CS348: Computer Networks

Error and Flow Control in TCP

Dr. Manas Khatua
Assistant Professor
Dept. of CSE, IIT Guwahati
E-mail: manaskhatua@iitg.ac.in

TCP Reliable Data Transfer

- Network layer service is unreliable, i.e. does not guarantee
  - datagram delivery
  - in-order delivery
  - data integrity

- TCP's reliable data transfer (RDT) service ensures that
  - the data stream that a process reads out of its TCP receive buffer is uncorrupted, without gaps, without duplication, and in sequence

TCP Error Control

- Error control in TCP is done by :
  - checksum
  - ACK
  - time-out

- TCP sends an ACK for the next packet that it is expecting

- When does a receiver generate ACK?

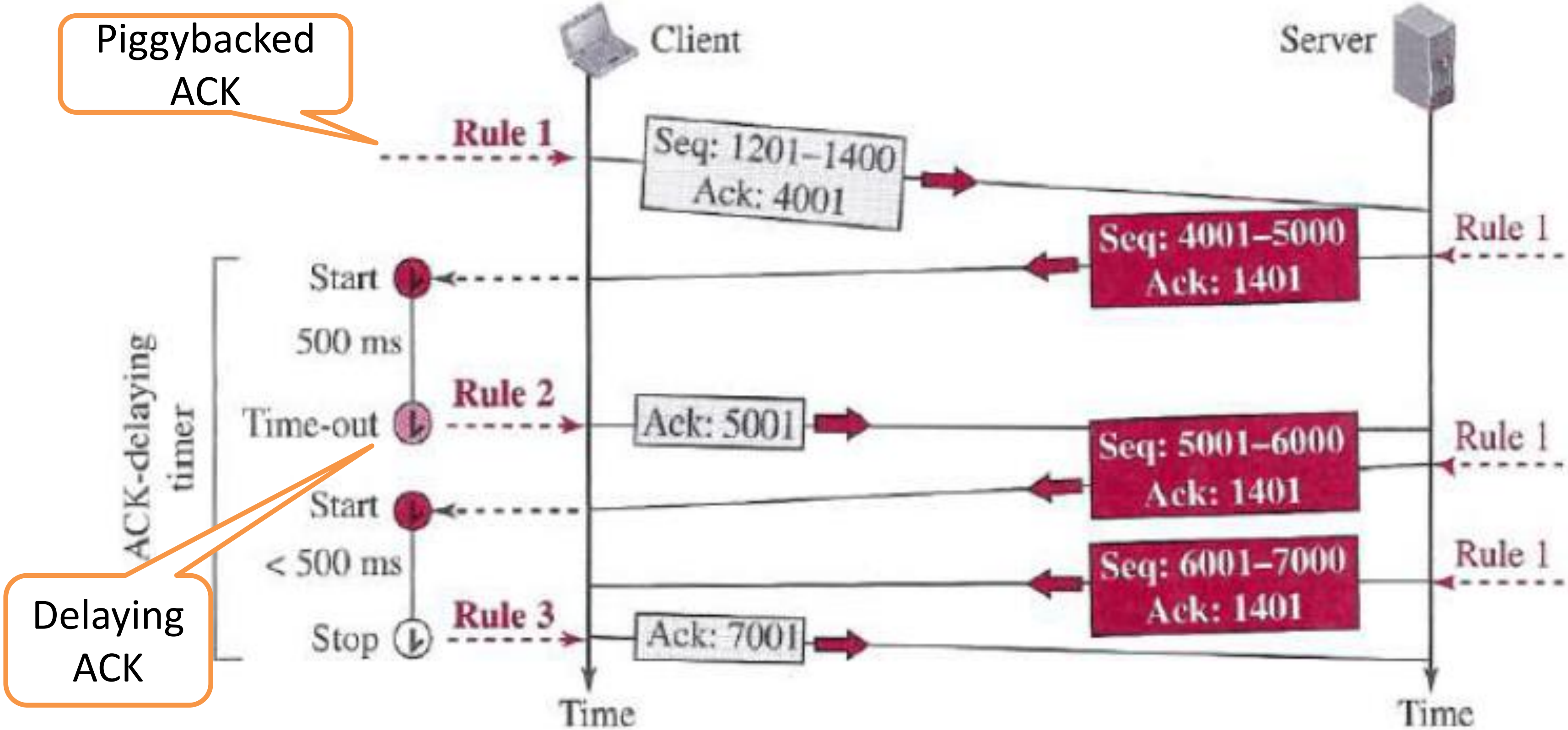
  - Rule-1: when node A sends data to node B, it piggybacks ACK
  - Rule-2: the receiver has no data to send and it receives an in-order segment, it delays sending ACK
  - Rule-3: there should not be more than two in-order unACKed segments at any time (it is delayed ACK)
  - Rule-4: when a segment arrives with an out-of-order sequence number; or, it is *fast retransmission* of missing segments
  - Rule-5: when a missing segment arrives
  - Rule-6: If a duplicate segment arrives

Cont…

- When retransmission happens?
  - After time-out
    - sending TCP maintains one retransmission time-out (RTO) for each connection
  - Three duplicate ACK rule
    - if three duplicate ACK (i.e., an original ACK and three exactly identical copies) arrive for a segment, the next segment is retransmitted without waiting for the time-out.
- Why three duplicate ACK?
    - Since TCP does not know whether a duplicate ACK is caused by a lost segment or just a reordering of segments, it waits for at max two duplicate ACK.
    - If three or more duplicate ACKs are received in a row, it is a strong indication that a segment has been lost.
- Data may arrive out-of-order and be temporarily stored by the receiving TCP.
- But, no out-of-order data are delivered to the process by TCP.

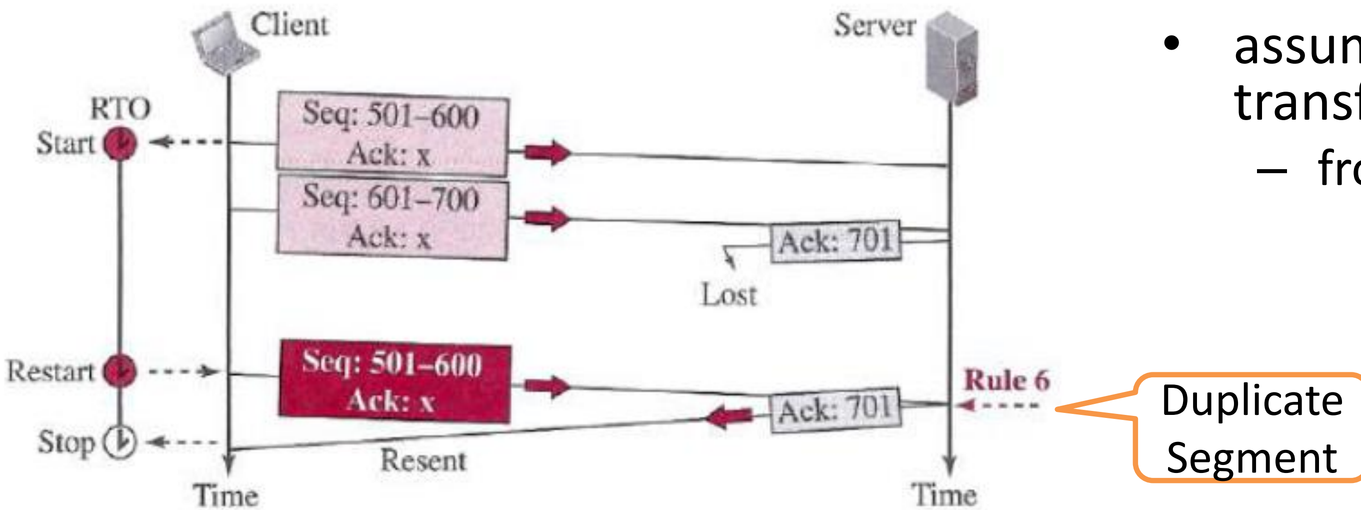
Example : Normal Scenario

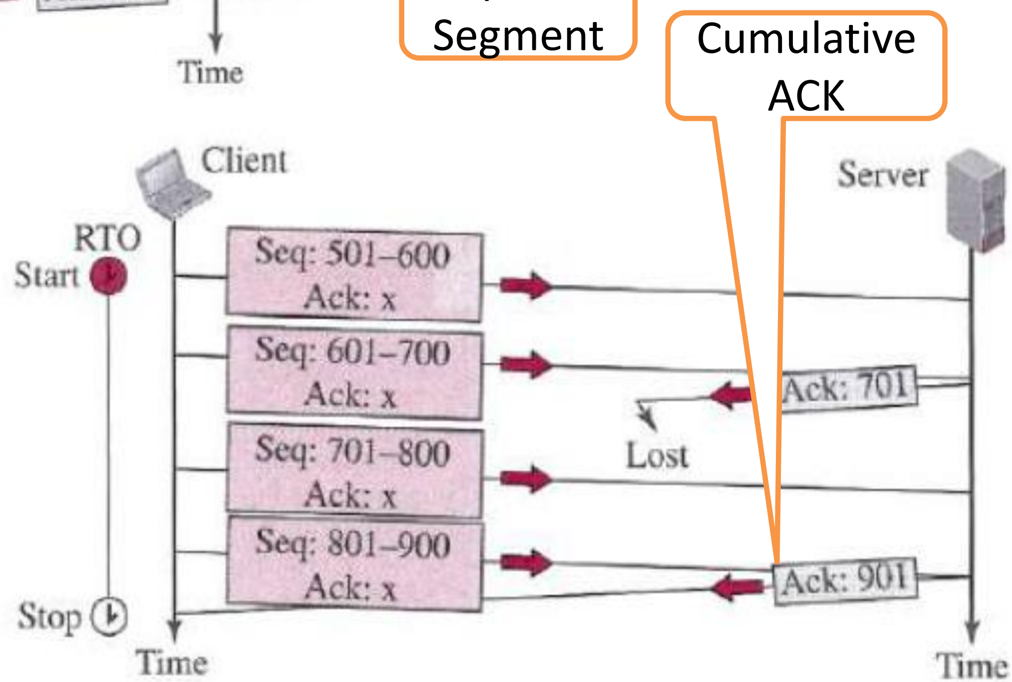
- client TCP sends one segment (2000 byte);
- server TCP sends three segments (3 x 1000 byte).



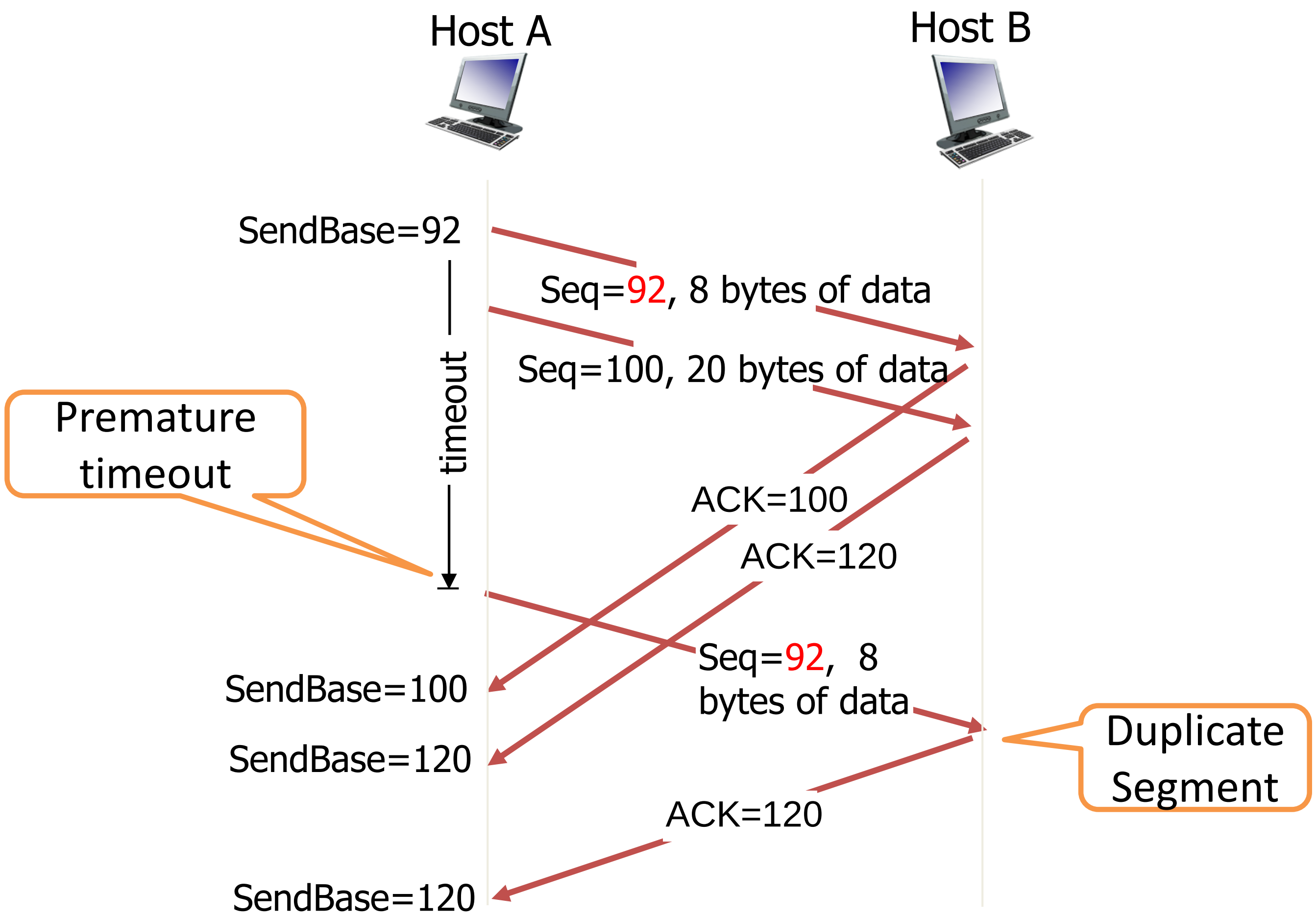
Example : Lost ACK, Cumulative ACK

- assume the data transfer is unidirectional
  - from client to server

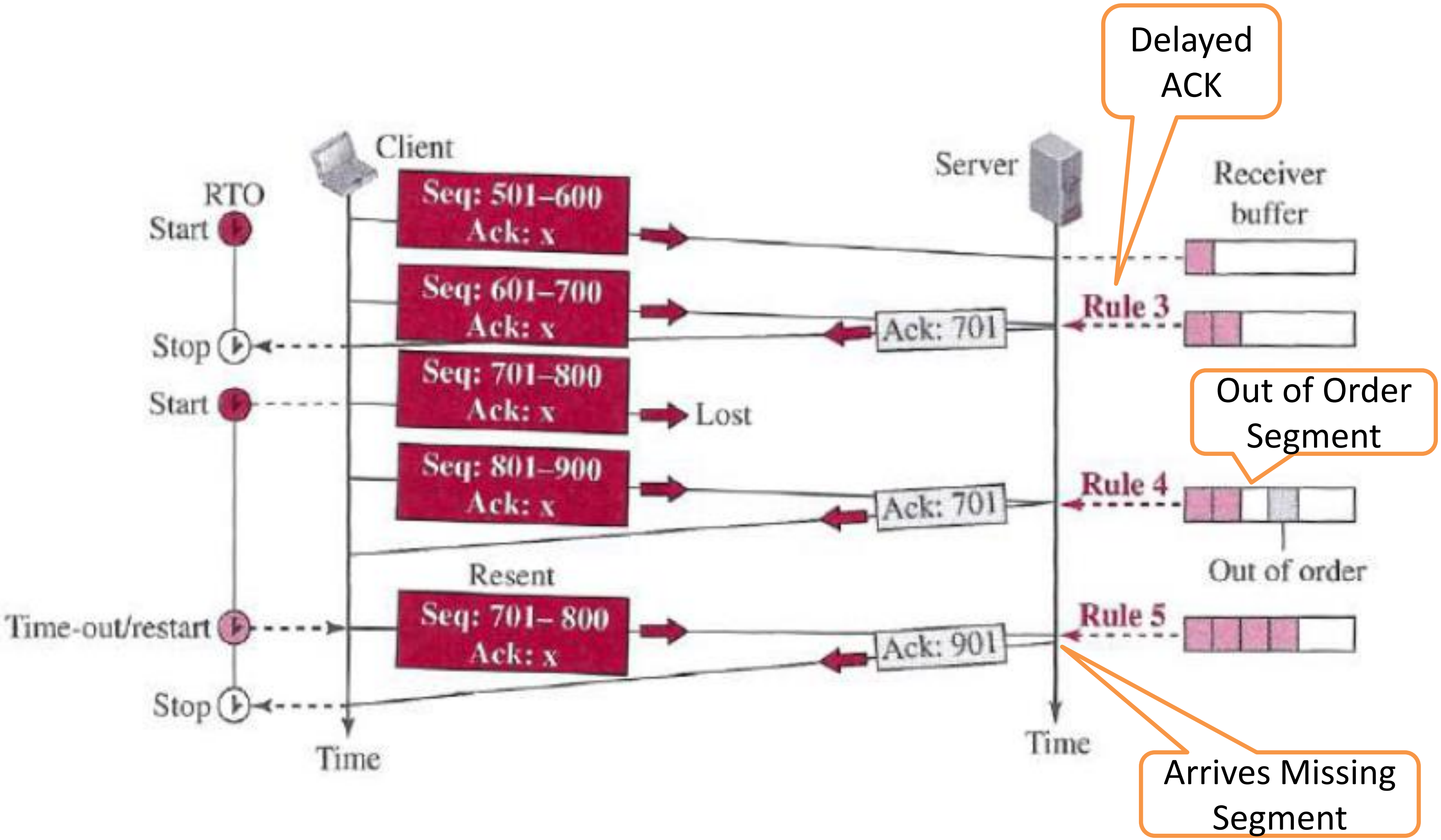




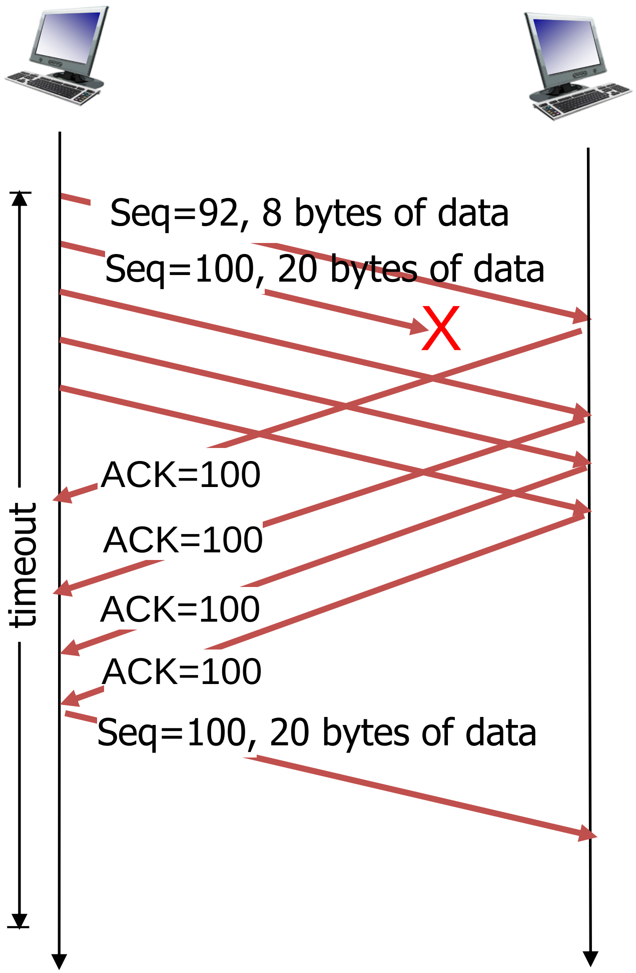
Example : Premature timeout

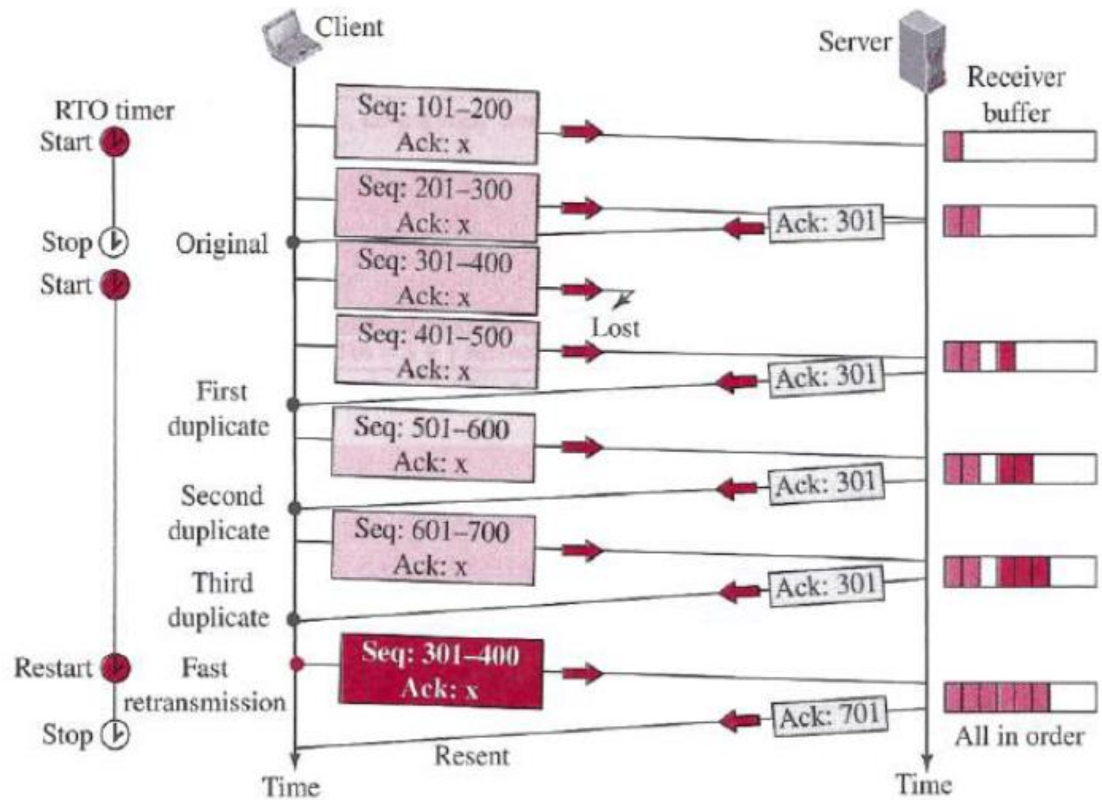


Example : Lost Segment



Example : Fast Retransmission after 3 duplicates

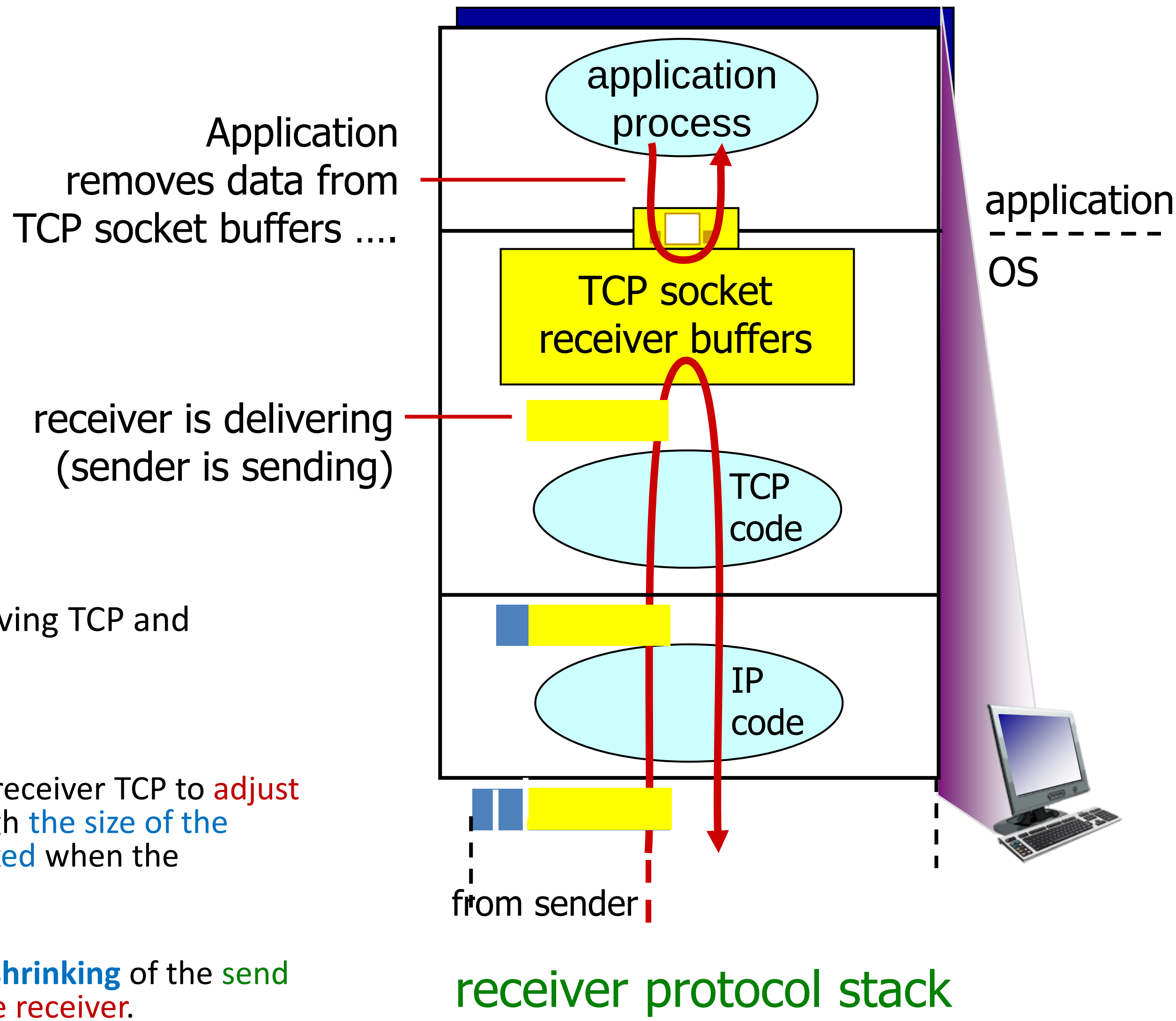




TCP Flow Control

flow control

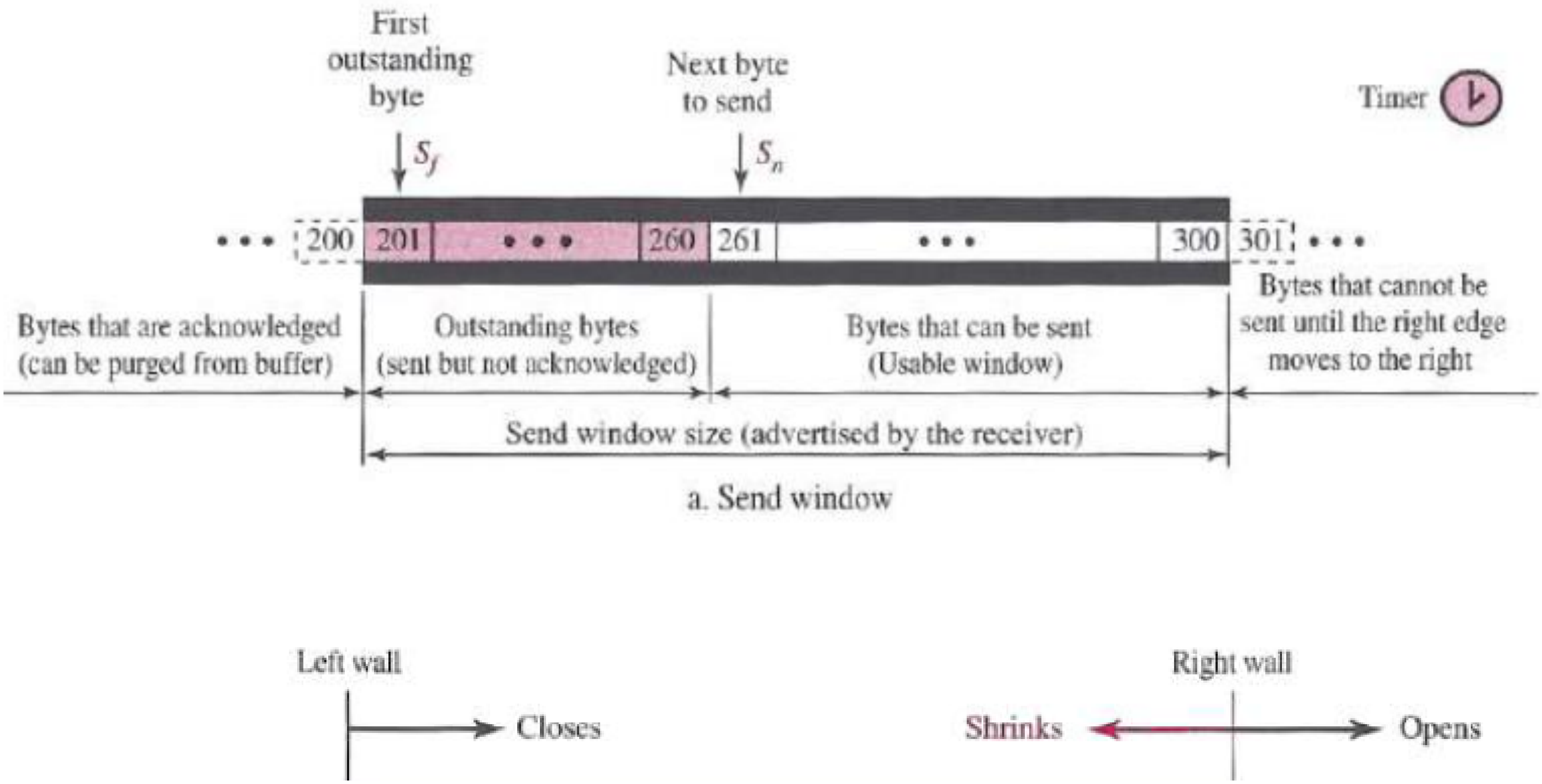
receiver controls sender, so sender won't overflow receiver's buffer by transmitting too much, too fast



- No flow control between receiving TCP and receiving process.
- To achieve flow control:
  - forces the sender TCP and receiver TCP to adjust their window sizes, although the size of the buffer for both parties is fixed when the connection is established.
  - The **opening**, **closing**, and **shrinking** of the send window is controlled by the receiver.

Send Window in TCP

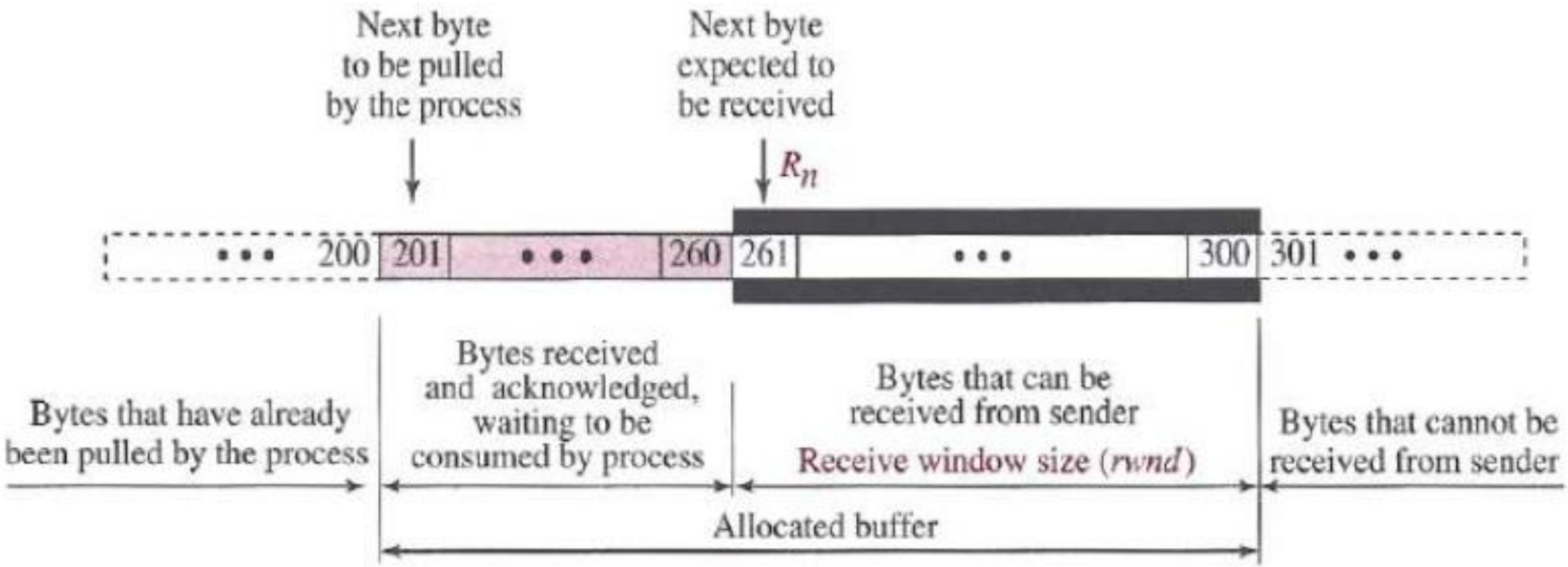
- Let send window size = 100



Modified SR (for Send Window)

- Sending window in TCP follows Selective-Repeat (SR) protocol with few modifications
  - The window size in SR = number of packets,
  - But, the window size in TCP = number of bytes.
  - SR requires individual ACK of each packet that was sent;
  - But, TCP sends ACK for the next packet that it is expecting
  - SR protocol may use several timers (for general transmission and selected retransmission)
  - But, TCP protocol uses only one timer.
  - Window size can be changed dynamically in TCP

Receive Window in TCP



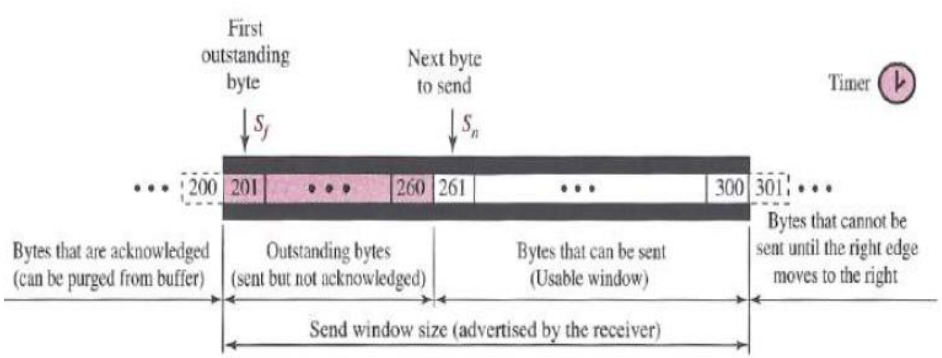
a. Receive window and allocated buffer

Modified SR (for Receive Window)

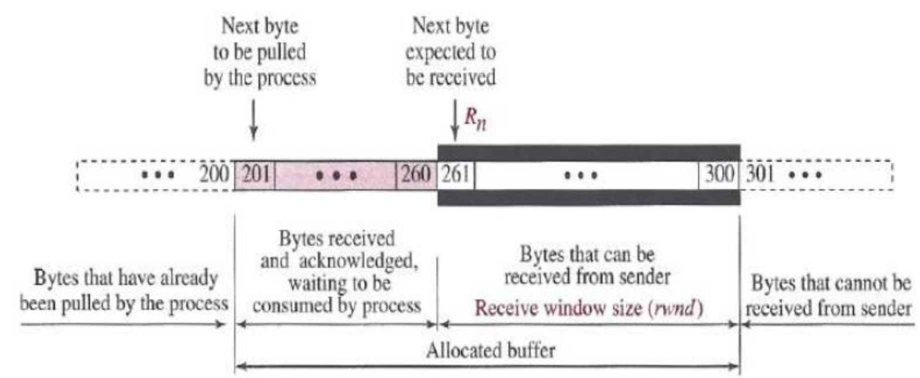
- Receive window in TCP is little different than that in SR
  - TCP allows the receiving process to pull data at its own pace.
  - The receive window size (*rwnd*) determines the number of bytes that the receive window can accept from the sender before being overwhelmed (**flow control**).

    rwnd = buffer size - number of bytes waiting to be pulled

  - ACK in SR is selective, but ACK in TCP is cumulative.
  - Retransmission is selective in SR, but oldest unACKed segment is retransmitted in TCP

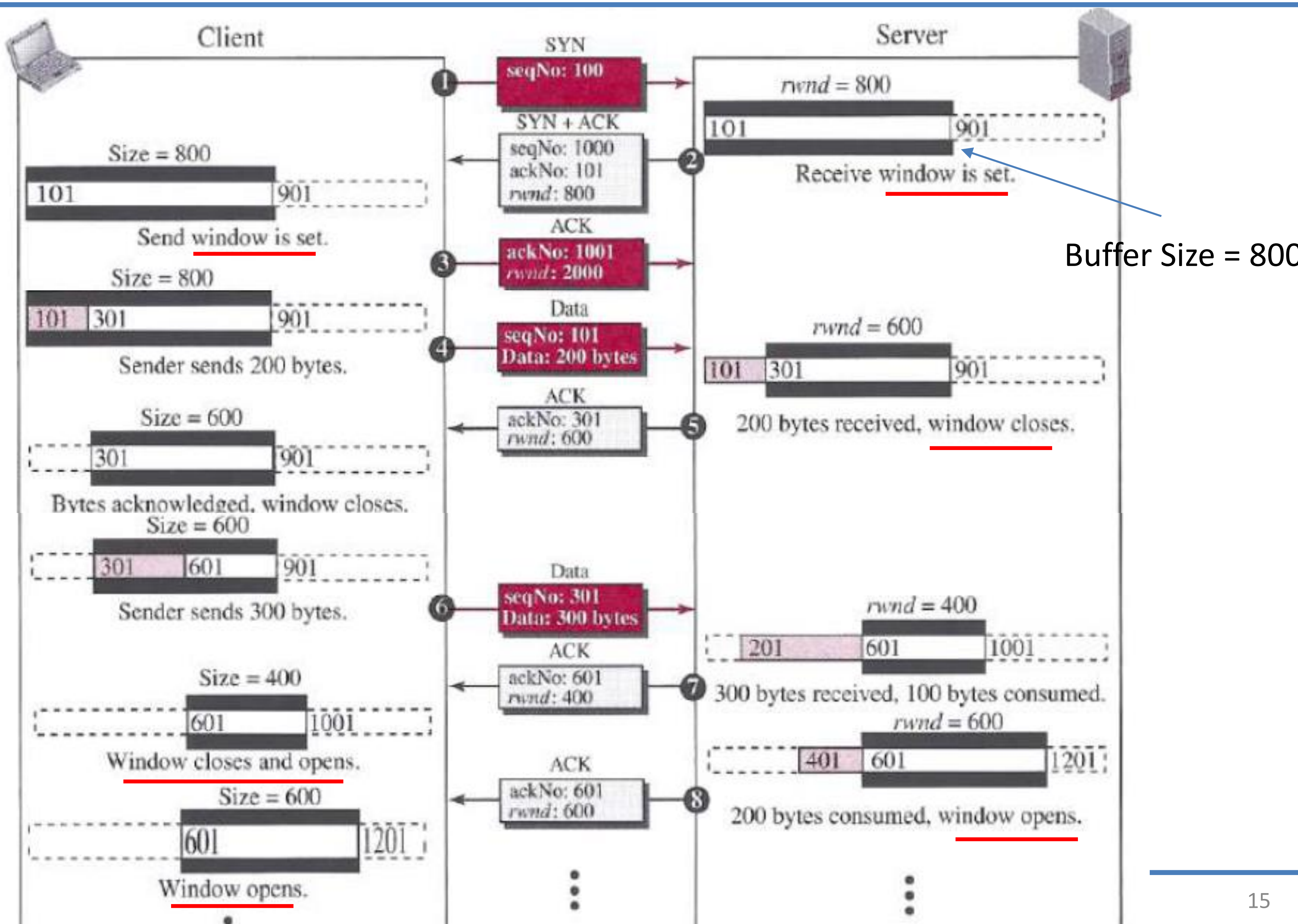


a. Send window



a. Receive window and allocated buffer

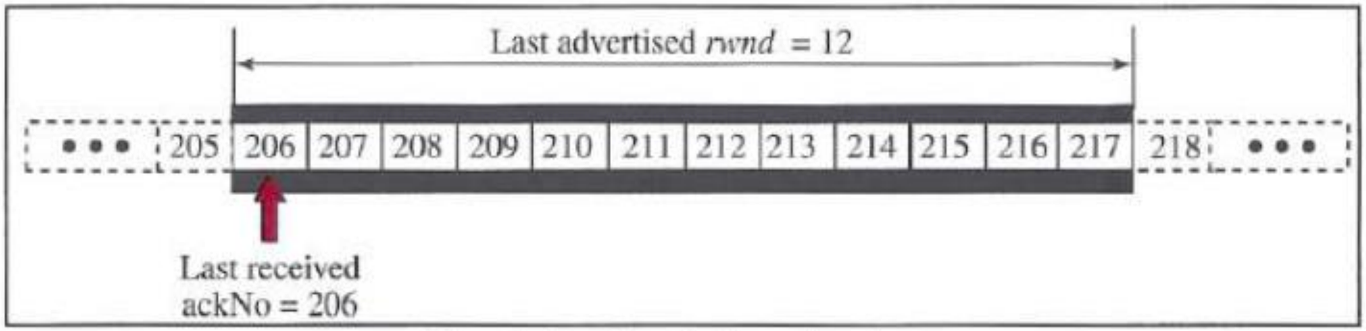
Example (oneway from client to server)



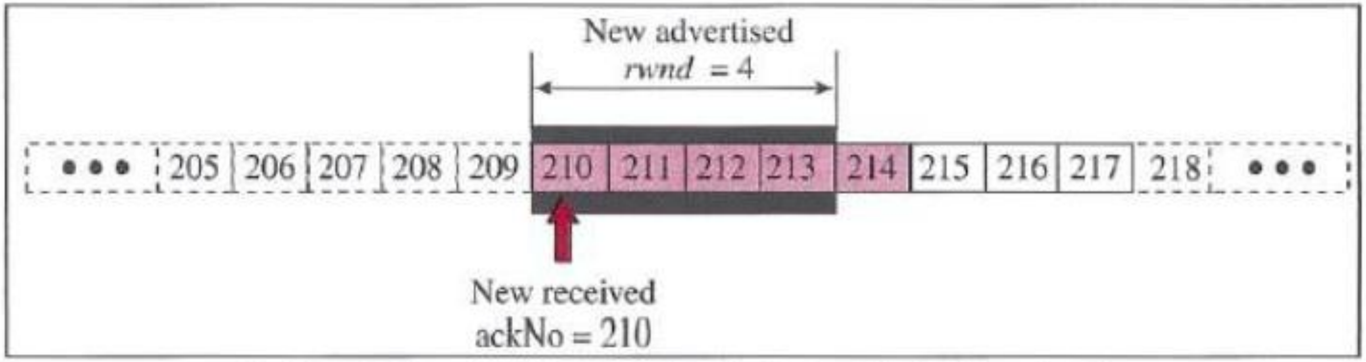
Shrinking of Windows

new ackNo + new *rwnd* >= last ackNo + last *rwnd*

If *rwnd*==0, it instructs for "window shutdown"



a. The window after the last advertisement



b. The window after the new advertisement; window has shrunk

Silly Window Syndrome

- Performance issue occurs when
  - Sending application program creates data slowly
  - OR, Receiving application program receives data slowly
  - OR, For both the above
- Example:
  - Sending process generating each byte very slowly
  - Sending TCP sends many 41 bytes segment (20 byte TCP header + 20 byte IP header + 1 byte data)
- Two types to address
  - Syndrome created by sender
  - Syndrome created by receiver

Solution

- Naïve solution faces a trade-off optimization
  - If TCP waits too long, it may delay the process
  - If TCP does not wait for long, it may end up sending small segment

- Better Solution for **sender**: Nagle's Algorithm
  - Sending TCP sends the 1st segment as it is.
  - 2nd segment onwards, the sending TCP accumulates data in sending buffer and waits until
    - Either the receiving TCP sends an ACK
    - Or enough data have accumulated to fill the maximum-size segment (MSS)

- Better Solution for **receiver**: Clark's two algorithms
- First algorithm:
  - send an ACK as soon as the data arrive,
  - but to announce a window size of zero until
    - either there is enough space to accommodate a segment of maximum size
    - or until at least half of the receive buffer is empty.
- Second algorithm:
  - delay sending the ACK.
  - The receiver waits until there is a decent amount of space in its incoming buffer before ACKing the arrived segments.

Thanks!

Content of this PPT are taken from:

1) **Computer Networks: A Top Down Approach**, by J.F. Kuros and K.W. Ross, 6th Eds, 2013, Pearson Education.

2) **Data Communications and Networking**, by B. A. Forouzan , 5th Eds, 2012, McGraw-Hill.

3) **Chapter 3 : Transport Layer,** PowerPoint slides of "Computer Networking: A Top Down Approach", 6th Eds, J.F. Kurose, K.W. Ross