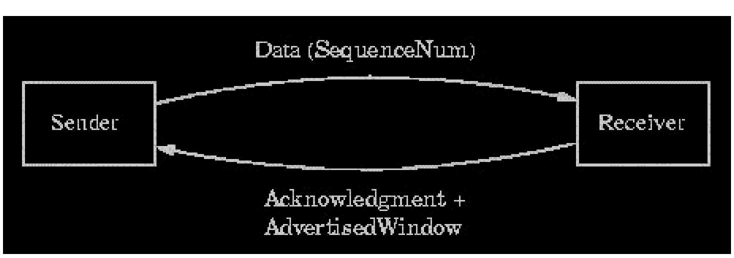
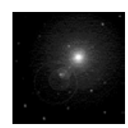
Flow Control

Buffer, Flow Control & Window

- TCP use windowing for flow control.
- Each site always advertises its remaining free buffer size.
- When receiver buffer is zero, sender stops.



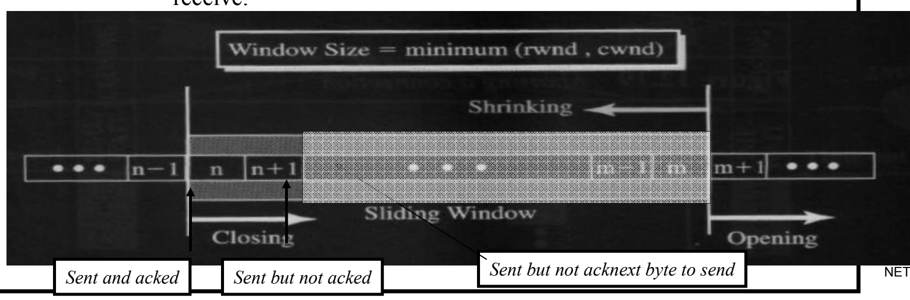


COMPUTER
COMMUNICATION
NETWORK

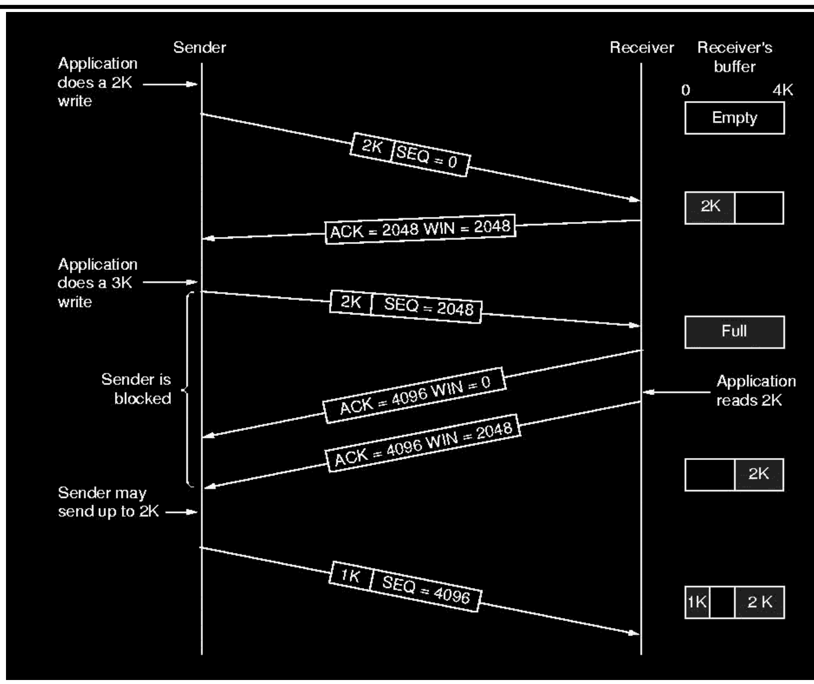
NET06F, javed@kent.edu

TCP Sliding Window

- A window operates on the sender buffer.
- Three operations are performed. All three operations are controlled by the receiver side.
  - OPEN: Sender is free to send more bytes (it has more capacity).
  - CLOSE: Receiver has received and acknowledged upto this so left wall moves right. The data can be discarded.
  - SHINK: Not used often.
  - Note: Exact meaning of ACK Value = next byte it is ready to receive.

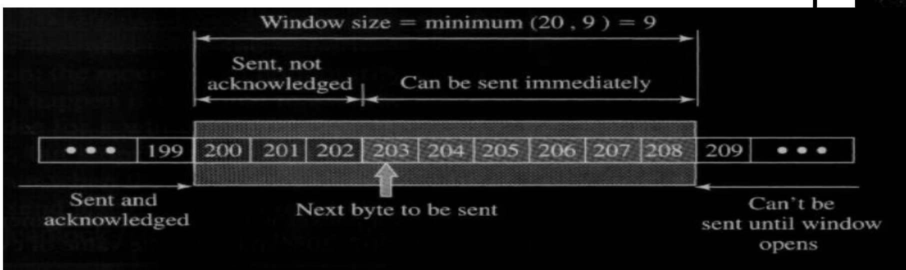


COMPUTER COMMUNICATION NETWORK



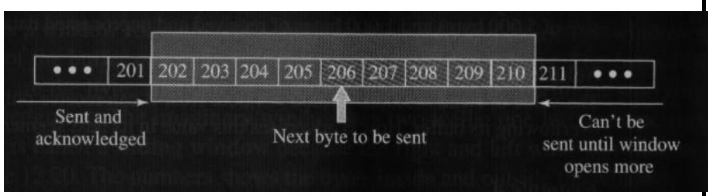
COMPUTER COMMUNICATION NETWORK

Quiz-1

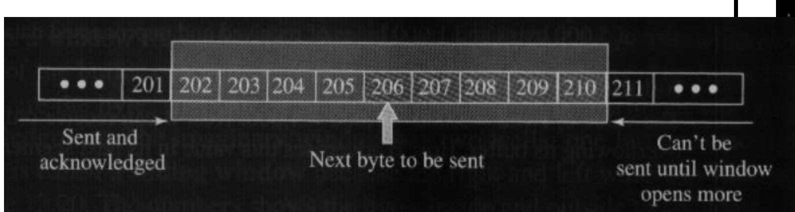


UTER
ICATION
ORK

- The sender receives a packet with an acknowledgement value of 202 and an rwnd of 9. The host has already sent bytes 203,204,and 205. The value of cwnd is still 20. Show the new window. (show the location of two windows and the next byte to send pointer).
- Ans:

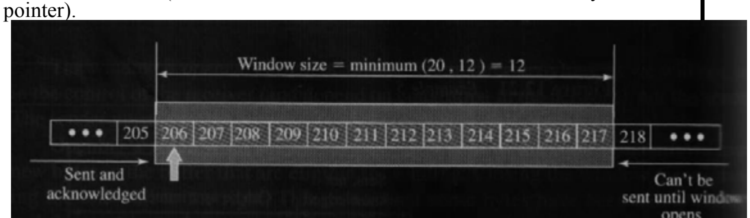


Quiz-2

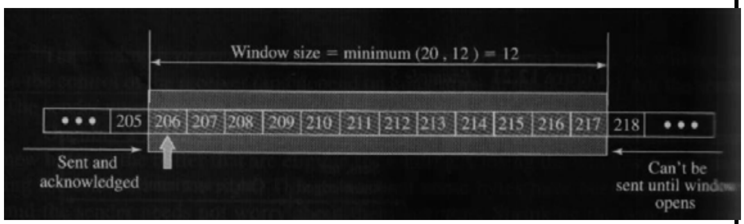


MPUTER
UNICATION
TWORK

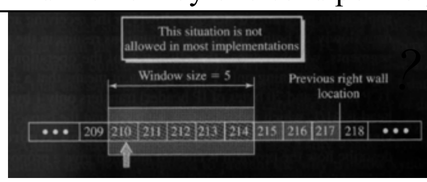
- The sender receives a packet with an acknowledgement value of 206 and an rwnd of 12. The host has not sent any new bytes. The value of cwnd is still 20. Show the new window. (show the location of two windows and the next byte to send pointer).
- Ans:



Quiz-3



- Quiz 605: The sender receives a packet with an acknowledgement value of 210 and an rwnd of 7. The host has sent bytes 206,207,208, 209, and 210. The value of cwnd is still 20. Show the new window. (show the location of two windows and the next byte to send pointer).



??

not collapse beyond left wall..

COMPUTER COMMUNICATION NETWORK

Window Shutdown: Zero Window Advertisement

- Shrinking of window is strongly discouraged. However, in some situations sender can advertise “zero window” if for some reason it wants sender to temporarily stop sending.
- In that case actually window is not changed, but sending is stopped until a new advertisement is received.
- Even though the sending is stopped by order of the receiver, but the sender can still send 1 byte segment to probe the receiver in case the new advertisement is lost to avoid deadlock. (we will learn more in TCP timer discussion).

COMPUTER COMMUNICATION NETWORK

Silly Window Syndrome!

- Slow Sender Scenario:
  - A TCP sender application generates data 1 byte at a time. If TCP sends immediately then there will be TCP segments containing 1 byte only. It will carry 40 byte header (20 byte TCP+ 20 byte IP header) for each byte of data!
  - But if it waits too long that will be bad for interactive applications.
- Solution: Nagle's Algorithm
  - Send first piece of data as first segment as soon as it is received from application- even if it is one byte.
  - For next segments- wait till ACK is received, or MTU unit data has been received from application.

Silly Window Syndrome!

- Slow Receiver Scenario:
  - A TCP receiver application consumes data 1 byte at a time. If TCP sends immediately ACK then the advertisements will be only for 1 byte. It will invite from sender, eagerly waiting, segments with 40 byte header (20 byte TCP+ 20 byte IP header) for each byte of data!
- Solution: Clark's Algorithm
  - Send ACK immediately as soon as data arrives, but advertise zero window until MTU unit buffer is space is available.
  - Also, delay acknowledgement till MTU unit space is available- but don't wait long so that sender times out (<500ms).

ACK (Error) Control

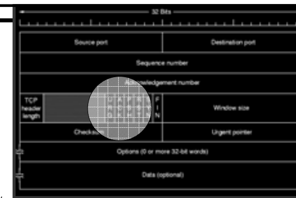
- Rule#1 Receiver would piggyback an acknowledgement on the next segment to send in opposite direction. Acknowledgement = the next sequence number it expects to receive.
- Rule#2: If there is no data to send in opposite direction wait a fixed amount of time (about 500ms) then send a segment just with Acknowledgement
- Rule#3: Keep no more than two segment unacked. If sender is holding back an Acknowledgement but the next segment arrives as well immediately send the Acknowledgement.
- Rule#4: If there are still missing previous data (hole) then Acknowledge the last segment up-to-which receiver has received all (connects to fast retransmit).
- Rule#5: When a missing segment arrives acknowledge to announce the next sequence number expected.
- Rule#6: If duplicate segment arrives, immediately send an Acknowledgement. This is on the guess that actually an Acknowledgement segment was lost.

Acknowledgement Types

- Accumulative Acknowledgement (ACK)
  - TCP was originally designed to receipt of segment accumulatively. The receiver advertises the next byte it expects to receive, ignoring all segments received out-of-order.
  - The 32 byte ACK field in TCP header is used for accumulative acknowledgments and it value is valid only when the ACK flag bit is set to 1.
- Selective Acknowledgment (SACK)
  - Modern implementation can send additional data as TCP option to sender. It reports the block of data that is out-of-order and the block of segments that is duplicated.

COMPUTER COMMUNICATION NETWORK

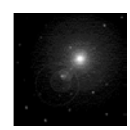
Immediate Data Transfer



NETWORK

- Pushing Data
  - Sending TCP as well and receiver TCP can delay the delivery of data. This is problematic for interactive application.
  - PSH flag can be set by the sender. The both TCP end-points are requested to sent the segment data immediately and deliver it to the application.
  - Most current implementations ignore them!
- Urgent Data
  - TCP is stream oriented and delivers bytes to receiver applications only when all bytes before it has been delivered. This can be problematic for control commands (such as abort a file transfer).
  - Sending TCP can set the URGENT flag. The urgent data is loaded at the head of next segment and normal data in remaining segment. The URGENT pointer tells where is the boundary of the URGENT data and normal data in this segment.
  - Receiver TCP extracts the urgent data and delivers it out of order to the receiving application.

TCP Timers



COMPUTER
COMMUNICATION
NETWORK

- Retransmission Timer
  - Keeps the timeout value based on smooth estimate of round trip time and Kern's algorithm (discussed earlier).
  - Uses binary back-off to adjust the congestion window (to be discussed later).
- Persistence Timer
  - To deal with zero window advertisement, sender TCP stops sending any data and starts this timer. If ACK is lost (the receiver (sender of ACK) does not track loss of ACK) then the sender may not resume ever and there would be a deadlock.
  - To avoid this if sender does not receive ACK (about 60 sec), it sends a PROBE segment until window is opened by an ACK from the receiver.
- TIME-WAIT Timer
  - 2MSL timer is user to manage connection termination (discussed already).

Congestion Control

- Congestion Collapse?
- When a message is lost, TCP assumes congestion!
- It sends one packet at a time even if the receiver advertised large buffer.
- If it receives acknowledgement, only then doubles the data per packet, and sends two additional packets.
- If these are acknowledged, then it sends four more packets..
- More in the next classes..

COMPUTER COMMUNICATION NETWORK

Some TCP Options

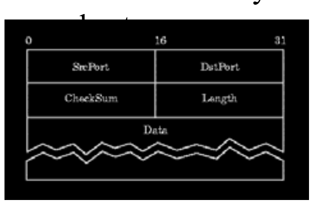
- Option#2: Maximum Segment Size (MSS):
  - It defines maximum bytes of payload data that can be sent in a segment (not the total segment size). Default is only 536 bytes. This is set during connection establishment.
- Option#3: Window Scale Factor
  - For long-fat-pipe segment size should be very large (much more than 65,535 bytes allowed in 16 bit window field).
  - In this option a scale x is sent. The window size is then interpreted as stated-size*2^x.
  - It is determined only once during connection establishment and can not change during operation.
- Option#5: Selective Acknowledgement
  - Sent to give detail information about acks. It contains a list of blocks describing received data blocks.
- Option#8: Timestamp
  - Can be used to determine the RTT.
  - Timestamp also allows the protection against wrapped sequence number.

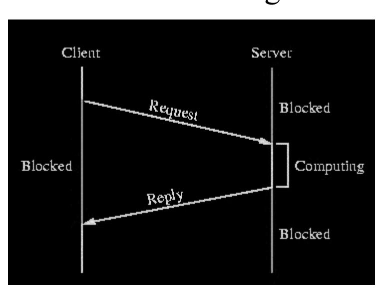
COMPUTER COMMUNICATION NETWORK

Other Transport Layer Protocols

Some Transport Protocols

- UDP:
  - when reliability is not an issue and messages are

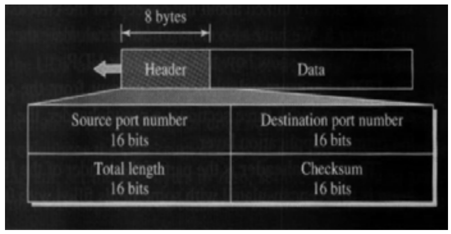




- RPC
  - request/response communication support.
- SCTP
  - Stream Control Transmission Protocol

COMPUTER COMMUNICATION NETWORK

User Datagram Protocol (UDP)



COMPUTER
COMMUNICATION
NETWORK

Few Well-known Ports

0-1024	Well Known Ports
1024-49,151	Registered
49,152-65,535	Dynamic or Private

UDP Ports

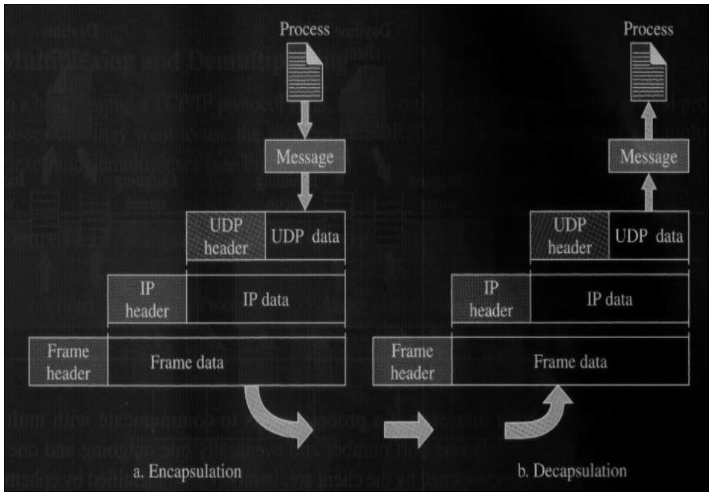
7	Echo
9	Discard
11	Users
13	Daytime
17	Quote
19	Chargen
53	Nameserver
67	BootPC
68	BootPC
69	TFTP
111	RPC
123	NTP
161	SNMP
162	SNMP

TCP Ports

7	Echo
9	Discard
11	Users
13	Daytime
17	Quote
19	Chargen
20	FTP (DATA)
21	FTP (CONTROL)
23	TELNET
25	SMTP
53	DNS
79	Finger
80	HTTP
111	RPC

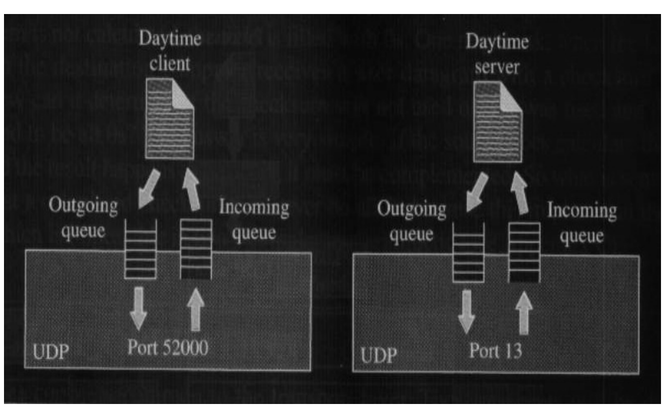
COMPUTER
COMMUNICATION
NETWORK

Encapsulation Decapsulation of UDP



COMPUTER
COMMUNICATION
NETWORK

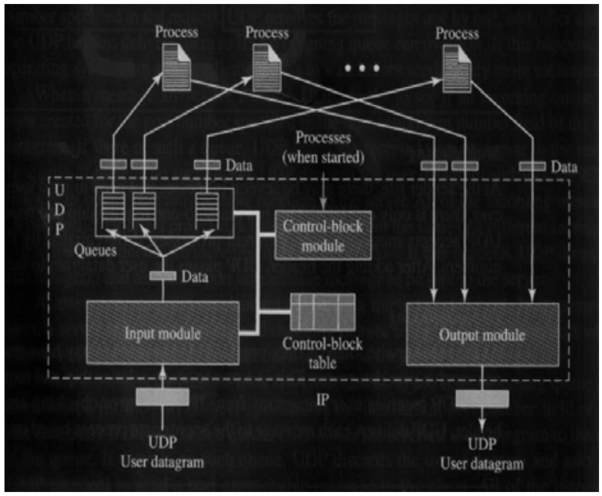
Client Server Communication using UDP



0-1024 Well Known Ports
1024-49,151 Registered
49,152-65,535 Dynamic or Private

COMPUTER
COMMUNICATION
NETWORK

UDP Control Block



COMPUTER
COMMUNICATION
NETWORK