# PART XIII

# Internetworking
# Part 4

## (Transport Protocols, UDP and TCP, Protocol Port Numbers)

# Transport Protocol

- Separate layer of protocol stack
- Conceptually between
  - Applications
  - IP

# Terminology

- IP
  - Provides computer-to-computer communication
  - Source and destination addresses are computers
  - Called *machine-to-machine*
- Transport protocols
  - Provide application-to-application communication
  - Need extended addressing mechanism to identify applications
  - Called *end-to-end*

# Transport Protocol Functionality

- Identify sending and receiving applications
- Optionally provide
  - Reliability
  - Flow control
  - Congestion control
- Note: not all transport protocols provide above facilities

# Two Transport Protocols Available

- *Transmission Control Protocol (TCP)*
- *User Datagram Protocol (UDP)*
- Major differences
  - Interface to applications
  - Functionality

# User Datagram Protocol (UDP)

- Provides unreliable transfer
- Requires minimal
  - Overhead
  - Computation
  - Communication
- Best for LAN applications

# UDP Details

- Connectionless service paradigm
  - Message-oriented interface
- Each message encapsulated in IP datagram
- UDP header identifies
  - Sending application
  - Receiving application

# Identifying An Application

- Cannot extend IP address
  - No unused bits
- Cannot use OS-dependent quantity
  - Process ID
  - Task number
  - Job name
- Must work on all computer systems

# Identifying An Application (continued)

- Invent new abstraction
  - Used only with TCP/IP
  - Identifies sender and receiver unambiguously
- Technique
  - Each application assigned unique integer
  - Called *protocol port number*

# Protocol Ports

- Server
  - Follows standard
  - Always uses same port number
  - Uses lower port numbers
- Client
  - Obtains unused port from protocol software
  - Uses higher port numbers

# Protocol Port Example

- Domain name server application is assigned port 53
- Application using DNS obtains port 28900
- UDP datagram sent from application to DNS server has
  - Source port number 28900
  - Destination port number 53
- When DNS server replies, UDP datagram has
  - Source port number 53
  - Destination port number 28900

# Transmission Control Protocol (TCP)

- Major transport protocol used in Internet
- Heavily used
- Completely reliable transfer

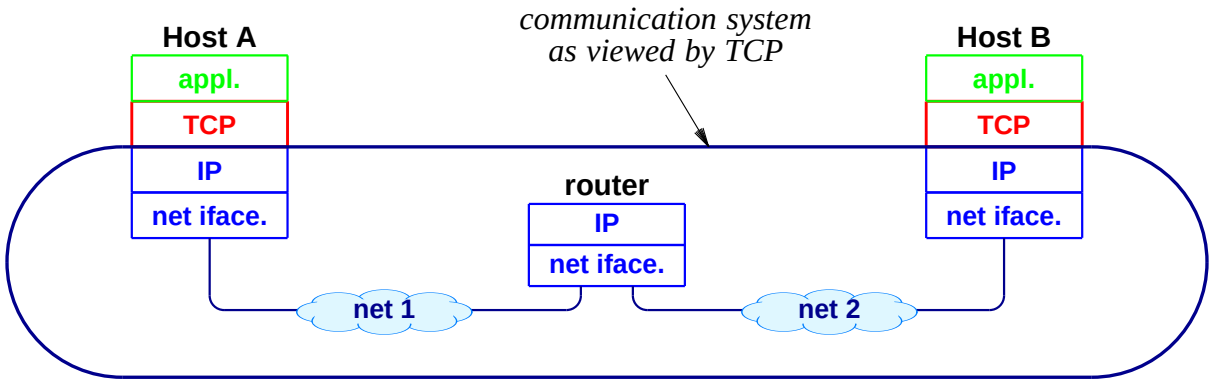
# TCP Features

- Connection-oriented service
- Point-to-point
- Full-duplex communication
- Stream interface
- Stream divided into segments for transmission
- Each segment encapsulated in IP datagram
- Uses protocol ports to identify applications

# TCP Feature Summary

*TCP provides a completely reliable (no data duplication or loss), connection-oriented, full-duplex stream transport service that allows two application programs to form a connection, send data in either direction, and then terminate the connection.*

# Relationship Between TCP And Other Protocols




- TCP on one computer uses IP to communicate with TCP on another computer

# Apparent Contradiction

- IP offers best-effort (unreliable) delivery
- TCP uses IP
- TCP provides completely reliable transfer
- How is this possible?

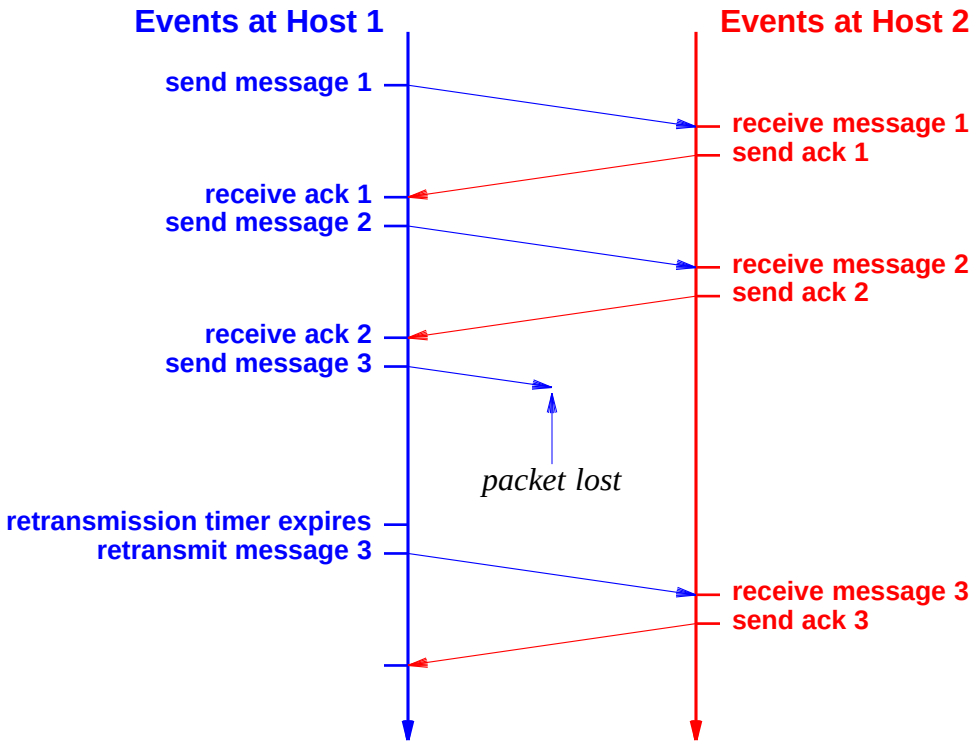
# Achieving Reliability

- Reliable connection startup
- Reliable data transmission
- Graceful connection shutdown

# Reliable Data Transmission

- Positive acknowledgment
  - Receiver returns short message when data arrives
  - Called *acknowledgment*
- Retransmission
  - Sender starts timer whenever message is transmitted
  - If timer expires before acknowledgment arrives, sender *retransmits* message

# Illustration Of Retransmission



# How Long Should TCP Wait Before Retransmitting?

- Time for acknowledgment to arrive depends on
  - Distance to destination
  - Current traffic conditions
- Multiple connections can be open simultaneously
- Traffic conditions change rapidly

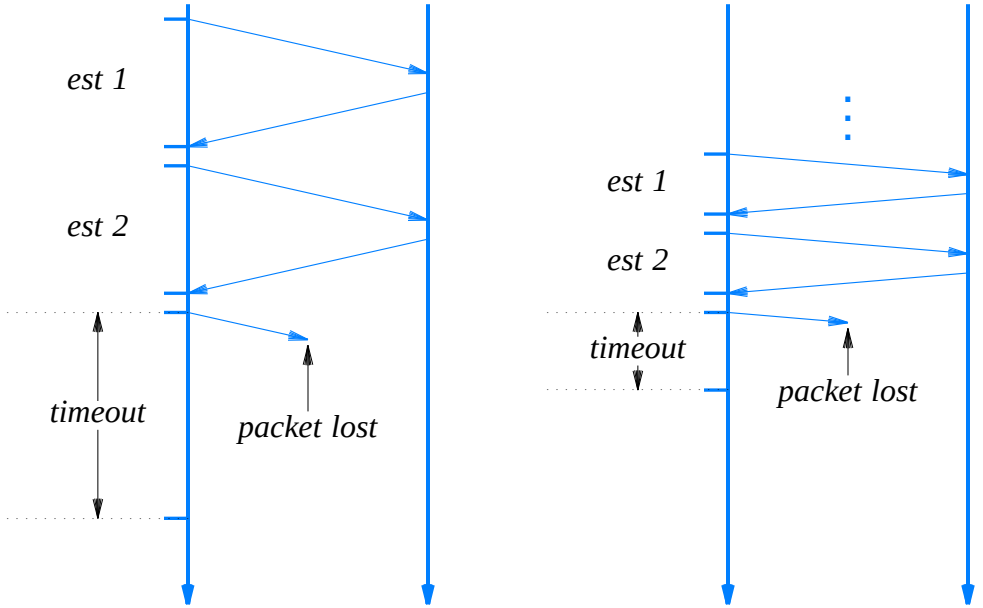
# Important Point

*The delay required for data to reach a destination and an acknowledgment to return depends on traffic in the internet as well as the distance to the destination. Because it allows multiple application programs to communicate with multiple destinations concurrently, TCP must handle a variety of delays that can change rapidly.*

# Solving The Retransmission Problem

- Keep estimate of round trip time on each connection
- Use current estimate to set retransmission timer
- Known as *adaptive retransmission*
- Key to TCP's success

# Illustration Of Adaptive Retransmission




- Timeout depends on current round-trip estimate

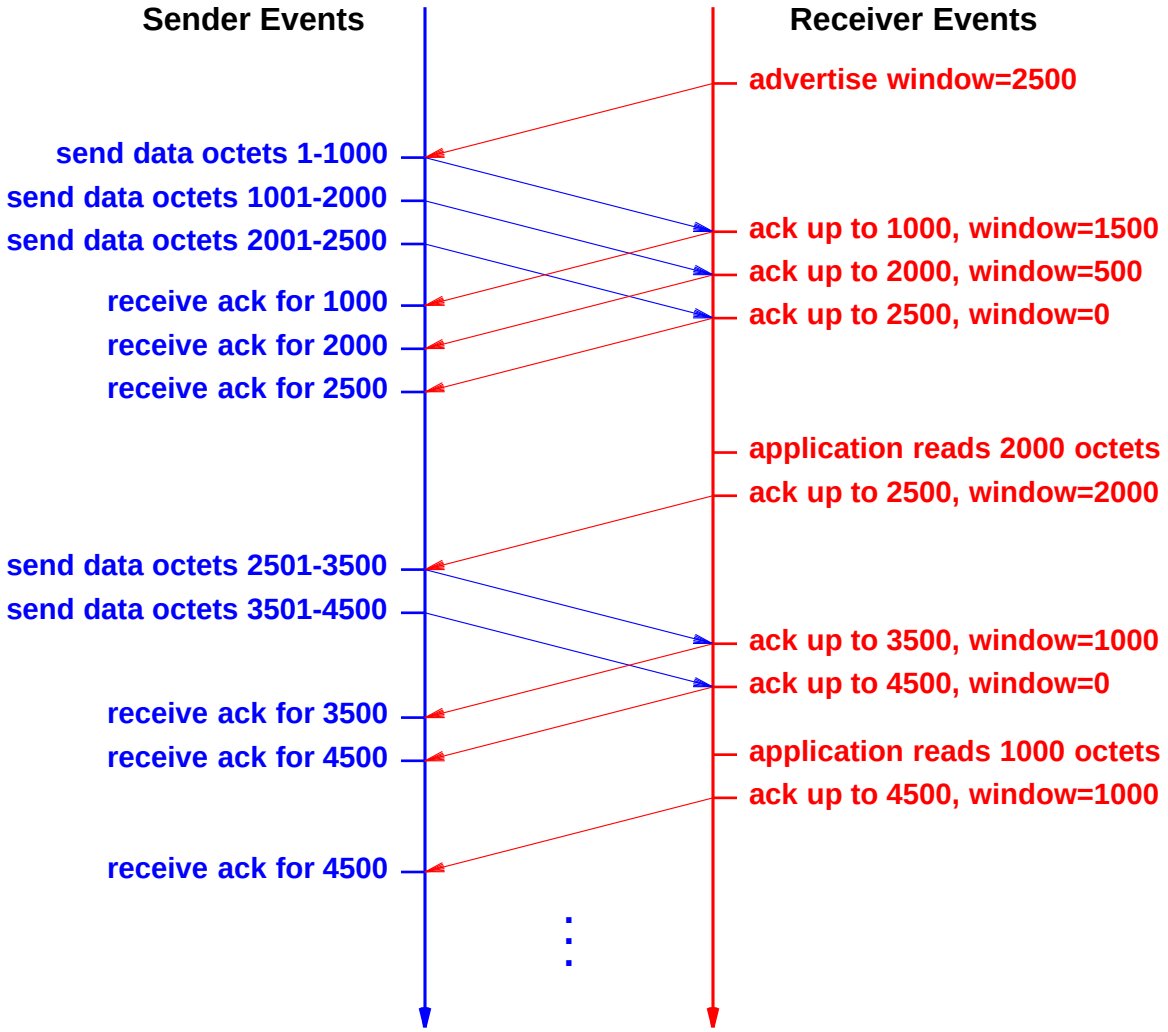
# TCP Flow Control

- Receiver
  - Advertises available buffer space
  - Called *window*
- Sender
  - Can send up to entire window before ack arrives

# Window Advertisement

- Each acknowledgment carries new window information
  - Called *window advertisement*
  - Can be zero (called *closed window*)
- Interpretation: I have received up through *X*, and can take *Y* more octets

# Illustration Of Window Advertisement



# Startup And Shutdown

- Connection startup
  - Must be reliable
- Connection shutdown
  - Must be graceful
- Difficult

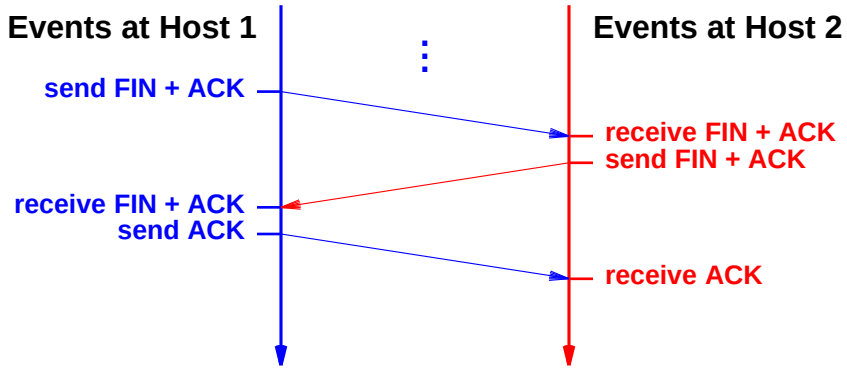
# Why Startup/Shutdown Difficult

- Segments can be
  - Lost
  - Duplicated
  - Delayed
  - Delivered out of order
  - Either side can crash
  - Either side can reboot
- Need to avoid duplicate ‘‘shutdown’’ message from affecting later connection

# TCP's Startup/Shutdown Solution

- Uses three-message exchange
- Known as *3-way handshake*
- Necessary and sufficient for
  - Unambiguous, reliable startup
  - Unambiguous, graceful shutdown
- *SYN* used for startup
- *FIN* used for shutdown

# Illustration Of 3-Way Handshake



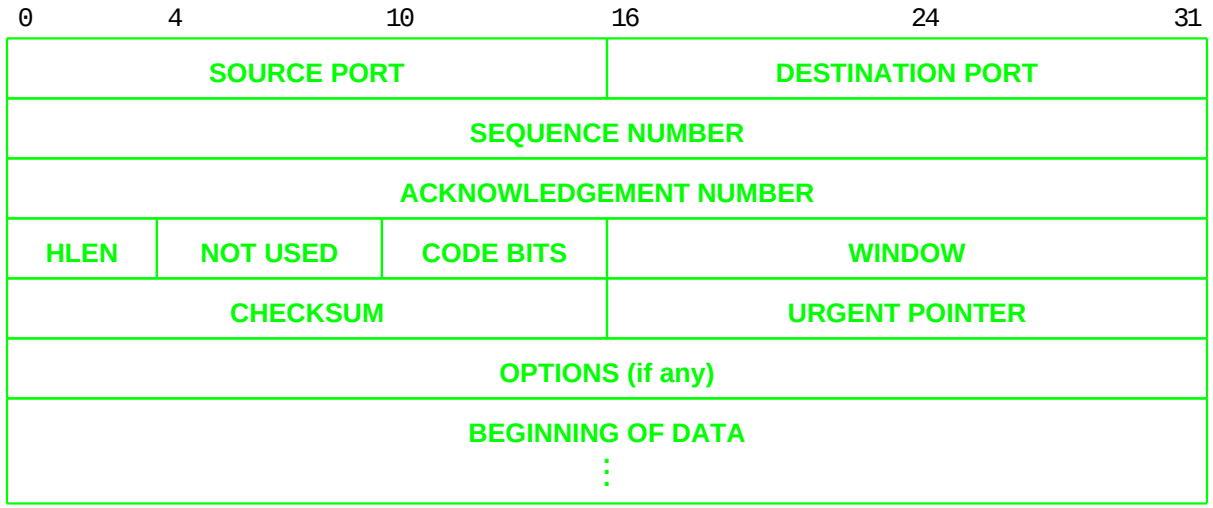
# TCP Segment Format

- All TCP segments have same format
  - Data
  - Acknowledgment
  - SYN (startup)
  - FIN (shutdown)
- Segment divided into two parts
  - Header
  - Payload area (zero or more bytes of data)

# TCP Segment Format
## (continued)

- Header contains
  - Protocol port numbers to identify
    - Sending application
    - Receiving application
  - Bits to specify items such as
    - SYN
    - FIN
    - ACK
  - Fields for window advertisement, acknowledgment, etc.

# Illustration Of TCP Segment




- Sequence number specifies where in stream data belongs
- Few segments contain options

# Summary

- Transport protocols fit between applications and Internet Protocol
- Two transport protocols in TCP/IP suite
  - User Datagram Protocol (UDP)
  - Transmission Control Protocol (TCP)
- UDP
  - Unreliable
  - Message-oriented interface

# Summary
# (continued)

- TCP
  - Major transport protocol used in Internet
  - Complete reliability
  - Stream-oriented interface
  - Uses adaptive retransmission

# Summary
# (continued)

- Protocol ports
  - Integers
  - Used to identify sending and receiving applications
  - Allow unambiguous, simultaneous communication with multiple applications