# Transport Layer Services

**Objectives:**

- To define process-to-process communication at the transport layer.
- To discuss the addressing mechanism at the transport layer, to discuss port numbers, and to define the range port numbers used for different purposes.
- To discuss the connectionless and connection-oriented services at the transport layer.
- To explain the format of a UDP packet, which is called a user datagram, and discuss the use of each field in the header.
- To introduce TCP as a protocol that provides reliable stream delivery service.

# Transport Layer Services

- The transport layer is responsible for providing services to the application layer; it receives services from the network layer.

## Process-to-Process Communication

- A process is an application-layer entity (running program) that uses the services of the transport layer.

# Addressing: Port Numbers

- A process on the local host, called a client, needs services from a process usually on the remote host, called a server.

**Internet Corporation for Assigned Names and Numbers (ICANN) Ranges:**

- Well-known ports: The ports ranging from 0 to 1,023 are assigned and controlled by ICANN. These are the well-known ports.
- Registered ports: The ports ranging from 1,024 to 49,151 are not assigned or controlled by ICANN. They can only be registered with ICANN to prevent duplication.
- Dynamic ports: The ports ranging from 49,152 to 65,535 are neither controlled nor registered.
- A transport-layer protocol in the TCP suite needs both the IP address and the port number, at each end, to make a connection. The combination of an IP address and a port number is called a socket address.

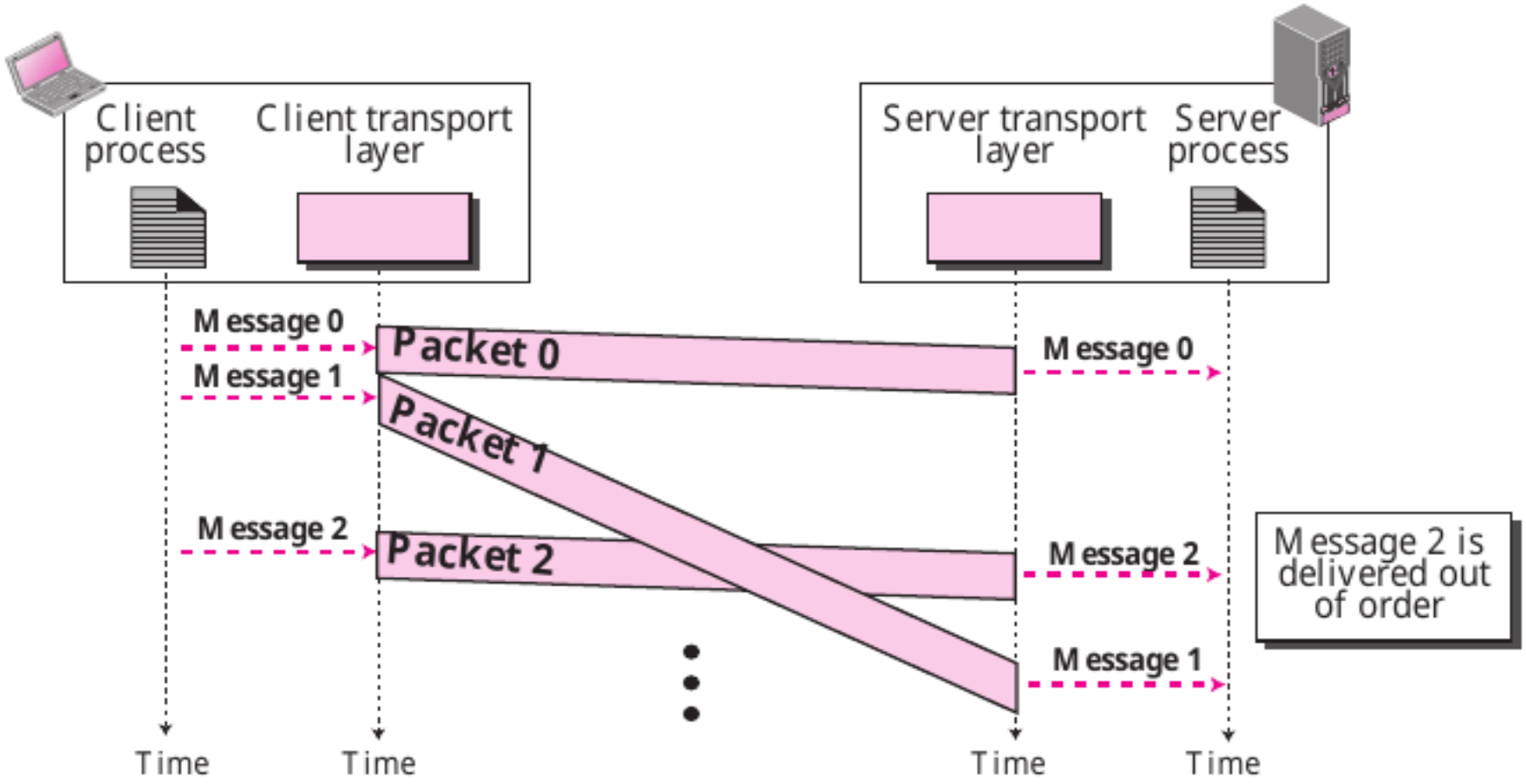
# Connectionless vs. Connection-Oriented Services

- Connectionless service at the transport layer means independency between packets.
- Connection-oriented means dependency.

## Connectionless Services

- In a connectionless service, the source process (application program) needs to divide its message into chunks of data of the size acceptable by the transport layer and deliver them to the transport layer one by one.
- The transport layer treats each chunk as a single unit without any relation between the chunks.
- When a chunk arrives from the application layer, the transport layer encapsulates it in a packet and sends it.

# Connectionless Process

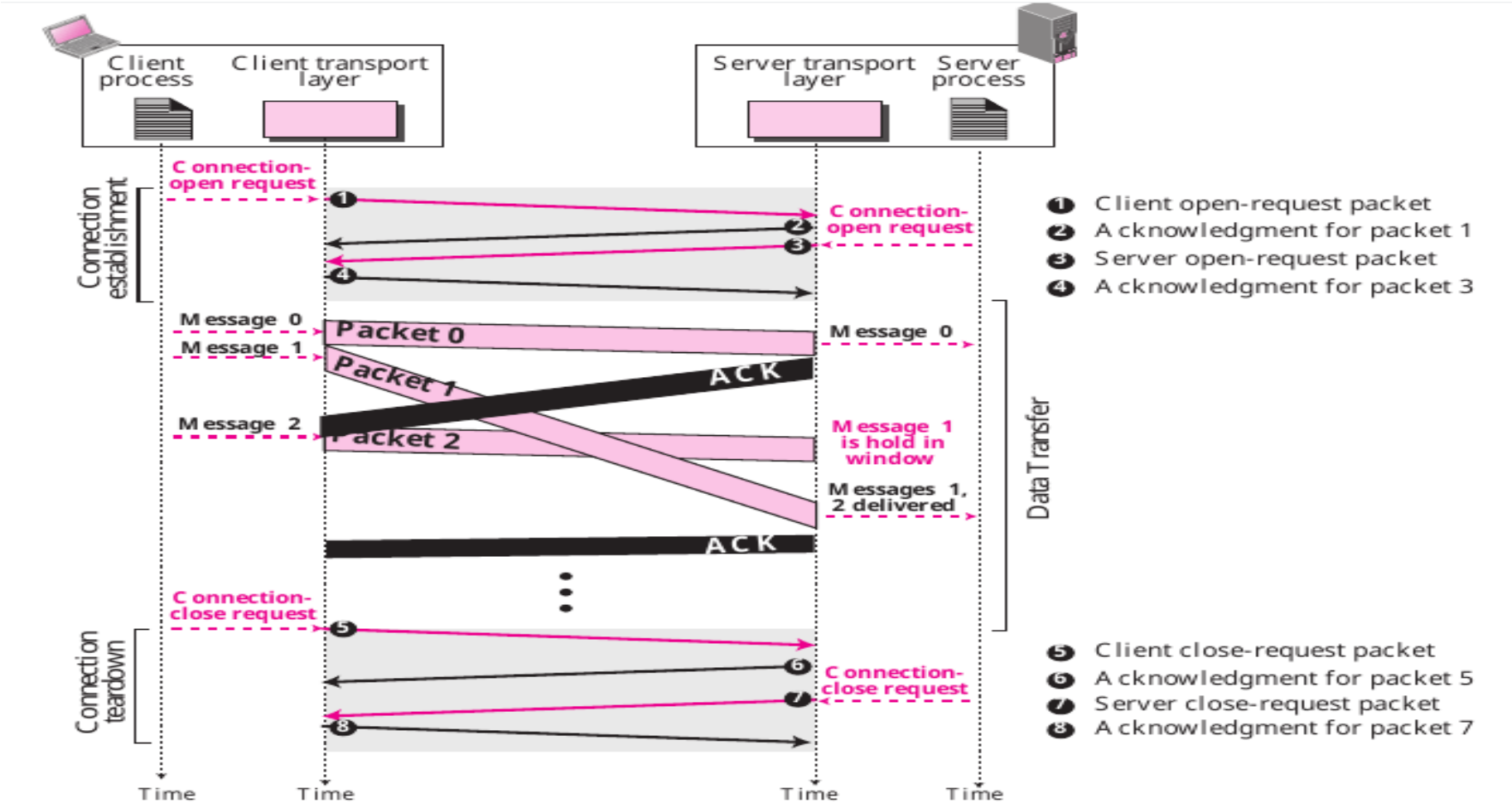


# Connection-Oriented Service

- In a connection-oriented service, the client and the server first need to establish a connection between themselves.

- The data exchange can only happen after the connection establishment. After data exchange, the connection needs to be teared down.

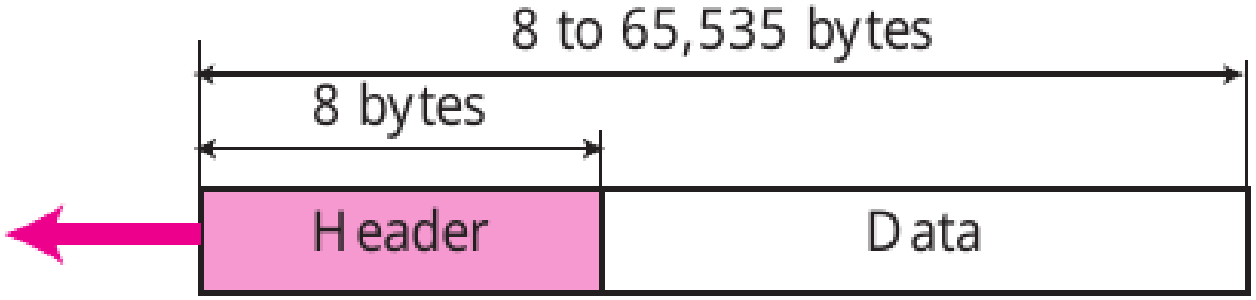
# Connection-Oriented Process



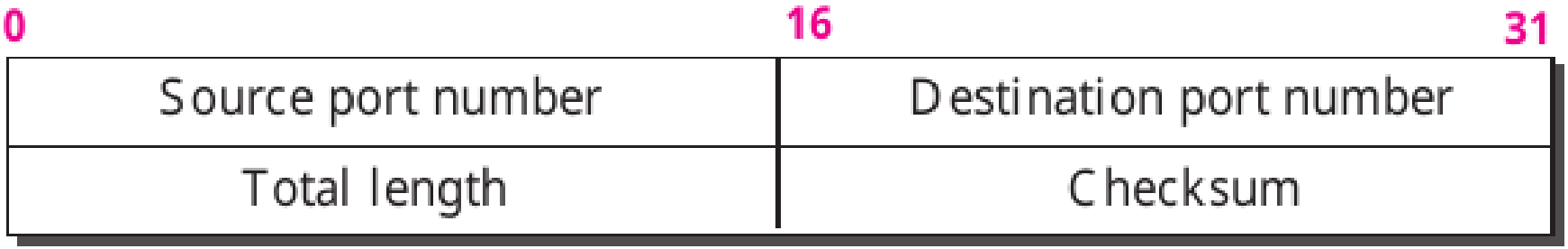
# User Datagram Protocol (UDP)

- UDP is a connectionless, unreliable transport protocol.
- It does not add anything to the services of IP except for providing process-to-process communication instead of host-to-host communication.

# UDP Format




a. UDP user datagram




b. Header format

# UDP Format in Detail

- Source port number: This is the port number used by the process running on the source host.
- Destination port number: This is the port number used by the process running on the destination host.
- Length: This is a 16-bit field that defines the total length of the user datagram, header plus data.
- Checksum: This field is used to detect errors over the entire user datagram (header plus data).

# Well-Known Ports used with UDP

| | | |
|---|---|---|
| 53 | Domain | Domain Name Service (DNS) |
| 67 | Bootps | Server port to download bootstrap information |
| 68 | Bootpc | Client port to download bootstrap information |
| 69 | TFTP | Trivial File Transfer Protocol |
| 111 | RPC | Remote Procedure Call |
| 123 | NTP | Network Time Protocol |
| 161 | SNMP | Simple Network Management Protocol |
| 162 | SNMP | Simple Network Management Protocol (trap) |

# Well-Known Ports used with TCP

| | | |
|---|---|---|
| 20 and 21 | FTP | File Transfer Protocol (Data and Control) |
| 23 | TELNET | Terminal Network |
| 25 | SMTP | Simple Mail Transfer Protocol |
| 53 | DNS | Domain Name Server |
| 67 | BOOTP | Bootstrap Protocol |
| 79 | Finger | Finger |
| 80 | HTTP | Hypertext Transfer Protocol |

# TCP Features

## Full-Duplex Communication

- TCP offers full-duplex service, where data can flow in both directions at the same time.

## Multiplexing and Demultiplexing

- TCP performs multiplexing at the sender and demultiplexing at the receiver.

# TCP Features Continue

## Connection-Oriented Service

- When a process at site A wants to send to and receive data from another process at site B, the following three phases occur:
  1. The two TCPs establish a virtual connection between them.
  2. Data are exchanged in both directions.
  3. The connection is terminated.

## Reliable Service

- TCP is a reliable transport protocol. It uses an acknowledgment mechanism to check the safe and sound arrival of data.

# TCP Features Continue

## Numbering System

- There are two fields called the sequence number and the acknowledgment number.

## Sequence Number

- TCP assigns a sequence number to each segment that is being sent.

## Acknowledgment Number

- Each party also uses an acknowledgment number to confirm the bytes it has received

# TCP Features Continue

## Flow Control

- The sending TCP controls how much data can be accepted from the sending process; the receiving TCP controls how much data can to be sent by the sending TCP.

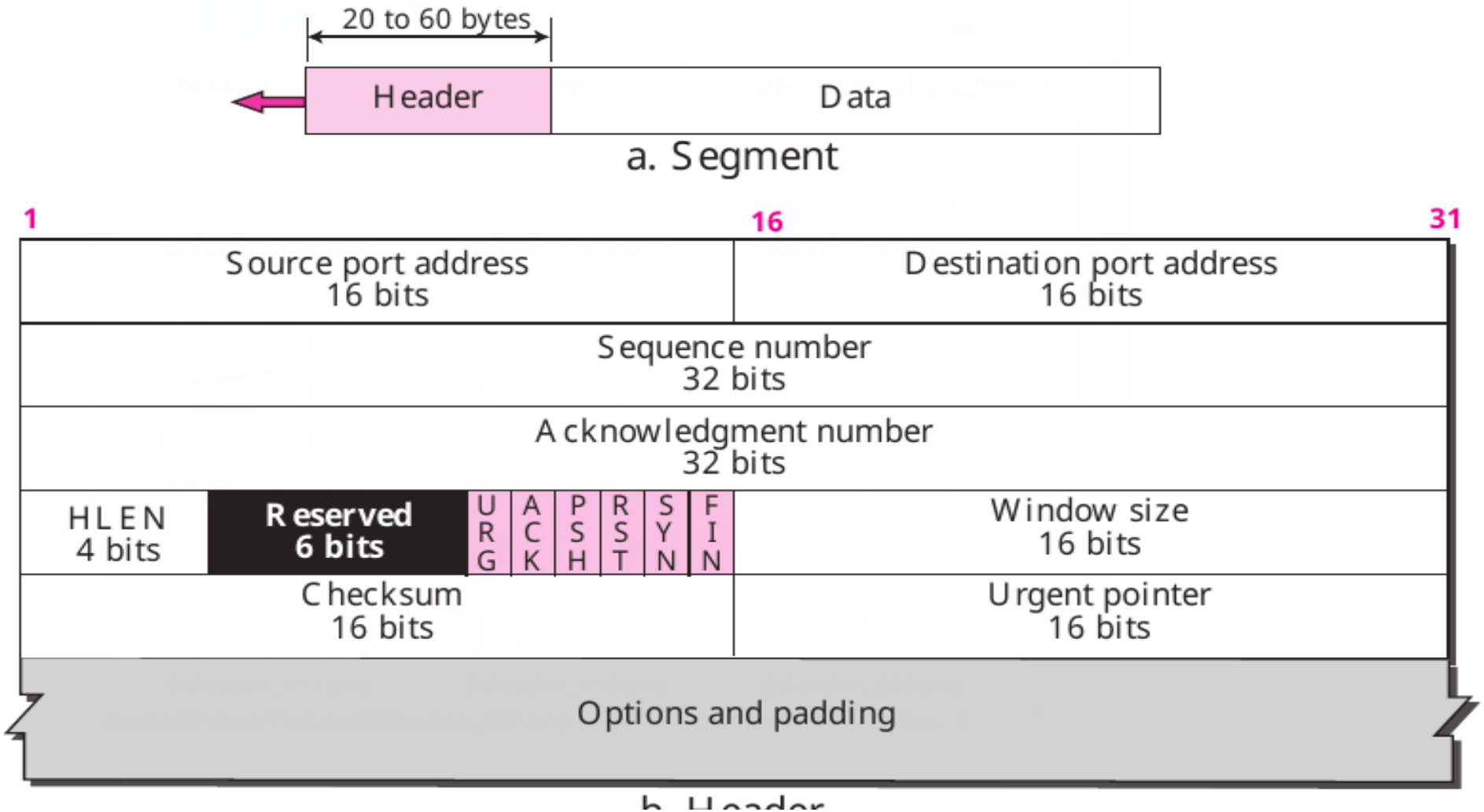
## Error Control

- To provide reliable service, TCP implements an error control mechanism. Although error control considers a segment as the unit of data for error detection (loss or corrupted segments), error control is byte-oriented.

## Congestion Control

- The amount of data sent by a sender is not only controlled by the receiver (flow control), but is also determined by the level of congestion, if any, in the network.

# TCP Format


20 to 60 bytes
Header
Data
a. Segment
1
16
31
Source port address
16 bits
Destination port address
16 bits
Sequence number
32 bits
Acknowledgment number
32 bits
HLEN
4 bits
Reserved
6 bits
URG
ACK
PSH
RST
SYN
FIN
Window size
16 bits
Checksum
16 bits
Urgent pointer
16 bits
Options and padding

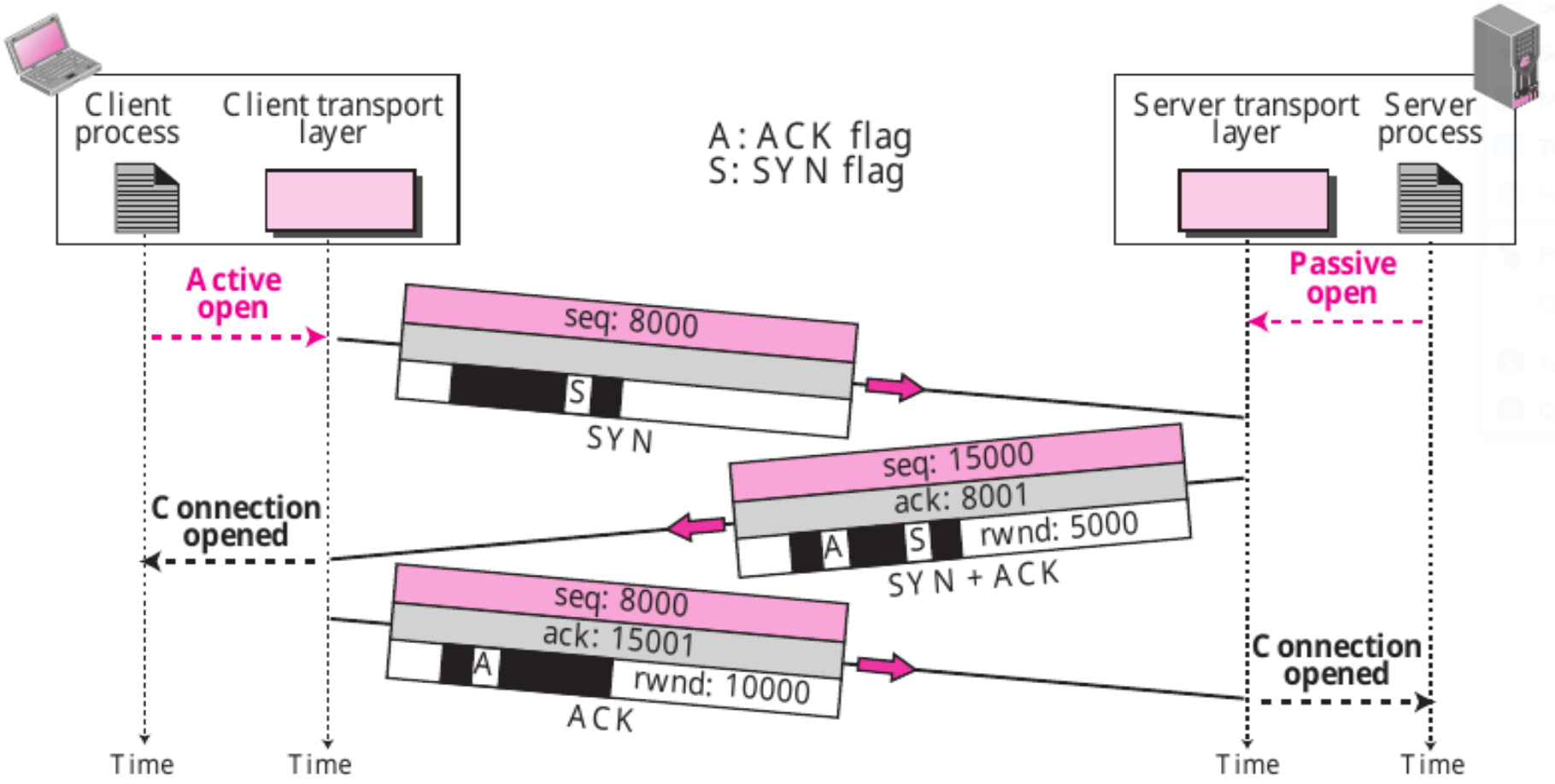
# TCP Segment Format

- Source port address: This is a 16-bit field that defines the port number of the application program in the host that is sending the segment.
- Destination port address: This is a 16-bit field that defines the port number of the application program in the host that is receiving the segment.
- Sequence number: This 32-bit field defines the number assigned to the first byte of data contained in this segment
- Acknowledgment number: This 32-bit field defines the byte number that the receiver of the segment is expecting to receive from the other party.
- Header length: This 4-bit field indicates the number of 4-byte words in the TCP header.

# TCP Segment Format Continue

- Reserved: This is a 6-bit field reserved for future use.
- Control: This field defines 6 different control bits or flags one or more of these bits can be set at a time.
- Window size: This field defines the window size of the sending TCP in bytes.
- Checksum: This 16-bit field contains the checksum.
- Urgent pointer: This 16-bit field, which is valid only if the urgent flag is set, is used when the segment contains urgent data.
- Options: There can be up to 40 bytes of optional information in the TCP header.

# Three-way Handshake



# End of Chapter 5
# :)

# Application Layer

**Objectives:**

- To introduce client-server paradigm.
- To discuss client-server communication using connectionless iterative service offered by UDP.
- To discuss client-server communication using connection-oriented concurrent service offered by TCP.
- To give the reasons why we need host configuration.
- To describe the purpose of DNS.
- To introduce SSH as an alternative to TELNET.
- To discuss FTP and two connections used in this protocol: control connection and data connection.

# Client-Server Paradigm

- The purpose of a network, or an internetwork, is to provide services to users.
- A user at a local site wants to receive a service from a computer at a remote site.

**Server**: A server is a program running on the remote machine providing service to the clients.

**Client**: A client is a program running on the local machine requesting service from a server

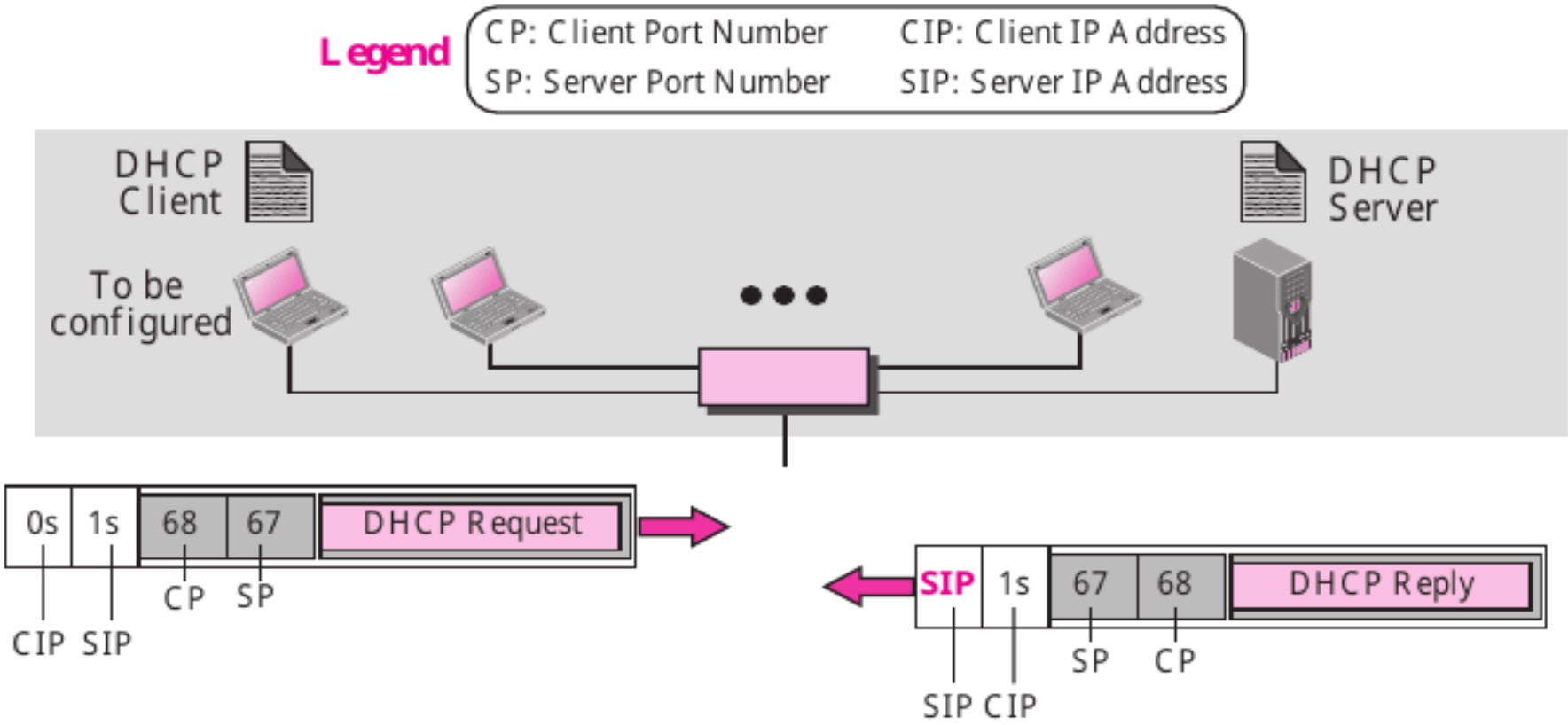
# DHCP: Dynamic Host Configuration Protocol

- Each computer that uses the TCP/IP protocol suite needs to know its IP address.
- Most computers today need two other pieces of information: the address of a default router to be able to communicate with other networks and the address of a name server to be able to use names instead of addresses.

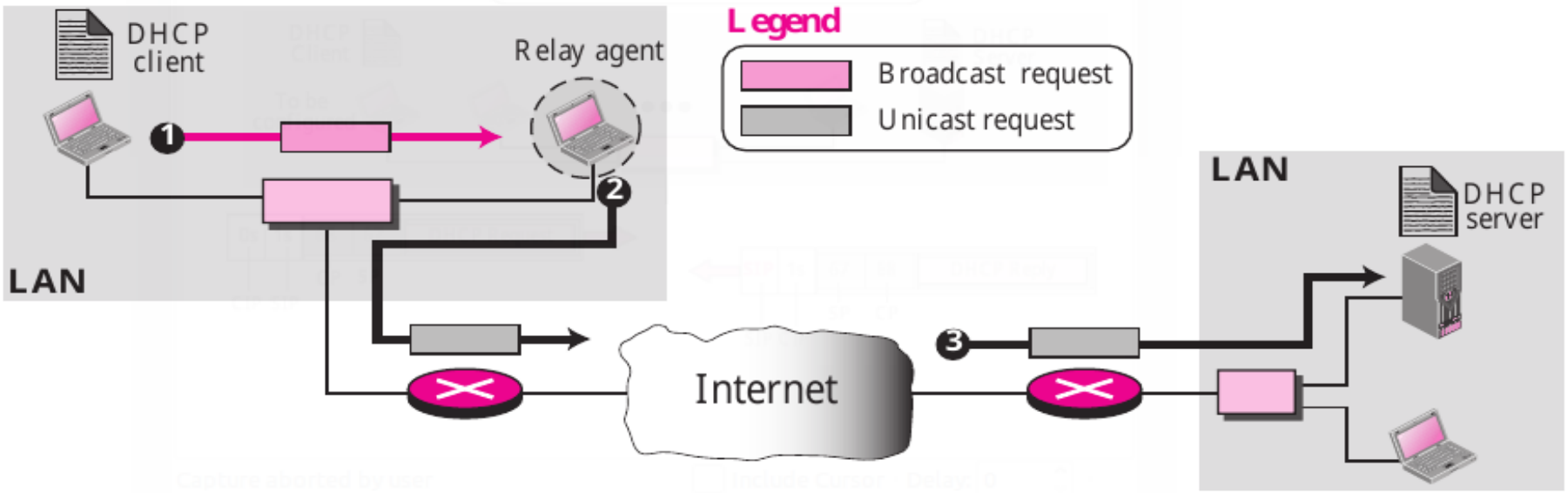
Four pieces of information are normally needed:

1) The IP address of the computer
2) The subnet mask of the computer
3) The IP address of a router
4) The IP address of a name server

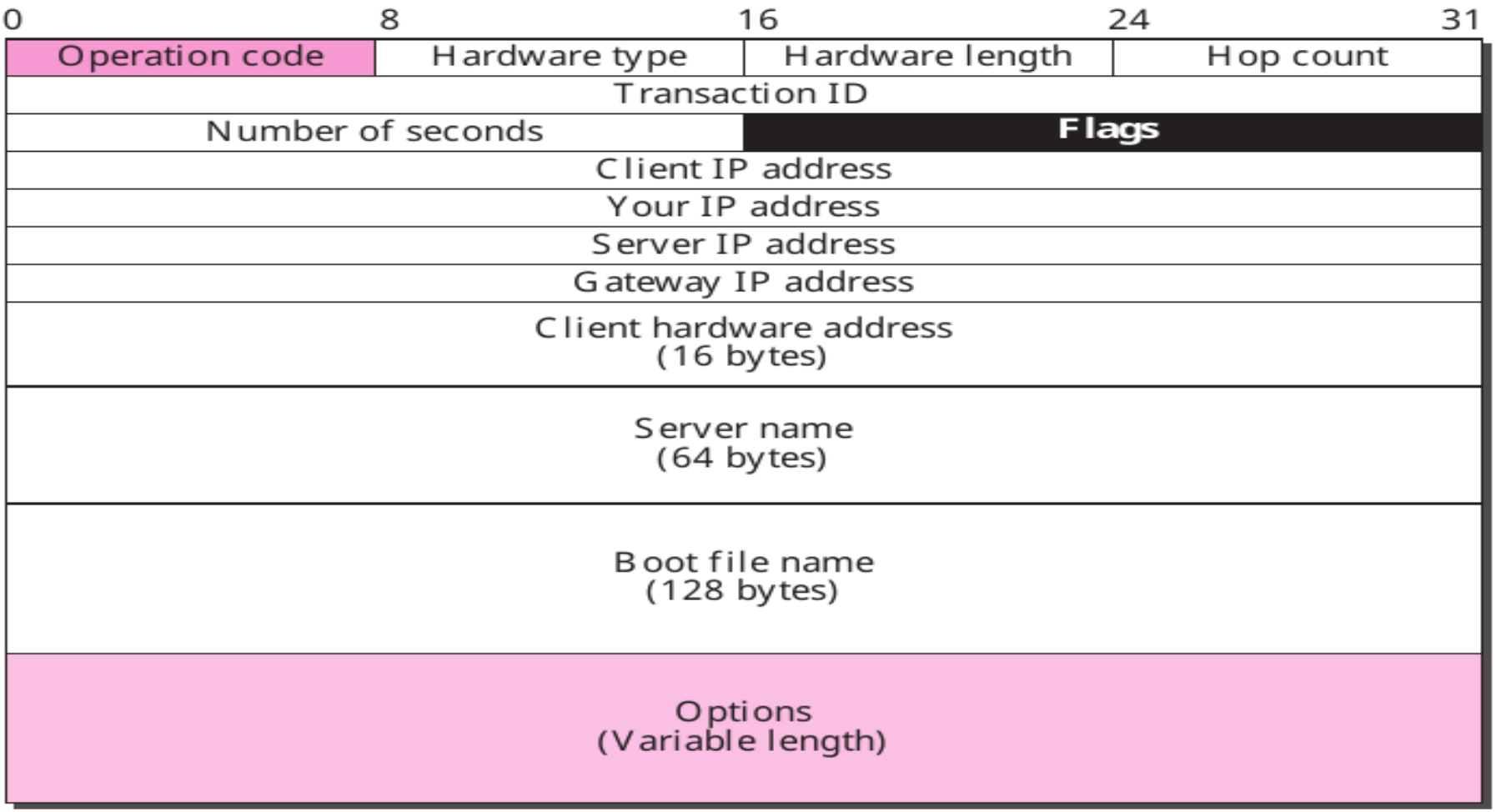
# Client & server on the same network



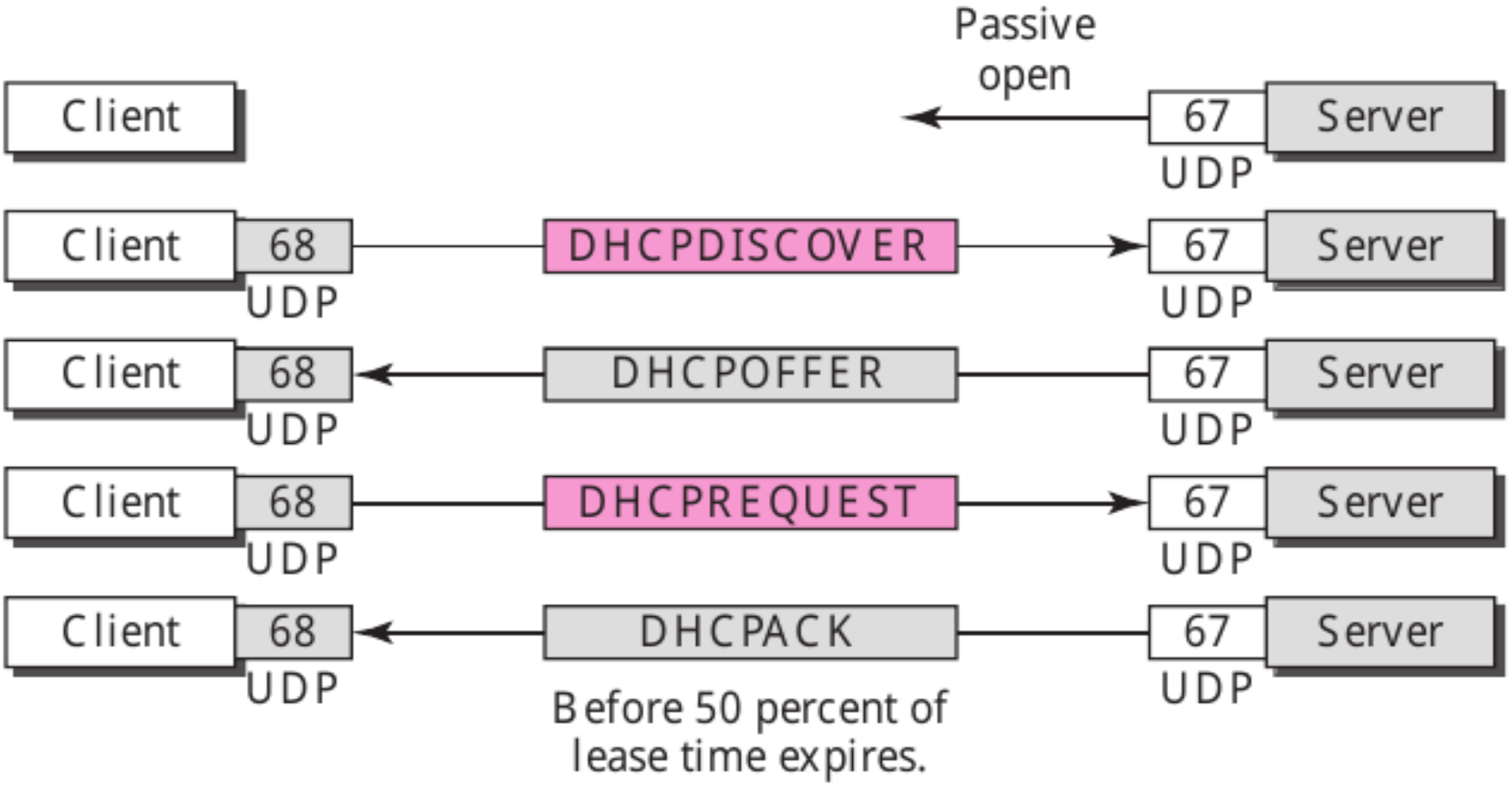
# Client & server on two different networks



# DHCP Packet Format



# DHCP Exchanging Messages



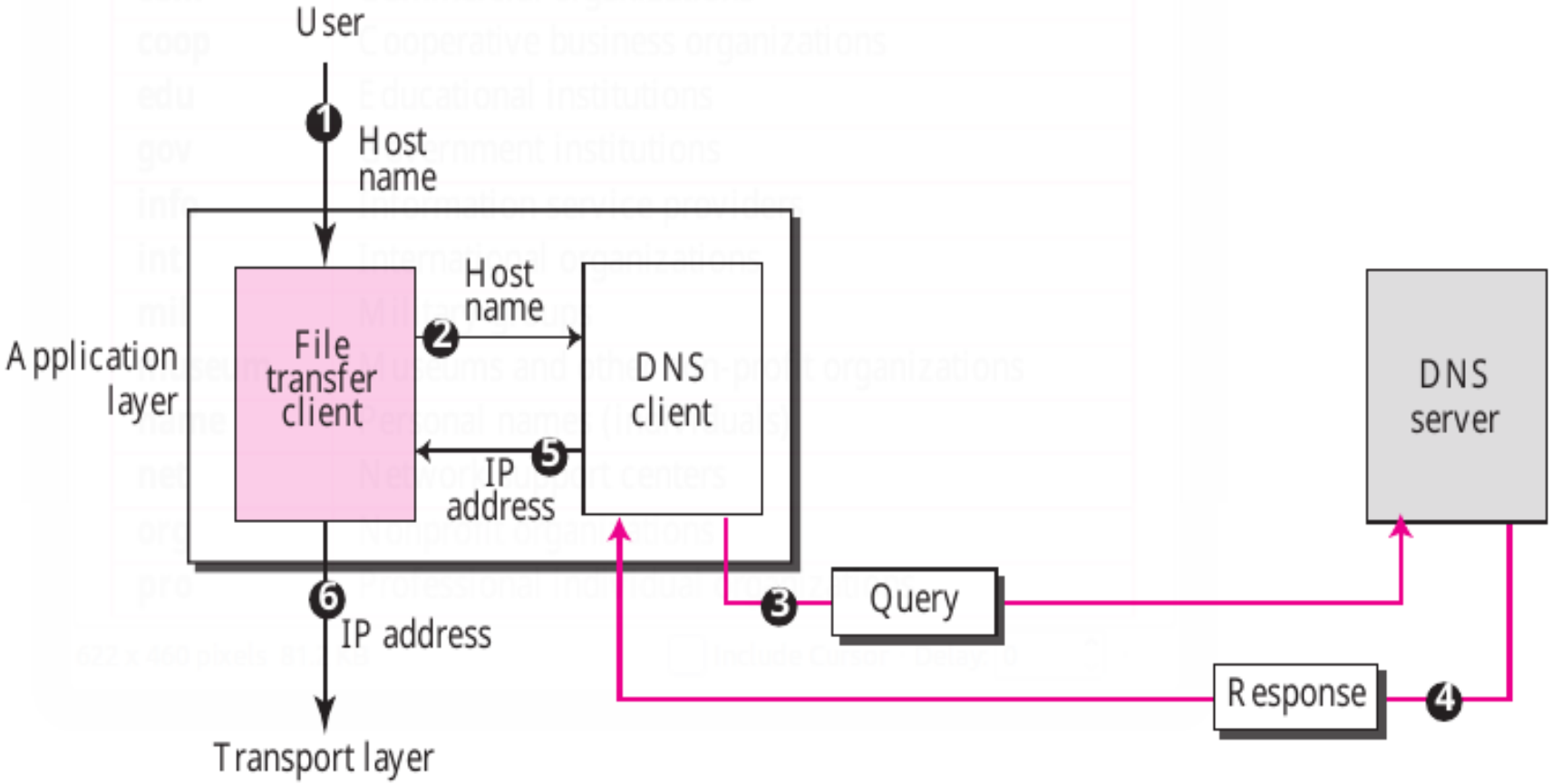
# Domain Name System (DNS)

- DNS maps IP address to names.

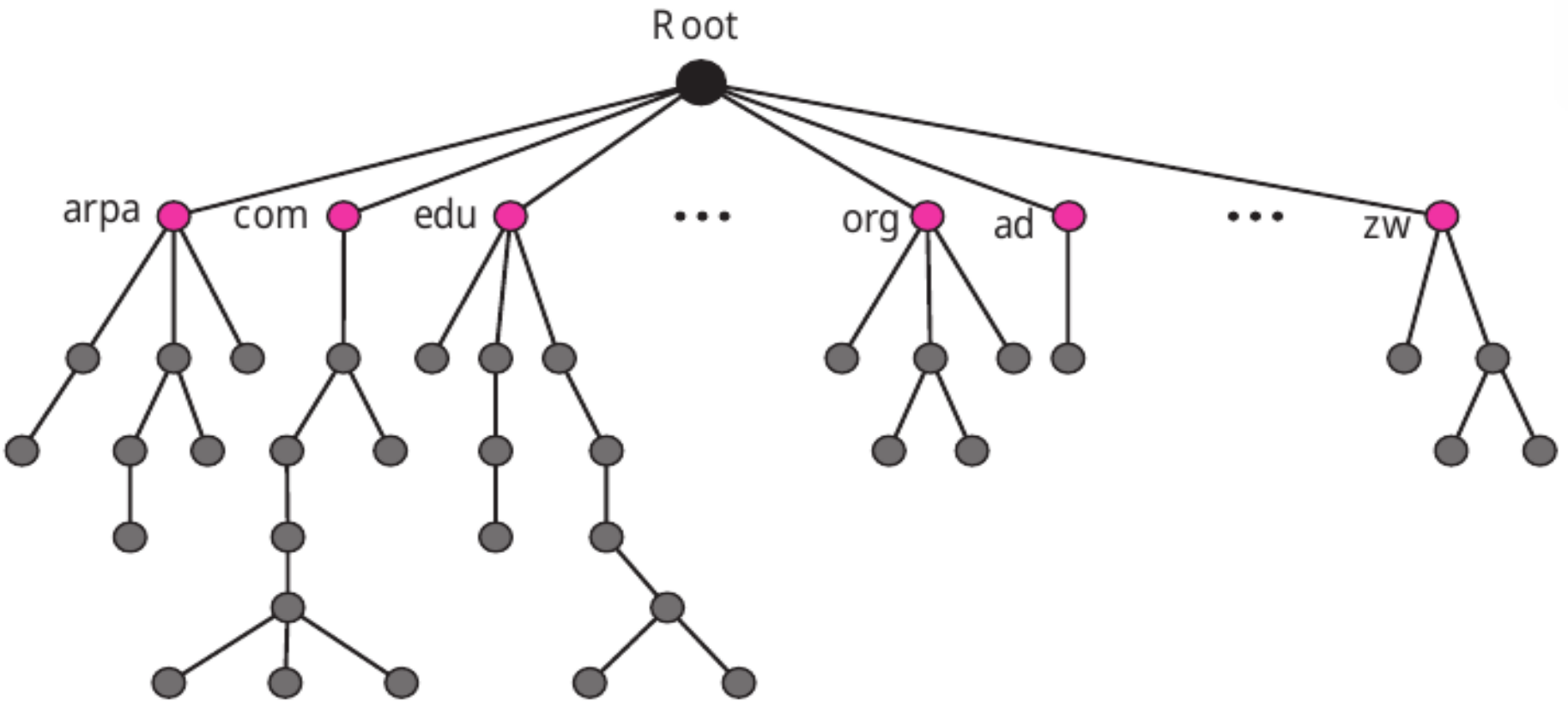
**Name Space:** maps each address to a unique name can be organized in two ways, flat or hierarchical

- Flat Name Space: a name is assigned to an address. A name in this space is sequence of characters without structure.
- Hierarchical Name Space: each name is made of several parts. The first part can define the nature of the organization, the second part can define the name of an organization, the third part can define the department of an organization and so on.

# Purpose of DNS



# Domain Name Space



# Generic Domain Labels

| *Label* | *Description* |
|---|---|
| **aero** | Airlines and aerospace companies |
| **biz** | Businesses or firms (similar to "com") |
| **com** | Commercial organizations |
| **coop** | Cooperative business organizations |
| **edu** | Educational institutions |
| **gov** | Government institutions |
| **info** | Information service providers |
| **int** | International organizations |
| **mil** | Military groups |
| **museum** | Museums and other non-profit organizations |
| **name** | Personal names (individuals) |
| **net** | Network support centers |
| **org** | Nonprofit organizations |
| **pro** | Professional individual organizations |

# Remote Login: TELNET & SSH

- Telnet: is an abbreviation of TErminaL NETwork.

- Telnet enables the establishment of a connection to a remote system in such a way that the local terminal appears to be a terminal at the remote system.

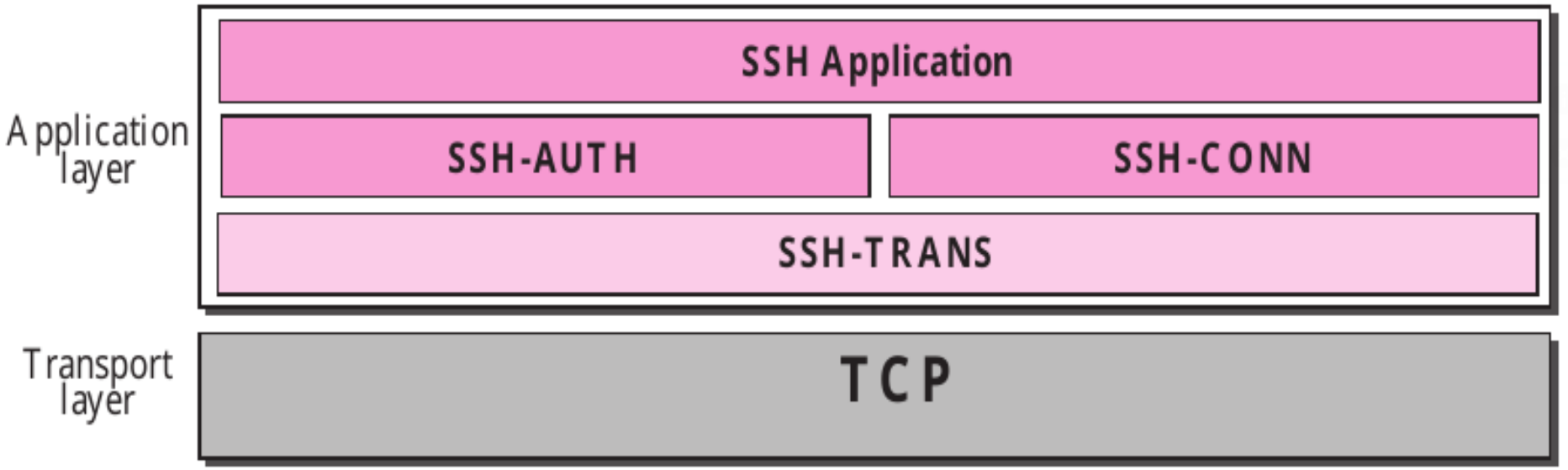
**Security Issues in Telnet:** TELNET suffers from security problems. Although TELNET requires a login name and password (when exchanging text), often this is not enough. A microcomputer connected to a broadcast LAN can easily eavesdrop using snooper software and capture a login name and the corresponding password (even if it is encrypted).

# Secure Shell (SSH)

- SSH is a remote login program uses TCP as the underlying transport protocol, and provides security and more services.

- There are two versions of SSH: SSH-1 and SSH-2 which are totally incompatible.

# Component of SSH



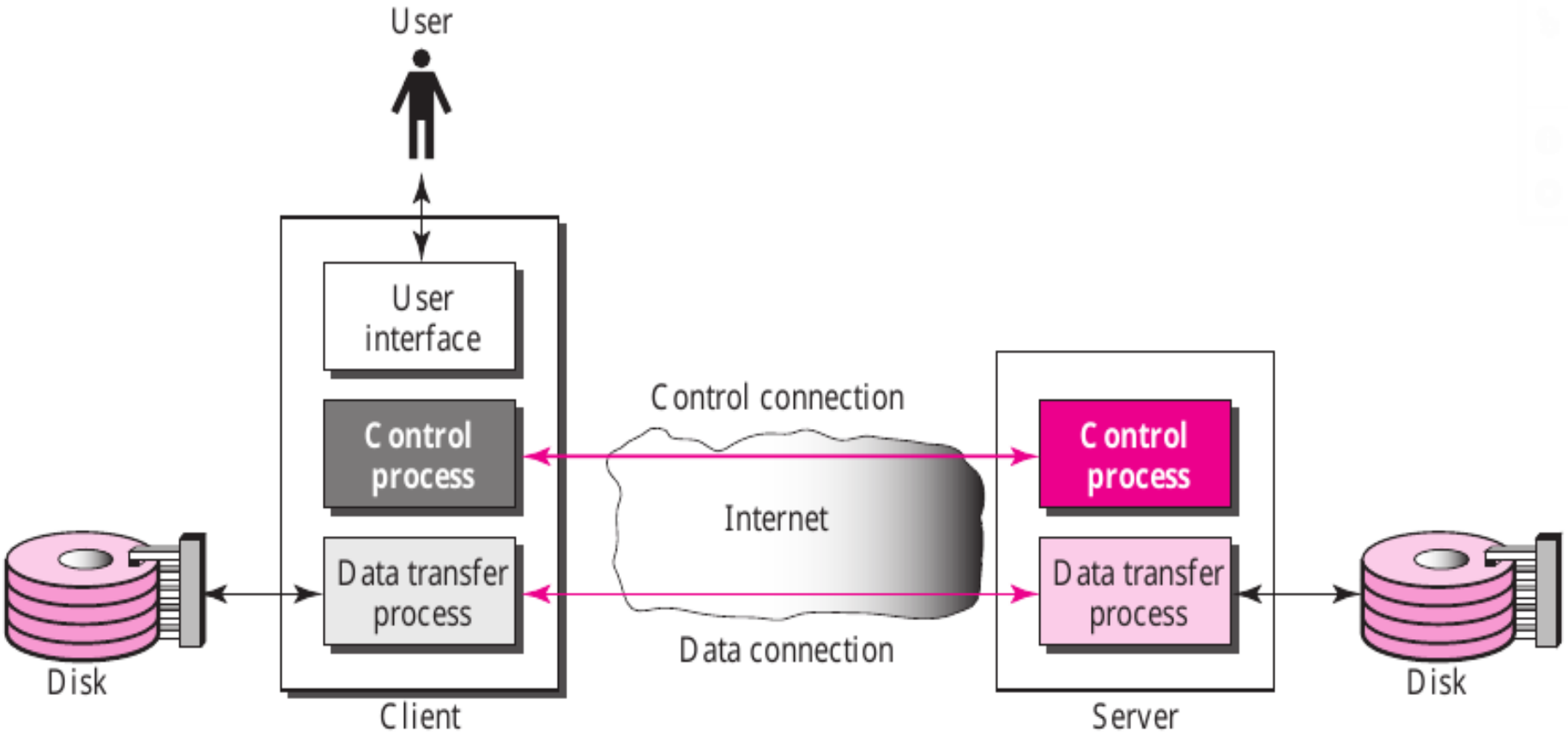
# Component of SSH Continue

- SSH Transport-Layer Protocol (SSH-TRANS): SSH first uses a protocol that creates a secured channel on the top of TCP. This new layer is an independent protocol referred to as SSH-TRANS.
- SSH Authentication Protocol (SSH-AUTH): After a secure channel is established between the client and the server and the server is authenticated for the client, SSH can call another software that can authenticate the client for the server.
- SSH Connection Protocol (SSH-CONN): One of the services provided by the SSH-CONN protocol is to do multiplexing. SSH-CONN takes the secure channel established by the two previous protocols and lets the client create multiple logical channels over it.
- SSH Applications: After the connection phase is completed, SSH allows several application programs to use the connection.

# File Transfer: FTP and TFTP

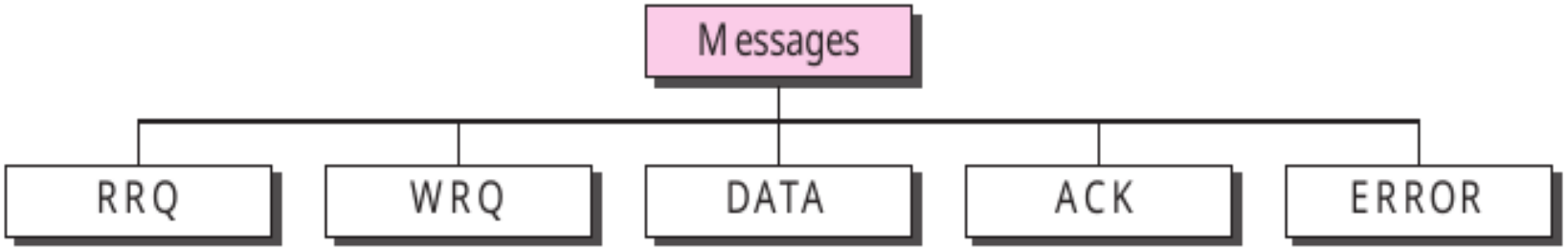
- FTP: is the standard mechanism provided by TCP/IP for copying a file from one host to another
- FTP differs from other client-server applications in that it establishes two connections between the hosts. One connection is used for data transfer, the other for control information (commands and responses).
- FTP uses two well-known TCP ports: Port 21 is used for the control connection, and port 20 is used for the data connection.

# File Transfer Protocol



# Trivial File Transfer Protocol (TFTP)

- TFTP can read or write a file for the client. Reading means copying a file from the server site to the client site. Writing means copying a file from the client site to the server site.
- TFTP uses the services of UDP on the well-known port 69.



# TFTP Message Categories

- RRQ: The read request (RRQ) message is used by the client to establish a connection for reading data from the server.
- WRQ: The write request (WRQ) message is used by the client to establish a connection for writing data to the server.
- DATA: The data (DATA) message is used by the client or the server to send blocks of data.
- ACK: The acknowledge (ACK) message is used by the client or server to acknowledge the receipt of a data block.
- Error: The ERROR message is used by the client or the server when a connection cannot be established or when there is a problem during data transmission.

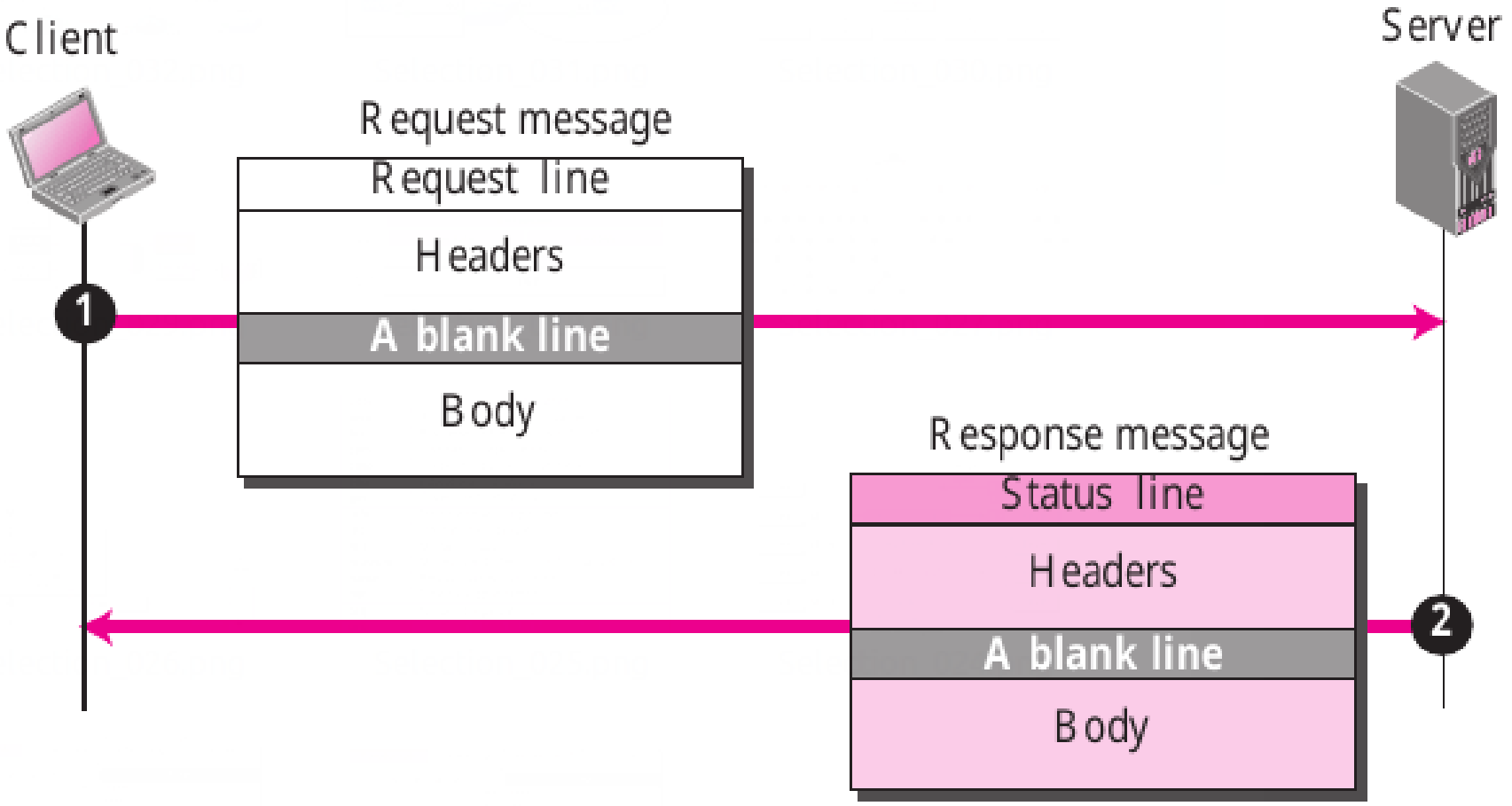
# World Wide Web and HTTP

- The WWW today is distributed client-server service, in which client using a browser can access a service using a server.
- The service provided is distributed over many locations called sites.
- Each site holds one or more documents, referred to as Web pages.
- Each Web page, however, can contain some links to other Web pages in the same or other sites.
- Web page can be simple or composite.
  - A simple Web page has no link to other Web pages;
  - A composite Web page has one or more links to other Web pages.

# Hypertext Transfer Protocol (HTTP)

- The Hypertext Transfer Protocol (HTTP) is a protocol used mainly to access data on the World Wide Web.
- HTTP uses the services of TCP on well-known port 80.
- HTTP Transaction: uses the services of TCP, HTTP itself is a stateless protocol, which means that the server does not keep information about the client. The client initializes the transaction by sending a request. The server replies by sending a response.
  - Request Message: A request message consists of a request line, a header, and sometimes a body.
  - Request Line: There are three fields in this line, methods, URL and Version

# HTTP Transaction



# HTTP Security

- HTTP can be run over the Secure Socket Layer (SSL).
- HTTP is referred to as HTTPS.
- HTTPS provides confidentiality, client and server authentication, and data integrity.

# End of Chapter 6
# :)