NET 363
Introduction to LANs

TCP

Greg Brewster
DePaul University

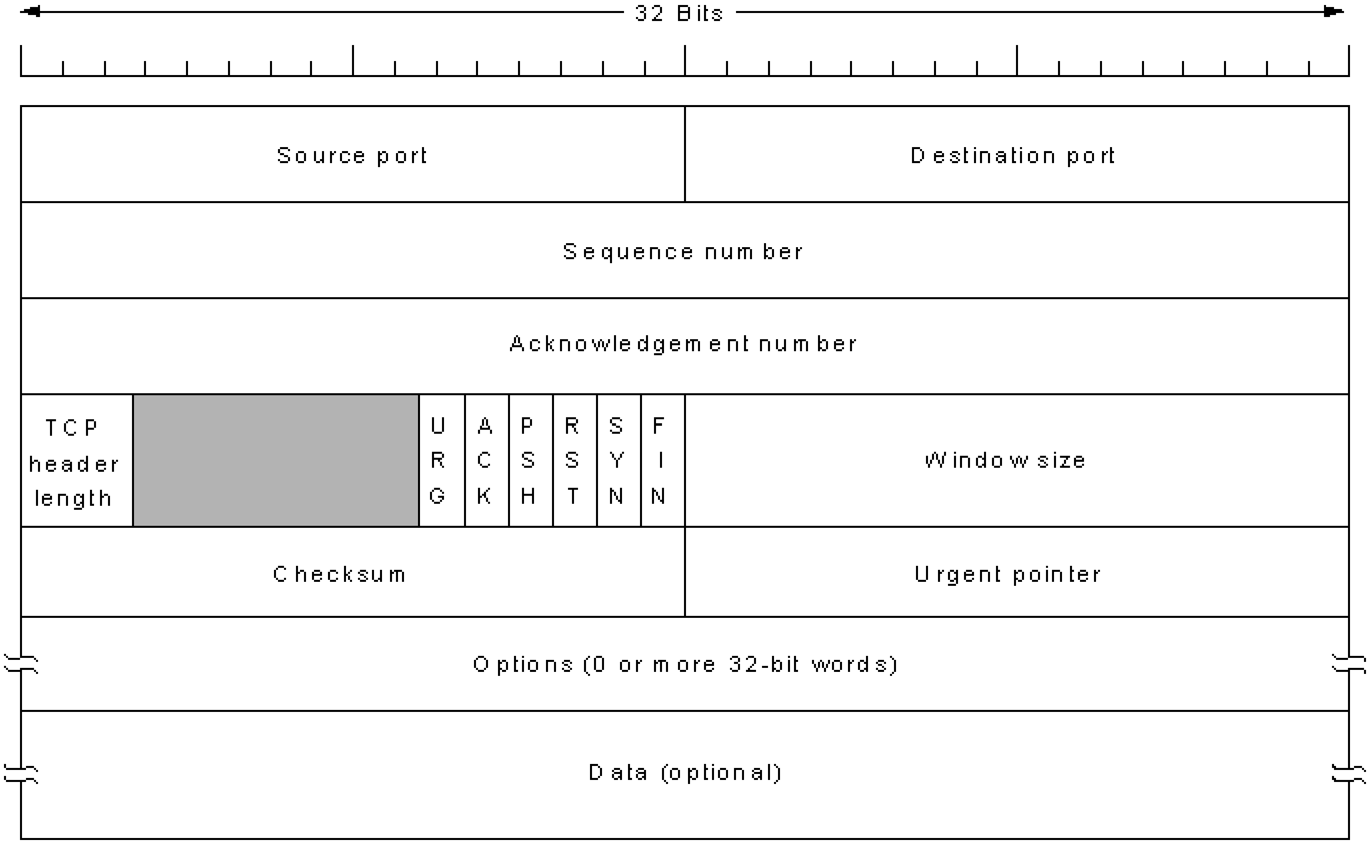
Transmission Control Protocol (TCP)

- Provides reliable data delivery services
  - Every byte of data is numbered (sequence number)
- Connection oriented
  - Requires the establishment of a connection between communicating nodes before the protocol will transmit data
- TCP segment
  - Holds the TCP data fields
  - Becomes encapsulated by the IP datagram

What does TCP do?

- Provides error-free in-order data delivery between Client and Server application software
  - Sets up connections across the Internet between Client and Server
  - Reorders data if it arrives out-of-order
  - Detects errors and re-transmits data if errors occur
  - Congestion Control: Automatically slows down if network is busy (packets dropped)

TCP Header



32 Bits

Source port

Destination port

Sequence number

Acknowledgement number

TCP header length

URG

ACK

PSH

RST

SYN

FIN

Window size

Checksum

Urgent pointer

Options (0 or more 32-bit words)

Data (optional)

TCP Header Fields

- **Source / Dest Port**: Source port number and destination port number for this packet
- **Sequence:** Number of the first byte of this segment.
- **Acknowledgement**: Number of the next Sequence number expected to be received. All bytes up this number have been received correctly.
- **TCP Header Length**: Indicates whether any options are used.

TCP Header Fields

- **Window Size:** the number of additional bytes the other end can send before it must wait for acknowledgement (Flow Control).
- **Checksum**: Calculated by sender and receiver to determine if any errors have occurred.
- **Urgent pointer**: Can indicate high-priority data within packet.

TCP Header Field Bits

- **URG bit:** Indicates whether there is urgent data in this packet
- **ACK bit:** Indicates whether this packet is acknowledging another packet
- **PSH bit:** Indicates whether this data should be quickly pushed up to application program at receiver.

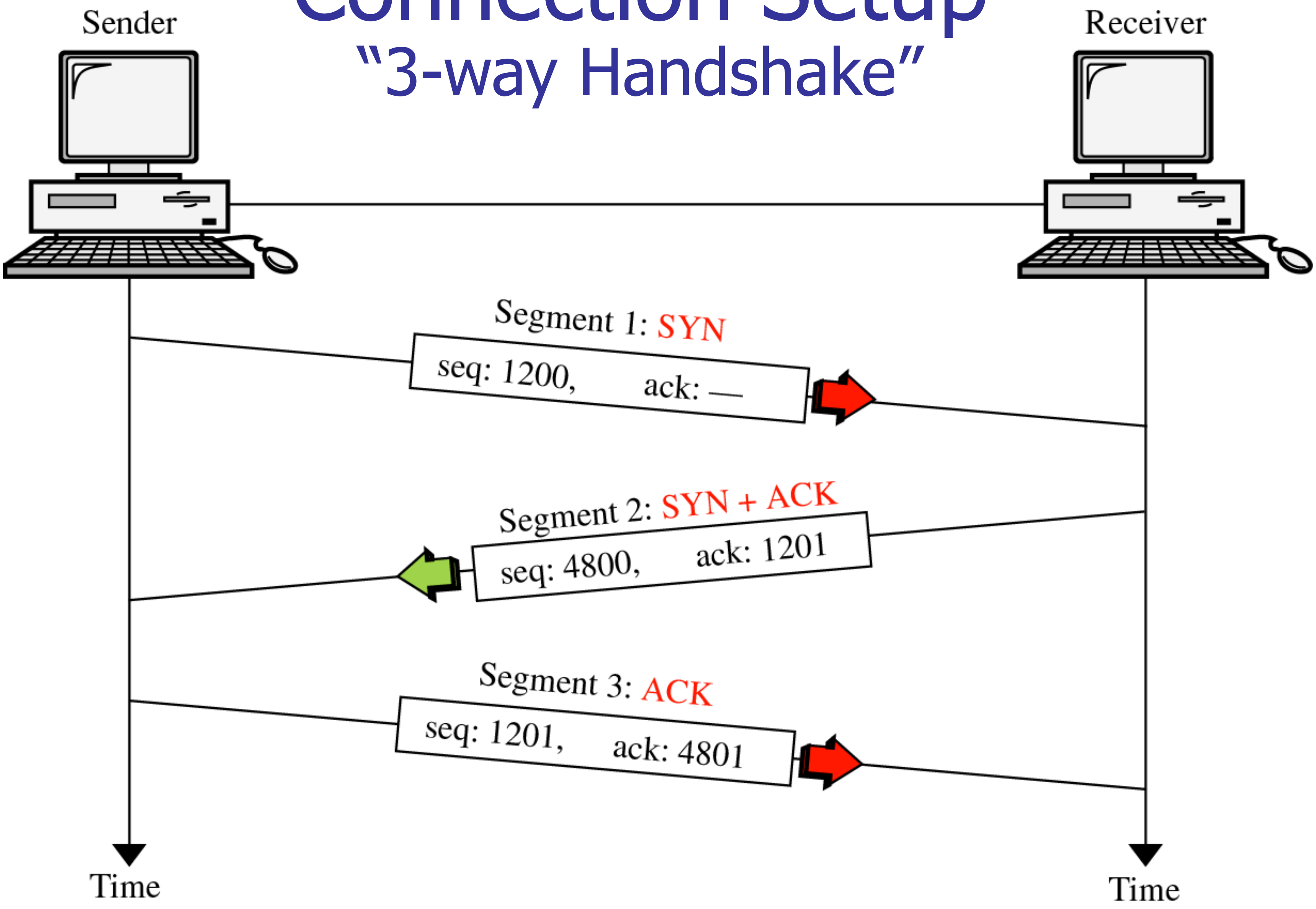
TCP Header Field Bits

- **RST bit:** Set to 1 to reset the TCP connection if error occurs
- **SYN bit:** Indicates a request to set up a new communications session (synchronize)
- **FIN bit:** Indicates final packet – closes down a communications session.

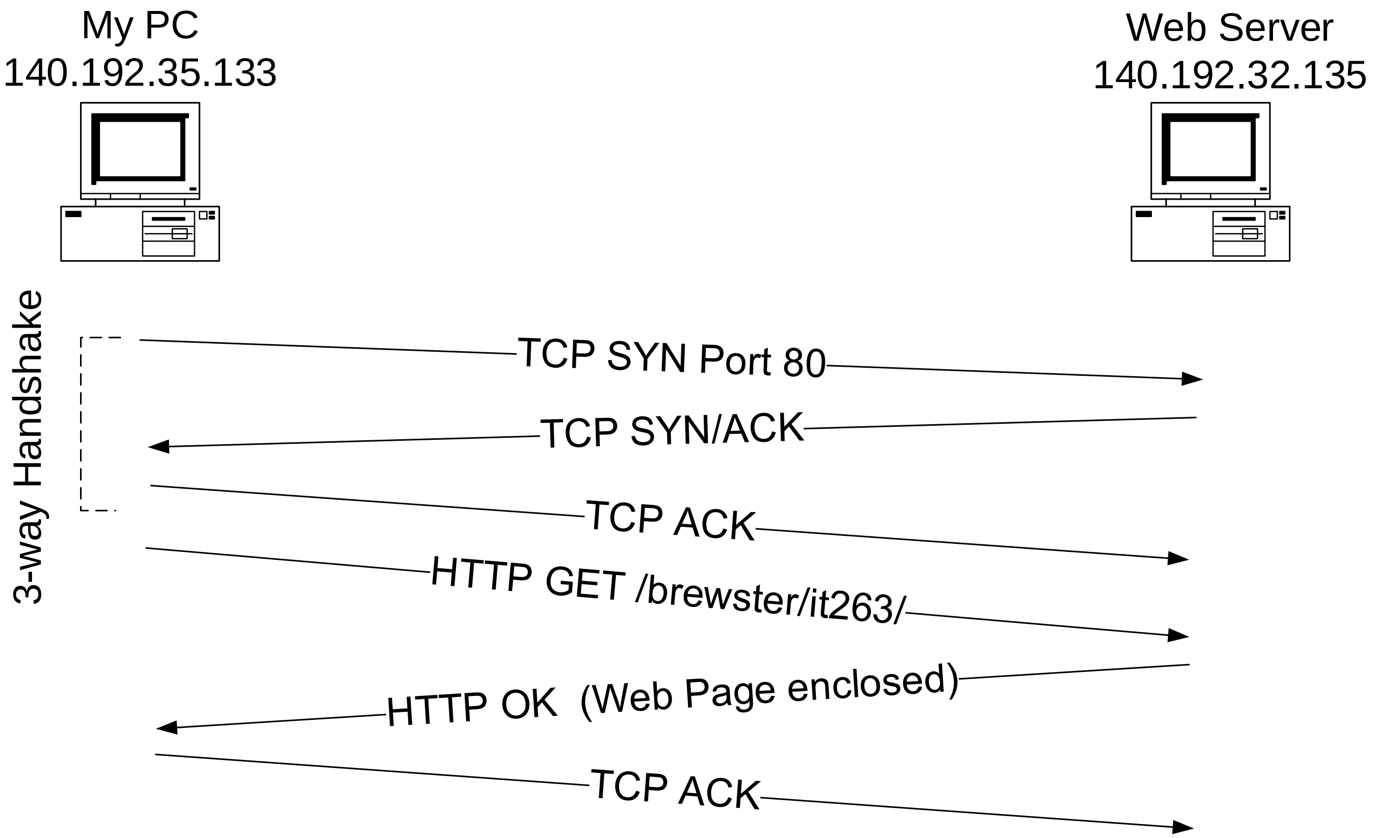
TCP Connection Setup
“3-way Handshake”

- Client and Server each choose a (random) **Initial Sequence Number (ISN)** for their data.
- Client sends “SYN” TCP packet
  - SYN bit = 1
  - Sequence Number = X (client ISN)
- Server sends “SYN/ACK” TCP packet
  - SYN bit = 1, ACK bit = 1
  - Sequence Number = Y (server ISN)
  - Acknowledgment = X+1 (ACKing client ISN)
- Client sends back “ACK” TCP packet
  - ACK bit = 1
  - Sequence Number = X+1
  - Acknowledgment = Y+1 (ACKing server ISN)

Connection Setup
“3-way Handshake”



Get Web Page - Packet Flow



My PC
140.192.35.133

Web Server
140.192.32.135

3-way Handshake

TCP SYN Port 80

TCP SYN/ACK

TCP ACK

HTTP GET /brewster/it263/

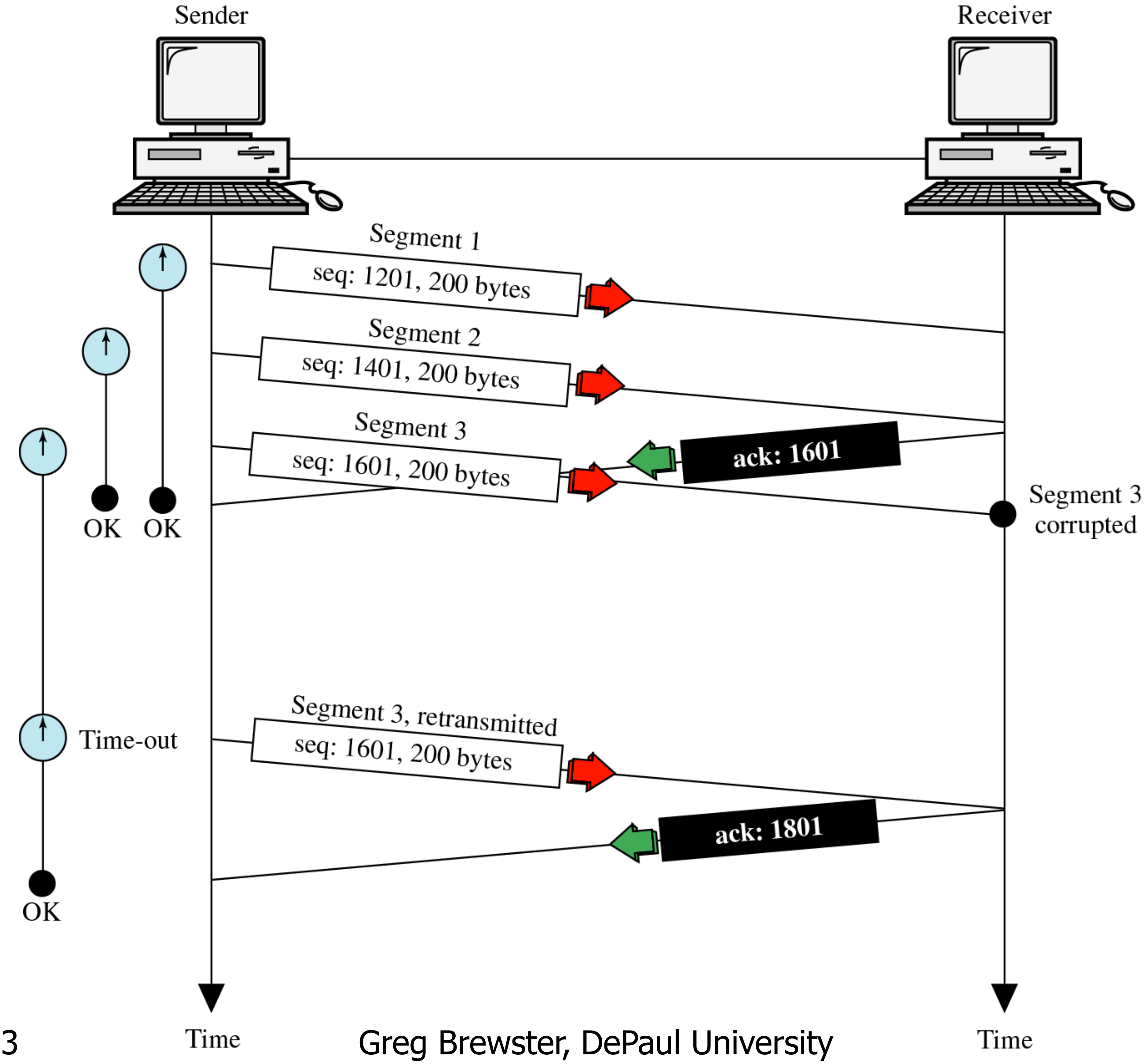
HTTP OK (Web Page enclosed)

TCP ACK

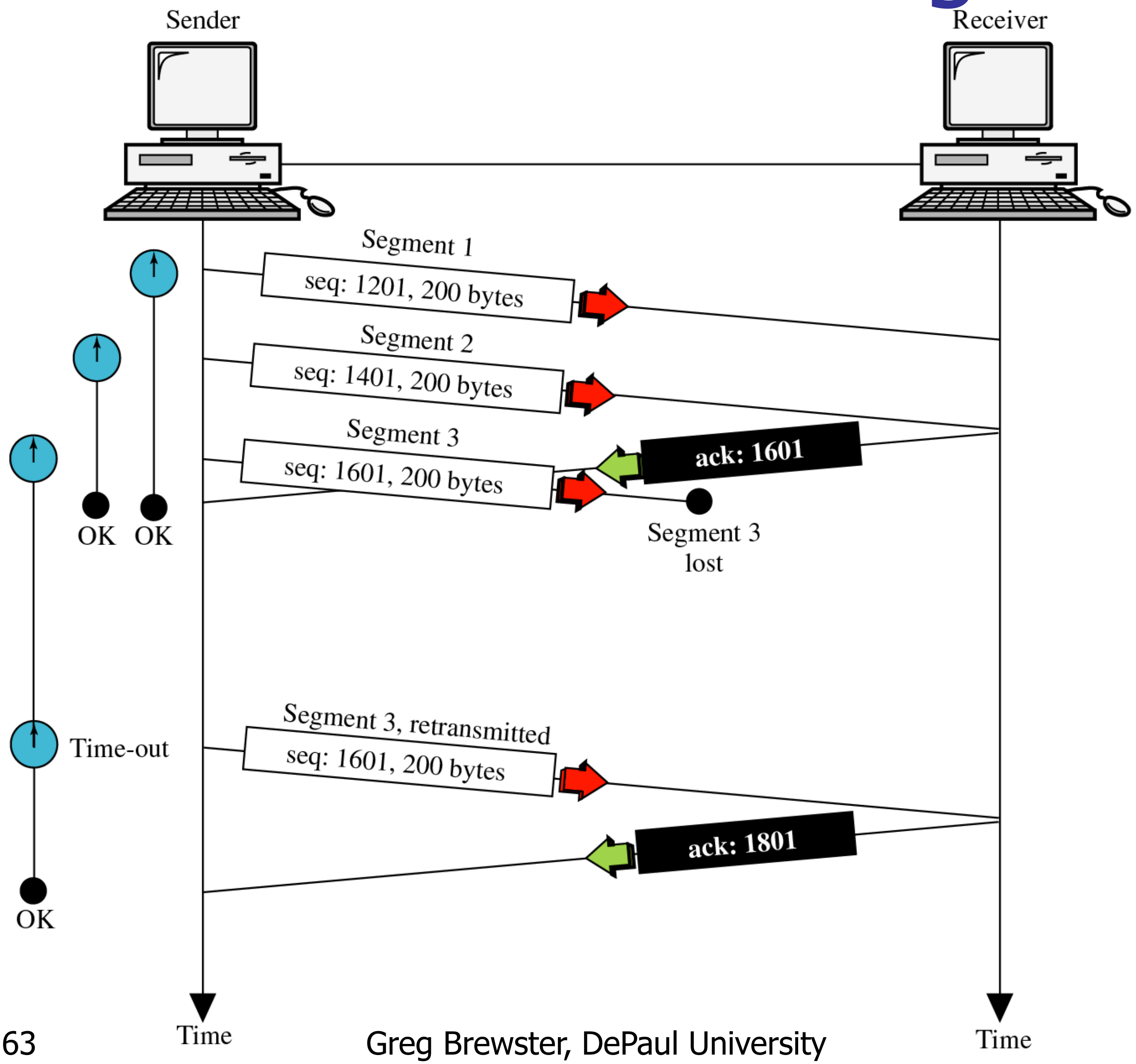
TCP Error Control

- If Receiver gets a corrupted or duplicated segment, it is discarded.
- Sender starts a Retransmission Timer for each transmitted segment
  - If ACK is received for this data before timer expires, then cancel timer
  - If Selective ACK is received indicating data was lost, then retransmit segment and restart timer
  - If timer expires before any ACK is received, then retransmit all data segments starting with timeout segment (Go-Back-N), and restart timer.

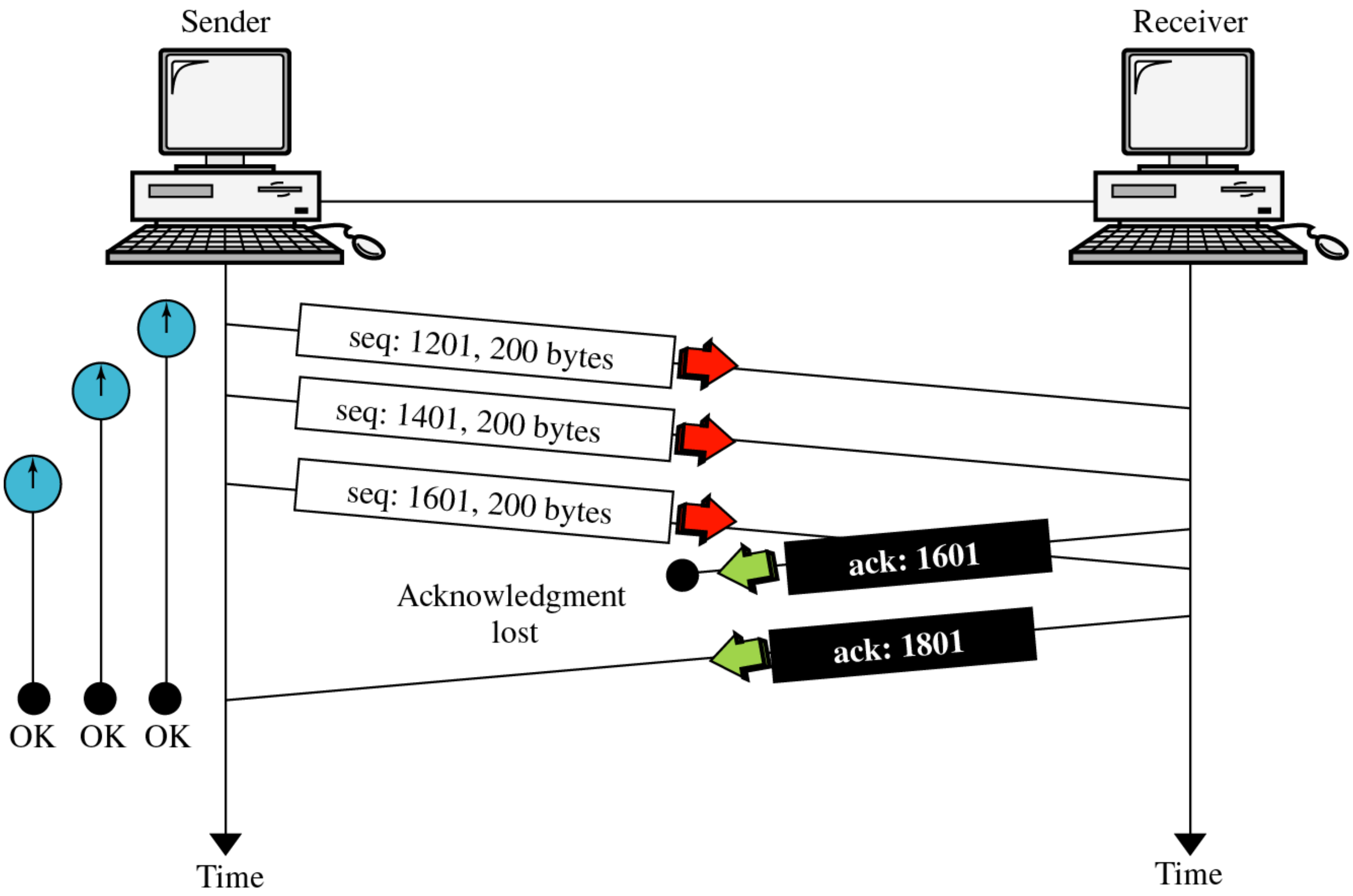
Error Control – Corrupted Segment



Error Control – Lost Segment



Error Control – Lost ACK



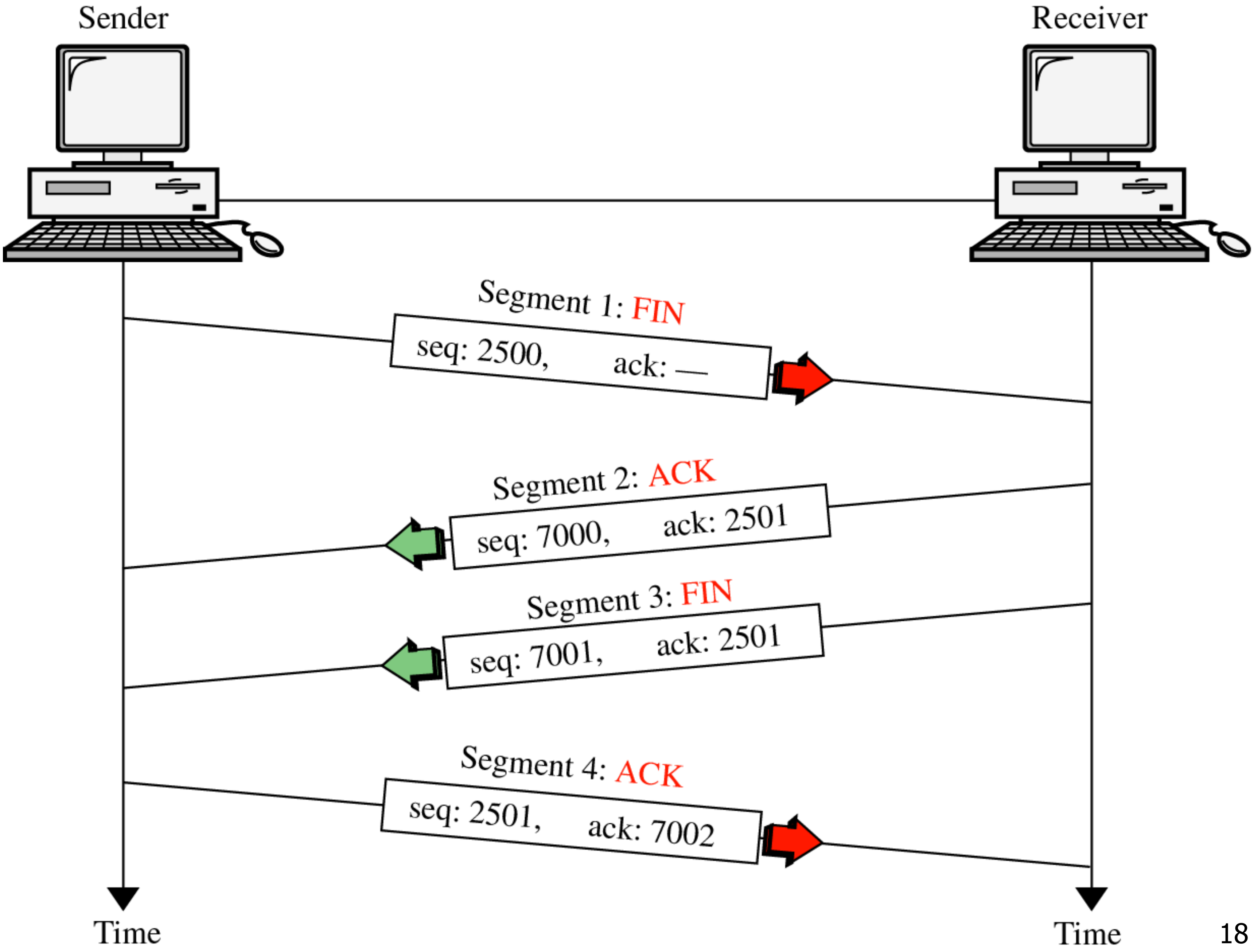
TCP Flow Control

- Flow Control prevents a TCP sender from overwhelming a receiver with too much data
- The TCP receiver sets the **Window** field of each Acknowledgement packet to hold the maximum number of additional bytes that can be sent beyond the Acknowledgement value.
  - **Ack Field** = next sequence number expected
  - **Window Field =** additional bytes that can be sent
  - **(Ack + Window) =** maximum sequence number that can be sent

TCP Connection Teardown

- A TCP connection consists of two 1-way data connections.
- For each 1-way connection:
  - Sender terminates communications by setting FIN bit = 1
  - Receiver acknowledges teardown by setting ACK bit = 1
- A TCP connection is not terminated until **both** 1-way connections are torn down
- This is a 4-way handshake (2 messages for each direction)

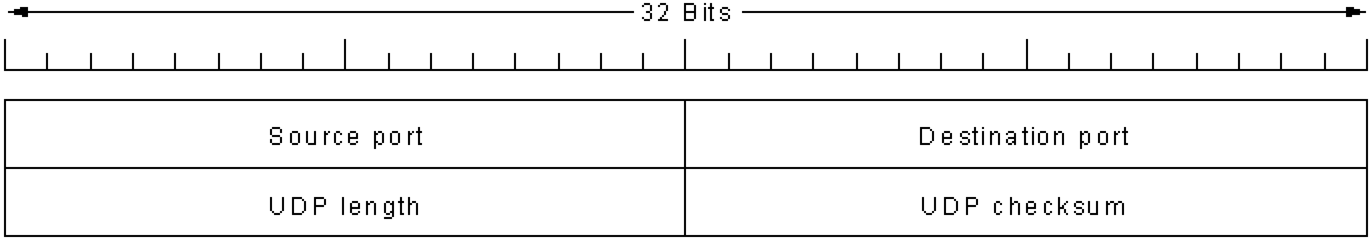
Figure 12-23


Sender
Receiver
Segment 1: FIN
seq: 2500, ack: —
Segment 2: ACK
seq: 7000, ack: 2501
Segment 3: FIN
seq: 7001, ack: 2501
Segment 4: ACK
seq: 2501, ack: 7002
Time
Time

User Datagram Protocol

- The User Datagram Protocol (UDP) is typically used by application programs that do not need Error Control
- UDP is a Transport Layer protocol
- Applications that use UDP rather than TCP: streaming audio or video transfer, network management applications
- UDP adds 8 bytes of ***UDP Header***

UDP Header



32 Bits

Source port	Destination port
UDP length	UDP checksum

Basically – just the port numbers.

TCP vs. UDP

- TCP provides end-to-end error checking and flow control. UDP does not.
- Applications that use TCP/IP
  - HTTP to access web pages
  - SMTP to send e-mail
- Applications that use UDP/IP
  - Streaming audio or video
  - Polling the status of a device