Midterm Exam I

CSE4175: Introduction to Computer Networks

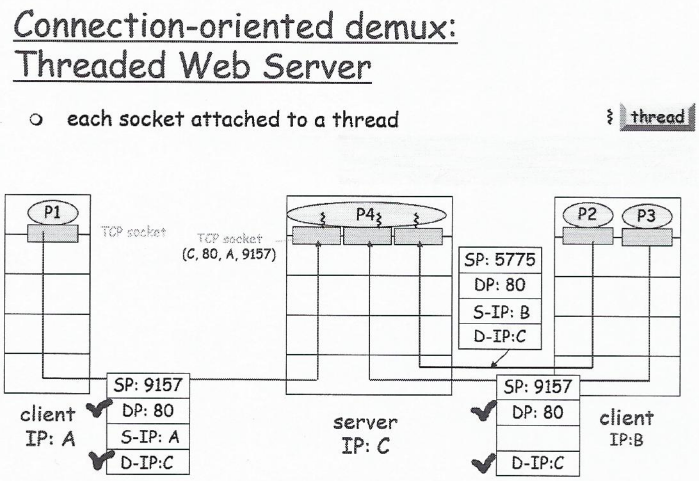
Dept. of Computer Science and Engineering

Spring 2017

Name: ____________________ Student Number: ______________________

Total points 100

Problem 1: (15 points) Transport protocol
We know that UDP socket is identified by two-tuple (dest. IP address, dest port number) but TCP socket is identified by 4-tuple (src IP address, src port number, dest IP address, dest port number). Explain, *in detail and specifically*, why the TCP socket needs to be identified by 4-tuple.
Answer:
Connection oriented service is to be provided and as shown below DP and D-IP can be identical but SP and S-IP need to be different for two TCP connections established to the same server.


Connection-oriented demux: Threaded Web Server

- each socket attached to a thread

Connection oriented

신뢰성

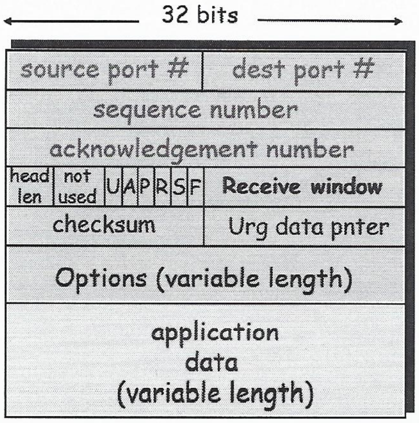
Reliable한 전송을 제공하기 위해
src port, IP로 구별이 필요!
(Sequence #이 각기 다르게 운영되므로)

Also, different logical TCP connections need to maintain their own parameters and buffer space.
For UDP, there is no connection concept. It is enough for the server to receive a request and send the response to it.

Problem 2: (10 points) TCP segment
When a sender side of TCP needs to establish a TCP connection, it will send a TCP segment. Describe key fields and their usage in the header of the segment which needs to be sent for this.
Answer:

- src port number 2점
- dest port number 2점
- sequence number – used as the initial sequence number 3점
- SYN flag – set, indicates that the corresponding segment is sent to initiate the TCP connection set-up. 3점
- (ACK flag – reset) → Optional. +1점.

Field 이름 잘못쓰면 -1점.



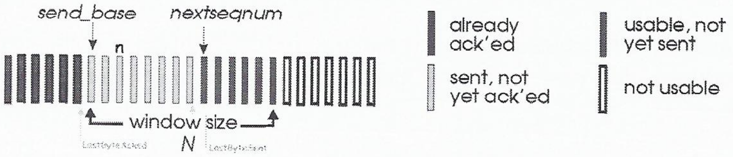
Problem 3: (15 points) ACK
Describe, *in detail and specifically*, when a sender side of TCP will receive duplicate ACKs.
Answer:

(Transmit) Window

Allows pipelined segment transmission

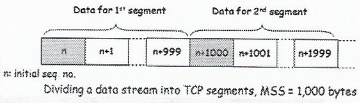
Sender:

- ❒ 32-bit seq # in seq # field
- ❒ "window" of up to N, consecutive unack'ed segments allowed



- ❒ ACK(n): ACKs all data bytes up to seq # n - 1: "cumulative ACK"
  - ○ When ACK(n) arrives, send_base ← n
  - ○ may receive duplicate ACKs

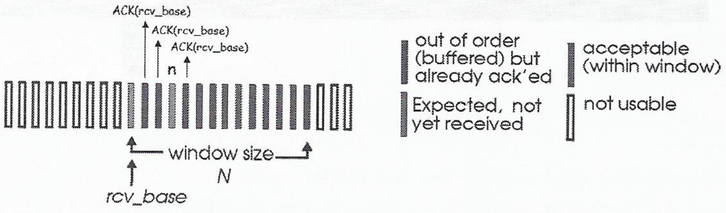
(Receive) Window



Dividing a data stream into TCP segments, MSS = 1,000 bytes

Receiver:

- ❒ receiver sends *cumulative* ACK for in-order and correctly received segments
  - ○ buffers segments, as needed (for out-of-order arrivals), for eventual in-order delivery to upper layer


ACK(rcv_base)
ACK(rcv_base)
ACK(rcv_base)
n
window size
N
rcv_base
out of order (buffered) but already ack'ed
Expected, not yet received
acceptable (within window)
not usable

receiver view of sequence numbers

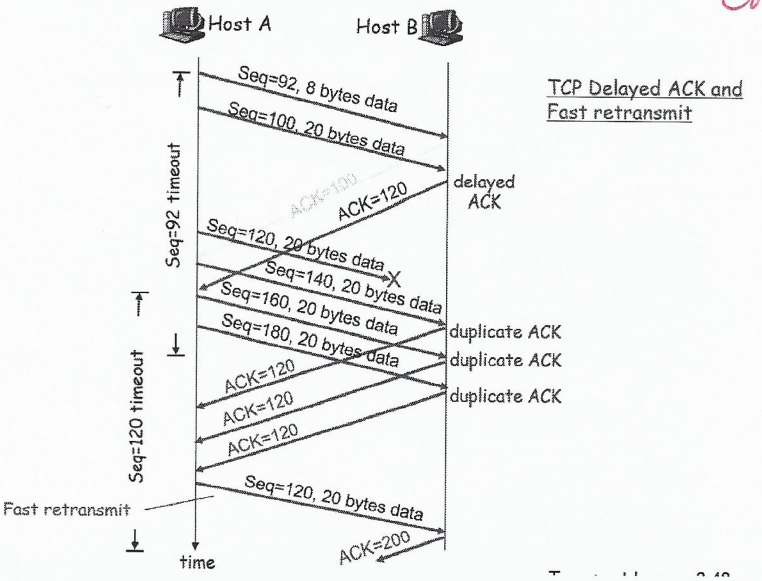
- ❒ If segment(rcv_base) arrives, rcv_base ← n

ex) Sequence #가 중간에 빠진 것이 있을 때 receiver가 duplicated ACK를 보낸다.

? 위와 같이 receiver가 duplicated ACK를 보내게 되는 부분의 설명이 미흡하면 −8점.

경위를 그림으로 그려 놓아도 설명을 하지 않으면 −~~8~~5점

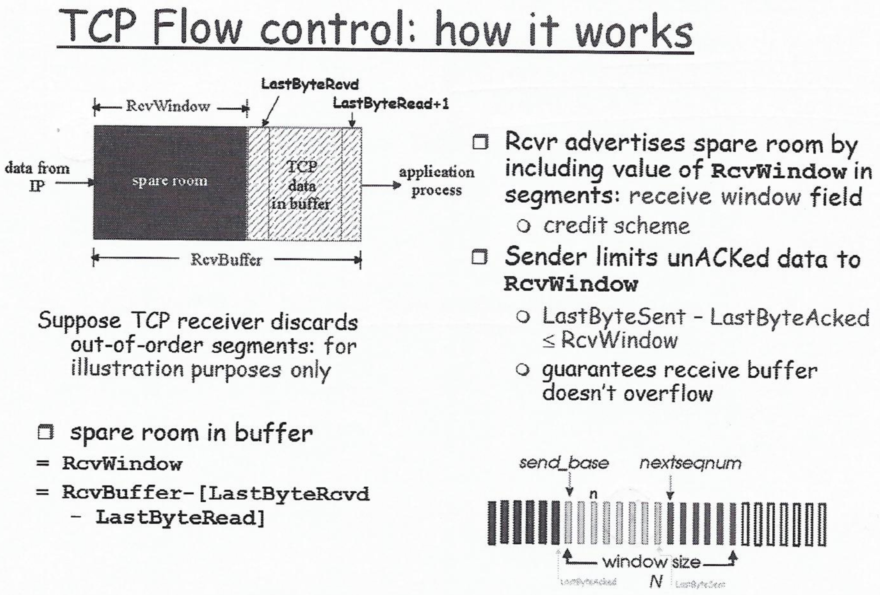
Let's see the usage of Duplicate ACKs



Problem 4: **(20 points) TCP flow control**

a) (10 points) Does the TCP flow control mechanism guarantee the no buffer overflow at the receiving side of the TCP? Answer in 'yes' or 'no'.
Answer: yes

b) (10 points) Justify your answer *specifically*.
Answer:
Initially, for predetermined RcvWindow, the sender will keep (LastByteSent – LastByteAcked) <= RcvWindow and there won't be any buffer overflow. Later when it receives a new RcvWindow which reflects the more recent buffer occupancy level, by keeping (LastByteSent – LastByteAcked) <= new RcvWindow, the buffer overflow can be avoided. Notice that if new RcvWindow is not to be sent, there will be no buffer overflow which is obvious.



TCP Flow control: how it works

Suppose TCP receiver discards out-of-order segments: for illustration purposes only

□ spare room in buffer
= `RcvWindow`
= `RcvBuffer-[LastByteRcvd - LastByteRead]`

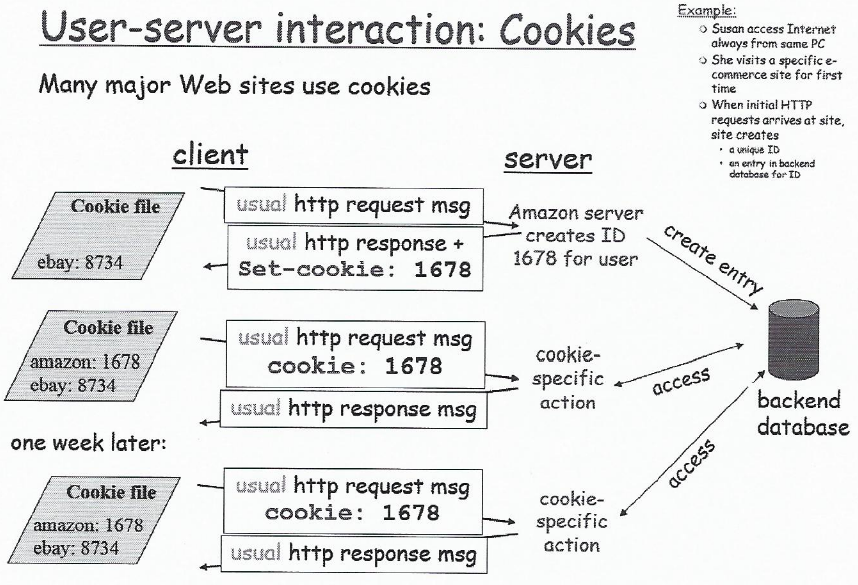
□ Rcvr advertises spare room by including value of `RcvWindow` in segments: receive window field
  ○ credit scheme

□ Sender limits unACKed data to `RcvWindow`
  ○ LastByteSent - LastByteAcked ≤ RcvWindow
  ○ guarantees receive buffer doesn't overflow

~~Receiver가 Sender의 RcvWindow를 갱신해 준다는 언급 없이 답안 작성 해도 되는지 ?~~

~~일단 언급 없을 시 (-5).~~

Problem 5: **(10 points) Cookie**
Consider an e-commerce site that wants to keep a purchase record for each of its customers. To make this possible, the site needs to send cookie information to clients. Describe how the cookie is sent to a corresponding client.
Answer:
via cookie header line in the HTTP response message. The example header line will look like 'set-cookie: 1678'.



User-server interaction: Cookies

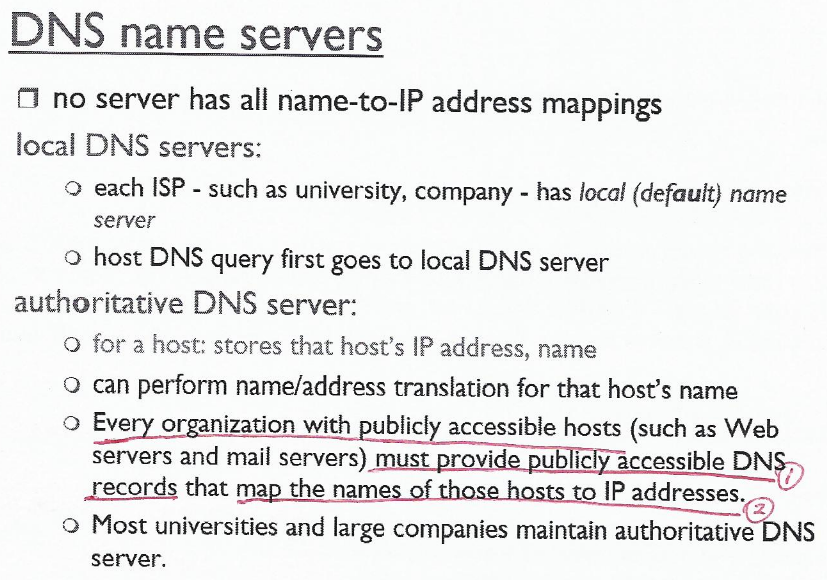
Many major Web sites use cookies

Example:
- Susan access Internet always from same PC
- She visits a specific e-commerce site for first time
- When initial HTTP requests arrives at site, site creates
  - a unique ID
  - an entry in backend database for ID

HTTP response message
온전히 언급해야 함.
불완전할 시 (-3)

Problem 6: **(15 points) DNS**
Describe the authoritative DNS server.

Answer:



DNS name servers

- ❒ no server has all name-to-IP address mappings

local DNS servers:

- ○ each ISP - such as university, company - has *local (default) name server*
- ○ host DNS query first goes to local DNS server

authoritative DNS server:

- ○ for a host: stores that host's IP address, name
- ○ can perform name/address translation for that host's name
- ○ Every organization with publicly accessible hosts (such as Web servers and mail servers) must provide publicly accessible DNS records ① that map the names of those hosts to IP addresses. ②
- ○ Most universities and large companies maintain authoritative DNS server.

①, ② 다 써야 15점.

① or ② 10점.

계층구조에 대한 언급 점수 줘야할지 ? 일단 점수를 주지 X.

Problem 7: (15 points) P2P File Distribution: BitTorrent
Describe the `.torrent` file.
Answer: content description ①, much smaller than content. contains address of tracker ② and list of chunks ③ that make up the content.

①, ②, ③ 각 5점.

Metadata 언급했을 시의 처리 ? 일단 점수를 주지 X.