CSc 461/561
Multimedia Systems
Review on TCP/IP Networking

Jianping Pan

Spring 2015

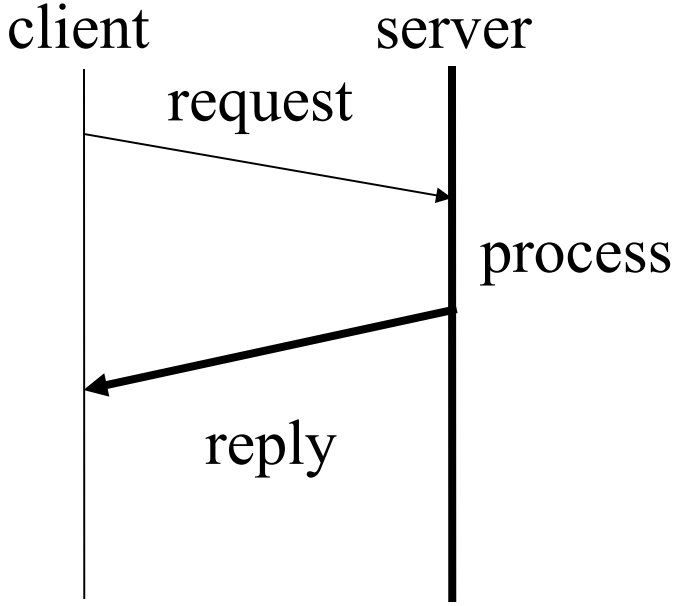
Application-oriented view

- (Old) Applications
  - remote login: e.g., telnet
  - file transfer: e.g., ftp
  - electronic mail: e.g., email
  - world-wide web: the Web!
- Requirements
  - move data from one location to another
  - elastic, error-free, in-sequence

* now: social networking, online audio/video streaming/gaming, etc

Client-server applications

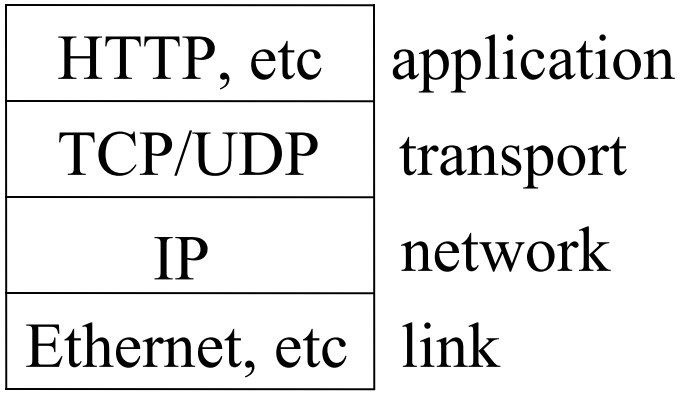
- E.g., HTTP
  - HTTP client (browser)
    - **GET** /index.html HTTP/1.1
    - Host: www.example.com
    - (parameters)
  - HTTP server (Web server)
    - HTTP/1.1 **200 OK**
    - (metadata)
    - (data)



* now: C/S->CDN, P2P, cloud, hybrid, etc, but mostly based on HTTP

Protocols to support

- TCP/IP
  - the Internet Protocol Suite
- TCP offers
  - connection oriented
  - reliable, in-sequence, stream-like data transfer
- IP offers
  - addressing and routing; connectionless
  - IP packets may be lost, corrupted, duplicated, reordered



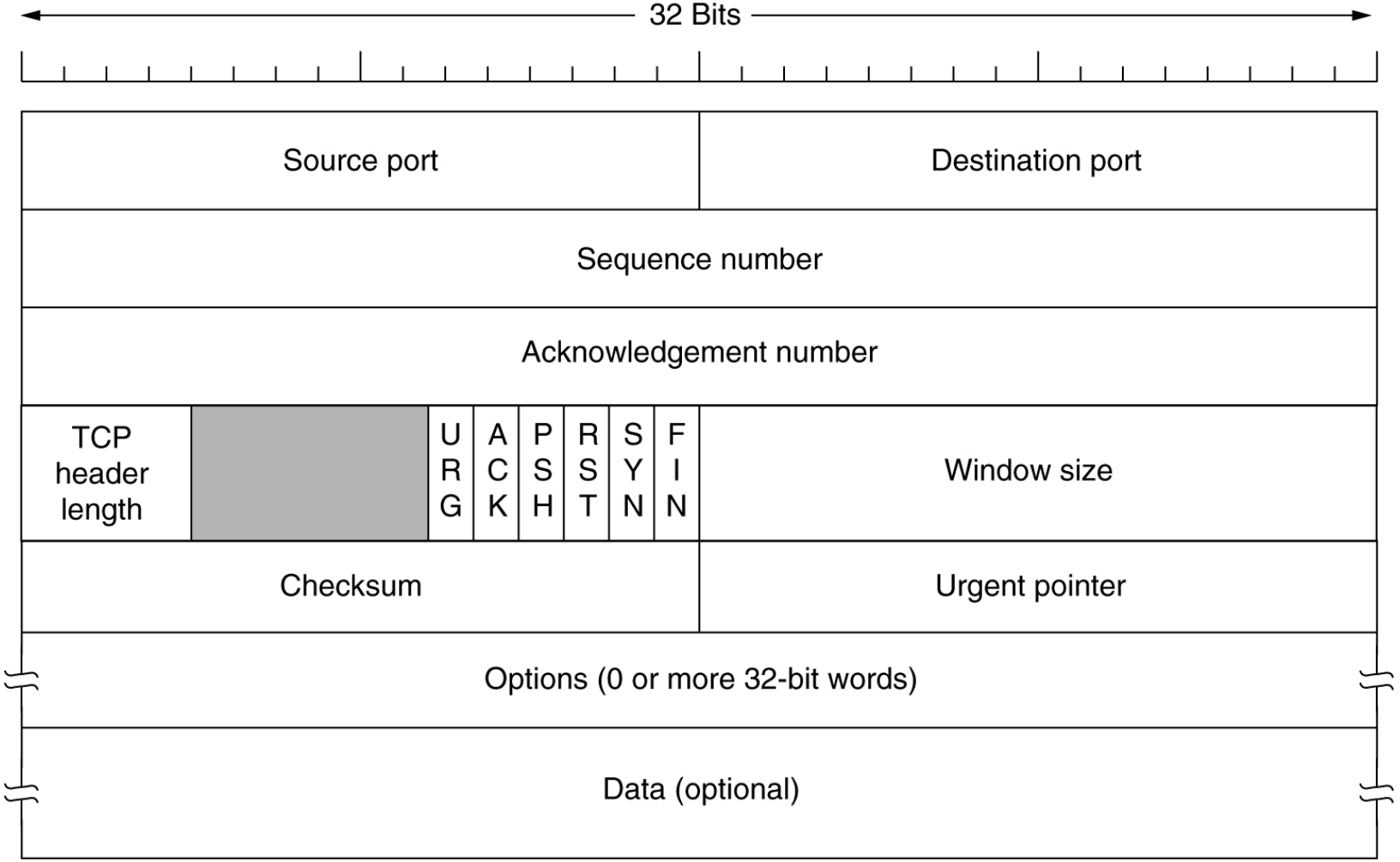
* and still mostly based on TCP/IP

TCP

- Connection management
  - through packet handshake (SYN, FIN, ACK)
  - Multiplexing: port number
- Flow, error, congestion control
  - sequence number
  - acknowledgment number
  - window size
  - checksum

* Transmission Control Protocol, since 1974

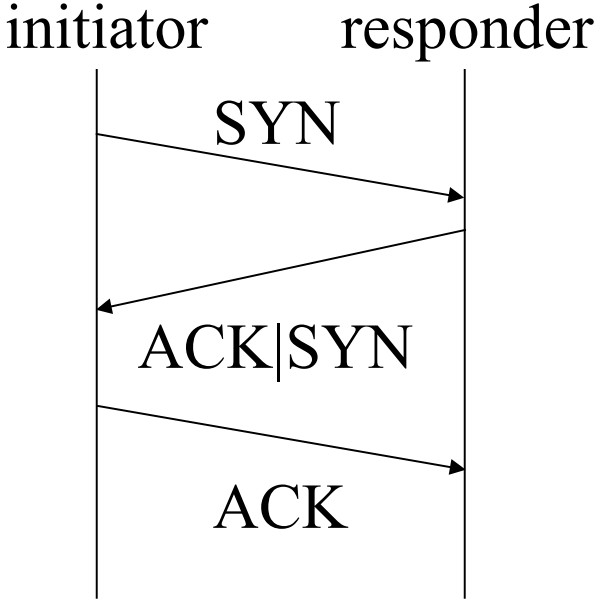
TCP header



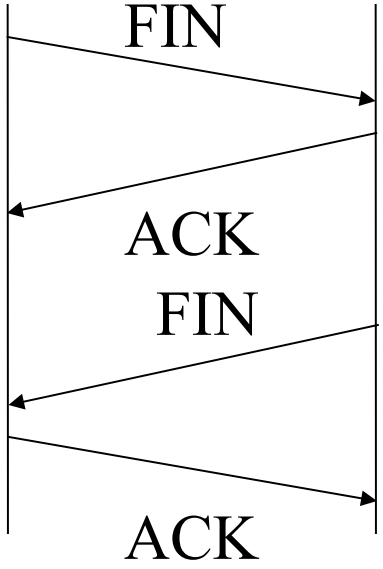
* reserved/unused fields? Urgent Point?

Packet handshake

connection establishment



connection release



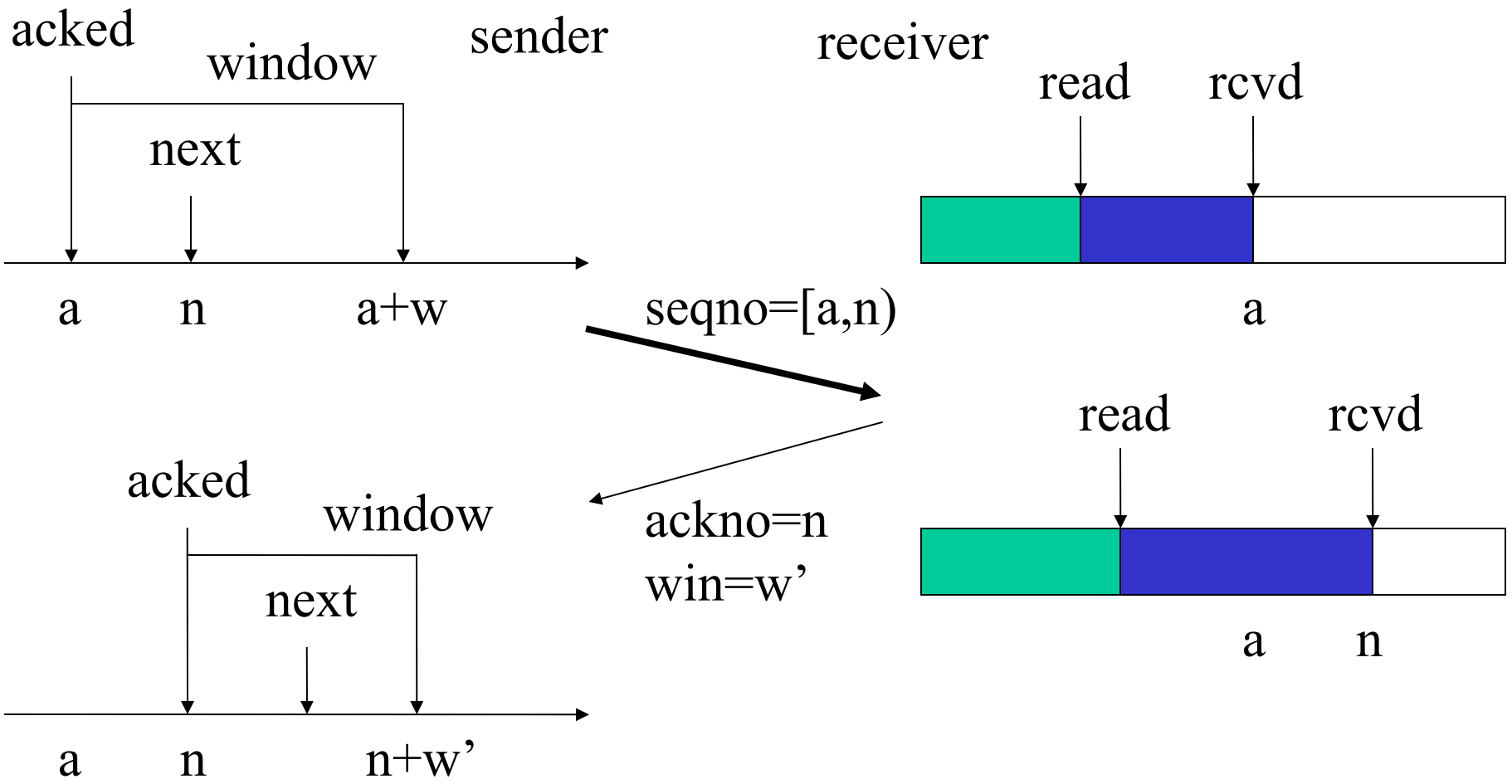
* overhead for short connections and HTTP/0.9->1.0->1.1, 2.0?

Flow control

- Purpose: pace sender and receiver
  - according to their buffer size
- Receiver's (advertised) window
  - available buffer space
- Sender's window
  - sliding window-based flow control
  - only send data within the window
  - sender's window < receiver's window

* other type of flow control: rate-based, better for MM applications?

Sliding window



Error control

- Error detection
  - receiver: sequence number, TCP checksum
  - sender: timeout
- Error notification
  - receiver=>sender: duplicate acknowledgment
- Error recovery
  - sender: end-to-end retransmission

* does MM applications need full reliability?

Go-back-N retransmission

- Sender: send packet 1, 2, 3, 4, 5, 6
- Receiver: receive packet 1, 2, 3, 5, 6
  - cumulatively acknowledge up to packet 3
- Sender: send packet 7, 8, 9
  - timeout for packet 4; retransmit packet 4
- Receiver
  - cumulatively acknowledge up to packet 9
- Sender: send packet 10, 11, 12, ...

* data may come too later or in duplication?

Congestion control

- Was not there when TCP was first designed
- Added to TCP since late 80s
  - heavily coupled with flow/error control
  - heavily explored research topics in decades!
- Purpose: pace sender and network
  - competing flows
  - overload network (e.g., output queue)
  - cause packet losses when queues overflow

* the centerpiece of Internet research, development and evolution

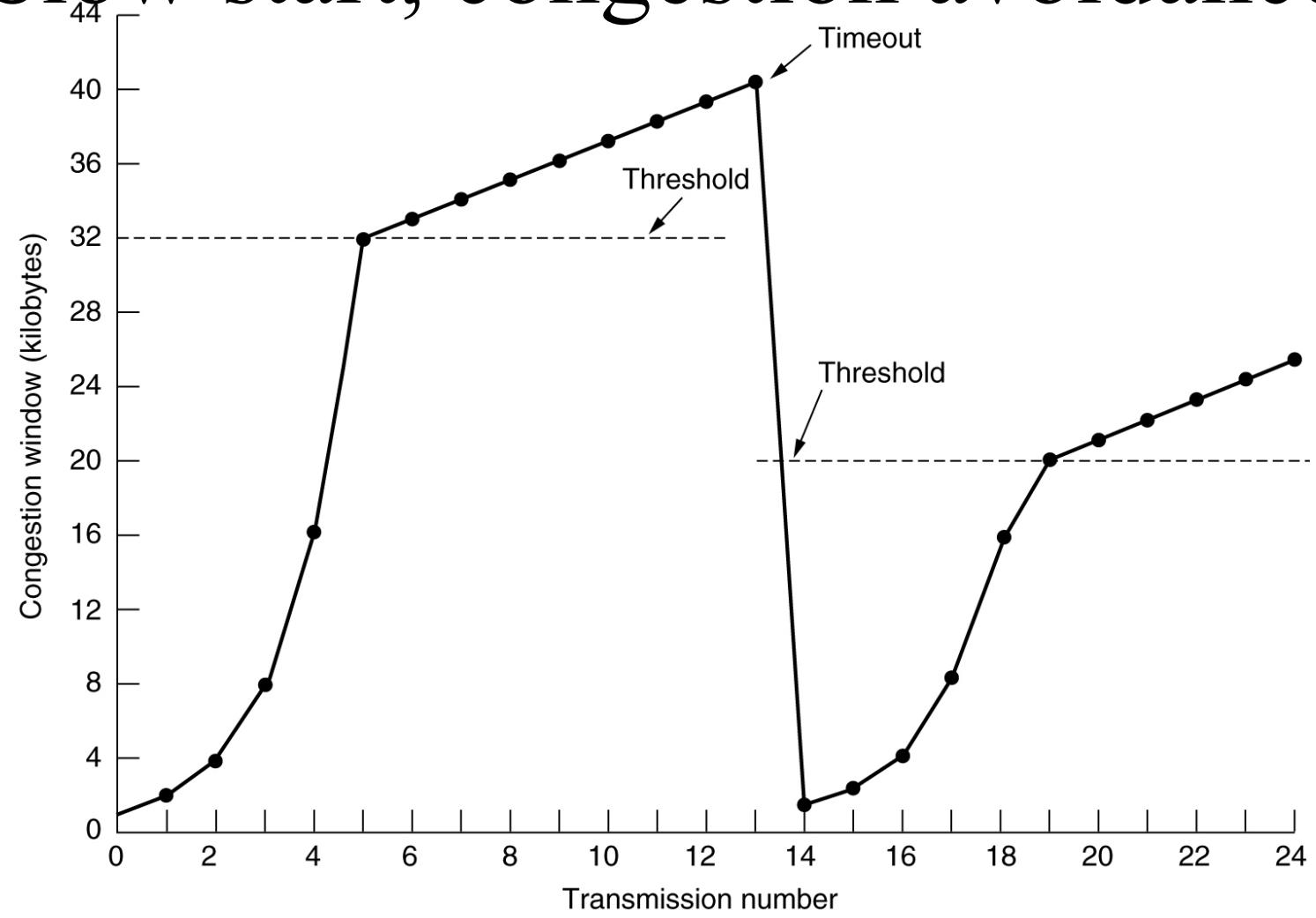
Congestion window

- Slow start
  - start with an initial congestion window (cwnd)
  - usually initial cwnd = 1 full-size packet
  - double window size every round-trip time until cwnd is above slow start threshold (ssthresh)
- Congestion avoidance
  - increase window size linearly

Congestion window: more

- Back-off
  - when timeout occurs, assume packet loss happened [error control]
  - ssthresh = 0.5 * current_cwnd
  - restart with the initial cwnd
  - retransmit with doubled timer [error control]

Slow start, congestion avoidance



* consequence on multimedia traffic?

Congestion control: more

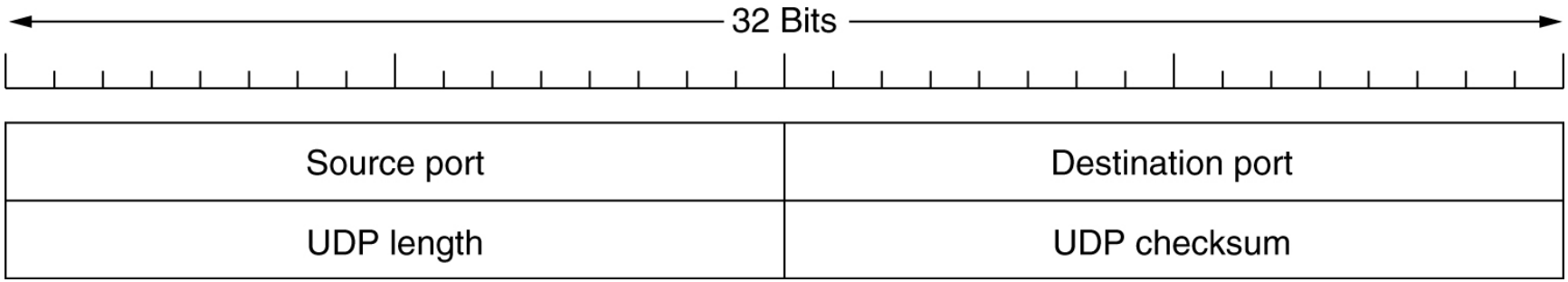
- Fast retransmit
  - retransmit with 3 duplicate acknowledgments
  - slow start threshold (ssthresh) to be a half of current congestion window (cwnd)
  - restart with the initial congestion window
- Fast recovery
  - similar to Fast retransmit
  - but restart with cwnd = ssthresh

UDP

- Why TCP is not enough?
  - sometimes TCP is an overkill
  - e.g., loss-tolerant, delay-sensitive applications
- UDP offers
  - connectionless, datagram-like data transfer
  - no reliability, in-sequence guarantee
  - *flexibility* to plug-in flow/error/congestion etc in application layers

* originally intended for packet audio/video

UDP header



* very small overhead too!

This lecture

- A quick review on
  - Internet Protocol Suite
  - TCP
  - UDP
- Explore further
  - Why TCP header has no “TCP length” field such as “UDP length” in UDP header?

* connex->wiki permissions issue has been fixed

Next lecture

- IP
- Link: wired and wireless
- Why multimedia networking is different, and difficulty?