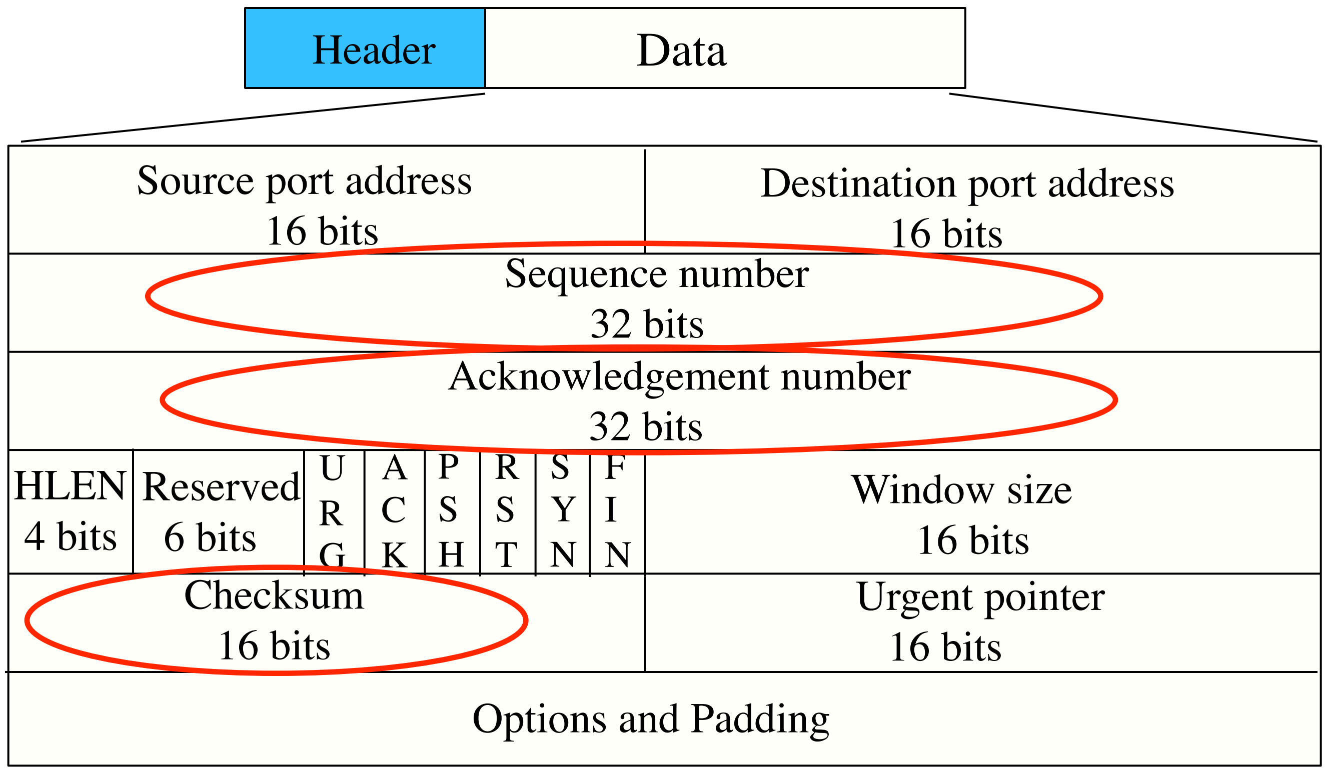
Lecture 20 Overview

- Last Lecture
  - Transport Control Protocol (1)
- This Lecture
  - Transport Control Protocol (2)
  - Source: chapters 23, 24
- Next Lecture
  - Internet Applications
  - Source: chapter 26

Error Control

- TCP is a reliable transport layer protocol
  - Deliver a stream of data in order, without error, and without any part lost or duplicated
  - Provide mechanisms for detecting
    - Lost segments
    - Duplicated segments
    - Out-of order segments
    - Corrupted segments
  - Provide a mechanism for error correction
- Error detection and correction in TCP is achieved by
  - Checksum
  - Acknowledgement
  - Time-out and retransmission

TCP Segment



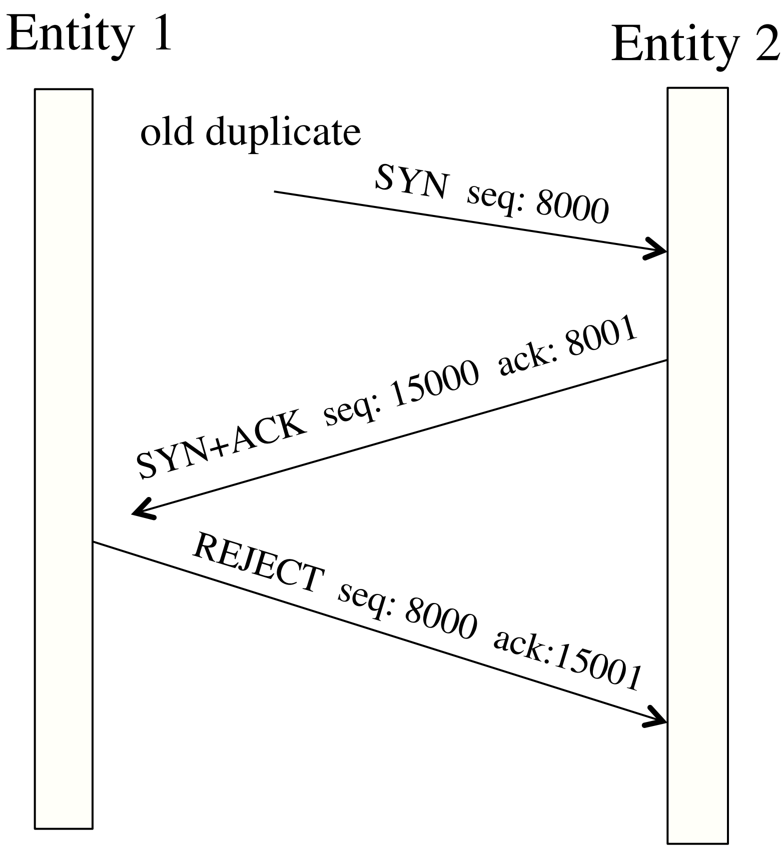
Header
Data
Source port address
16 bits
Destination port address
16 bits
Sequence number
32 bits
Acknowledgement number
32 bits
HLEN
4 bits
Reserved
6 bits
URG
ACK
PSH
RST
SYN
FIN
Window size
16 bits
Checksum
16 bits
Urgent pointer
16 bits
Options and Padding

Error Control (cont.)

- Checksum
  - Each segment includes a checksum field to check for a corrupted segment
  - A corrupted segment is **discarded** and considered as lost
- Acknowledgement
  - Each segment except ACK segment is acknowledged on receipt
- Retransmission
  - Retransmission time-out (RTO) timer: retransmit when the timer expires. The timer is set based on the round-trip time (RTT).
  - Three-duplicate-ACKs rule: retransmit after receiving three duplicate ACK segments. This is applied when the receiver receives many out-of-order segments that can not be buffered. This feature is referred to as **fast retransmission**.

Duplicate Segments

- Duplicate segments can be detected and discarded using sequence number.



The same for data segments

Out-of-order Segments

- Out-of-order segments can be detected using the sequence number.
- Out-of-order segments are not discarded. Instead the receiver maintains a sliding window that temporally buffers the out-of-order segments until the missing segment arrives.
- Out-of-order segments will not be delivered to the process.

Lost/corrupted Segments

- A lost segment and a corrupted segment are treated in the same way by the receiver. Both are considered lost.
- Time-out + Retransmission
  - The sender sets the RTO timer when sending a segment. If the sender does not receive the corresponding acknowledgement when the timer fires, it assumes the segment is lost and retransmits the segment.
- What will happen if the FIN and ACK segments in three-way handshaking for connection termination are lost?

The Two Army Problem

- The set-up
  - A white army is encamped in a valley.
  - On both of the surrounding hillsides are blue armies.
  - The white army is larger than either of the blue armies alone, but together they are larger than the white army.
  - If either blue army attacks by itself, it will be defeated, but if both blue armies attack simultaneously, they will be victorious.
  - The communication medium between the two blue armies is to send messengers on foot down into the valley, where they might be captured and the message lost.

The Two Army Problem (cont.)

- The question
  - Is there a protocol that guarantees the blue armies will win?

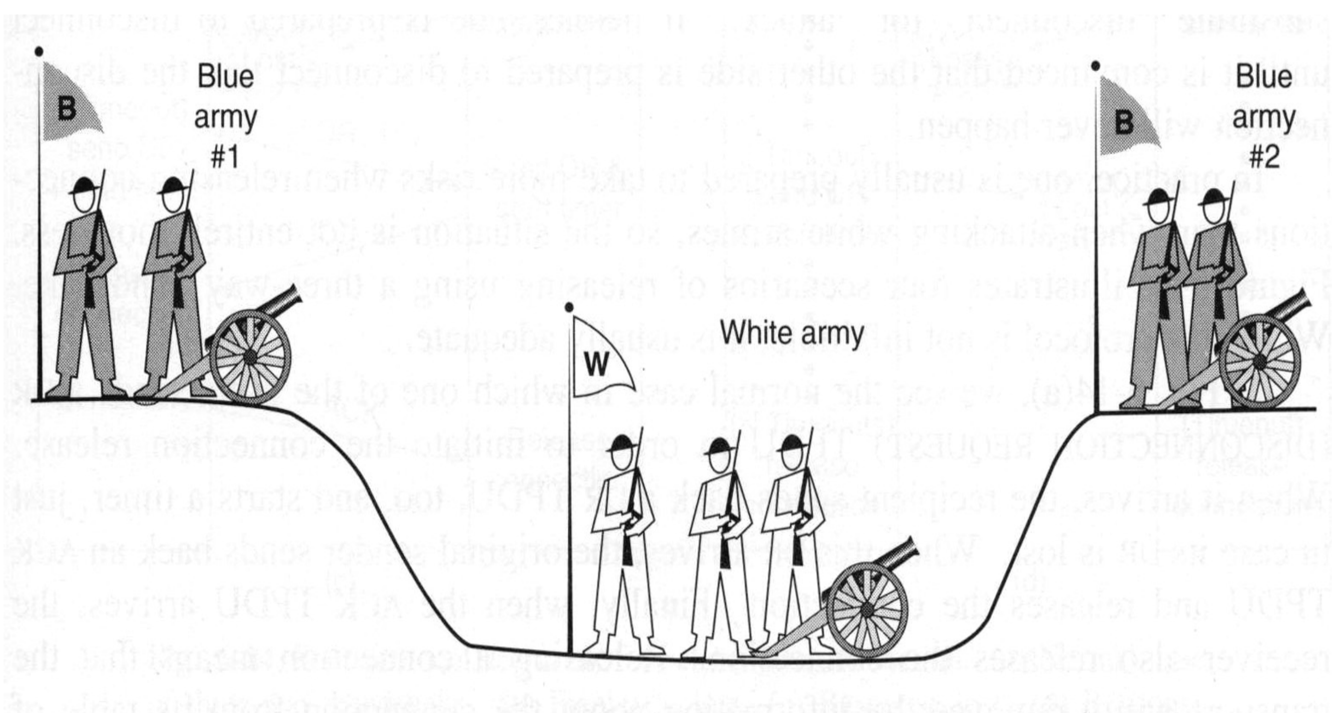


Fig. 6-13. The two-army problem.

Three-way Handshake Disconnect with a Timer

- Timer is used - If there is no TPDU from the other party for some time, disconnect anyway.
- The protocol can fail if all transmissions except the initial connection request are all lost.
  - The sender will give up and delete the connection, while the other side knows nothing about the attempts to disconnect and is still fully active.
- This situation is called half-open connection.

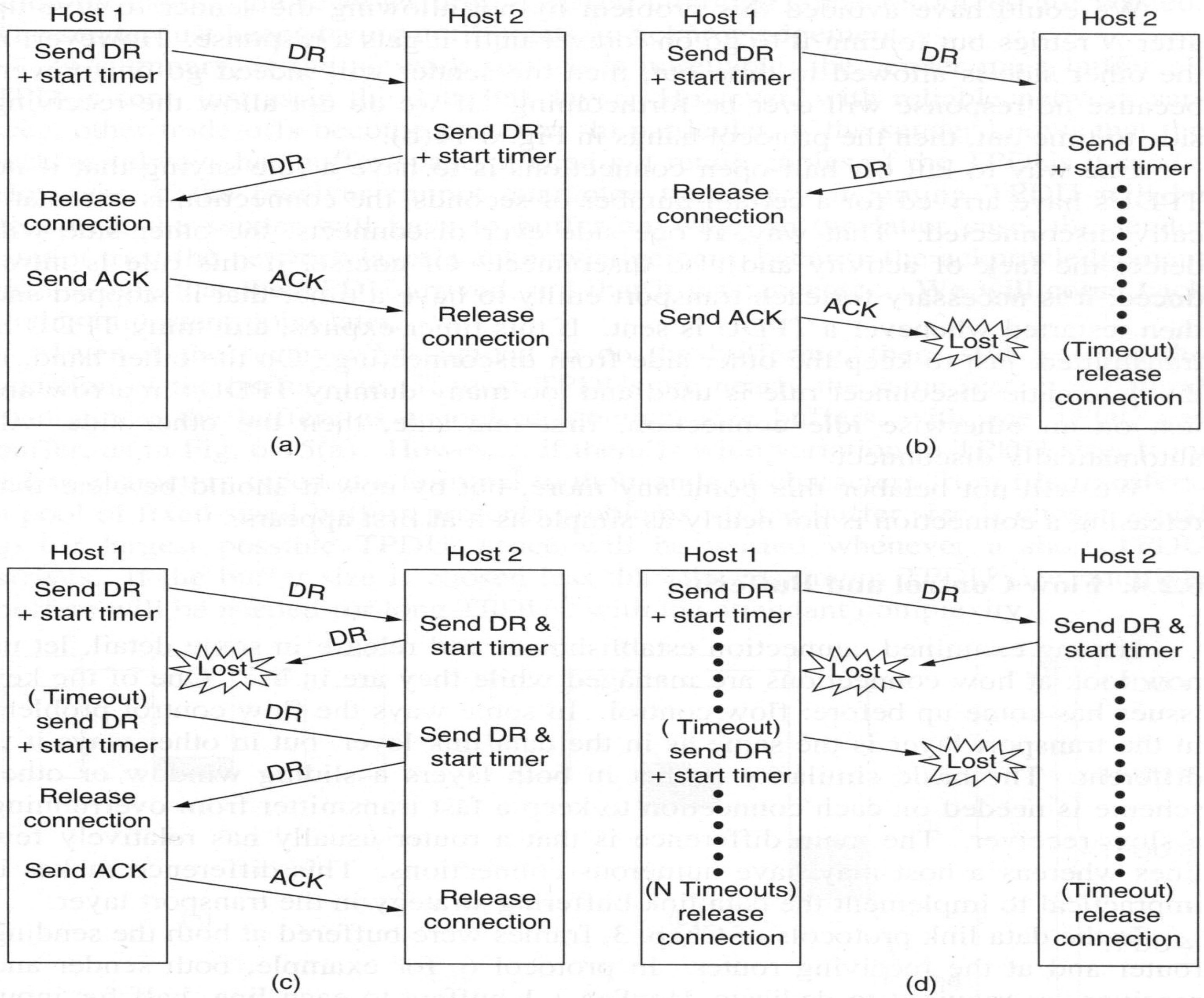


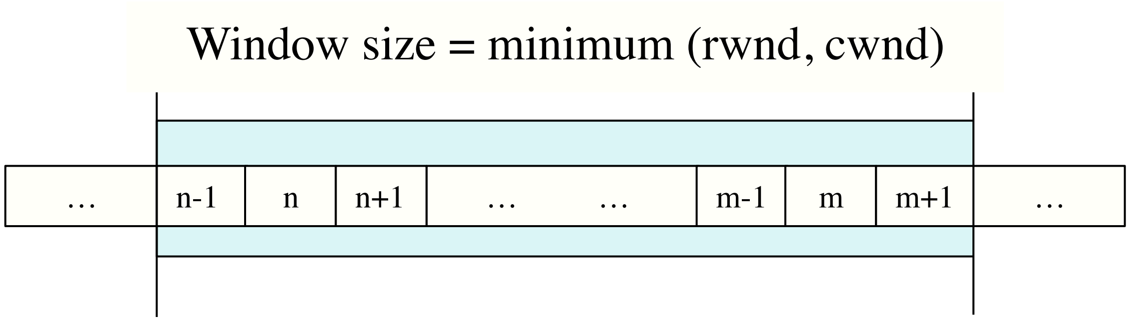
Fig. 6-14. Four protocol scenarios for releasing a connection. (a) Normal case of three-way handshake. (b) Final ACK lost. (c) Response lost. (d) Response lost and subsequent DRs lost.

Flow Control

- Transport protocols resemble the data link protocols
  - Flow control, error control
  - A sliding window is used for flow control
- Why cannot use the same flow control scheme at data link layer?
  - A router usually has only a few links to others, while a transport entity may have numerous connections. This difference makes it impractical to implement a data link buffering strategy in the transport layer.

Flow Control (cont.)

- TCP sliding window
  - is **byte-oriented,** not frame-oriented
  - has a **variable size**.



  - The size of the window is determined by:
    - Receiver window (rwnd)
    - Congestion window (cwnd)

Flow Control (cont.)

- Dynamic buffer management - credit mechanism
  - Initially, the sender requests a certain number of buffers (credit), based on its perceived needs.
  - The receiver grants as many of these as it can afford.
  - Every time the sender transmits a TPDU, it decreases its allocation (credit), stopping when the allocation (credit) reaches zero.
  - The receiver returns both acknowledgments and buffer allocations (credit).
    - May be piggybacked

Flow Control (cont.)

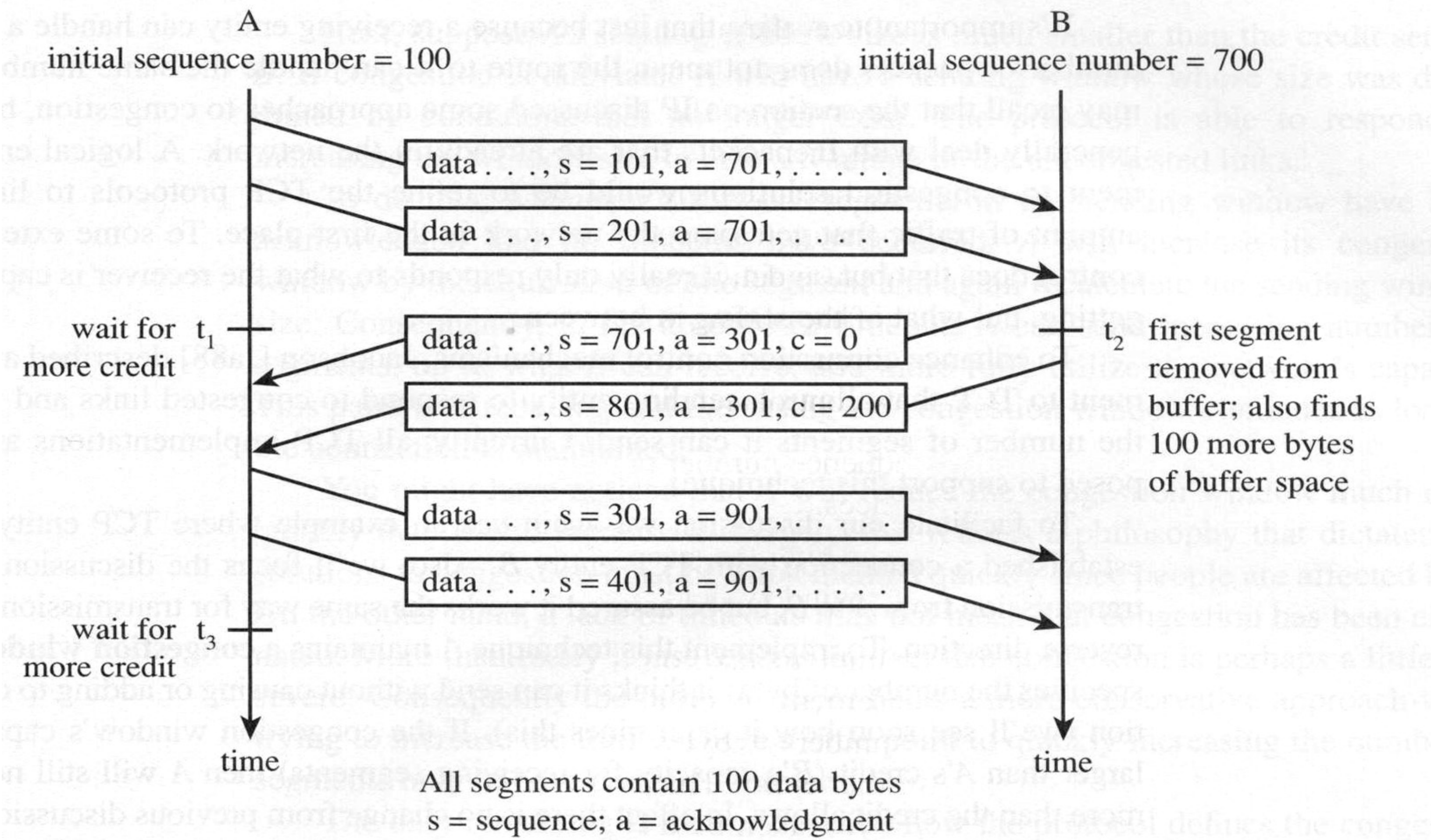


Figure 7.46 Flow Control Using a Credit Mechanism

Flow Control (cont.)

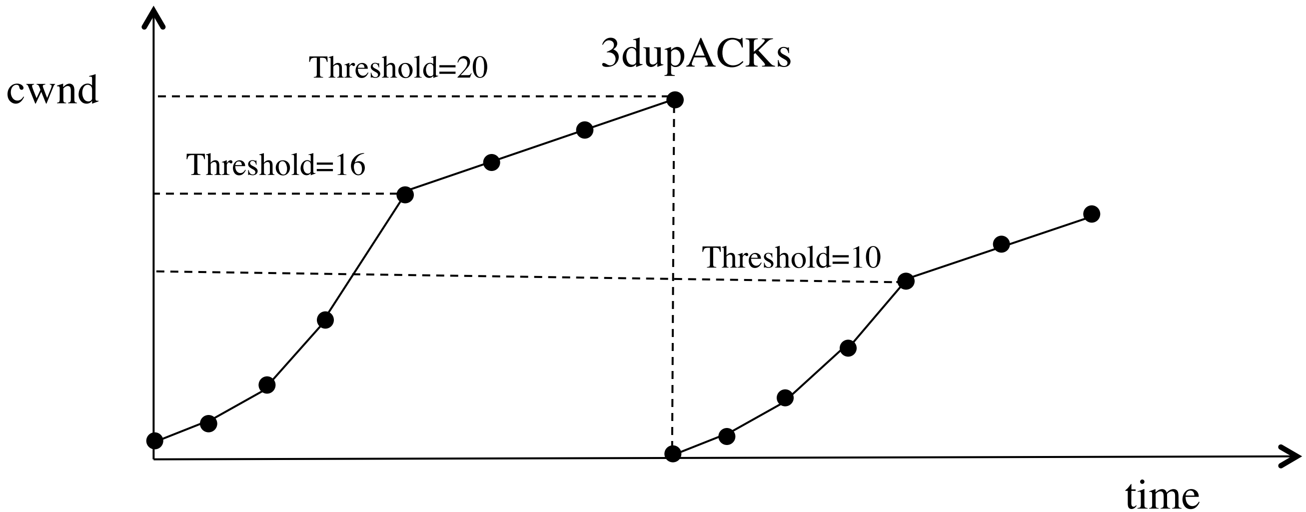
- Assume that when the connection is established, entities A and B have initial sequence numbers 100 and 700 respectively, and each entity agrees to a credit of 200 bytes.
- Assume entities send 100 bytes in each TPDU.
- A starts by sending two TPDUs and then waits because it has used up its credit.
- At time *t2*, B removes the first received TPDU from the buffer and also finds 100 more bytes of buffer space; so B now has 200 bytes space available and tells A in the credit field of a TPDU.
- After A receives the 200 bytes credit, it can send two additional TPDUs.

Congestion Control in TCP

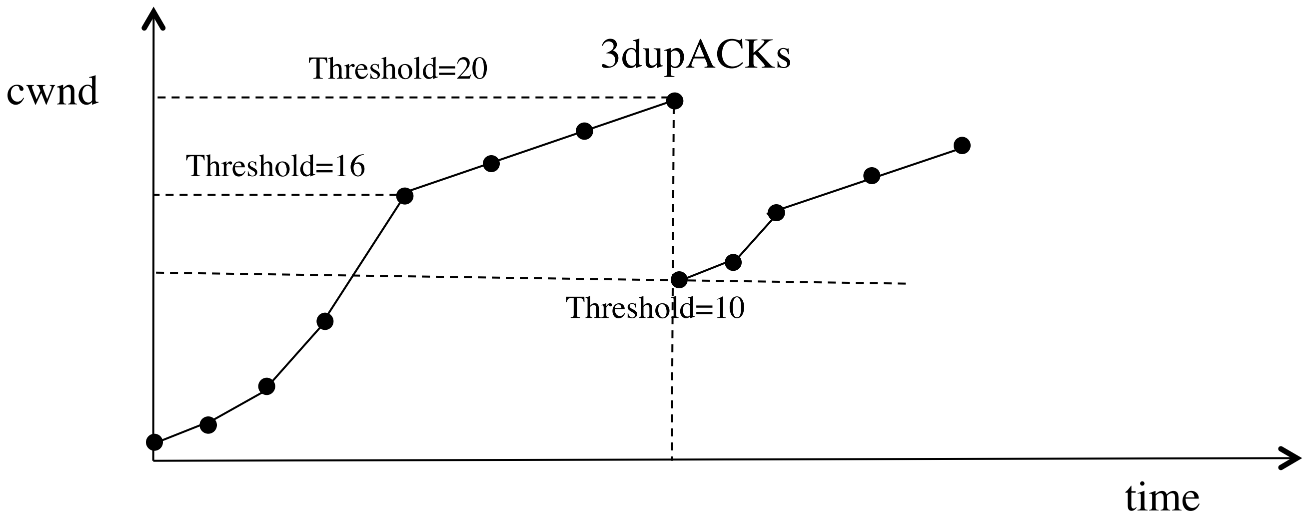
- Congestion policy has three phases:
  - Slow start
    - The cwnd starts with one maximum segment size (MSS)
    - The cwnd increases one MSS each time an ACK is received.
    - Start slowly, but grows **exponentially**.
  - Congestion avoidance
    - Use **additive increase** instead of exponential increase.
    - When the size of the congestion window reaches the slow-start threshold, the slow-start phase stops, and the additive phase begins.
  - Congestion detection
    - **Multiplicative decrease**: the size of the cwnd is dropped to one-half.

Congestion Control in TCP (cont.)

Taho TCP

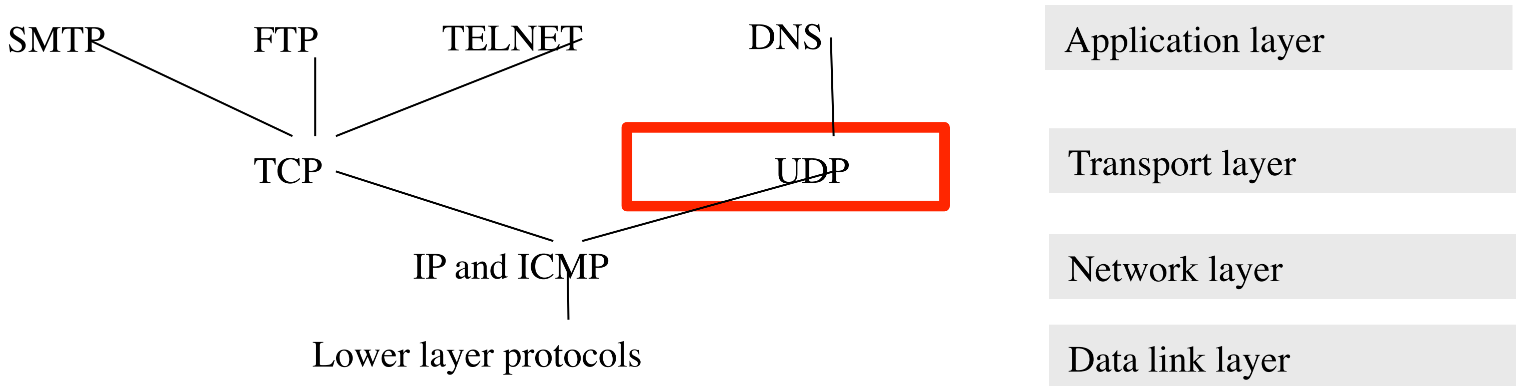


Reno TCP



User Datagram Protocol (UDP)

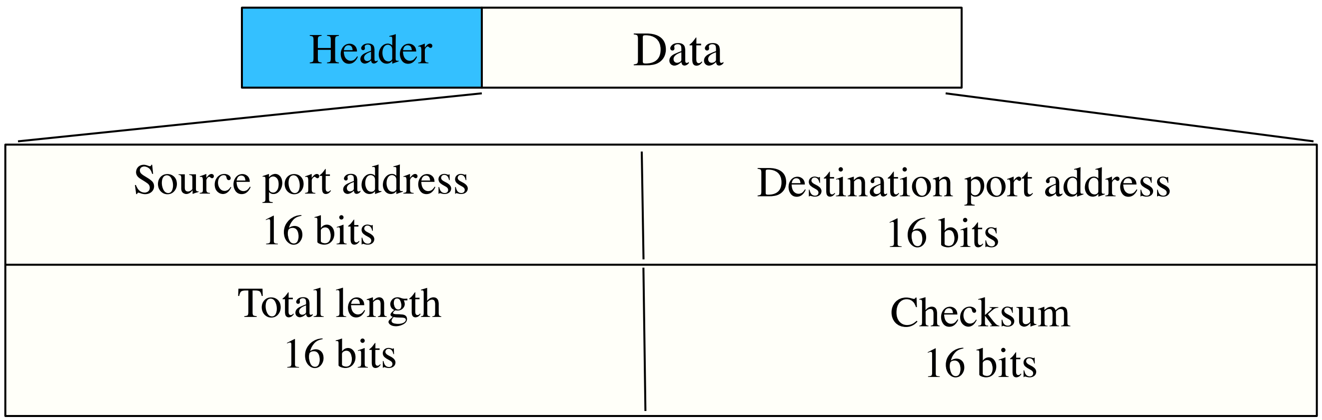
- Connectionless unreliable transport protocol



- Simple interface between IP and higher layer protocols
  - Add nothing to the services of IP except to provide process to process communication.
- Why needs this protocol?
  - A simple protocol using a minimum of overhead
  - Useful in applications that require simple request-response communication with little concern for flow and error control.

User Datagram Protocol (cont.)

- User datagram format



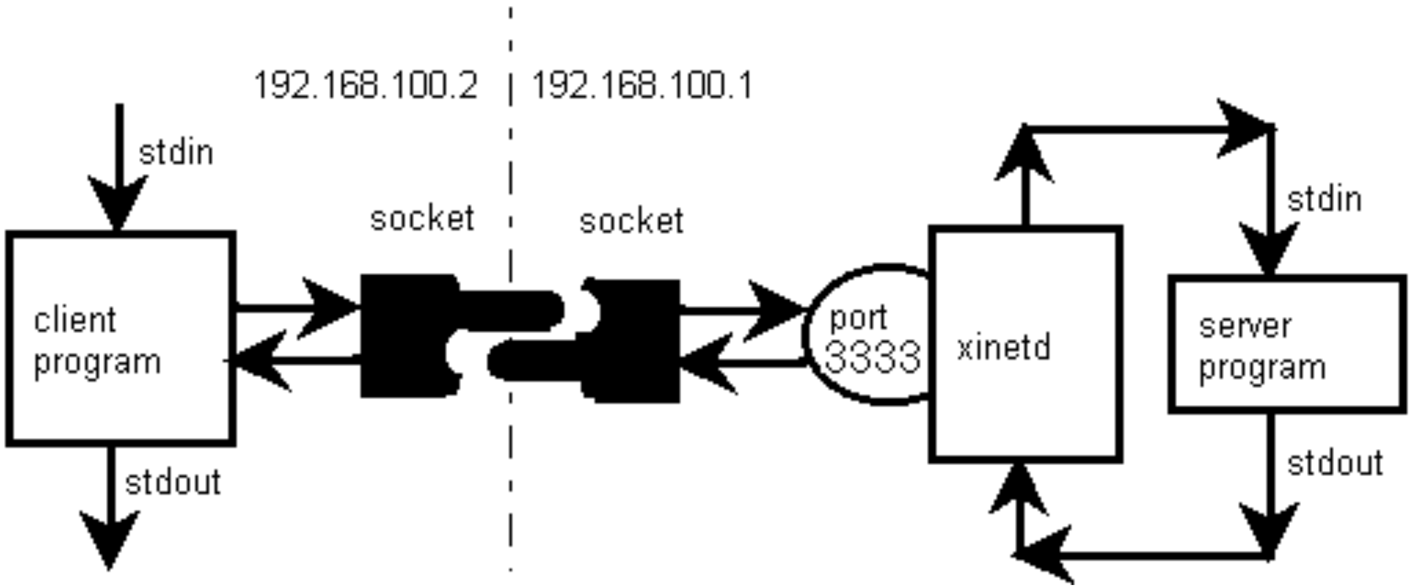
- No flow control mechanism
- No error control mechanism except the checksum

How to Develop the Program?

- Socket Programming
- What is a socket?

  - Sockets represent endpoints in a line of communication.
  - A socket is a **software component** characterized by a unique combination of
    - Local socket address: local IP address and port number
    - Remote socket address: only for TCP sockets
    - Protocol: TCP, UDP

Socket Programming



- Socket Address: the combination of an IP address and a port number (a 16-bit unsigned integer, ranging from 0 to 65535).
- Socket API: an application programming interface, usually provided by the operating system.

Socket Programming (cont.)

- Primitives for socket
  - socket - create a communication end point
  - (bind - attach a local address to a socket)
  - (listen - wait for a connection)
  - accept - accept a connection request
  - (connect - attempt to establish a connection)
  - send/write - send data over the connection
  - recv/read - receive data from the connection
  - close - close the connection

Summary

- Concepts
  - TCP
  - TCP segment and its format
  - Credit mechanism
  - UDP
  - Socket
- Flow control and error control in transport layer
  - Differences of flow control between transport and data link layers
  - Why error control in transport layer
- Sockets