


Chapter 2 - Part 1

A thick, horizontal yellow brushstroke with a textured, painterly appearance, spanning across the width of the slide below the chapter title.

The TCP/IP Protocol: The Language of the Internet

Protocols



- ◆ A protocol is a “language” or set of rules that two or more computers use to communicate

Protocol Analogy: Phone Call



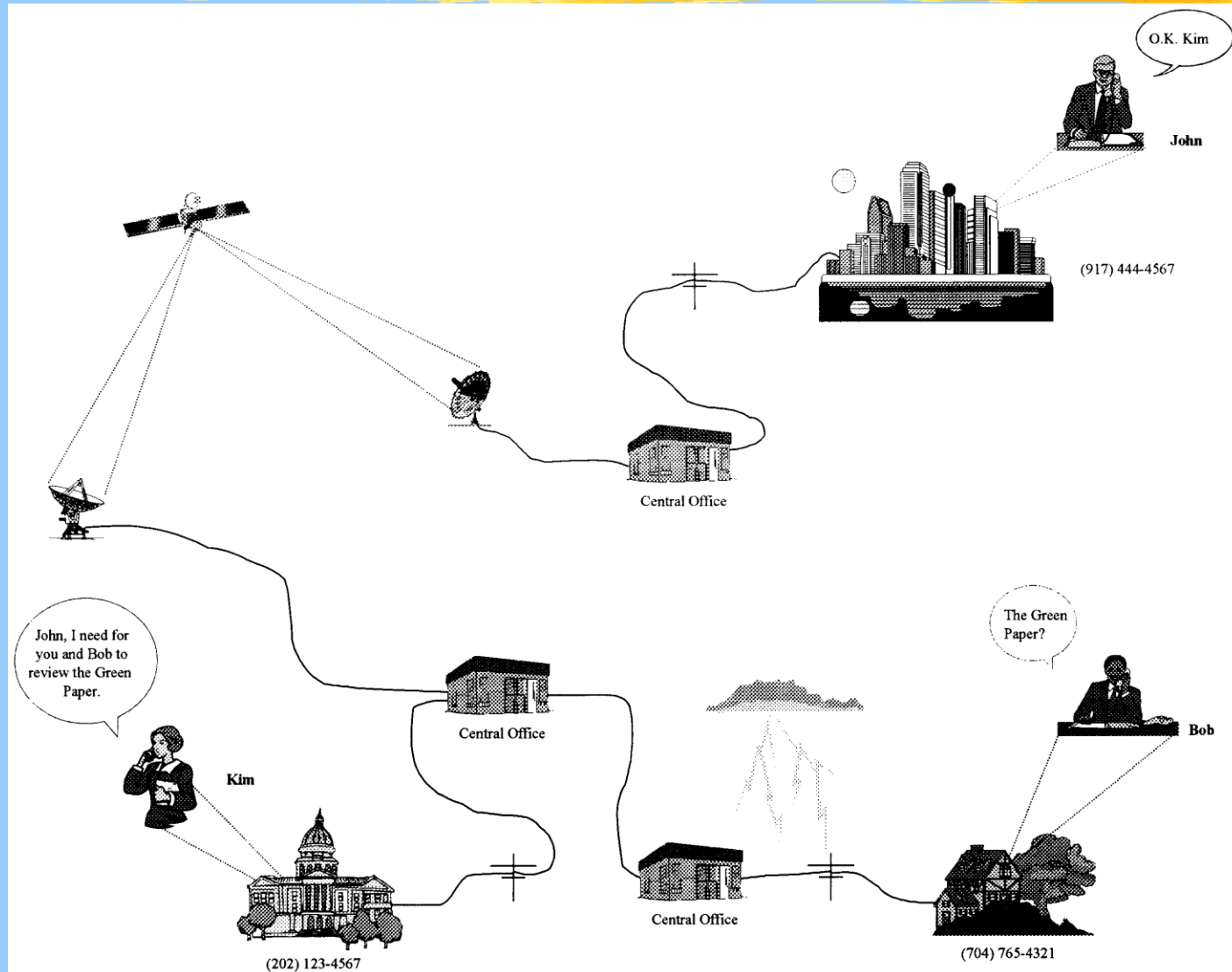
- ◆ Parties in phone call are located at physical addresses
- ◆ Phone numbers are like logical addresses
 - ◆ Area code and prefix subdivide the network
- ◆ Physical addresses are fixed, logical addresses can be reassigned
- ◆ In computer networking
 - ◆ Physical addresses are Media Access Control (MAC) addresses
 - ◆ Logical addresses are Internet Protocol (IP) addresses

Protocol Analogy: Phone Call



- ◆ Conversation has its own protocols
 - ◆ Listen while others are talking
 - ◆ Repeat a sentence if other person does not hear it clearly

Protocol Analogy: Phone Call



Layered Network Protocols



- ◆ **Communication between computers is a difficult problem**
- ◆ **Protocols provide the structure and rules for computers to communicate**
- ◆ **Specialized network communication software provides an interface between software applications and the network interface card (NIC)**
- ◆ **Communication software is divided into pieces in a layered fashion**
 - ◆ **Creates modular structure**

Layered Network Protocols



- ◆ Each piece implements a separate protocol that when combined with the other pieces, forms the complete protocol *suite* or *stack*
- ◆ Each layer depends only on the services provided by lower layers
- ◆ Each protocol layer can be developed and tested independently of the other layers
- ◆ It only has to interface properly with the layers above and below it

OSI Model



- ◆ **Open-System Interconnection (OSI) model is a reference for designing protocol standards**
- ◆ **Developed by International Organization for Standardization (ISO)**
- ◆ **Has 7 layers**

OSI Model Layers



◆ Layer 7: Application

- ◆ Programs with which users interact
- ◆ Often uses a graphical user interface (GUI)
- ◆ Example: WWW browser

◆ Layer 6: Presentation

- ◆ Presents information to the application layer in formatted structure
- ◆ Handles encryption and compression

◆ Layer 5: Session

- ◆ Manages sessions between hosts

OSI Model Layers



◆ Layer 4: Transport

- ◆ Provides reliable end-to-end communication between hosts
- ◆ Handles packet sequencing, flow control, error recovery
- ◆ Examples: Transmission Control Protocol (TCP) and Sequenced Packet Exchange (SPX)

OSI Model Layers



◆ Layer 3: Network

- ◆ Provides end-to-end delivery of information between networks
 - ◆ Reliability not guaranteed
 - ◆ Uses network addresses
- ◆ Permits creation of internetworks (internets)
- ◆ Examples: Internet Protocol (IP) and Internetwork Packet Exchange (IPX)
- ◆ Output of this layer is a *packet*

OSI Model Layers



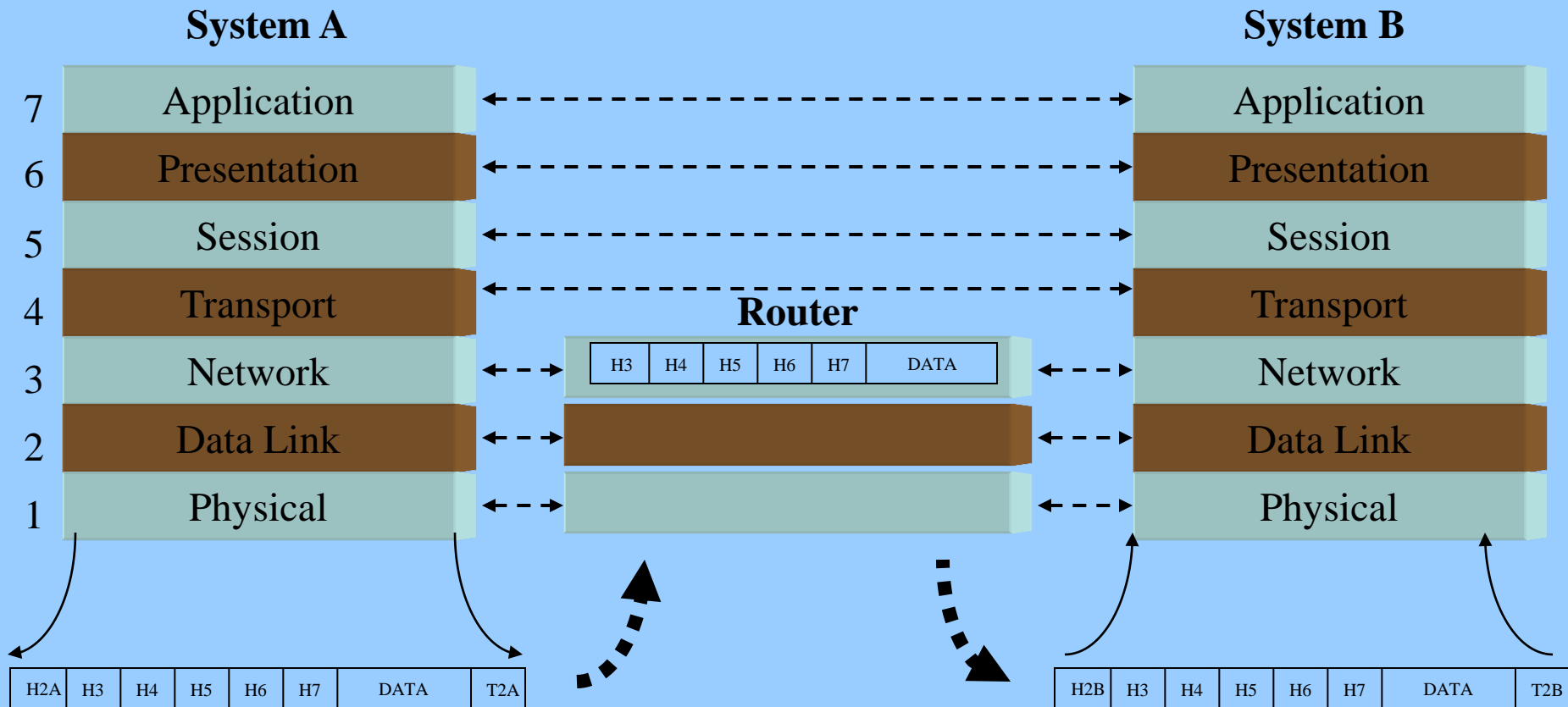
◆ Layer 2: Data Link

- ◆ Produces encoding pattern for physical layer
- ◆ Handles some flow control, error control
- ◆ Addresses for this layer are known as MAC addresses
- ◆ Output of this layer is a *frame*

◆ Layer 1: Physical

- ◆ Electrical and mechanical interface to the network
- ◆ Transmits and receives a bit stream

Communication and the OSI Model



Protocol Data Units (PDU)



- ◆ The combination of data and header information from a layer in a protocol suite is a PDU

OSI Model PDUs

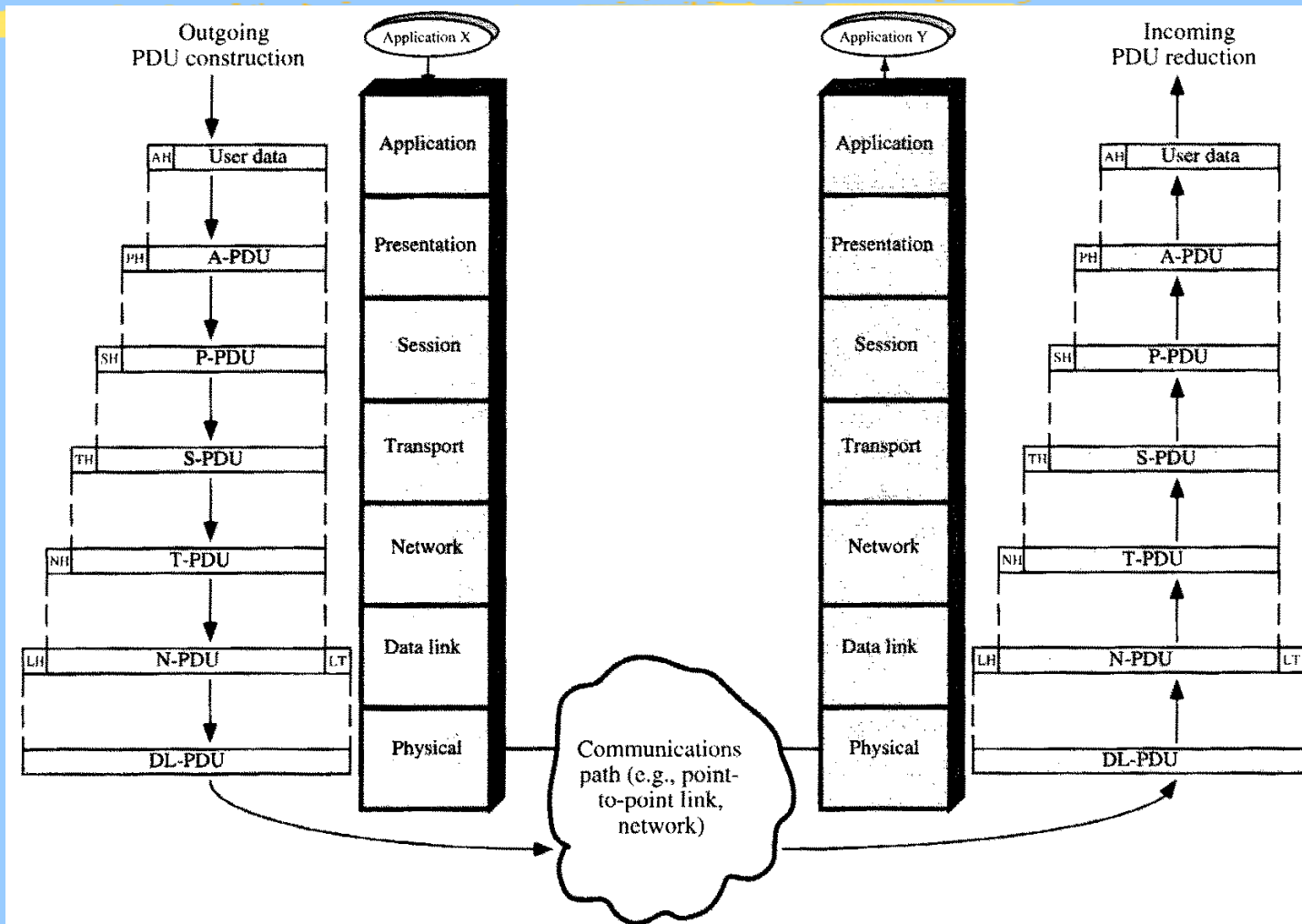


Figure 2.6 The OSI Environment

Peer Layer Interaction



- ◆ Consider two entities communicating
- ◆ Layer 7 on Host A communicates , or interacts, only with Layer 7 on Host B
- ◆ Same with layers 4, 5, and 6
- ◆ Layers 1, 2, and 3 interact only with the same layers on the Host B or the next-hop entity
- ◆ Result is Peer-to-Peer interaction between model layers

PDU Flow



- ◆ PDUs flow up and down the model depending on their direction
- ◆ Example
 - ◆ Application on Host A requires information from application on Host B
 - ◆ Host A application layer appends a header to its data and sends the application layer PDU to the presentation layer
 - ◆ Presentation layer appends its header and sends presentation layer PDU to session layer
 - ◆ Process continues until physical layer appends its header, creating physical layer PDU, and sends PDU out on network

PDU Flow



- ◆ Physical layer on Host B receives Host A's physical layer PDU
- ◆ Host B physical layer strips off the Host A physical layer header and passes the resulting data link PDU up to the Host B data link layer
- ◆ Process continues until Host B's application layer finally receives Host A's application layer PDU
- ◆ Process reverses when Host B replies

Layer Interaction - Services



- ◆ Protocol standards must define the *services* provided by each layer to the layers above and below

OSI Architecture and Services

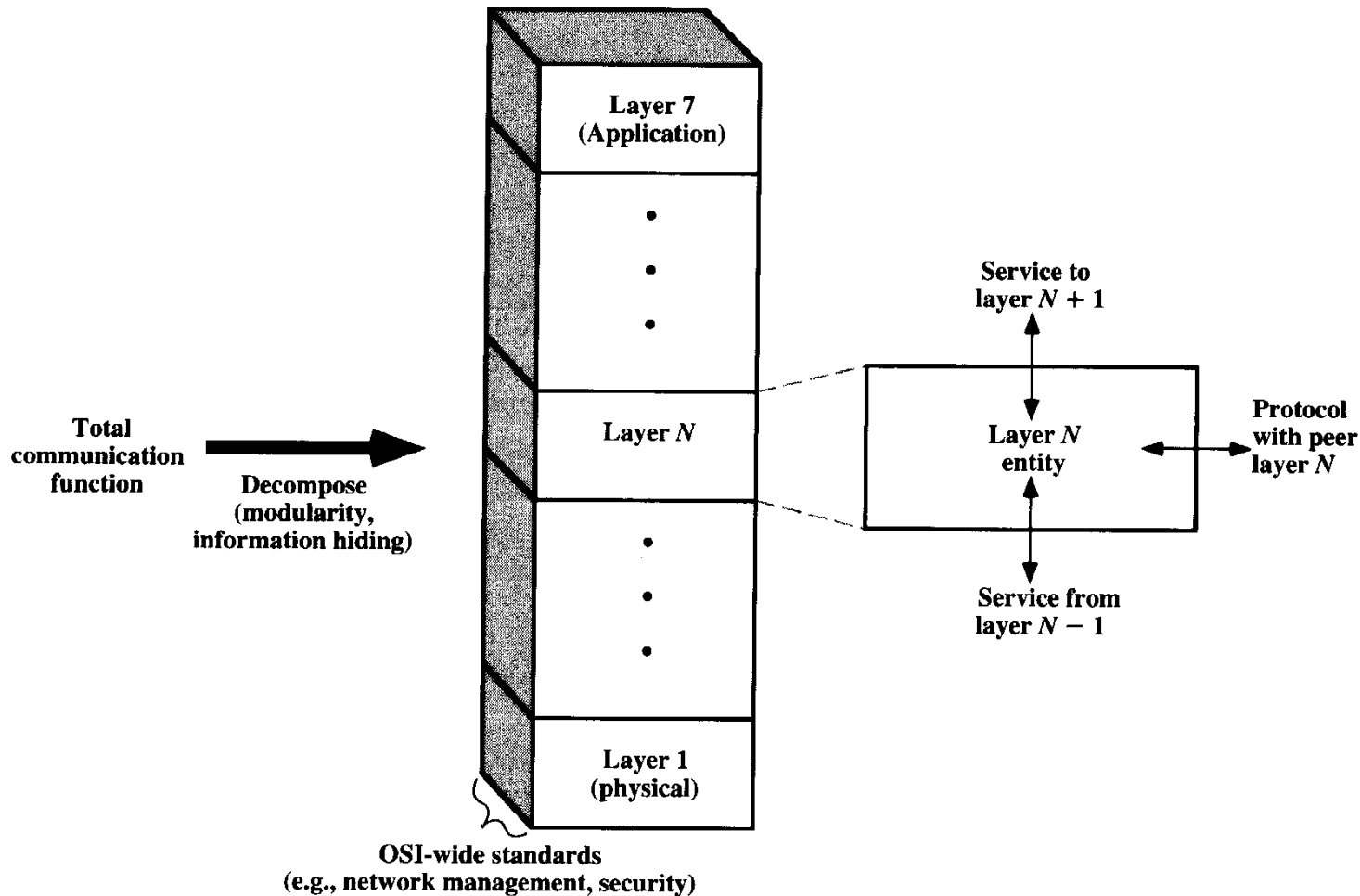


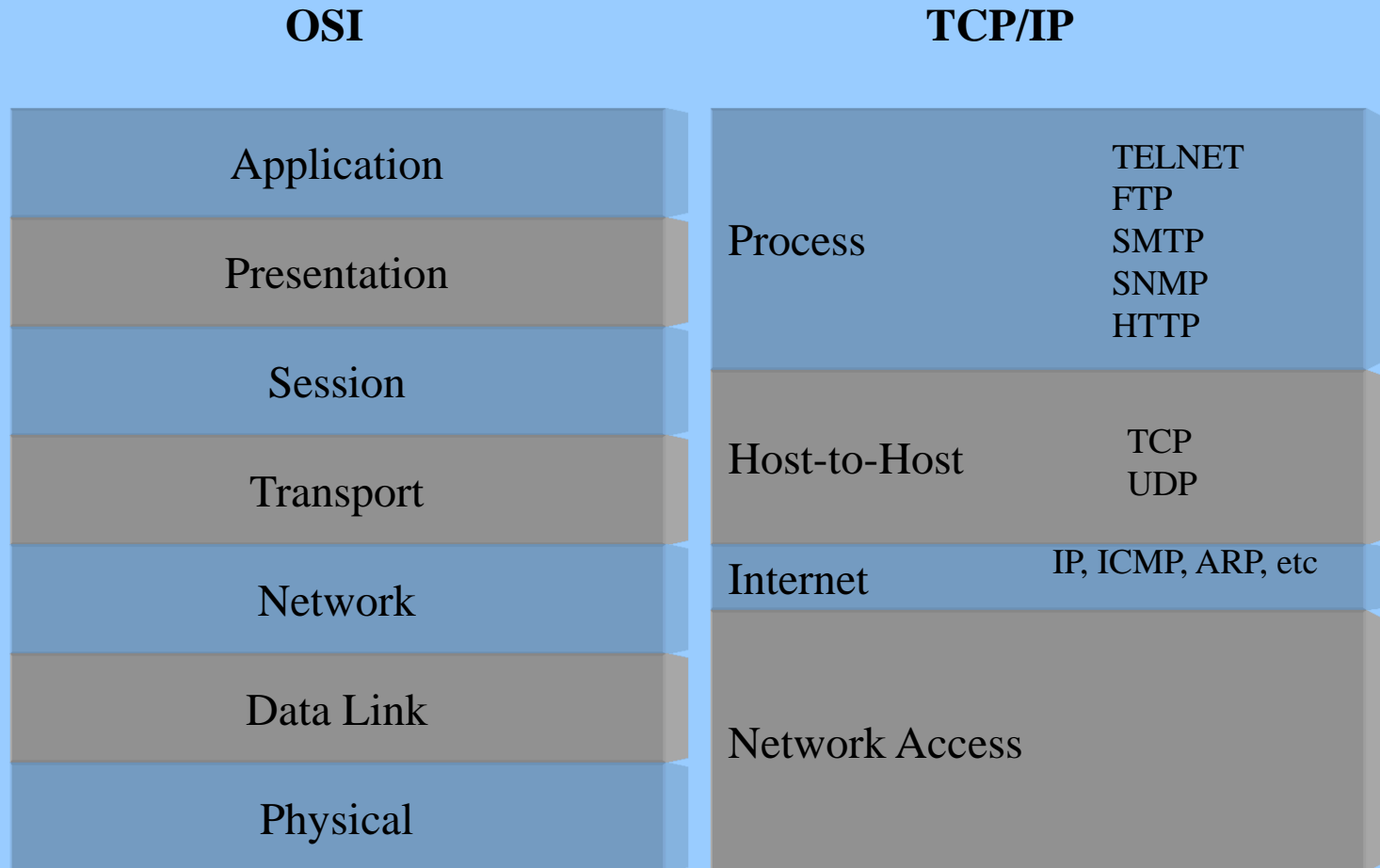
Figure 2.7 The OSI Architecture as a Framework for Standardization

TCP/IP Internet Protocols



- ◆ TCP - Transmission Control Protocol.
- ◆ IP - Internet Protocol.
- ◆ Most popular protocol suite used for interconnecting networks.
 - ◆ Others include IPX/SPX, VINES, AppleTalk.
- ◆ Was the internet protocol for ARPANET.

TCP/IP Protocol Stack vs. OSI Model



TCP/IP Protocol Stack vs. OSI Model



- ◆ **Some authors do the following:**
 - ◆ **Call the TCP/IP Process layer the Application layer**
 - ◆ **Call the Host-to-Host layer the Transport layer**
 - ◆ **Split Network Interface layer into the Network Access and Physical layers (creates a 5-layer model)**

TCP/IP Layers



4. Process (Application)

- ◆ Provides communication between processes or applications on separate hosts
- ◆ Combines OSI presentation and application layers
- ◆ OSI session layer omitted

TCP/IP Layers



3. Host-to-Host (Transport)

- ◆ End-to-end data transfer
- ◆ Logical connection between source and destination
- ◆ Two main functions
 - ◆ Flow control using sliding windows
 - Communicating hosts negotiate how much data is to be transmitted each time
 - ◆ Reliability using sequence numbers and acknowledgements
 - Guarantees delivery of each packet

TCP/IP Layers



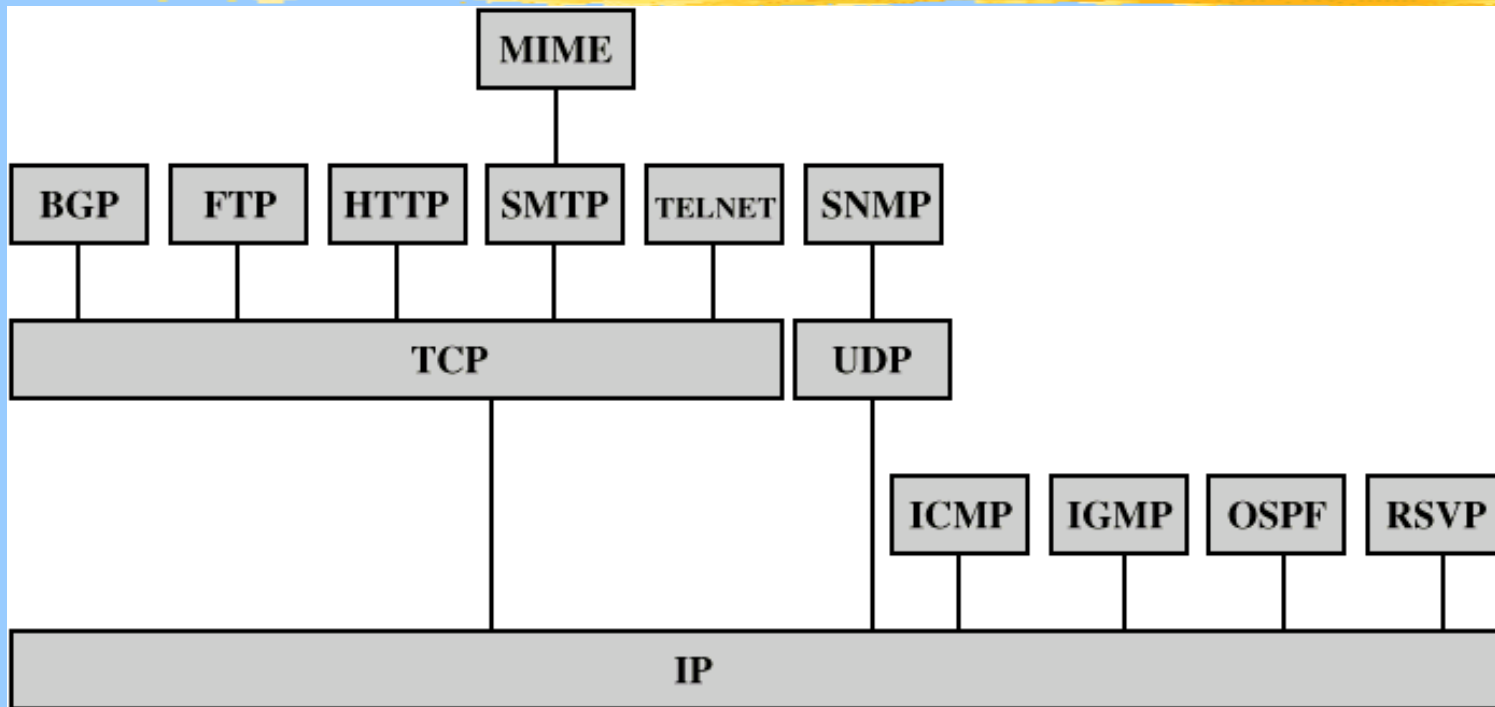
2. Internet

- ◆ Routing data through one or more networks using routers
- ◆ Provides logical (not physical) addressing and path selection

1. Network Access (Network Access + Physical)

- ◆ Organize packets into frames for transmitting
- ◆ Encodes bit stream
- ◆ Provides some error detection
- ◆ Similar to OSI Data Link and Physical layers

Some TCP/IP Protocols



BGP = Border Gateway Protocol

FTP = File Transfer Protocol

HTTP = Hypertext Transfer Protocol

ICMP = Internet Control Message Protocol

IGMP = Internet Group Management Protocol

IP = Internet Protocol

MIME = Multi-Purpose Internet Mail Extension

OSPF = Open Shortest Path First

RSVP = Resource ReSerVation Protocol

SMTP = Simple Mail Transfer Protocol

SNMP = Simple Network Management Protocol

TCP = Transmission Control Protocol

UDP = User Datagram Protocol

- ◆ W. Stallings, Data and Computer Communications, 6 ed., Upper Saddle River, NJ, Prentice Hall, 2000

Some TCP/IP Process Layer Protocols



- ◆ **Telnet**
 - ◆ Virtual terminal utility enabling access to remote hosts
- ◆ **FTP - File Transfer Protocol**
 - ◆ File transfer utility
- ◆ **SMTP - Simple Mail Transfer Protocol**
 - ◆ Main protocol responsible for transmission and distribution of e-mail
- ◆ **SNMP - Simple Network Management Protocol**
 - ◆ Main protocol responsible for transmission of vital network and system statistics and commands

Some TCP/IP Process Layer Protocols



- ◆ **HTTP - Hypertext Transfer Protocol**

- ◆ **Main protocol responsible for transmission of information throughout the World Wide Web**

Some TCP/IP Host-to-Host Layer Protocols



◆ TCP

- ◆ Connection oriented
- ◆ Reliable
- ◆ Connection established between both ends before data transfer begins
- ◆ Creates a virtual circuit between end-user applications
- ◆ Responsible for
 - ◆ Breaking messages into segments (segmentation)
 - ◆ Reassembling messages at destination (reassembly)
 - ◆ Handles retransmitting dropped packets

Some TCP/IP Host-to-Host Layer Protocols



◆ UDP - User Datagram Protocol

◆ Connectionless protocol

◆ No acknowledgements

◆ Unreliable

◆ No checking for segment delivery and sequencing

◆ Depends on higher layers for reliability

Some TCP/IP Internet Layer Protocols



◆ IP - Internet Protocol

- ◆ Main protocol of the entire TCP/IP protocol suite responsible primarily for addressing and routing of packets
- ◆ Connectionsless, best-effort delivery
- ◆ Fragmentation and reassembly
 - ◆ Routers will break up packets if necessary (Fragmentation) and reassemble and a downstream router (Reassembly)
 - Some networks have maximum packet sizes
 - ◆ Do not confuse with TCP's Segmentation and Reassembly

Some TCP/IP Internet Layer Protocols



◆ ICMP - Internet Control Message Protocol

- ◆ Provides network error and diagnostic feedback to networking nodes (hosts, routers)
- ◆ Note: error in text puts ICMP in Host-to-Host layer

◆ ARP - Address Resolution Protocol

- ◆ Broadcast protocol responsible for converting IP addresses to MAC addresses

◆ RARP - Reverse ARP

- ◆ Broadcast protocol responsible for converting a MAC address to an IP address

Some TCP/IP Internet Layer Protocols



◆ RIP - Routing Information Protocol

- ◆ Interior routing protocol used to disseminate routing information within an autonomous system

- ◆ Autonomous system

- Group of routers exchanging information via a common routing protocol
- Group of routers and networks managed by a single organization
- Always a connection between any two nodes (except during failure)

- ◆ Distance-vector protocol

- ◆ Hop count determines shortest path

- ◆ Often used in LANs

Some TCP/IP Internet Layer Protocols



◆ OSPF - Open Shortest Path First

- ◆ Interior routing protocol used to disseminate routing information within an autonomous system

- ◆ Link state protocol

 - ◆ Algorithm finds shortest path including speed of link and number of hops

 - ◆ Can perform load balancing

◆ BGP - Border Gateway Protocol

- ◆ Exterior routing protocol that exchanges information between border routers of *different* autonomous systems

Some TCP/IP Internet Layer Protocols



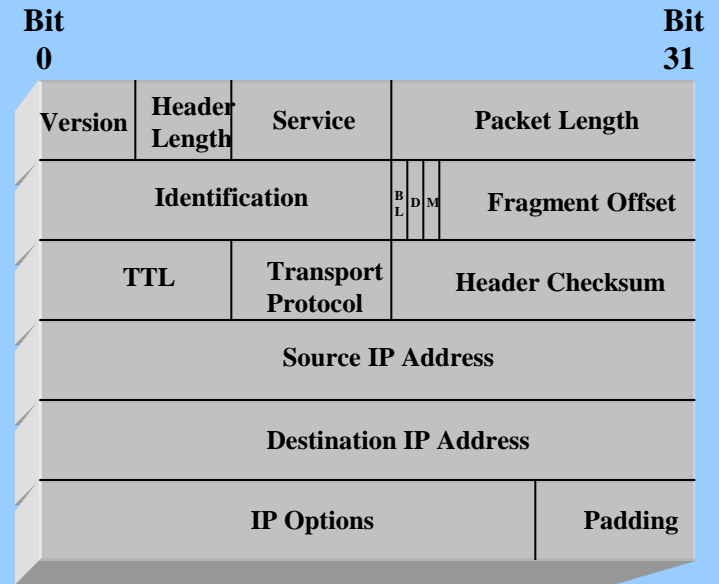
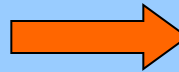
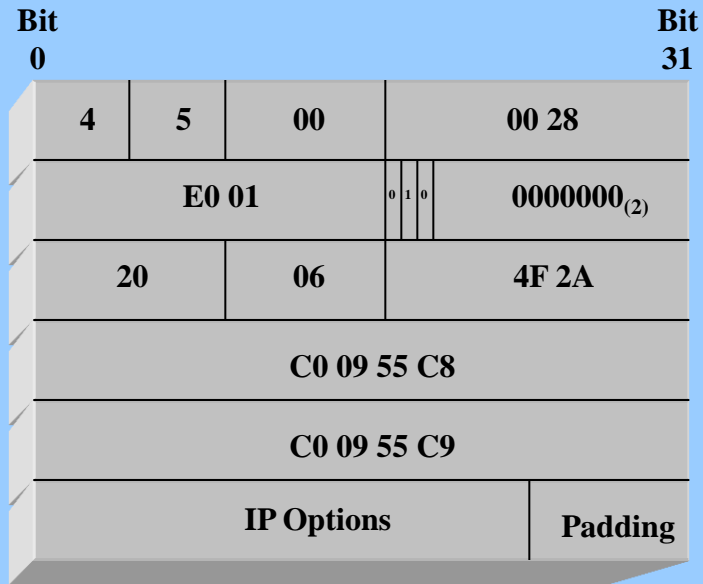
- ◆ **EGP - Exterior Gateway Protocol**
 - ◆ Exterior routing protocol used to exchange information between autonomous systems
 - ◆ BGP is an EGP

IP, TCP, and UDP Headers



- ◆ Protocol analyzers are used to analyze traffic on networks
 - ◆ Interpret the protocol fields
- ◆ Workstation-based analyzers are software packages that use the workstation's NIC to capture frames on the network
 - ◆ Ethernet card programmed to respond to all MAC addresses, not just its own (and broadcasts)
 - ◆ Strips off MAC framing and passes IP information up to Internet (Network) layer
 - ◆ Called “promiscuous” mode

IP Header



IP Header Fields



◆ Version

- ◆ Version 4 is current

◆ Length

- ◆ Length of header in 32-bit words

◆ Priority and Service

- ◆ How the datagram should be handled, such as priority

◆ Packet length

- ◆ Length of packet in bytes

IP Header Fields



◆ Identification

- ◆ Relates a fragment to the IP packet from which it came

◆ Flags

- ◆ Specifies whether fragmenting should occur

◆ Fragment offset

- ◆ Fragment's position relative to original packet from which it came

IP Header Fields



◆ TTL (Time To Live)

- ◆ Time in seconds that a packet can remain valid

- ◆ Internet nodes must decrease by 1 second

 - ◆ When TTL reaches zero, packet is discarded

 - ◆ Device discarding sends packet back ICMP message saying packet was dropped

 - ◆ Since internet nodes process packets in less than 1 second, TTL is essentially the maximum hop count

◆ Transport protocol

- ◆ Number identifying the TCP protocol in the payload

IP Header Fields



- ◆ **Header checksum**
 - ◆ Integrity check for header only
- ◆ **Source IP address**
 - ◆ 32-bit source IP address
- ◆ **Destination IP address**
 - ◆ 32-bit destination address

IP Header Fields



◆ Options

◆ Source route

- ◆ Pre-determined route

◆ Security specifications

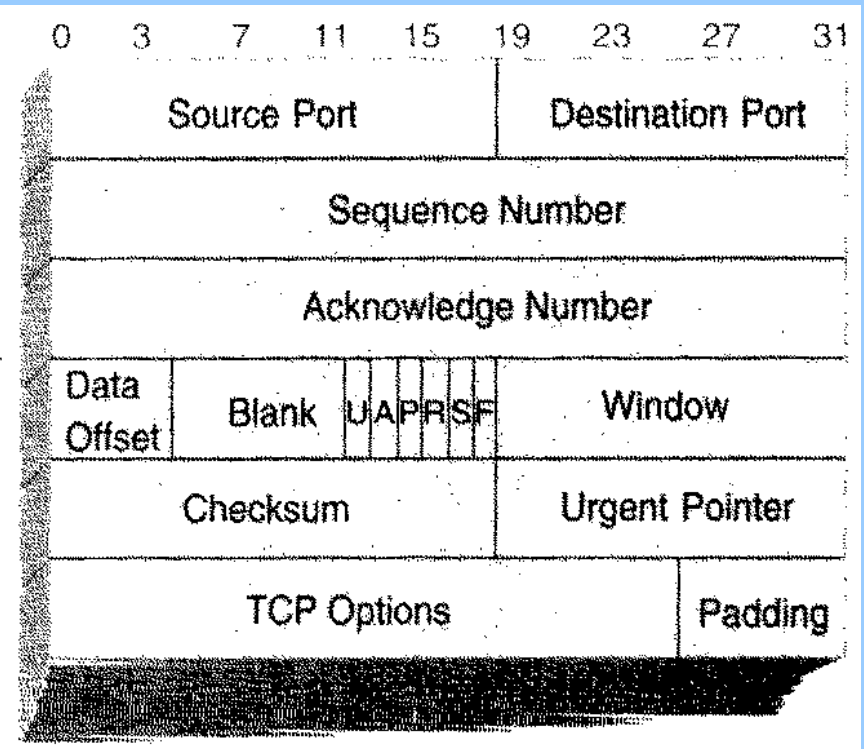
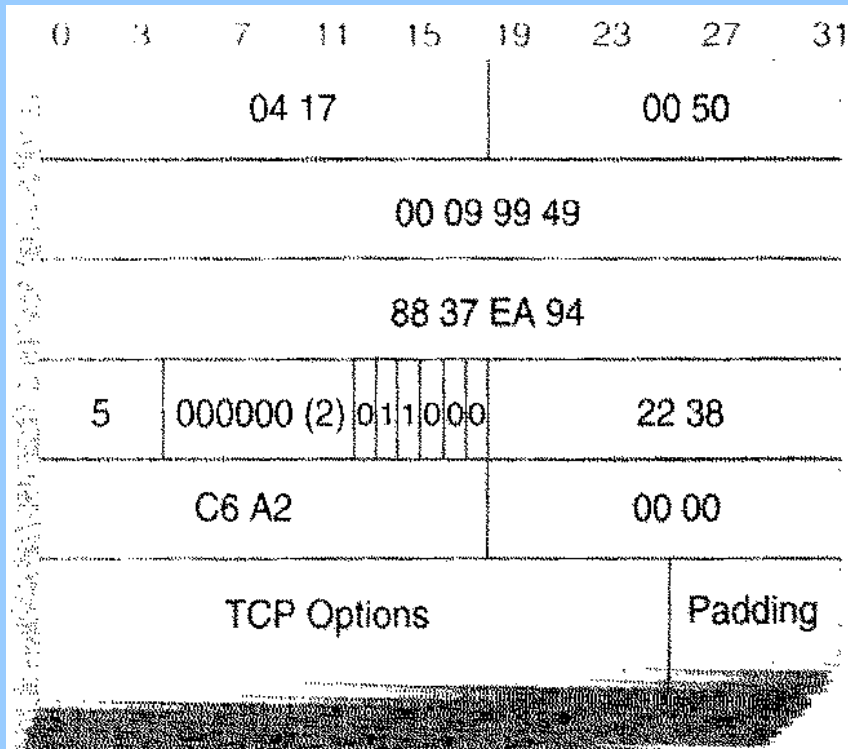
◆ Record route

- ◆ Addresses of intermediate nodes are recorded and sent back to source

◆ Padding

- ◆ Bits added to create a valid number of 16-bit entries in header

TCP Header



TCP Header Fields



- ◆ **Minimum header size is 20 bytes**
 - ◆ **More if options are used**

TCP Header Fields



◆ Source port

- ◆ Number of calling port
- ◆ Initiator of request is *client process*
- ◆ Generally not significant

◆ Destination port

- ◆ Number of called port
- ◆ Grantor of request is the *server process*
- ◆ Identifies one of the destination processes such as HTTP

TCP Header Fields



◆ Sequence number

- ◆ Number of bytes transmitted to other end node of TCP connection
- ◆ Unique numbers selected at beginning of connection and incremented as total number of bytes transmitted increases
- ◆ Used to ensure correct sequencing of information at destination and for retransmitting from source

◆ Acknowledgement number

- ◆ Next expected TCP octet

TCP Header Fields



- ◆ **Data offset / Header length**
 - ◆ Length of TCP header
- ◆ **Flags**
- ◆ **Window**
 - ◆ Number of octets destination says it is willing to accept
 - ◆ Source uses this to decide how many packets to send
 - ◆ Window size negotiated at connection start up and adjusted depending on network conditions
- ◆ **Checksum**
 - ◆ Checksum of header and data fields

TCP Header Fields



- ◆ **Urgent pointer**

- ◆ Indicates end of urgent data

- ◆ **Options**

- ◆ Only one defined is Maximum Segment Size

- ◆ **Padding**

- ◆ Used to fill out header

- ◆ **Data**

- ◆ Upper layer protocol data

TCP/UDP Ports



- ◆ Port numbers are used to pass information to upper layers
- ◆ They are also used to keep track of different connections running at the same time
- ◆ Internet Assigned Numbers Authority (IANA) assigns port numbers
 - ◆ Numbers below 1024 are “well-known ports”
 - ◆ Numbers above 1024 are dynamically assigned ports
 - ◆ Some of these are registered for vendor-specific applications

Popular Port Numbers

Service	Port	Transport	Comment
FTP	21	TCP	File transfer
Telnet	23	TCP	Remote terminal
SMTP	25	TCP	e-mail
HTTP	80	TCP	Hypertext transport protocol
TFTP	69	UDP	Trivial file transfer
SNMP	161	UDP	Simple network management
RIP	520	UDP	Routing information protocol
DNS	53	TCP/UDP	Domain name system

TCP Services



- ◆ **Connection-oriented**
 - ◆ Application must establish session with destination
- ◆ **Point-to-point communication**
 - ◆ Only 2 endpoints
- ◆ **Complete reliability**
 - ◆ Guarantees error-free delivery
- ◆ **Full duplex communication**

TCP Services



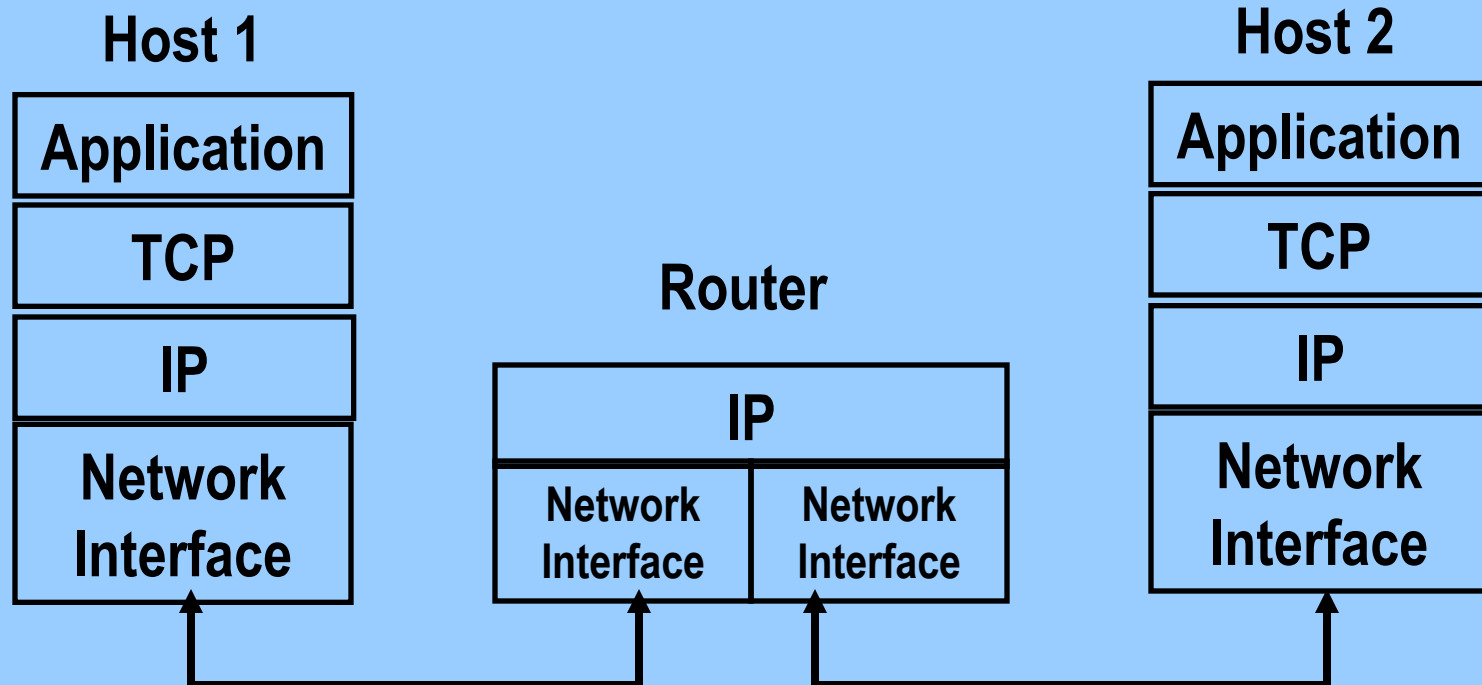
- ◆ **Has a stream interface mode**
 - ◆ **Application sends continuous sequence of octets**
 - ◆ **No guarantee that data at receiver is partitioned same as at sender**
- ◆ **Reliable connection startup**
- ◆ **Graceful connection shutdown**
 - ◆ **Data will be delivered before closing connection**

End-to-End Delivery



- ◆ TCP is an end-to-end protocol
 - ◆ Provides connection between applications on two computers
 - ◆ *Virtual connections* - connections created by software
- ◆ IP carries messages for TCP
 - ◆ TCP message is the IP payload
- ◆ At destination, IP passes data to TCP

End-to-End Delivery



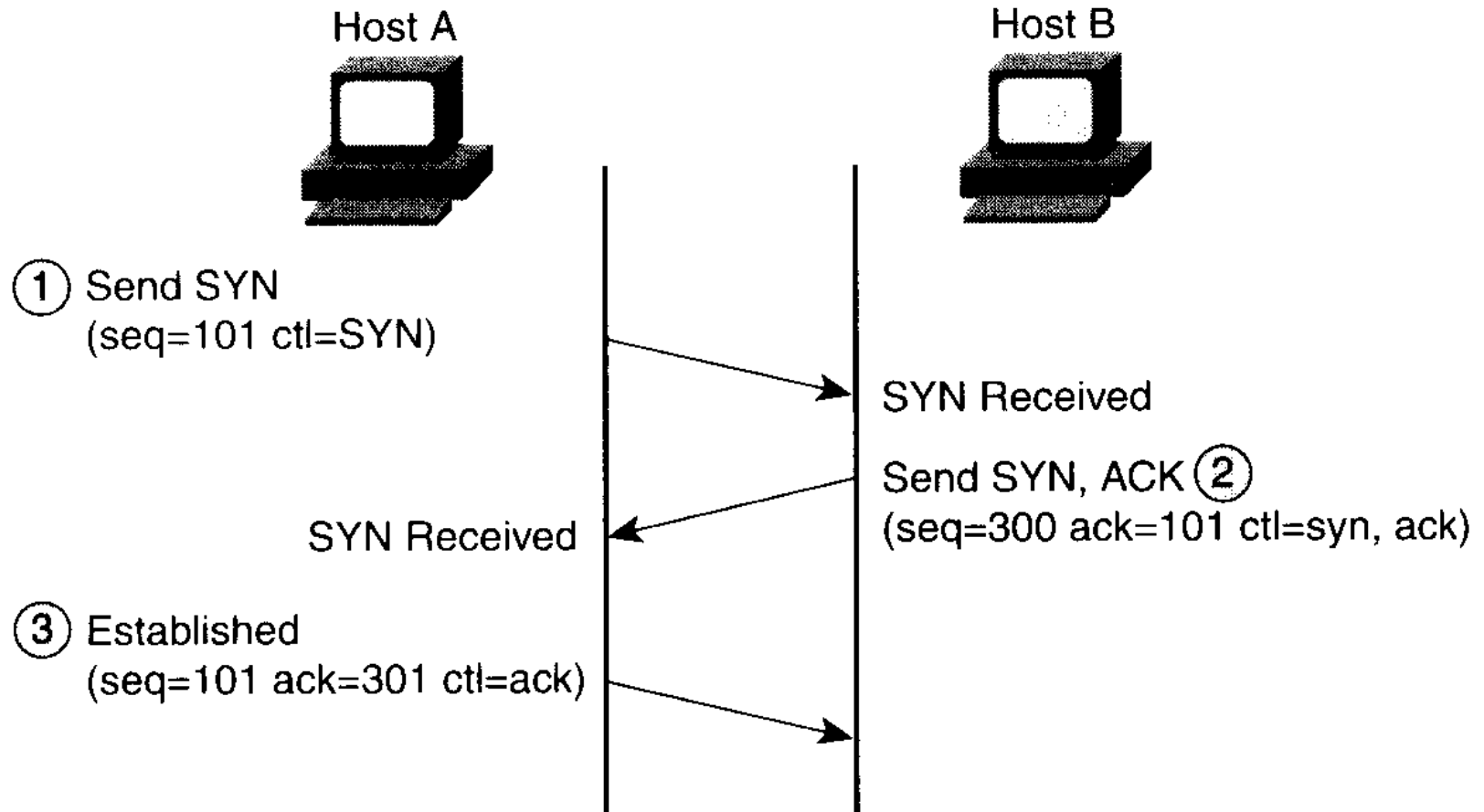
TCP Connection Establishment



- ◆ Source and destination exchange SYN (synchronize) and ACK (acknowledgement) packets
 - ◆ Packets contain initial sequence numbers
- ◆ Uses *3-Way Handshake* technique
 - ◆ Source sends SYN packet, destination sends back ACK
 - ◆ Happens 3 times
 - ◆ Avoids problems that can occur with late delivery of an old SYN from a previous session
 - ◆ 2-way handshake is susceptible to this

Three-Way Handshake

Three-Way Handshake



Flow Control



- ◆ Data overrun can occur when sender transmits data faster than receiver can process it
- ◆ Protocols use flow control mechanisms so receiver can control rate of transmission
- ◆ Two key types:
 - ◆ Sliding window
 - ◆ Stop-and-go
 - ◆ Same as sliding window with window size = 1

Stop and Go Flow Control



- ◆ Receiver sends small ACK packet when it is ready for next
- ◆ Sender waits for ACK before sending next packet
- ◆ Inefficient if network transit time is large

Example - Stop and Go Flow Control



◆ Given

- ◆ Packet length = 1000 octets
- ◆ One-way travel time = 50 ms
- ◆ Network capacity = 2 Mbps
- ◆ Transmission takes $50 \text{ ms} + 0.5 \text{ ms}$ (packet length in seconds)
- ◆ Receiver sends acknowledgement, which arrives at sender 50 ms later

Example - Stop and Go Flow Control



◆ Sender can only send packets every 100.5 ms

◆ Effective data rate is

$$8 \frac{\text{bits}}{\text{octet}} \times 1000 \text{ octets} \times \frac{1}{100.5 \text{ ms}} = 79.6 \text{ kbps}$$

◆ This is 4% of capacity

TCP Flow Control - Sliding Window



- ◆ TCP uses a sliding window method of flow control
 - ◆ Allows sender to transmit several packets before receiving an acknowledgment
 - ◆ TCP defines window size as the number of packets that can be sent before an acknowledgement is returned
 - ◆ Window size of 1 means each segment must be acknowledged before next one is sent
 - Very inefficient
 - ◆ As ACKs arrive at the sender, the window is moved along the buffer of data packets to be sent
 - ◆ This is the *sliding window*

Sliding Window Flow Control

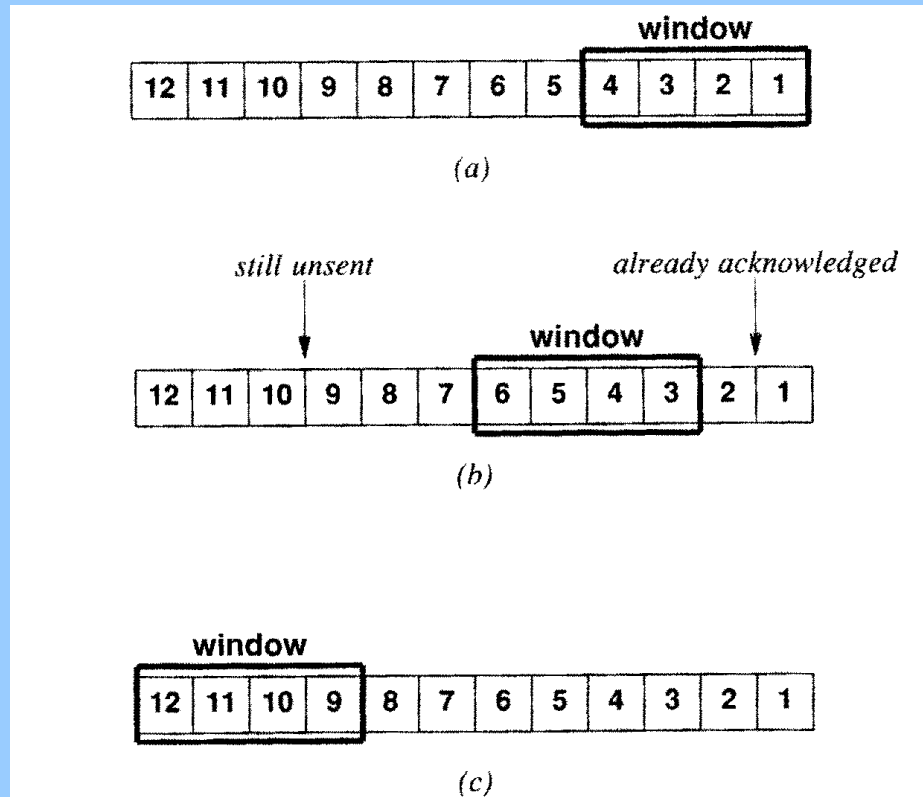


Figure 14.6 A 4-packet window sliding through outgoing data. The window is shown (a) when transmission begins, (b) after two packets have been acknowledged, and (c) after eight packets have been acknowledged. The sender can transmit all packets in the window.

TCP Flow Control - Sliding Window

- ◆ Sliding window protocol is about W times faster than stop-and-go protocol

$$T_w = T_g \times W$$

- ◆ T_w is the throughput of sliding window protocol, T_g is the throughput of stop and go protocol, and W is the window size
- ◆ Max speed is the network capacity

Stop-and-Go vs. Sliding Window

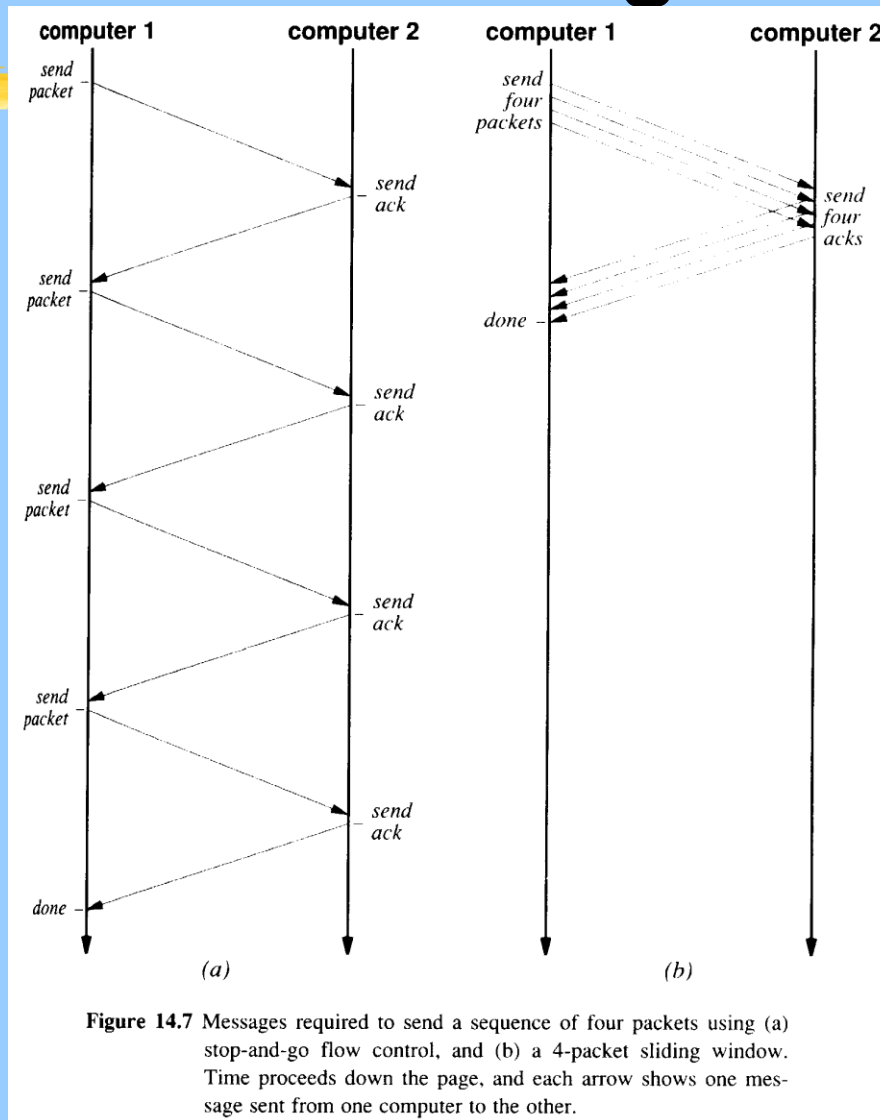


Figure 14.7 Messages required to send a sequence of four packets using (a) stop-and-go flow control, and (b) a 4-packet sliding window. Time proceeds down the page, and each arrow shows one message sent from one computer to the other.

TCP Flow Control - Sliding Window



- ◆ TCP window sizes are variable during a connection
 - ◆ Each acknowledgement from destination contains a window size saying how many bytes destination can accept
 - ◆ Sender adjusts its window size to match the size the destination wants
 - ◆ Exception is during network congestion

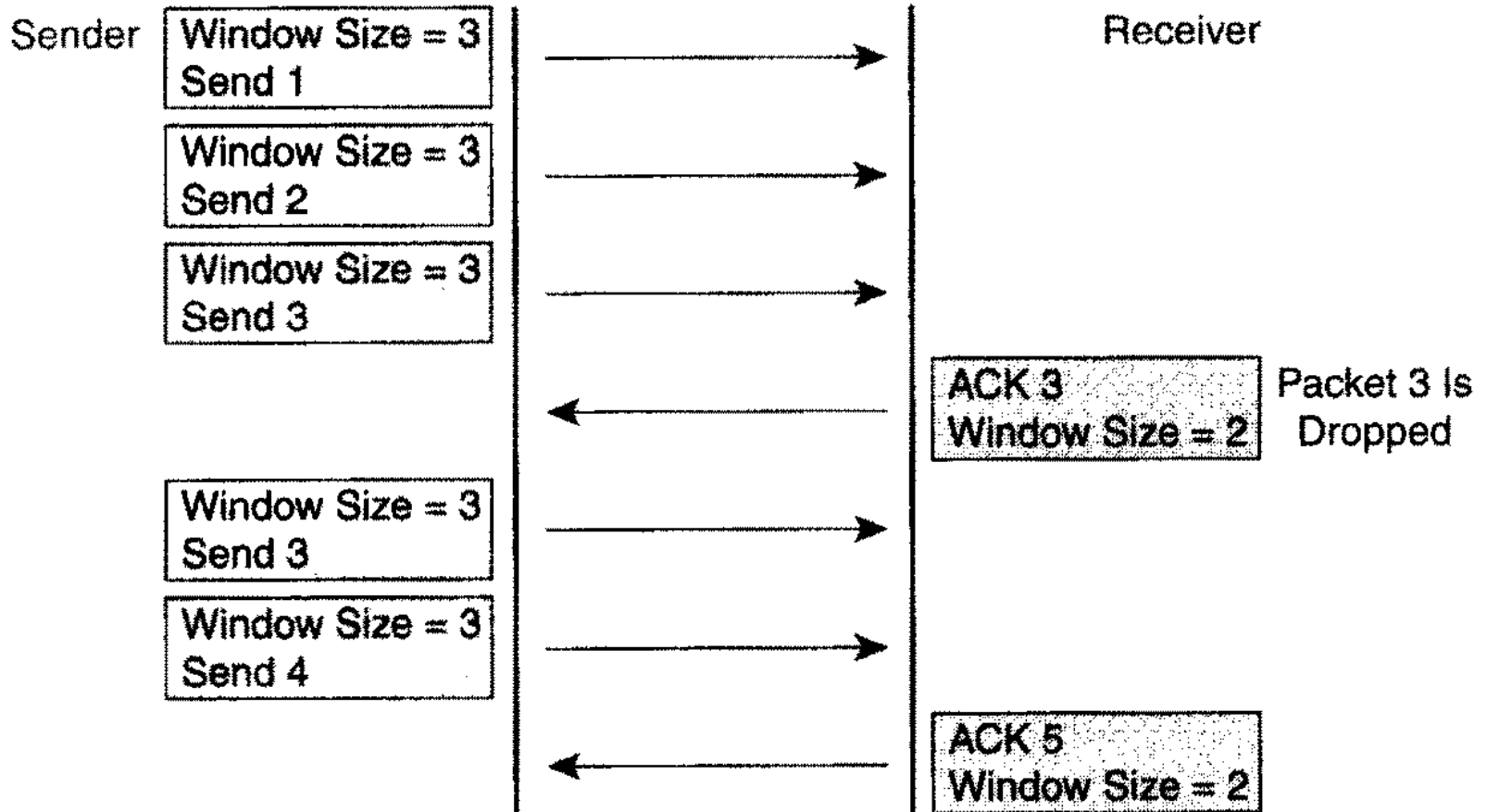
TCP Flow Control - Sliding Window



- ◆ **Sender also has a congestion control window that is normally the size of the receiver's window**
 - ◆ **However, can be reduced if segments are dropped or lost**
 - ◆ **When this happens, sender uses a much smaller congestion control window size**
 - ◆ **If segments are not lost this time, congestion window is increased**
 - ◆ **Process repeats until the sender's congestion window reaches the size of the receiver's buffer**

Window Size Adjustment

TCP Windowing



Reliability and Retransmission



- ◆ **Retransmission is one of the most important ways TCP achieves reliability**
- ◆ **When TCP sends data, it starts the *Retransmission Timer***
 - ◆ **If timer expires before an acknowledgement is received from destination, TCP retransmits the segment**
 - ◆ **TCP adjusts the timer to account for network conditions**
 - ◆ **Short timer for LANs**
 - ◆ **Longer timer for long distances**
 - ◆ **Longer timer for congested networks**
 - ◆ **This technique reduces retransmissions and congestion**

Retransmission

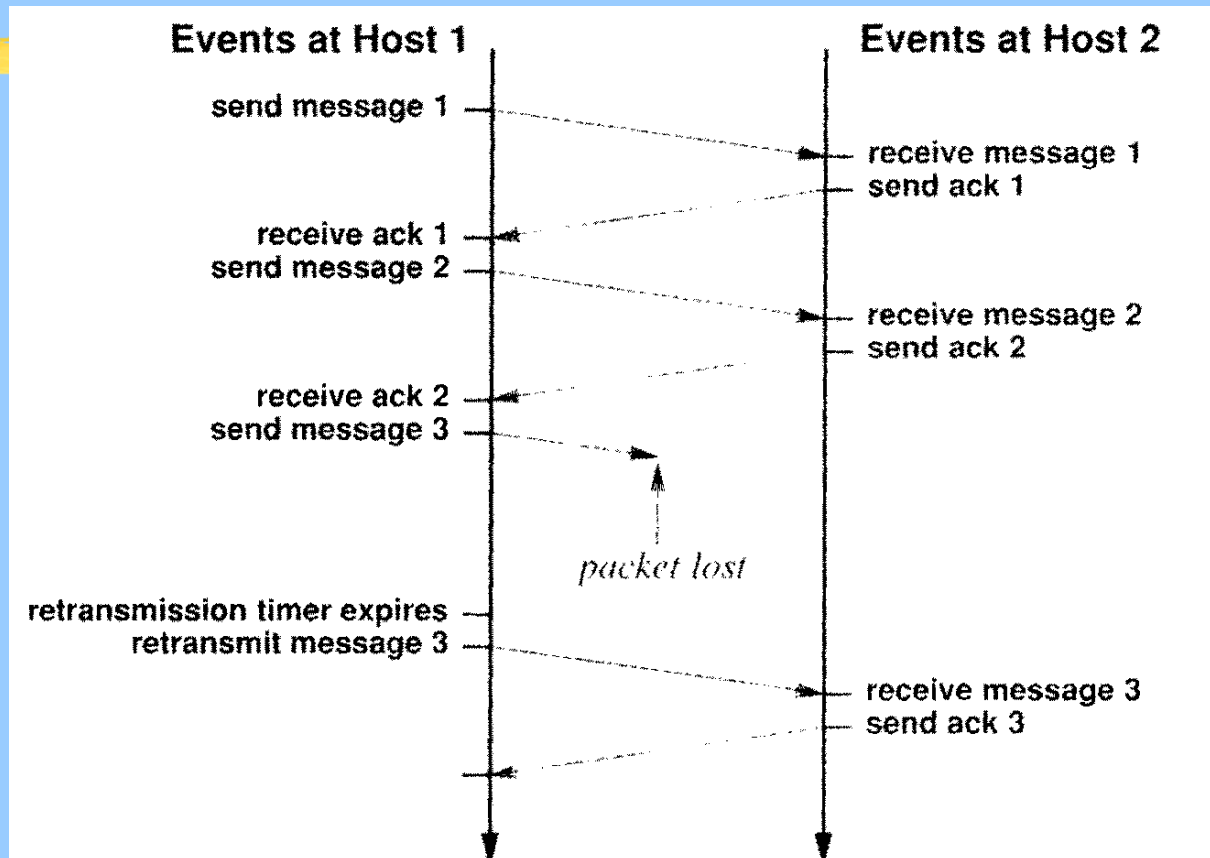


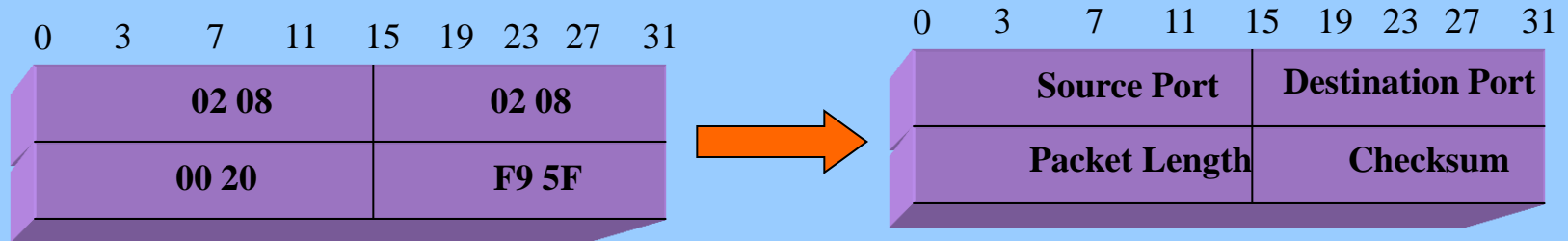
Figure 22.2 Example of retransmission. Items on the left correspond to events in a computer sending data, items on the right correspond to events in a computer receiving data, and time goes down the figure. The sender retransmits lost data.

UDP



- ◆ **Lower overhead than TCP**
 - ◆ **No provisions for retransmission**
 - ◆ **No windows**

UDP Header



UDP Header



- ◆ **Source and destination port**
 - ◆ Similar to TCP
- ◆ **Packet length**
 - ◆ Length of UDP header and all higher-layer information
- ◆ **Checksum**
 - ◆ Similar to TCP

IP Addresses



- ◆ They are the key to communicating on the Internet
- ◆ They are logical addresses
 - ◆ Can't use hardware addresses
- ◆ 32-bits (4 octets) long
- ◆ Unique across the Internet
- ◆ IP addresses of the source and destination (not the next hop) are included in an IP packet

IP Address Hierarchy



- ◆ Each address has a prefix and a suffix
- ◆ Prefix
 - ◆ Also called *Network Number*
 - ◆ Identifies the network
- ◆ Suffix
 - ◆ Also called *Host Number*
 - ◆ Identifies host on that prefix's network
- ◆ Uniqueness
 - ◆ No two networks can have the same network number
 - ◆ No two hosts on a given network can have the same host number

IP Address Classes



- ◆ Five address classes: A, B, C, D, E
- ◆ First four bits determine the class
- ◆ A, B, and C are *primary classes*
 - ◆ A has an 8-bit network number
 - ◆ First bit = 0
 - ◆ B has a 16-bit network number
 - ◆ First 2 bits = 10
 - ◆ C has a 32-bit network number
 - ◆ First 3 bits = 110
- ◆ Class D is for IP multicasting
- ◆ Class E is for research

IP Address Classes



- ◆ **Class A addresses are for networks with many hosts**
 - ◆ Not many of these networks
- ◆ **Class C addresses are for networks with a few hosts**
 - ◆ Many of these networks

IP Address Classes

Address Classes

	8 Bits	8 Bits	8 Bits	8 Bits
Class A:	Network	Host	Host	Host
Class B:	Network	Network	Host	Host
Class C:	Network	Network	Network	Host
Class D:	Multicast			
Class E:	Research			

IP Address Classes

Address Classification

	Bits: 1	8 9	16 17	24 25	32
Class A:	0NNNNNNN	Host	Host	Host	
	Range (1—126)				
Class B:	10NNNNNN	Network	Host	Host	
	Range (128—191)				
Class C:	110NNNN	Network	Network	Host	
	Range (192—223)				

IP Address Classes

IP Address Classes

Class A Address	Class B Address	Class C Address
The first bit is 0.	The first two bits are 10.	The first three bits are 110.
Range of network numbers: 1.0.0.0 to 126.0.0.0.	Range of network numbers: 128.0.0.0 to 191.255.0.0.	Range of network numbers: 192.0.0.0 to 223.255.255.0.
Number of possible networks: 127 (1 through 126 are usable; 127 is reserved).	Number of possible networks: 16,384.	Number of possible networks: 2,097,152.
Number of possible values in the host portion: 16,777,216. (The number of usable hosts is two less than the total number possible because the host portion must be nonzero and cannot be all 1s.)	Number of possible values in the host portion: 65,536. (The number of usable hosts is two less than the total number possible because the host portion must be nonzero and cannot be all 1s.)	Number of possible values in the host portion: 256. (The number of usable hosts is two less than the total number possible because the host portion must be nonzero and cannot be all 1s.)

IP Address Classes D and E



◆ Class D: Multicast

- ◆ Range: 224.0.0.0 - 239.255.255.255

- ◆ First 4 bits are 1110

◆ Class E: Research

- ◆ 240.0.0.0 - 247.255.255.255

- ◆ First 5 bits are 11110

Dotted Decimal Notation



- ◆ Divide up 32-bit address into 8-bit segments
- ◆ Convert each segment to decimal
- ◆ Separate decimal numbers by dots

Dotted Decimal Notation

IP Address Format

