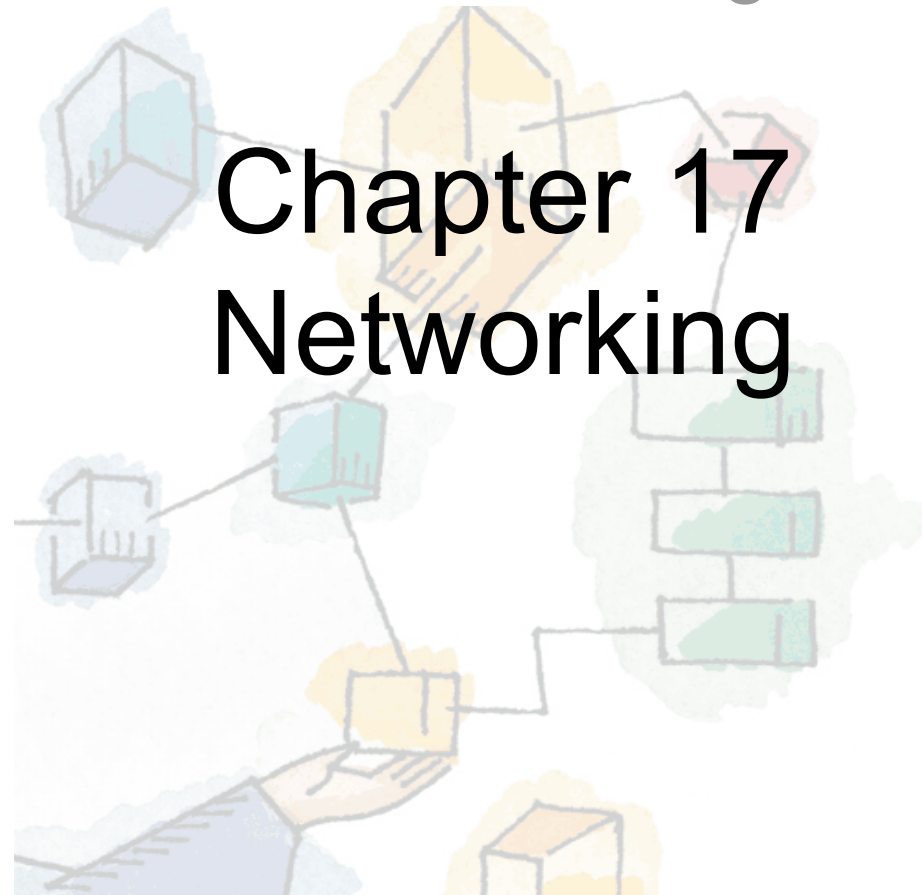
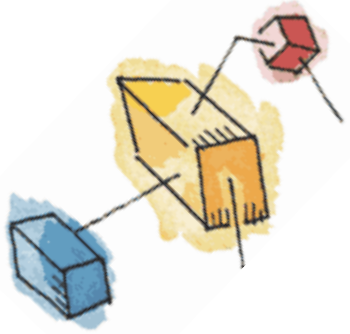


*Operating Systems:  
Internals and Design Principles, 6/E*  
William Stallings



# Chapter 17 Networking

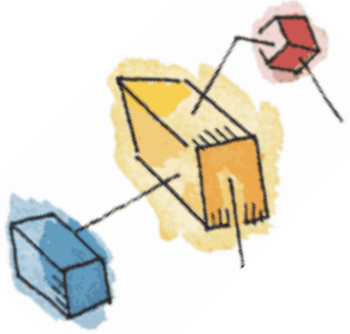
Patricia Roy  
Manatee Community College, Venice, FL  
©2008, Prentice Hall



# Distributed Capabilities

- Communications architectures
  - Software that supports a group of networked computers
- Network operating system
  - Each computer has its own private operating system
- Distributed operating system
  - Common operating system shared by a network of computers

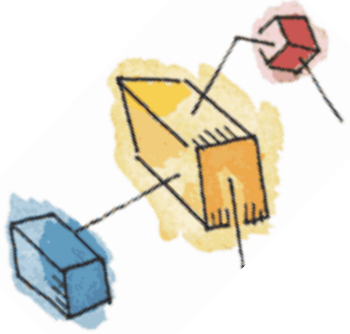




# The Need for a Protocol Architecture

- Computer communications
  - Exchange of information between computers for the purpose of cooperative action
- Computer network
  - When two or more computers are interconnected via a communication network





# Two Concepts

- Protocol
  - Used for communication between entities in different systems
- Protocol architecture
  - Broken into subtasks, each of which is implemented separately





# Key Elements of a Protocol

- Syntax
- Semantics
- Timing



# File Transfer

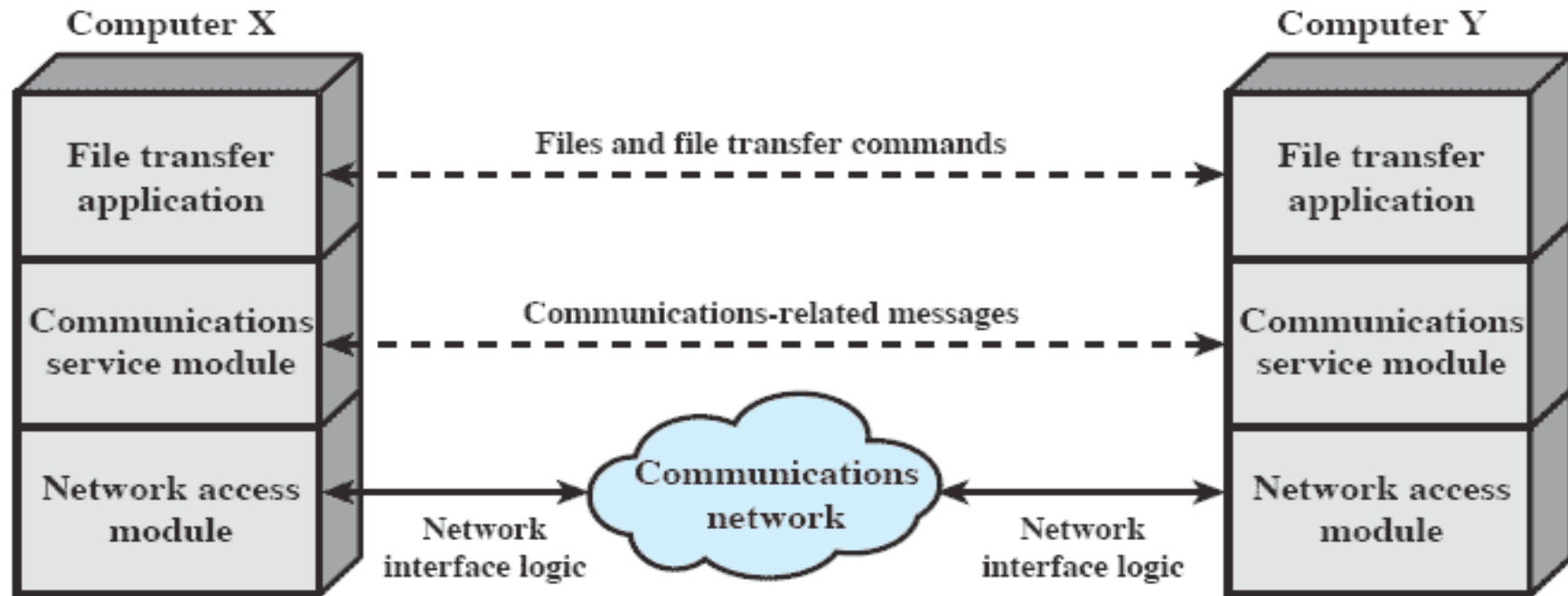
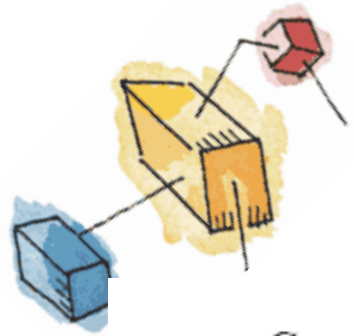
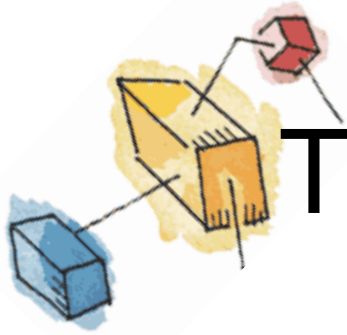


Figure 17.1 A Simplified Architecture for File Transfer



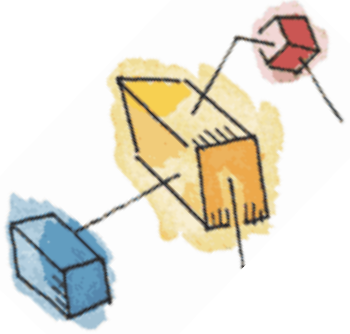


# TCP/IP Protocol Architecture

- Five relatively independent layers
  - Physical
  - Network access
  - Internet
  - Host-to-host, or transport
  - Application

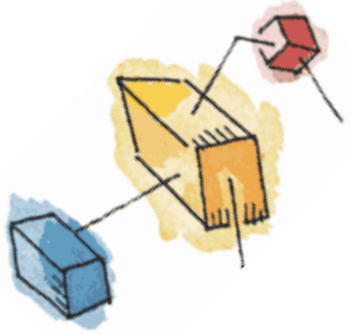


# Physical Layer



- Specifying
  - the characteristics of the transmission medium
  - Nature of the signals
  - Data rate

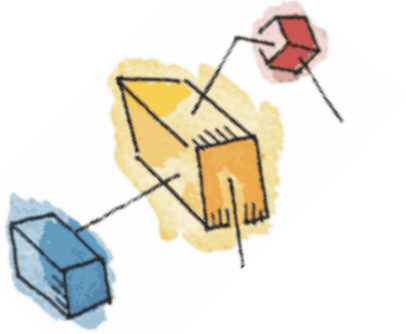




# Network Access Layer

- Concerned with the exchange of data between an end system and the network
- Different standards
  - Circuit switching
  - Packet switching (frame relay)
  - LANs (Ethernet)

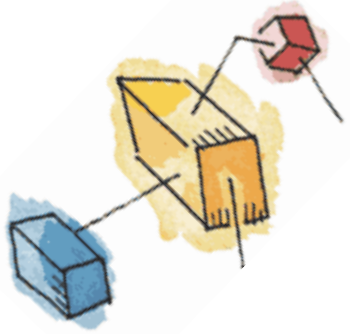




# Internet Layer

- Procedures for data to traverse different networks
- Implemented in the end systems and routers

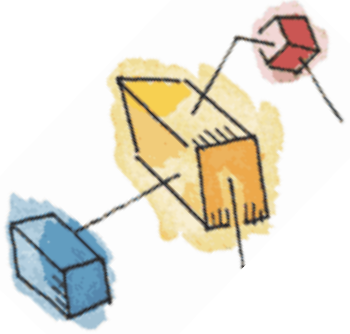




# Transport Layer

- Ensures all data arrives at the destination and in the order sent
- TCP

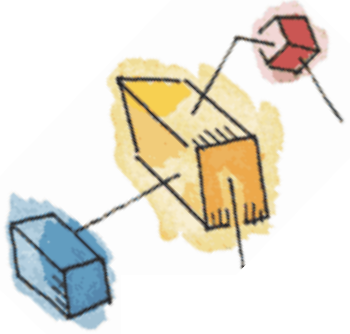




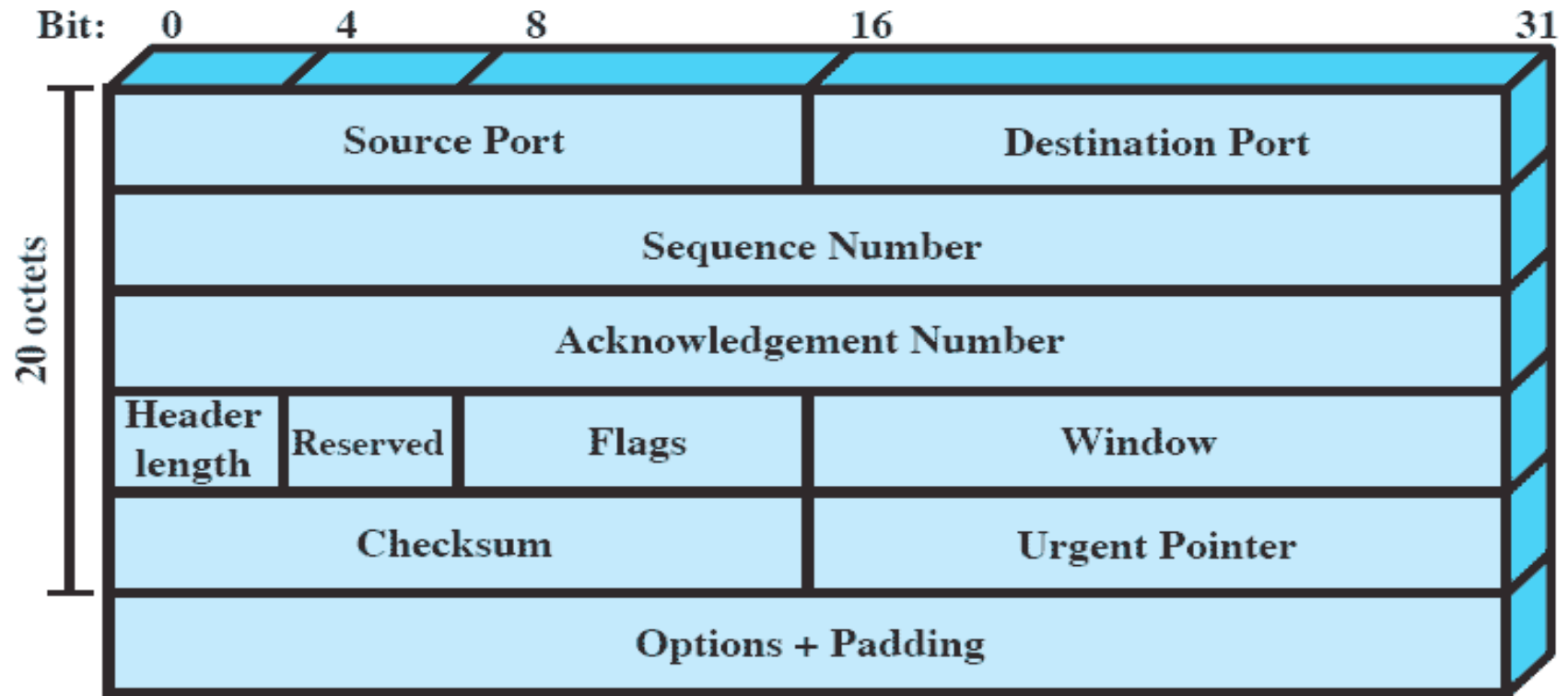
# Application Layer

- Supports various user application
- Example: file transfer





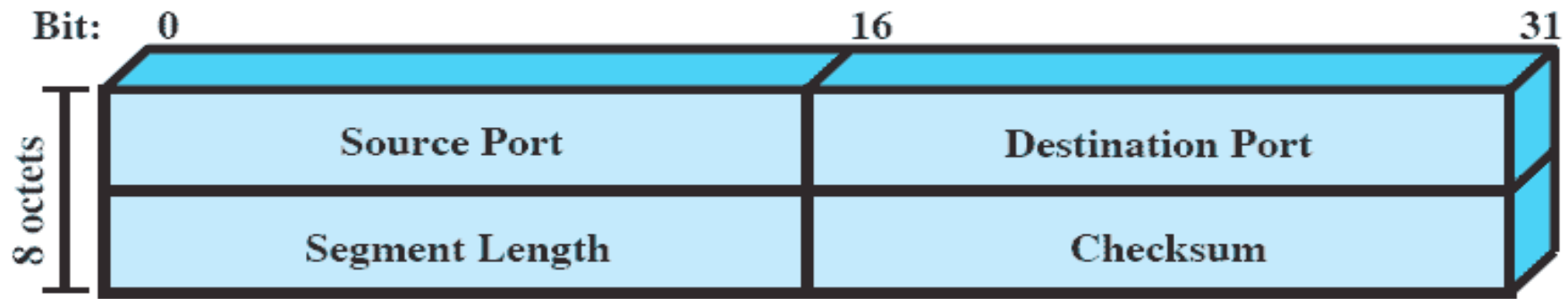
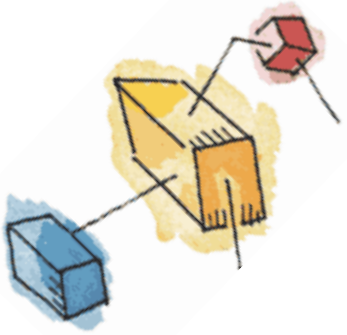
# TCP Header



(a) TCP Header

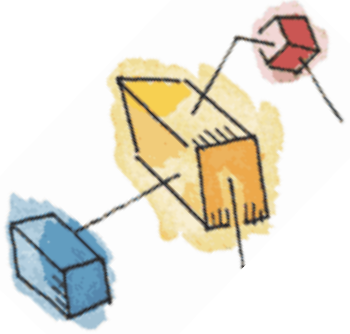


# UDP Header



(b) UDP Header

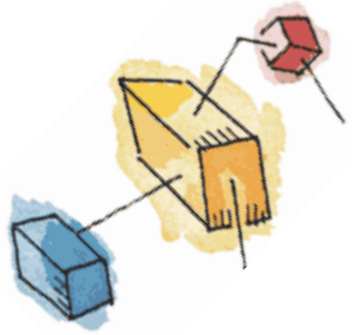




# IP Datagram

- IP appends a header of control information
- Example: destination host address

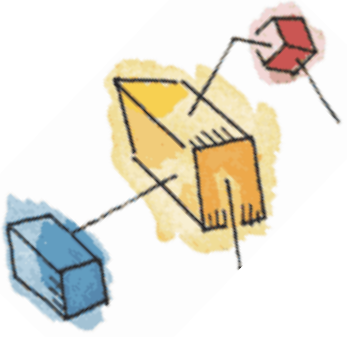




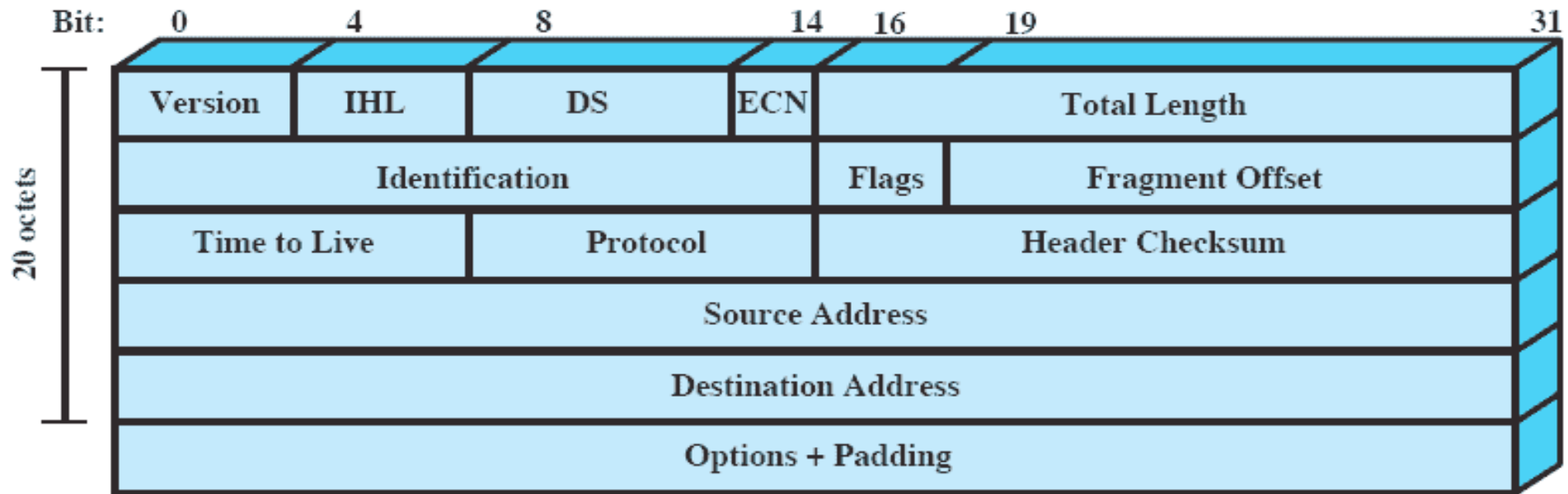
# IPv6

- Provides enhancements over existing IP
- Designed to accommodate higher speeds of a mix of data streams, graphic and video
- Provides more addresses
- Includes 128-bits for addresses
  - IP uses 32-bit address





# IPv4 Header

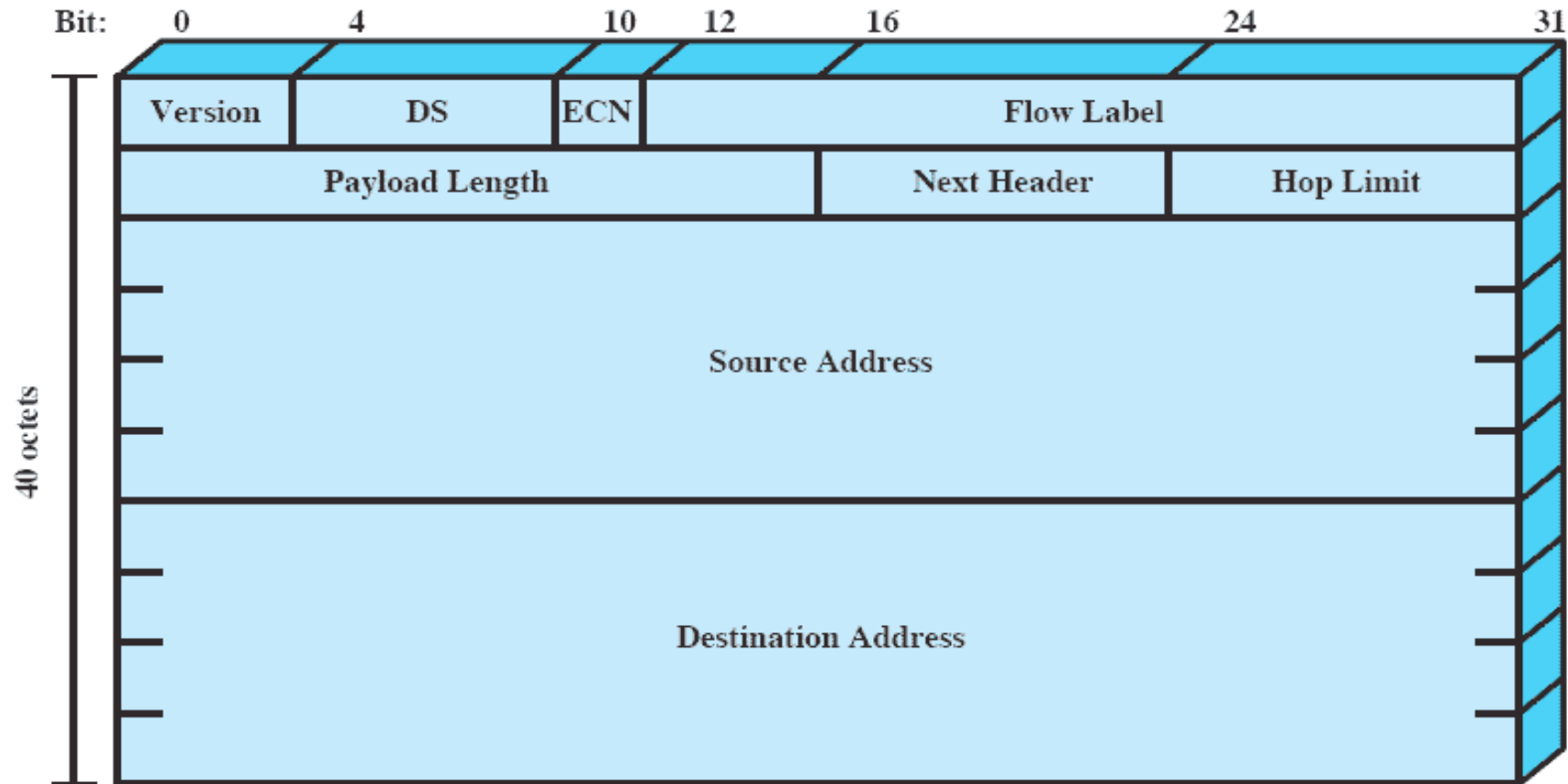


(a) IPv4 Header





# IPv6 Header



(b) IPv6 Header

DS = Differentiated services field

ECN = Explicit congestion notification field

Note: The 8-bit DS/ECN fields were formerly known as the Type of Service field in the IPv4 header and the Traffic Class field in the IPv6 header.



# TCP/IP operation

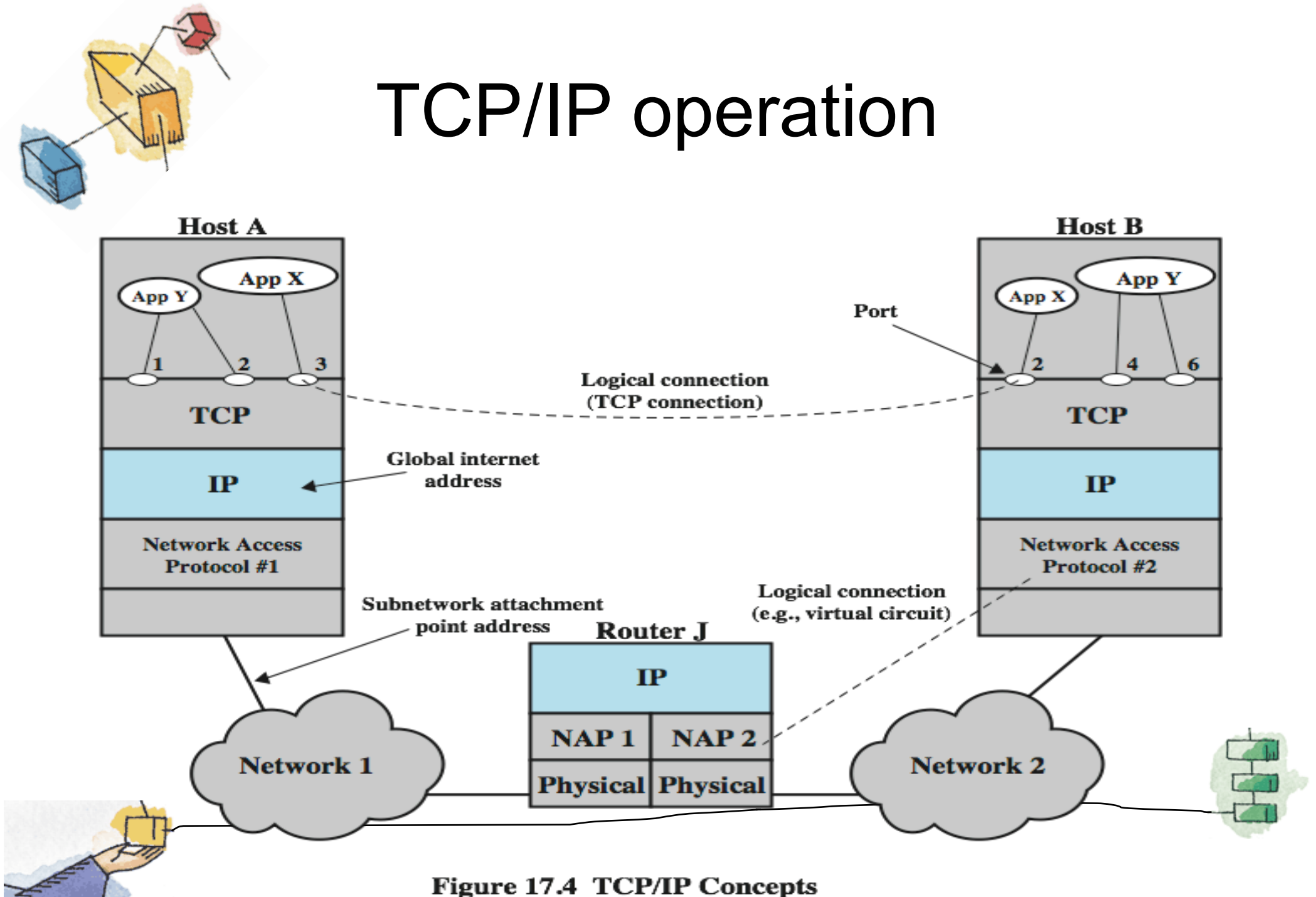
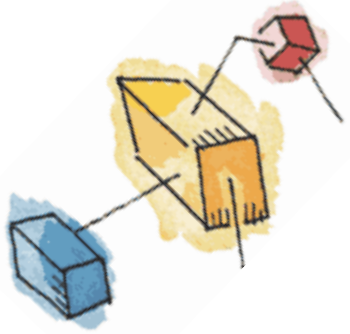


Figure 17.4 TCP/IP Concepts



# TCP/IP Applications

- Simple mail transfer protocol (SMTP)
- File transfer protocol (FTP)
- TELNET



# Protocol Data Units

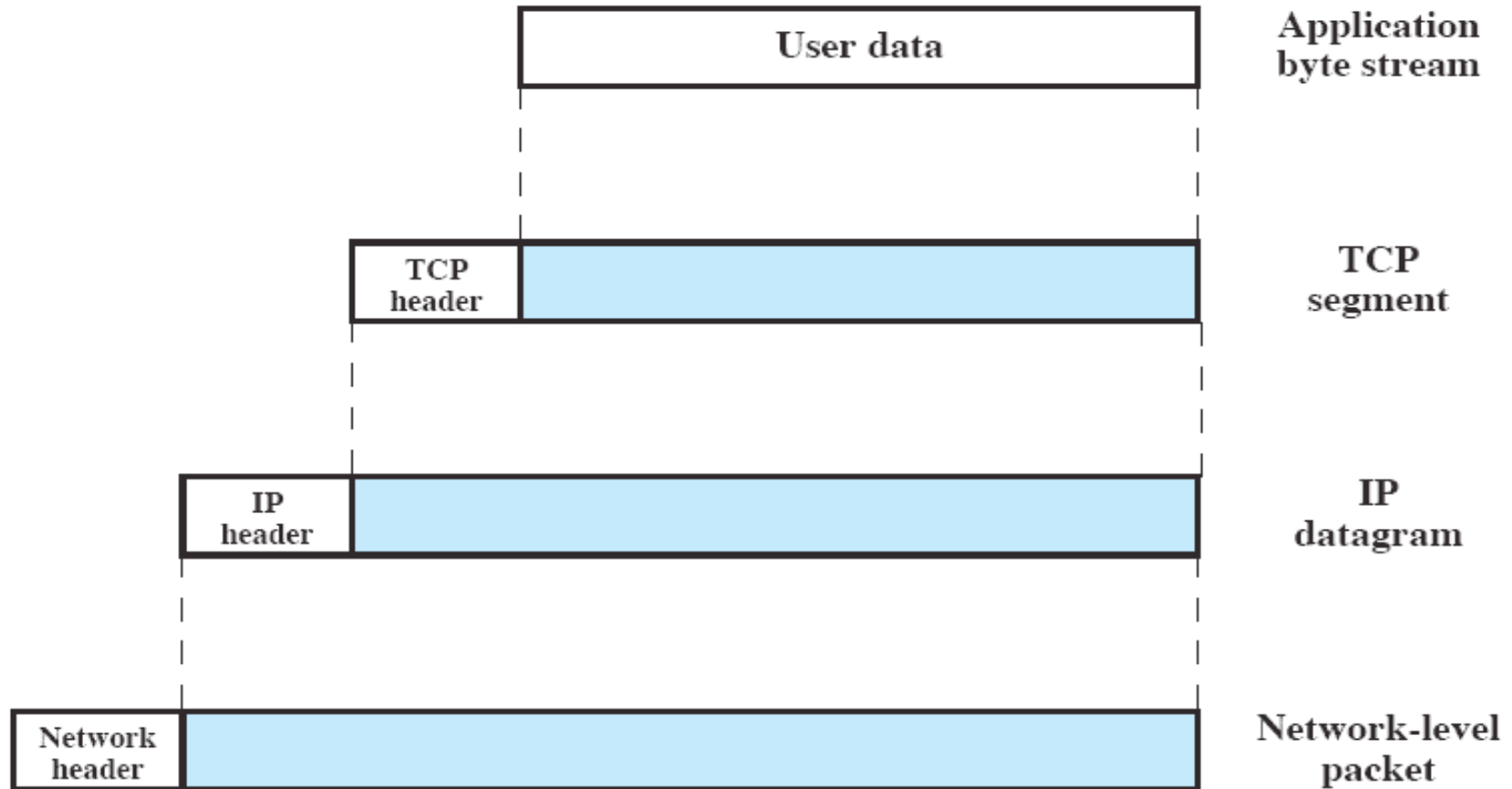
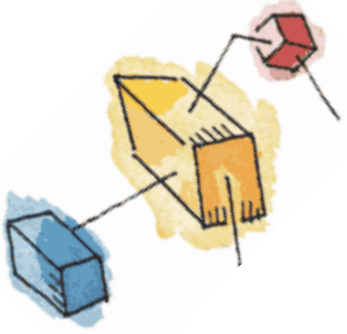


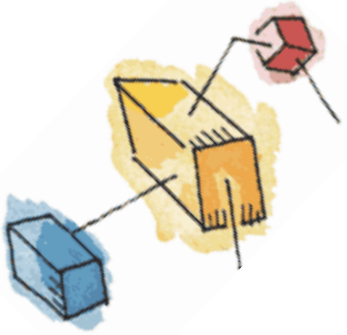
Figure 17.5 Protocol Data Units (PDUs) in the TCP/IP Architecture

# Sockets



- Enable communication between a client and server
- Endpoint in communication



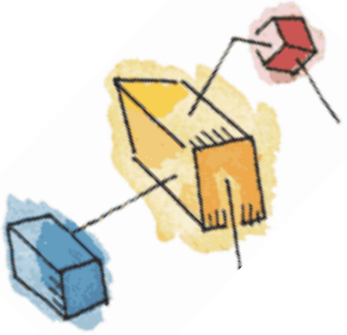


# Windows Socket

- Based on Berkeley specification
- Provides generic access to interprocess communication services

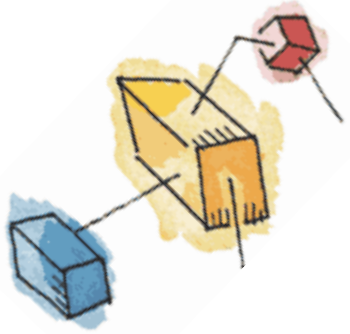


# Socket



- Used to define an API
- Generic communication interface for writing programs that use TCP or UDP

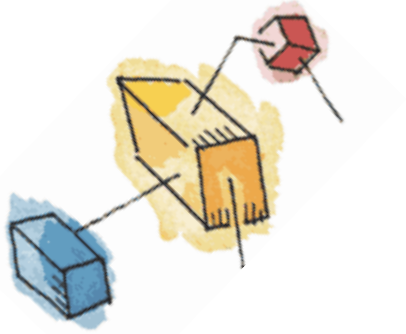




# Types of Sockets

- Stream sockets
  - Use TCP
  - Reliable data transfer

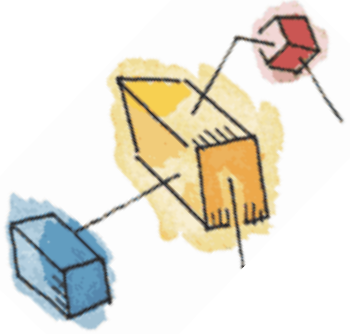




# Types of Sockets

- Datagram sockets
  - Use UDP
  - Delivery is not guaranteed

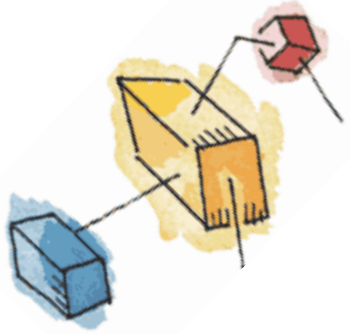




# Types of Sockets

- Raw sockets
  - Allow direct access to lower layer protocols

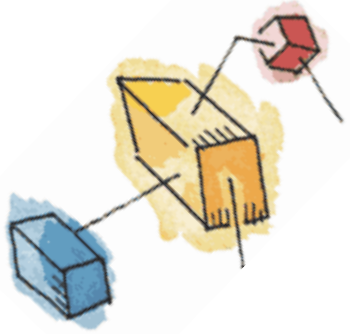




# Socket Setup

- socket() command
- Three parameters
  - Protocol family is always PF\_INET for TCP/IP
  - Type specifies whether stream or datagram
  - Protocol specifies either TCP or UDP

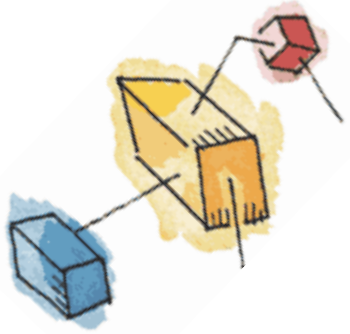




# Socket Connection

- One side is client
  - Requests connection
- Other side is server





# Socket System Calls

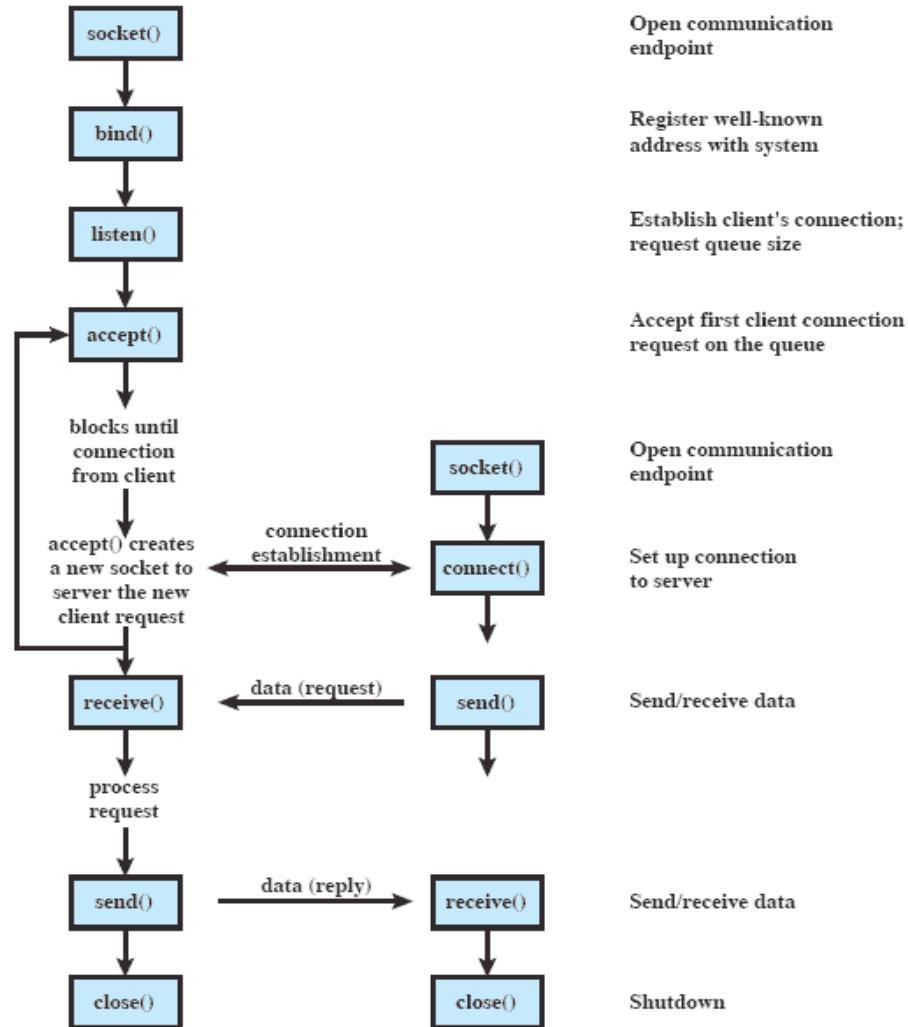
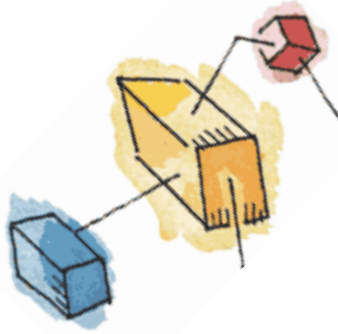


Figure 17.6 Socket System Calls for Connection-Oriented Protocol





# Linux Kernel Components

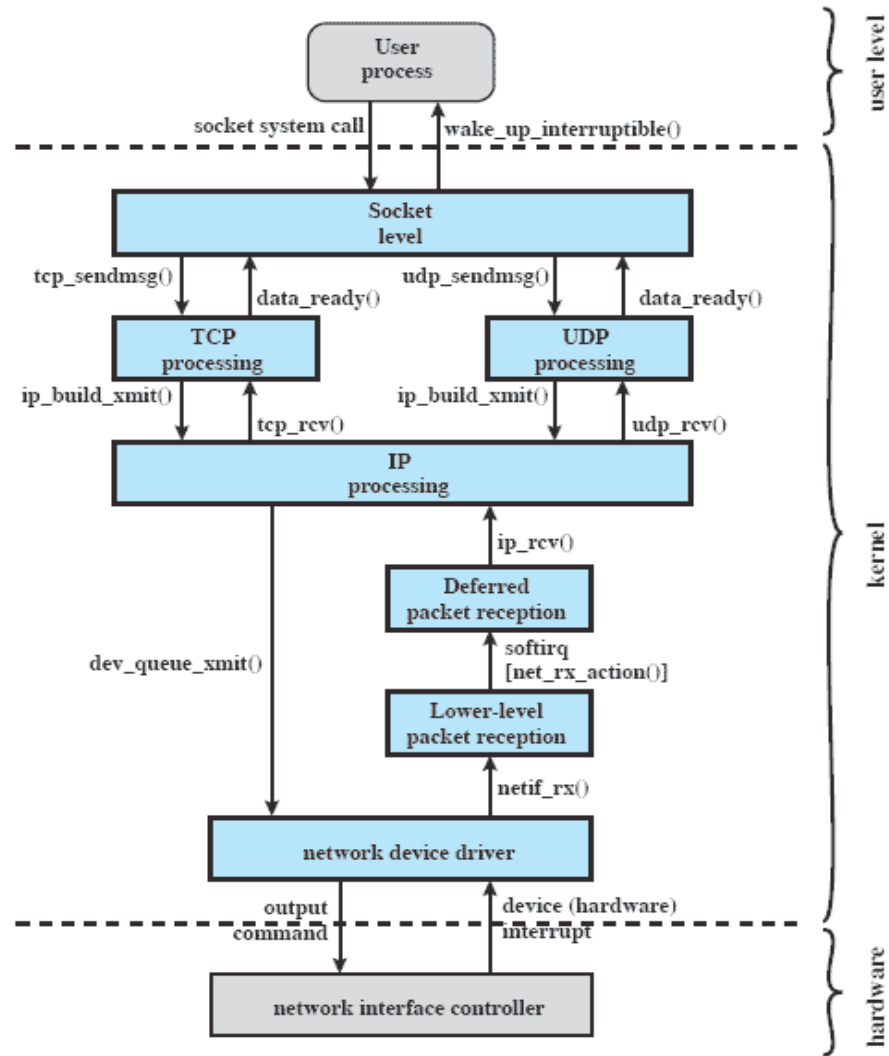


Figure 17.7 Linux Kernel Components for TCP/IP Processing