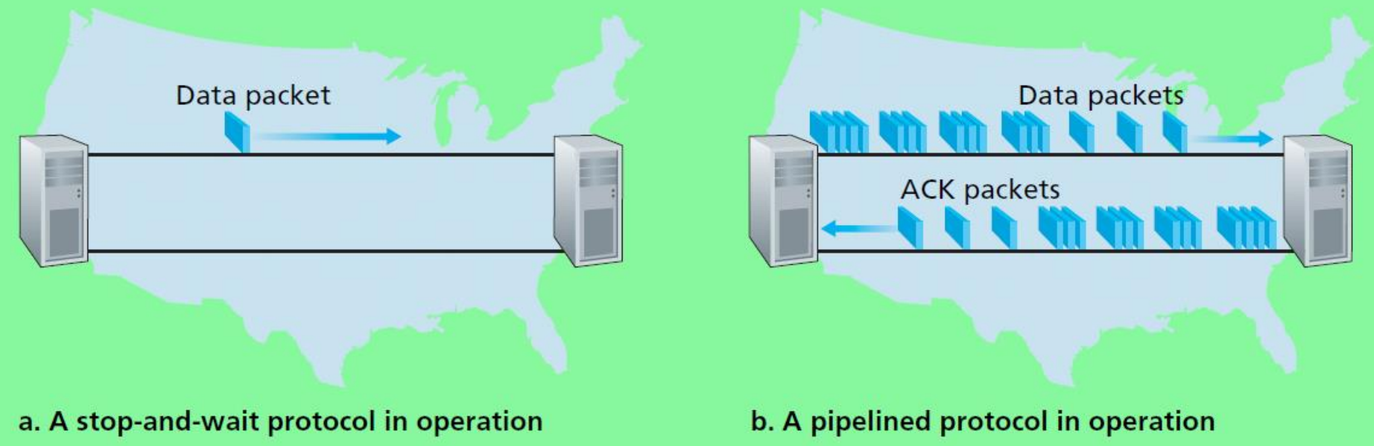
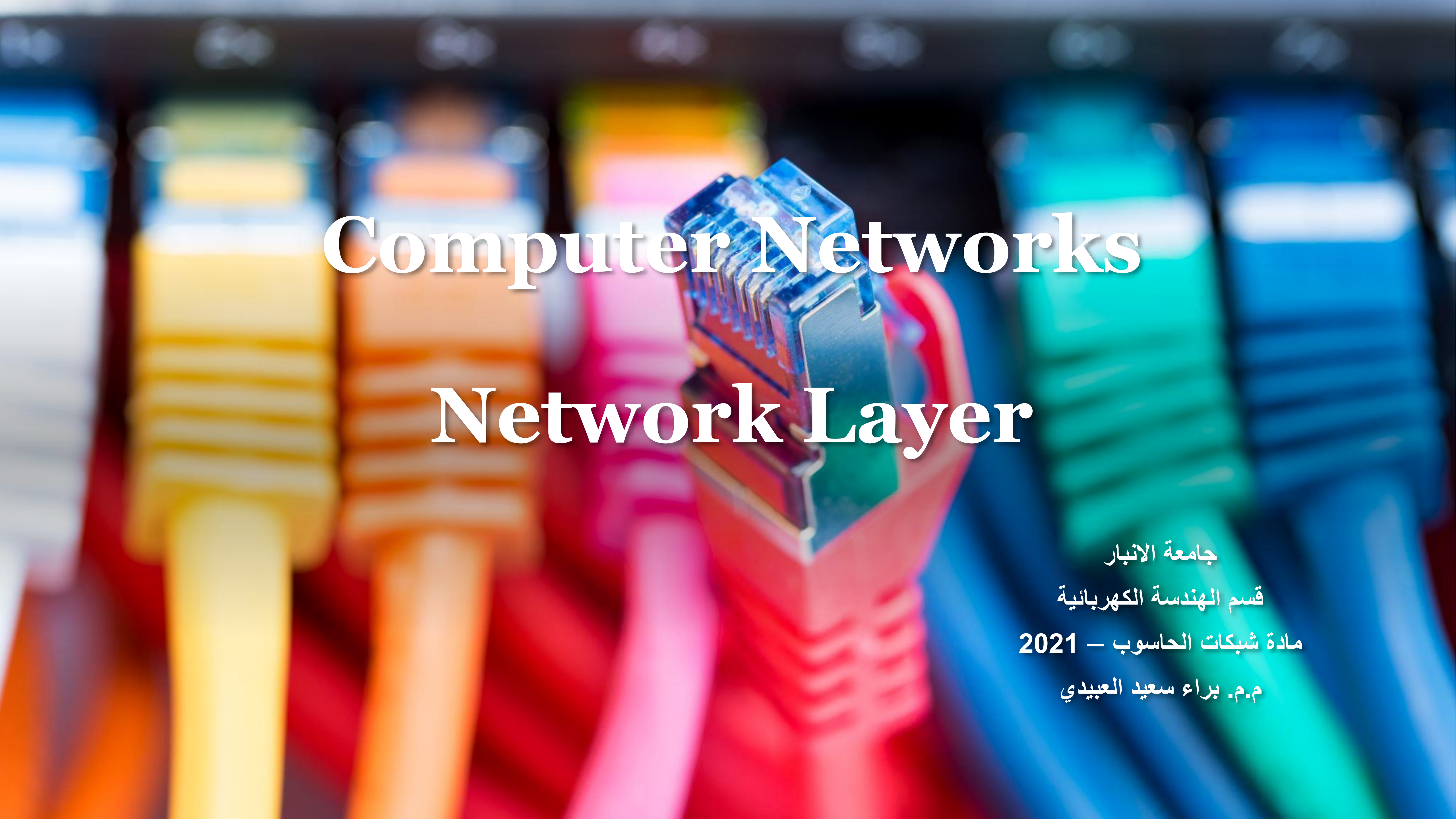


Figure 3.17 ♦ Stop-and-wait versus pipelined protocol

Application	Application-Layer Protocol	Underlying Transport Protocol
Electronic mail	SMTP	TCP
Remote terminal access	Telnet	TCP
Web	HTTP	TCP
File transfer	FTP	TCP
Remote file server	NFS	Typically UDP
Streaming multimedia	typically proprietary	UDP or TCP
Internet telephony	typically proprietary	UDP or TCP
Network management	SNMP	Typically UDP
Routing protocol	RIP	Typically UDP
Name translation	DNS	Typically UDP

Figure 3.6 ♦ Popular Internet applications and their underlying transport protocols



Computer Networks

Network Layer

جامعة الانبار

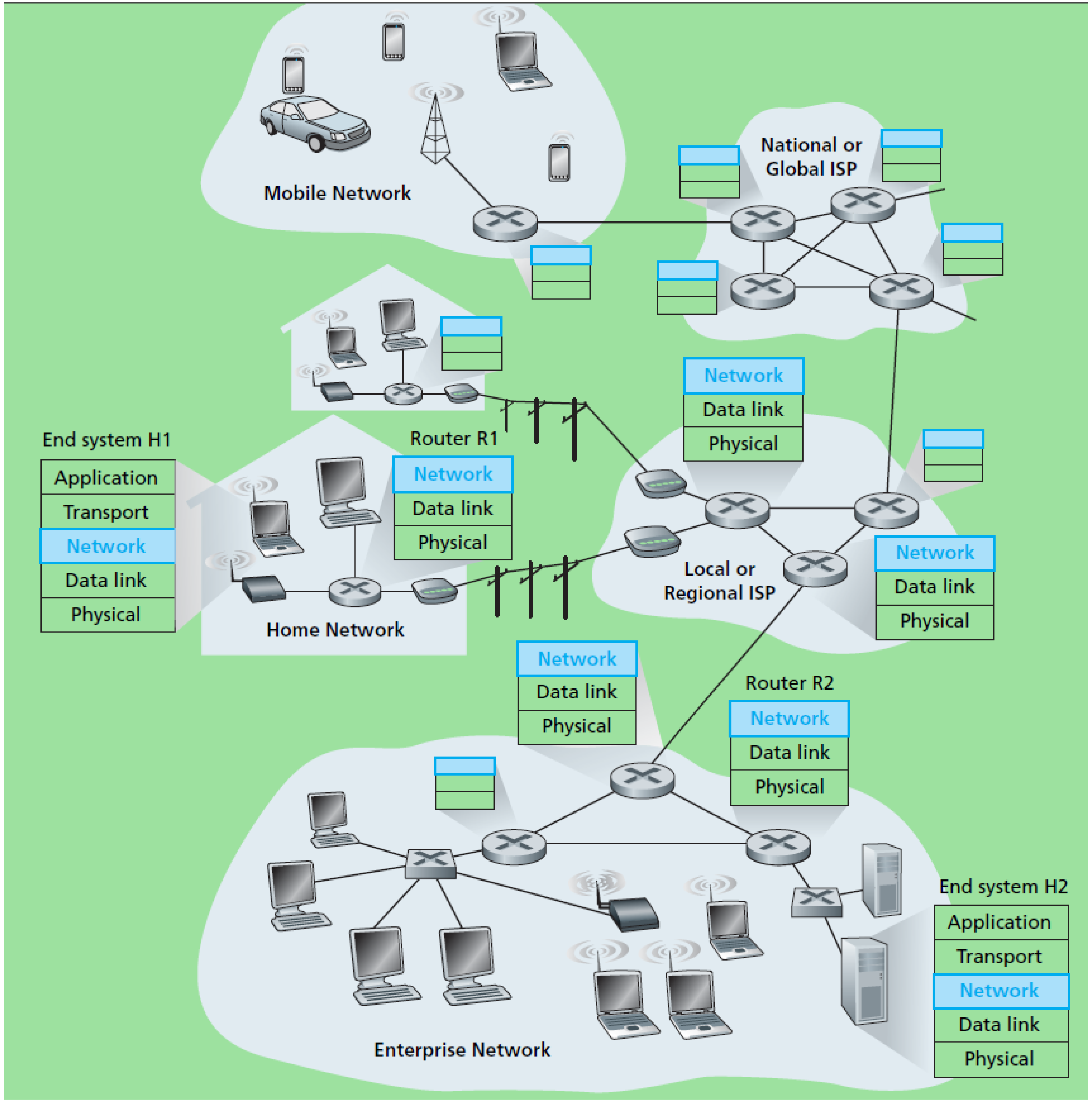
قسم الهندسة الكهربائية

مادة شبكات الحاسوب – 2021

م.م. براء سعيد العبيدي

Network layer

- The network layer implements the host-to-host physical communication service. It is responsible for the actual communication between the sender and the receiver.
- The network layer implements forwarding and routing.
  - Forwarding: the router-local action of transferring a packet from an input link interface to the appropriate output link interface.
  - Routing: refers to the network-wide process that determines the end-to-end paths that packets take from source to destination.
- The main device in this layer is the router.
- Every router has a forwarding table.
- Routing algorithms determine values in forwarding tables.
- The main protocol in this layer is the IP (Internet Protocol).



Mobile Network
National or
Global ISP
End system H1
Application
Transport
Network
Data link
Physical
Router R1
Network
Data link
Physical
Network
Data link
Physical
Home Network
Local or
Regional ISP
Network
Data link
Physical
Network
Data link
Physical
Router R2
Network
Data link
Physical
Enterprise Network
End system H2
Application
Transport
Network
Data link
Physical

IP Address

- IP address is 32 bits.
- **dotted-decimal notation**: each byte of the address is written in its decimal form and is separated by a period (dot) from other bytes in the address. For example:

  11000001 00100000 11011000 00001001 ➔ 193.32.216.9

- **Subnet mask:**

- The first 24 bits of the 32-bit quantity define the subnet address

  223.19.2.X 223.19.2.0/24

Ex.:

00000000.00000000.00000000.X ==> 0.0.0.0

11111111.11111111.11111111.X ==> 255.255.255.255

maximum number of subnets = 2 to power 24 = 16,777,216

maximum number of hosts = 2 to power 8 = 256

IP Address

- The first 16 bits of the 32-bit quantity define the subnet address

  244.123.100.0/16

  223.19.X.X

Ex.:

00000000.00000000.X.X ==> 0.0.X.X

11111111.11111111.X.X ==> 255.255.X.X

maximum number of subnets = 65,536 subnets

maximum number of hosts = 65,536 hosts

IP Address

- The first 8 bits of the 32-bit quantity define the subnet address

223.X.X.X 223.19.2.0/8

Ex.:

00000000.X.X.X ==> 0.X.X.X

11111111.X.X.X ==> 255.X.X.X

maximum number of subnets = 2 to power 8 = 256

maximum number of hosts= 2 to power 24 = 16,777,216

Example

Q1: Design a network that can provide 65,563 IP address i.e. the network can host up to 59000 hosts) given that you need only 255 subnets. Divide the IP address accordingly.

A1:

The IP address should be divided as follows:

16 bits (2 bytes) for the subnet part, which gives us 65,536 subnets.

16 bits (2 bytes) for the hosts part, which gives us 65,536 hosts.

For the subnets:	for the hosts:
00000000.00000000.X.X	X.X.00000000.00000000
through	through
11111111.11111111.X.X	X.X.11111111.11111111

An Example of a Forwarding Table

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

Router Architecture

A high-level view of a generic router architecture is shown in Figure 4.6. Four router components can be identified:

- *Input ports:* An input port performs several key functions. It performs the physical layer function of terminating an incoming physical link at a router; An input port also performs link-layer functions needed to interoperate with the link layer at the other side of the incoming link. The lookup function is also performed at the input port.
- *Switching fabric:* The switching fabric connects the router's input ports to its output ports.
- *Output ports:* An output port stores packets received from the switching fabric and transmits these packets on the outgoing link by performing the necessary link-layer and physical-layer functions.

Router Architecture

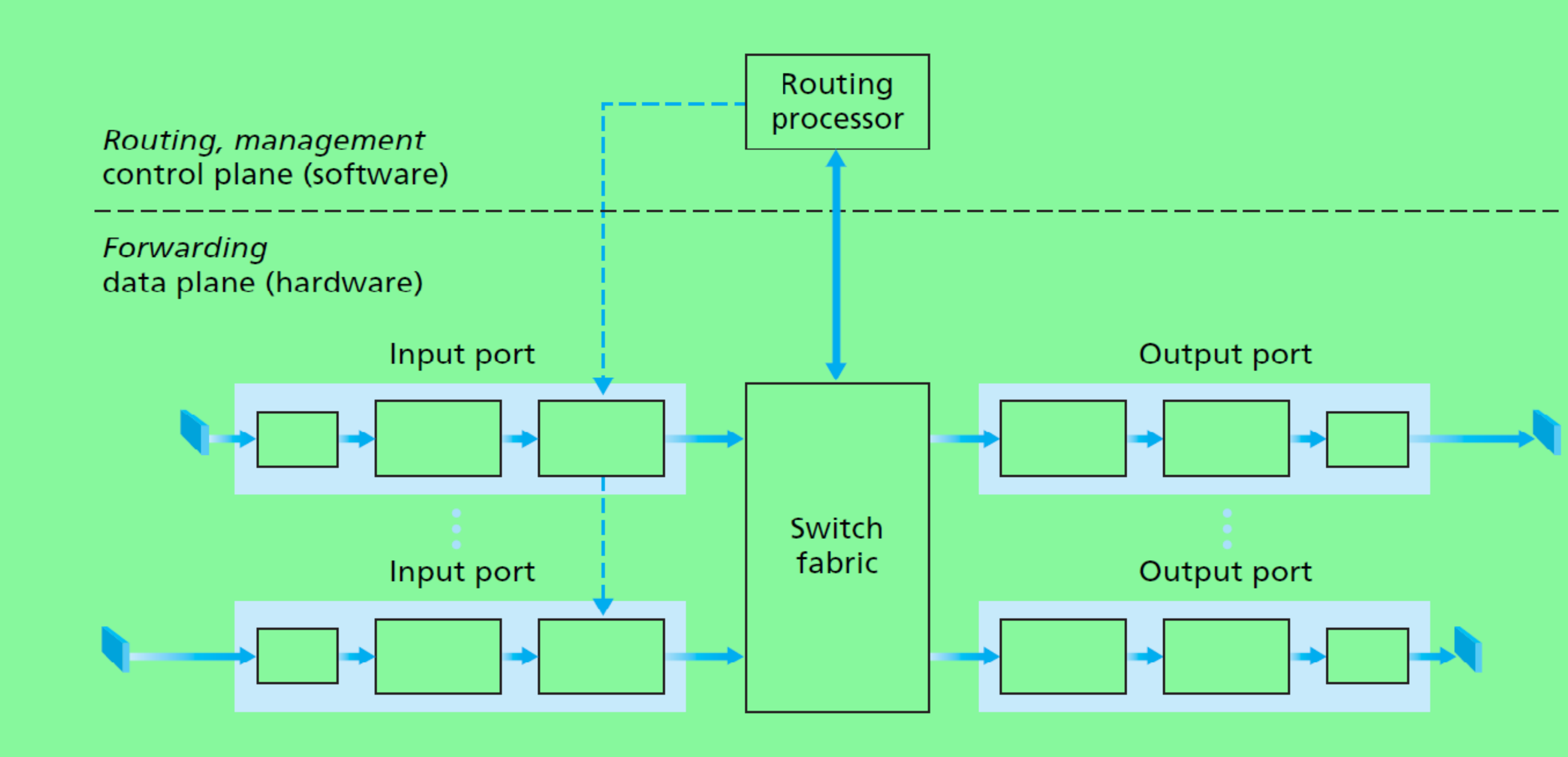


Figure 4.6 ♦ Router architecture

Three subnets connected by one router

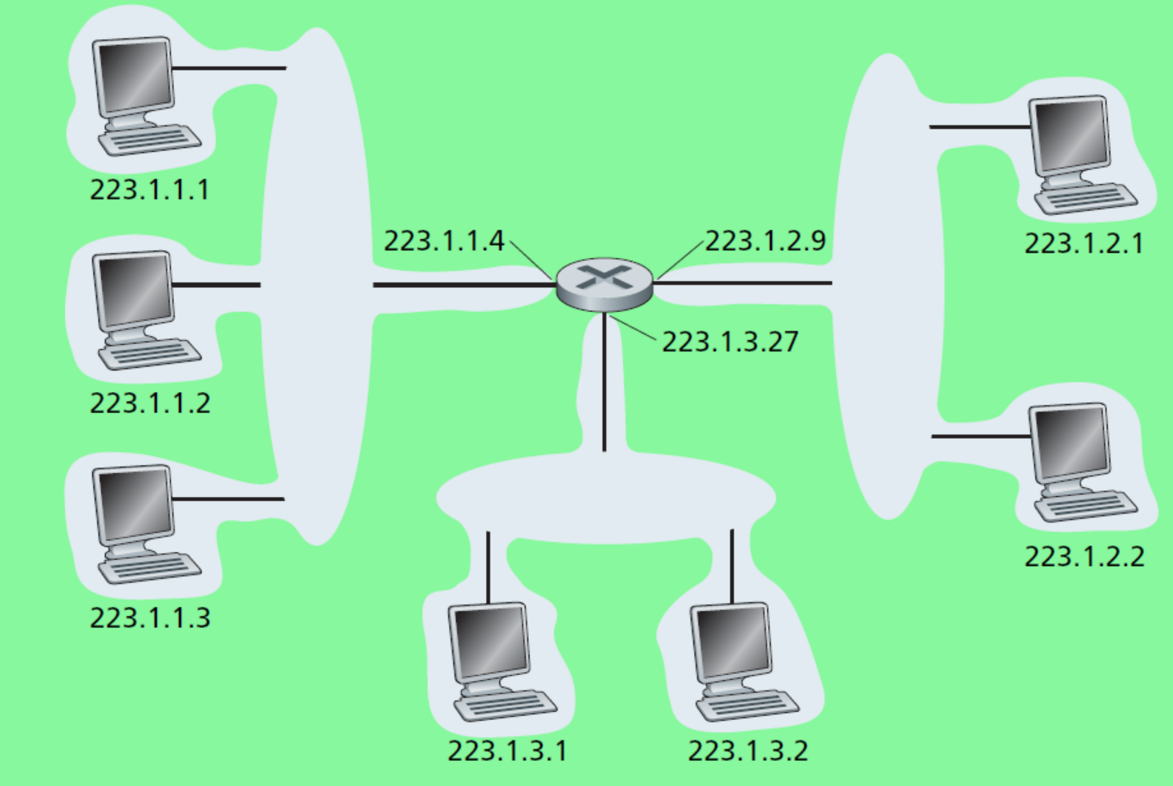


Figure 4.15 ♦ Interface addresses and subnets

How forwarding is done?

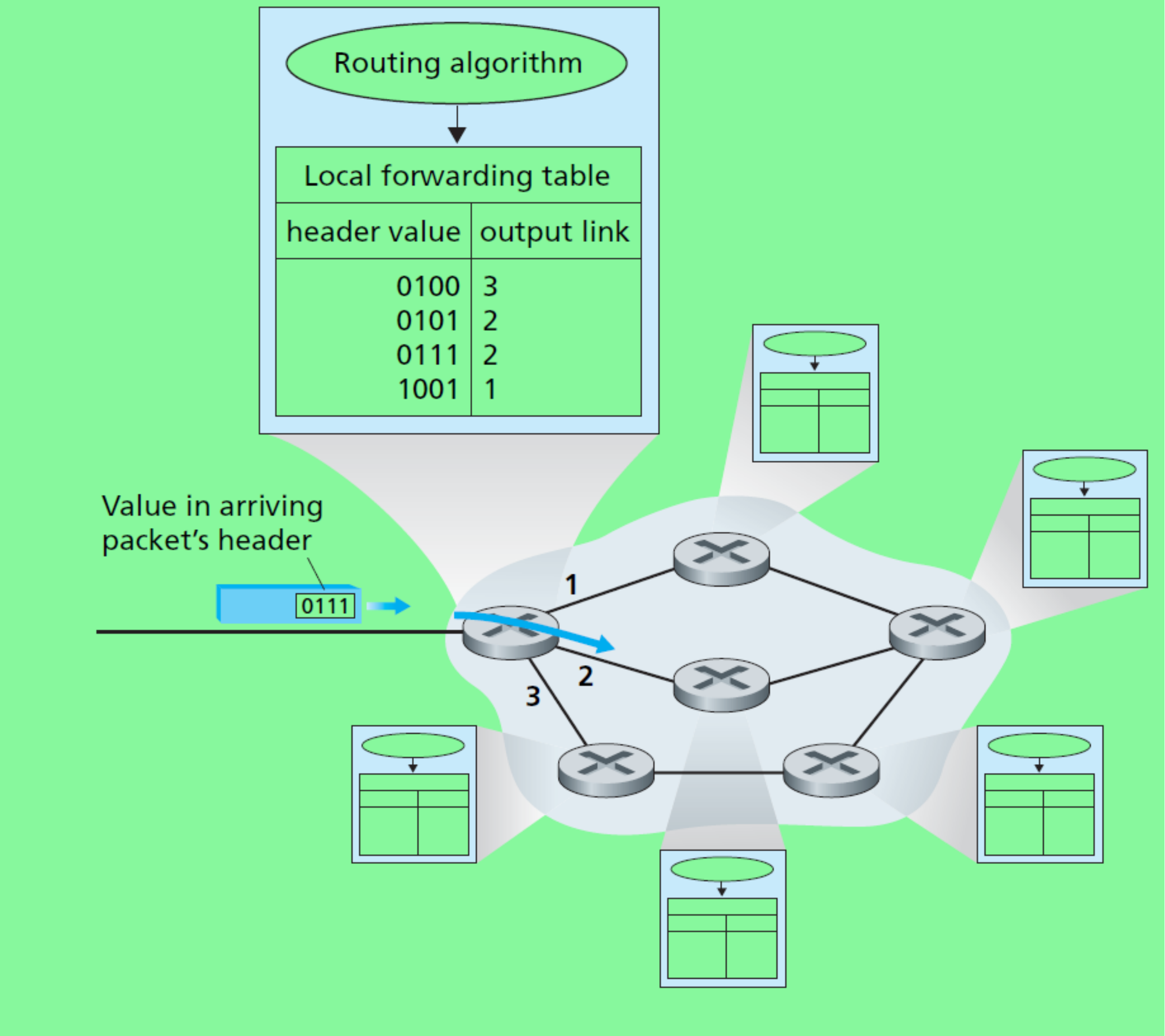


Figure 4.2 ♦ Routing algorithms determine values in forwarding tables

Link Layer

جامعة الانبار

قسم الهندسة الكهربائية

مادة شبكات الحاسوب – 2021

م.م. براء سعيد العبيدي

Main function, protocols and data unit.

The main function of the link layer is node to node delivery.

The main protocols: PPP, Ethernet, FDM, TDM, CDMA.

The data unit is called frame.

Where is the link layer implemented?

- The link layer is implemented in the NICs (network interface cards) of nodes.
- Nodes include hosts, routers, switches, and WiFi access points.

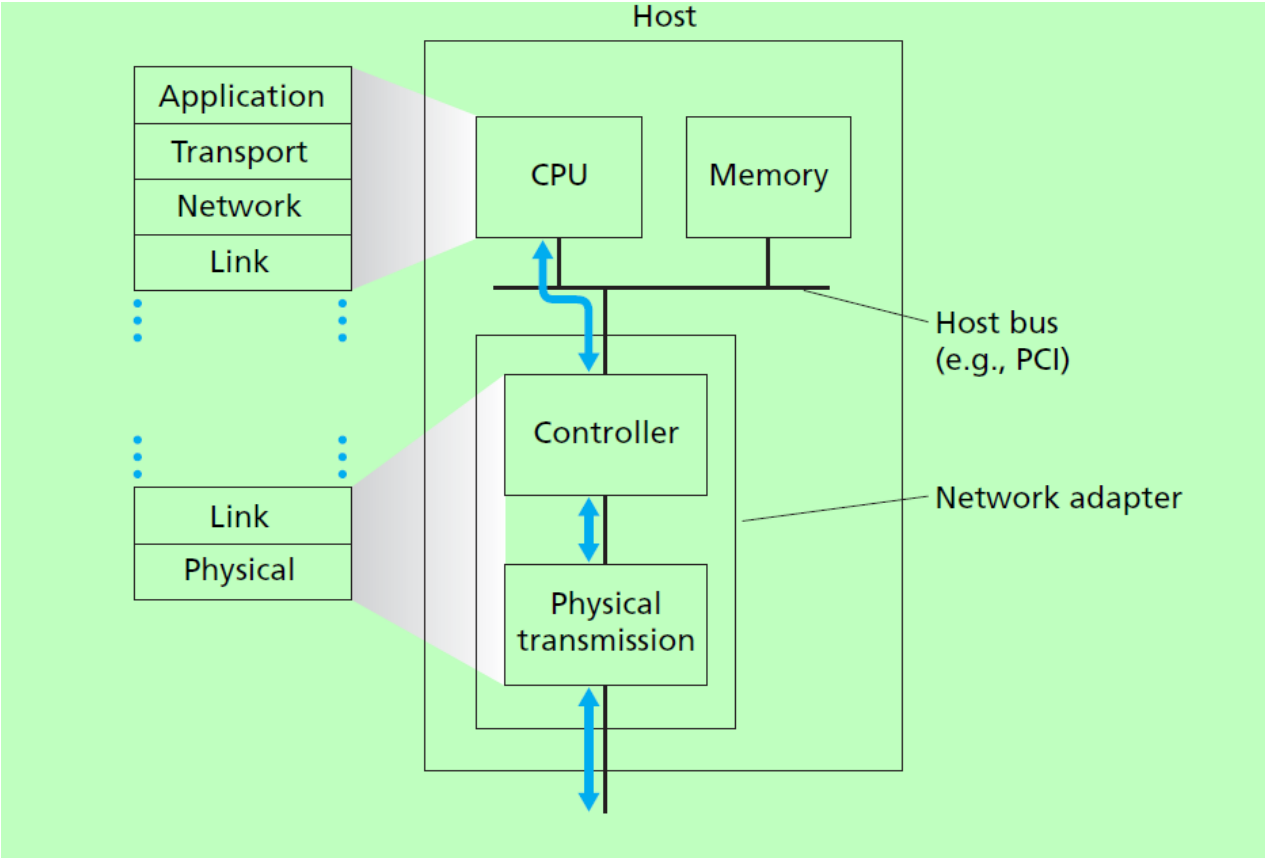


Figure 5.2 ♦ Network adapter: its relationship to other host components and to protocol stack functionality

Multiple Access Links and Protocols

TDM (Time-Division Multiplexing): TDM divides time into time frames and further divides each time frame into N time slots.

FDM (Frequency- Division Multiplexing): FDM divides the R bps channel into different frequencies (each with a bandwidth of R/N) and assigns each frequency to one of the N nodes.

CDMA (Code Division Multiple Access): CDMA assigns a different code to each node. Each node then uses its unique code to encode the data bits it sends.