

# CSCE 515: Computer Network Programming ----- TCP Details

Wenyuan Xu

Department of Computer Science and  
Engineering  
University of South Carolina

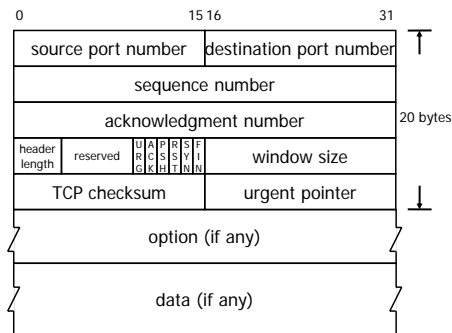
## TCP

- TCP provides the end-to-end reliable connection that IP alone cannot support
- The TCP protocol
  - Segment format
  - Connection Creation
  - Flow control
  - Congestion control
  - Connection termination

9/15/2008

CSCE515 – Computer Network Programming

## TCP Segment Format



9/15/2008

CSCE515 – Computer Network Programming

## More...

- Every TCP segment includes a *Sequence Number* that refers to the first byte of *data* included in the segment.
- Every TCP segment includes a *Request Number (Acknowledgement Number)* that indicates the byte number of the next data that is expected to be received.
  - All bytes up through this number have already been received.

9/15/2008

CSCE515 – Computer Network Programming

## And more...

- MSS: Maximum segment size (A TCP option)
- Window: Every ACK includes a Window field that tells the sender how many bytes it can send before the receiver will have to toss it away (due to fixed buffer size).

9/15/2008

CSCE515 – Computer Network Programming

## And more...

- There are a bunch of control flags:
  - URG: urgent data included.
  - ACK: this segment is (among other things) an acknowledgement.
  - RST: error - abort the session.
  - SYN: Used to establish connection; synchronize Sequence Numbers (setup)
  - FIN: polite connection termination.

9/15/2008

CSCE515 – Computer Network Programming

## TCP Lingo

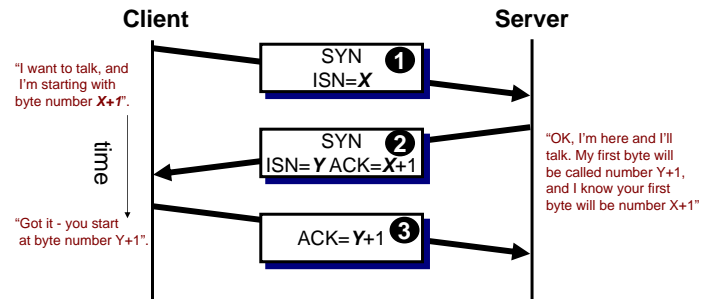
- When a client requests a connection, it sends a “SYN” segment (a special TCP segment) to the server port.
- SYN stands for *synchronize*. The SYN message includes the client’s ISN.
- ISN is Initial Sequence Number.

9/15/2008

CSCE515 – Computer Network Programming

## TCP Connection Establishment

– Three-way handshake



9/15/2008

CSCE515 – Computer Network Programming

## TCP Connection Creation

- A *server* accepts a connection.
  - Must be looking for new connections!
- A *client* requests a connection.
  - Must *know* where the server is!

9/15/2008

CSCE515 – Computer Network Programming

## Client Starts

- A client starts by sending a SYN segment with the following information:
  - Client’s ISN (generated pseudo-randomly)
  - Maximum Receive Window for client.
  - Optionally (but usually) MSS (largest datagram accepted).
  - No payload! (Only TCP headers)

9/15/2008

CSCE515 – Computer Network Programming

## Server Response

- When a waiting server sees a new connection request, the server sends back a SYN segment with:
  - Server’s ISN (generated pseudo-randomly)
  - Request Number is Client ISN+1
  - Maximum Receive Window for server.
  - Optionally (but usually) MSS
  - No payload! (Only TCP headers)

9/15/2008

CSCE515 – Computer Network Programming

## Finally

- When the Server’s SYN is received, the client sends back an ACK with:
  - Request Number is Server’s ISN+1

9/15/2008

CSCE515 – Computer Network Programming

## Why 3-Way?

- Why is the third message necessary?
- HINTS:
  - TCP is a reliable service.
  - IP delivers each TCP segment.
  - IP is not reliable.

9/15/2008

CSCE515 – Computer Network Programming

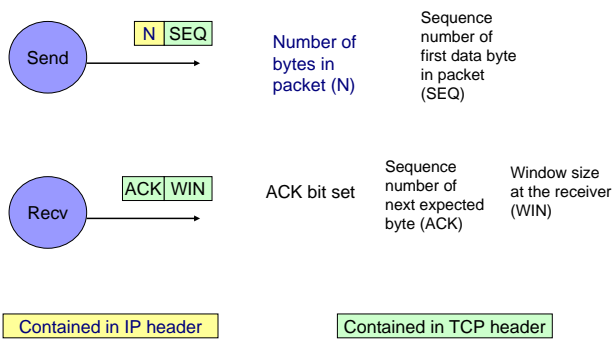
## TCP Data and ACK

- Once the connection is established, data can be sent.
- Each data segment includes a **sequence number** identifying the first **byte** in the segment.
- Each ACK segment includes a **request number** indicating what data has been received. (**bytes** instead of packets)

9/15/2008

CSCE515 – Computer Network Programming

## Important Information in TCP/IP packet headers



9/15/2008

CSCE515 – Computer Network Programming

## Buffering

- The TCP layer doesn't know when the application will ask for any received data.
- Both the client and server **allocate buffers** to hold incoming and outgoing data (TCP does this)
  - incoming data -> it's ready when application asks for it.
  - outgoing data-> it can be retransmitted should error occurs
- Both the client and server announce with every ACK how much buffer space remains (the Window field in a TCP segment).

9/15/2008

CSCE515 – Computer Network Programming

## Send Buffers

- The application gives the TCP layer some data to send.
- The data is put in a send buffer, where it stays until the data is ACK'd.
  - it has to stay, as it might need to be sent again!
- The TCP layer won't accept data from the application unless (or until) there is buffer space.

9/15/2008

CSCE515 – Computer Network Programming

## Recv Buffers

- The received data is put in the recv buffer, where it stays until the application reads data
- The Recv Buffer won't accept data from network unless (or until) there is buffer space, and the sequence number is within its sliding windows.
- The recv buffer can store out of order data as long as there is buffer space and the sequence number is within its sliding windows

9/15/2008

CSCE515 – Computer Network Programming

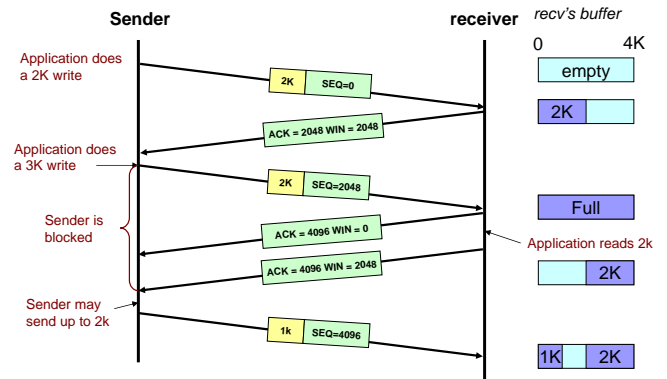
## ACKs

- A receiver doesn't have to ACK every segment (it can ACK many segments with a single ACK segment).
- Each ACK can also contain outgoing data (piggybacking).
- If a sender doesn't get an ACK after some time limit (MSL) it resends the data.

9/15/2008

CSC515 - Computer Network Programming

## TCP Flow Control



9/15/2008

CSC515 - Computer Network Programming

## TCP Segment Order

- Most TCP implementations will accept out-of-order segments (if there is room in the buffer).
- Once the missing segments arrive, a single ACK can be sent for the whole thing.
- Remember: IP delivers TCP segments, and IP is not reliable - IP datagrams can be lost or arrive out of order.

9/15/2008

CSC515 - Computer Network Programming

## Termination

- The TCP layer can send a RST segment that terminates a connection if something is wrong.
- Usually the application tells TCP to terminate the connection politely with a FIN segment.

9/15/2008

CSC515 - Computer Network Programming

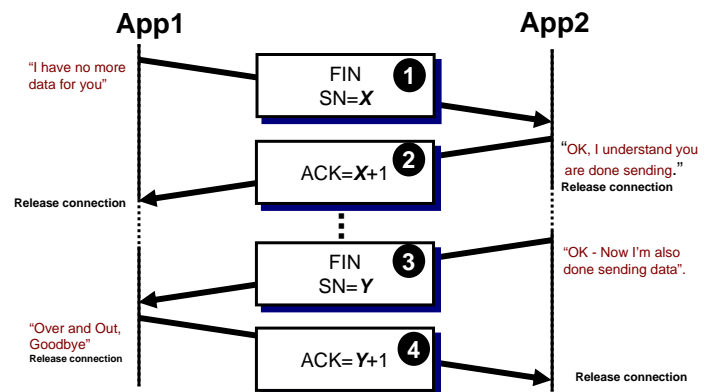
## FIN

- Either end of the connection can initiate termination.
- A FIN is sent, which means the application is done sending data.
- The FIN is ACK'd.
- The other end must now send a FIN.
- That FIN must be ACK'd.

9/15/2008

CSC515 - Computer Network Programming

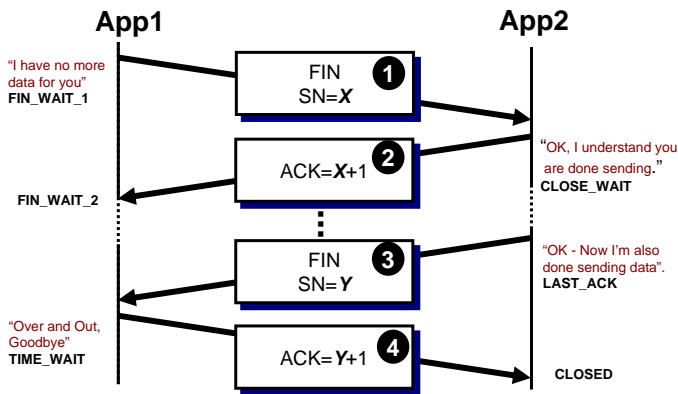
## TCP Termination



9/15/2008

CSC515 - Computer Network Programming

## TCP Termination



9/15/2008

CSCE515 - Computer Network Programming

## TCP TIME\_WAIT

- Once a TCP connection has been terminated (the last ACK sent) there is some unfinished business:
  - What if the ACK is lost? The last FIN will be resent and it must be ACK'd.
  - What if there are lost or duplicated segments that finally reach the destination after a long delay?
- TCP hangs out for a while to handle these situations.

9/15/2008

CSCE515 - Computer Network Programming

## Test Questions

- Why is a 3-way handshake necessary?
- Who sends the first FIN - the server or the client?
- Once the connection is established, what is the difference between the operation of the server's TCP layer and the client's TCP layer?
- What happens if a *bad guy* can guess ISNs?

9/15/2008

CSCE515 - Computer Network Programming

## Assignment & Next time

- Reading:
  - UNP 2.4, 2.6, 2.7 \*\*
- Next Lecture:
  - TCP Socket Programming

9/15/2008

CSCE515 - Computer Network Programming