UDP: User Datagram Protocol [RFC 768]
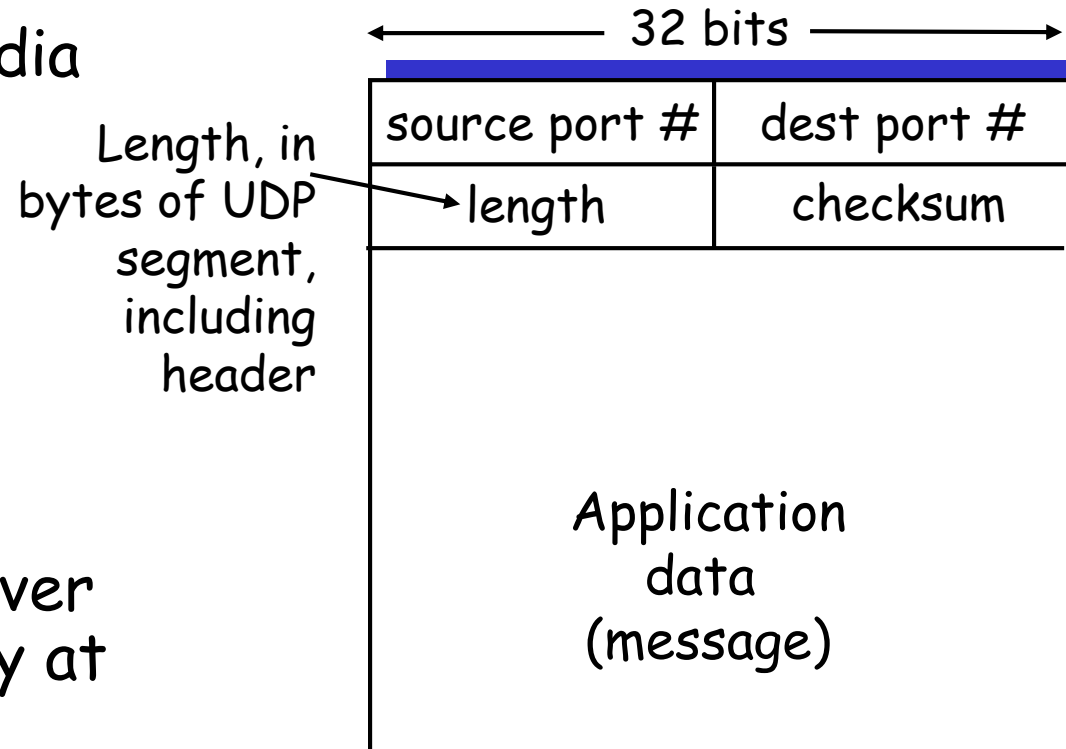
- "no frills," "bare bones" Internet transport protocol
- "best effort" service, UDP segments may be:
  - lost
  - delivered out of order to app
- *connectionless:*
  - no handshaking between UDP sender, receiver
  - each UDP segment handled independently of others

Why is there a UDP?

- no connection establishment (which can add delay)
- simple: no connection state at sender, receiver
- small segment header
- no congestion control: UDP can blast away as fast as desired

UDP (cont'd)

- often used for streaming multimedia apps
  - loss tolerant
  - rate sensitive

- other UDP uses
  - DNS
  - SNMP

- reliable transfer over UDP: add reliability at application layer
  - application-specific error recovery!

Length, in bytes of UDP segment, including header

32 bits	
source port #	dest port #
length	checksum
Application data (message)	

UDP segment format

UDP Checksum

Goal: detect "errors" (e.g., flipped bits) in transmitted segment

Sender:

- treat segment contents as sequence of 16-bit integers
- checksum: addition (1's complement sum) of segment contents
- sender puts checksum value into UDP checksum field

Receiver:

- compute checksum of received segment
- check if computed checksum equals checksum field value:
  - NO - error detected
  - YES - no error detected. But maybe errors nonetheless? More later ….

Checksum: Example (from book)

arrange data segment
in sequences of
16-bit words

0110011001100000

0101010101010101

\+   1000111100001100

binary addition,
*with overflow
wrapped around*

sum: 0100101011000010

checksum(1's complement):   1011010100111101

verify by adding: 1111111111111111