

TCP

Raj Jain

Professor of CIS

The Ohio State University

Columbus, OH 43210

Jain@ACM.Org

<http://www.cis.ohio-state.edu/~jain/>



- q Key features, Header format
- q Mechanisms, Implementation choices
- q Slow start congestion avoidance,
Fast Retransmit/Recovery
- q Selective Ack and Window scaling options
- q UDP

Ref: RFCs, Thomas

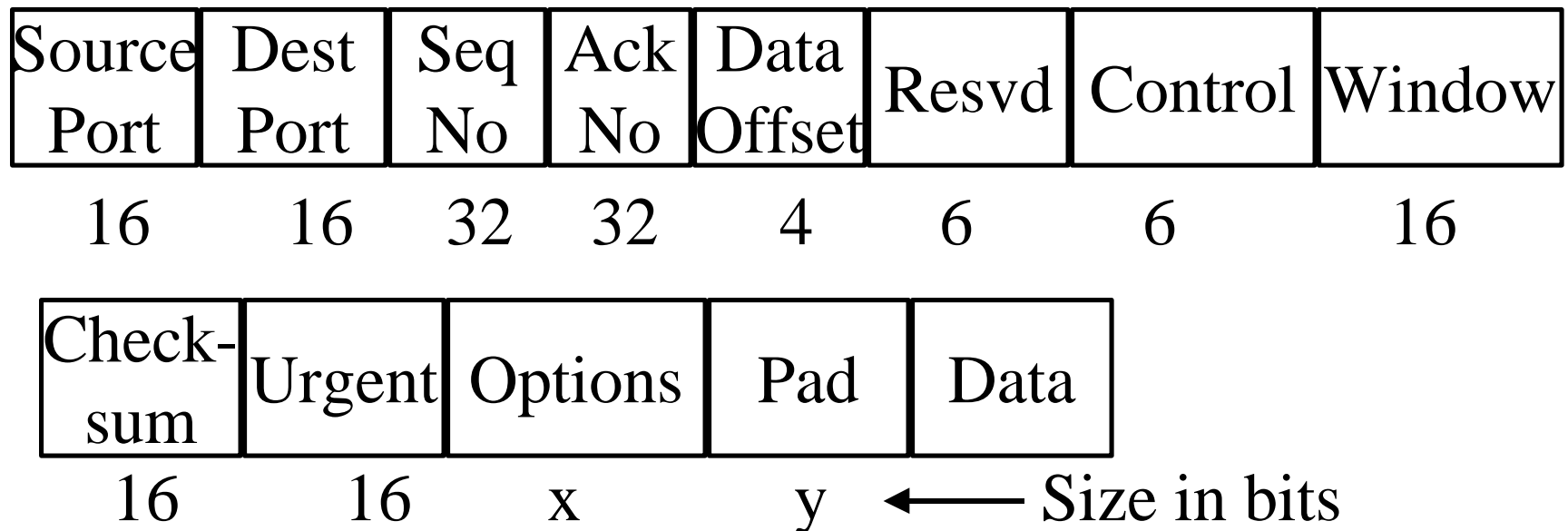
Key Features of TCP

- q Connection oriented
- q Point-to-point communication: Two end-points
- q Reliable transfer: Data is delivered in order
- q Full duplex communication
- q Stream interface: Continuous sequence of octets
- q Reliable connection startup: Data on old connection does not confuse new connections
- q Graceful connection shutdown: Data sent before closing a connection is not lost.

Transport Control Protocol (TCP)

- q Key Services:
 - q Send: Please send when convenient
 - q Data stream push: Please send it all now, if possible.
 - q Urgent data signaling: Destination TCP! please give this urgent data to the user
(Urgent data is delivered in sequence. Push at the should be explicit if needed.)
 - q Note: Push has no effect on delivery.
Urgent requests quick delivery

TCP Header Format

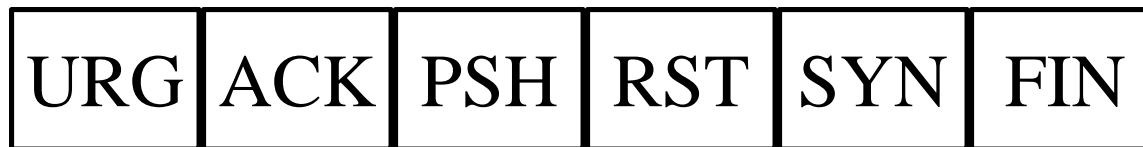


TCP Header

- q Source Port (16 bits): Identifies source user process
20 = FTP, 23 = Telnet, 53 = DNS, 80 = HTTP, ...
- q Destination Port (16 bits)
- q Sequence Number (32 bits): Sequence number of the first byte in the segment. If SYN is present, this is the initial sequence number (ISN) and the first data byte is ISN+1.
- q Ack number (32 bits): Next byte expected
- q Data offset (4 bits): Number of 32-bit words in the header
- q Reserved (6 bits)

TCP Header (Cont)

- q Control (6 bits): Urgent pointer field significant,
Ack field significant,
Push function,
Reset the connection,
Synchronize the sequence numbers,
No more data from sender



- q Window (16 bits): Will accept [Ack] to [Ack]+[window]

TCP Header (Cont)

- q Checksum (16 bits): covers the segment plus a pseudo header. Includes the following fields from IP header: source and dest adr, protocol, segment length. Protects from IP misdelivery.
- q Urgent pointer (16 bits): Points to the byte following urgent data. Lets receiver know how much data it should deliver right away.
- q Options (variable):
Max segment size (does not include TCP header, default 536 bytes), Window scale factor, Selective Ack permitted, Timestamp, No-Op, End-of-options

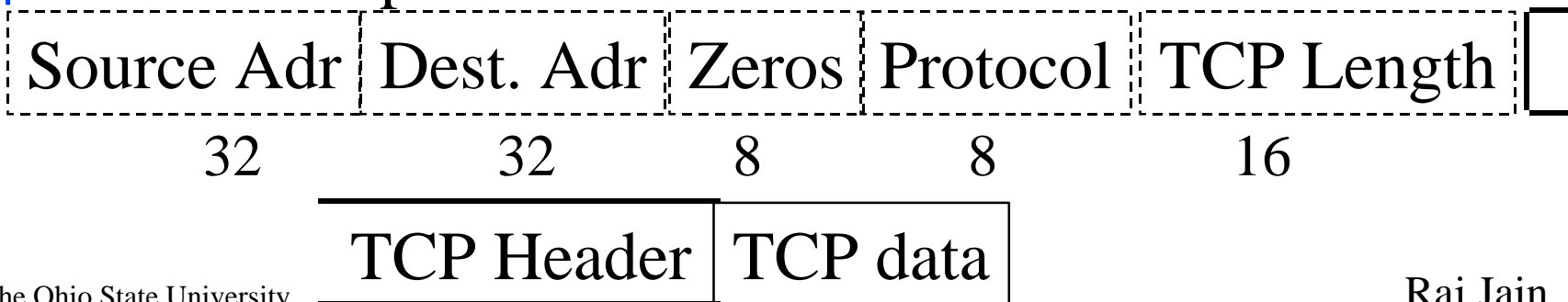
TCP Options

Kind	Length	Meaning
0	1	End of Valid options in header
1	1	No-op
2	4	Maximum Segment Size
3	3	Window Scale Factor
8	10	Timestamp

- q End of Options: Stop looking for further option
- q No-op: Ignore this byte. Used to align the next option on a 4-byte word boundary
- q MSS: Does not include TCP header

TCP Checksum

- q Checksum is the 16-bit one's complement of the one's complement sum of a pseudo header of information from the IP header, the TCP header, and the data, padded with zero octets at the end (if necessary) to make a multiple of two octets.
- q Checksum field is filled with zeros initially
- q TCP length (in octet) is not transmitted but used in calculations.
- q Efficient implementation in RFC1071.



TCP Service Requests

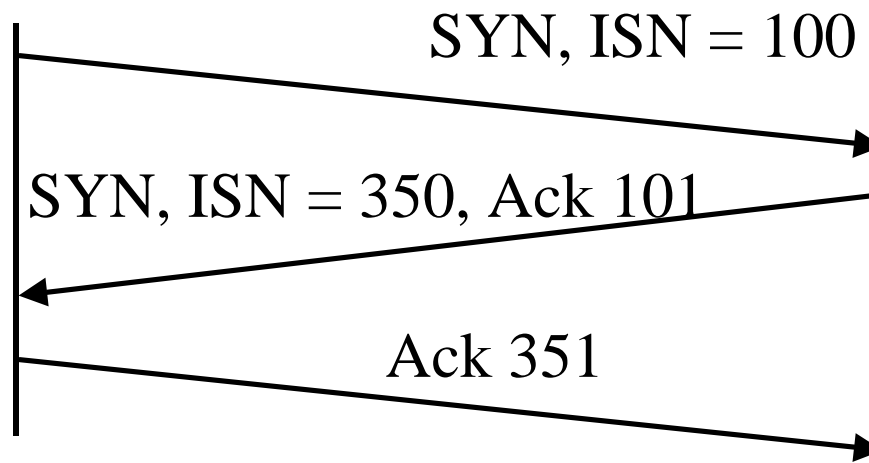
- q Unspecified passive open:
Listen for connection requests from any user (port)
- q Full passive open:
Listen for connection requests from specified port
- q Active open: Request connection
- q Active open with data: Request connection and transmit data
- q Send: Send data
- q Allocate: Issue incremental allocation for receive data
- q Close: Close the connection gracefully
- q Abort: Close the connection abruptly
- q Status: Report connection status

TCP Service Responses

- q Open ID: Informs the name assigned to the pending request
- q Open Failure: Your open request failed
- q Open Success: Your open request succeeded
- q Deliver: Reports arrival of data
- q Closing: Remote TCP has issued a close request
- q Terminate: Connection has been terminated
- q Status Response: Here is the connection status
- q Error: Reports service request or internal error

TCP Mechanisms

- q Connection Establishment
 - q Three way handshake
 - q SYN flag set \Rightarrow Request for connection



- q Connection Termination
 - q Close with FIN flag set
 - q Abort

Three-Way Handshake

- q 3-way handshake for opening and closing connections. Necessary and sufficient for unambiguity despite loss, duplication, and delay

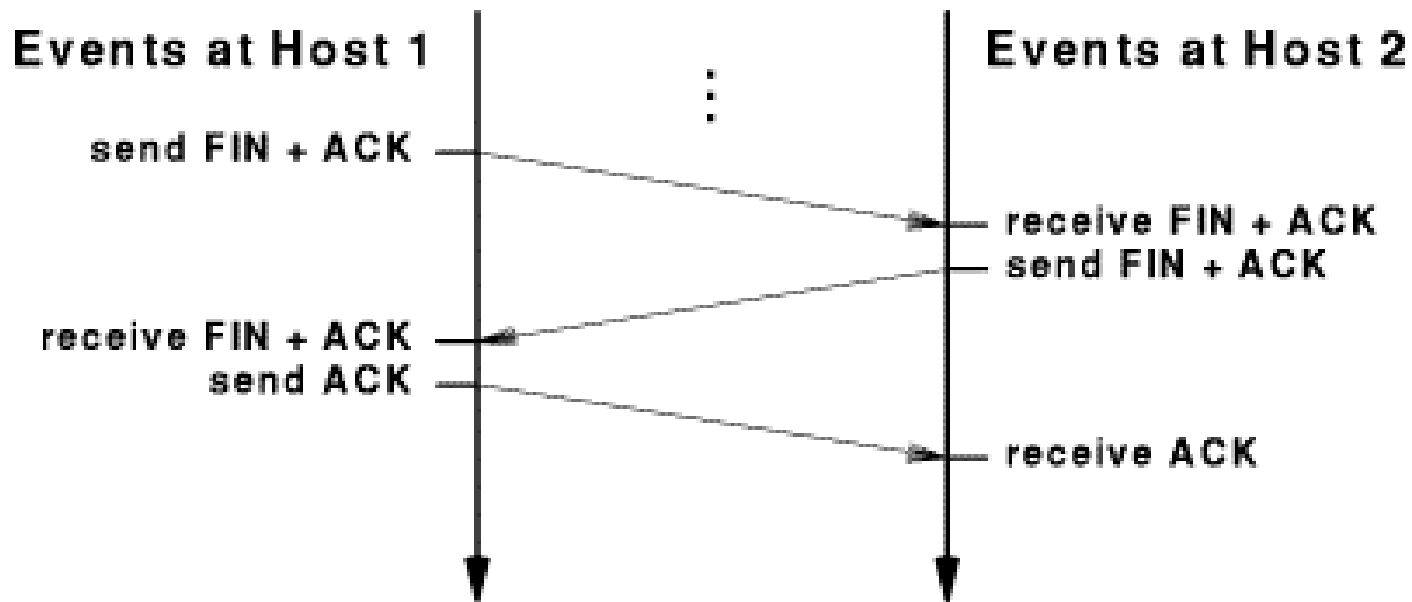
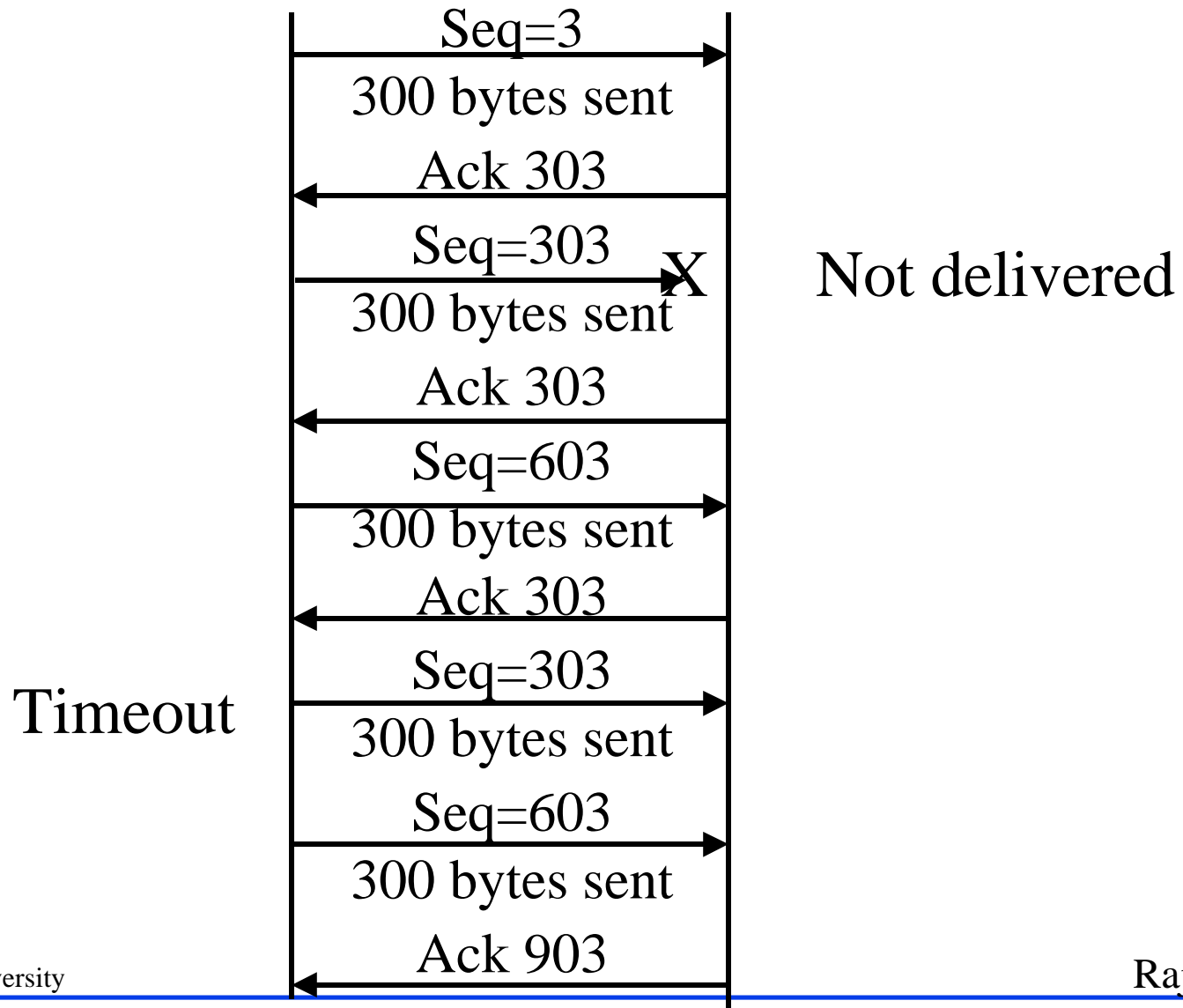


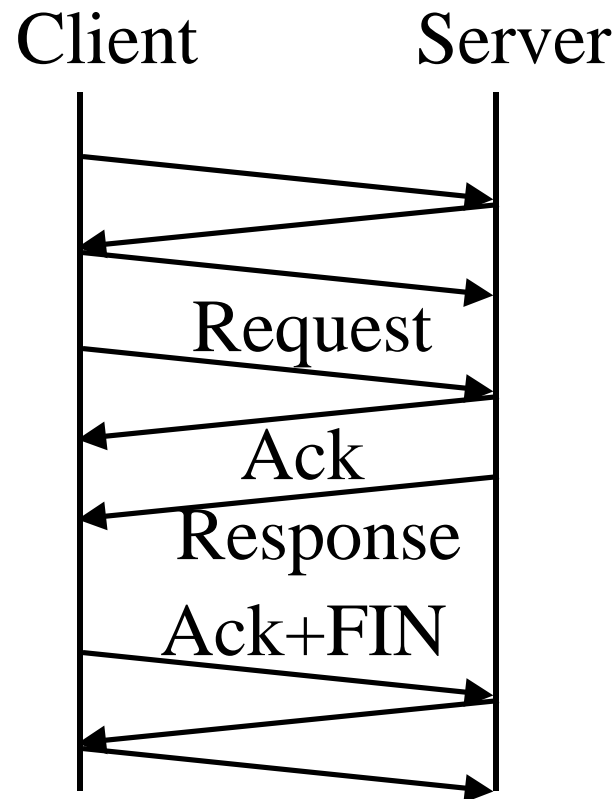
Fig 20.5

TCP Retransmission



T/TCP: Transaction Oriented TCP

- q Three-way handshake \Rightarrow Long delays for transaction-oriented (client-server) applications.
T/TCP avoids 3-way handshakes [RFC 1644].



Data Transfer

- q Stream: Every byte is numbered modulo 2^{32} .
- q Header contains the sequence number of the first byte
- q Flow control: Credit = number of bytes
- q Data transmitted at intervals determined by TCP
Push \Rightarrow Send now
- q Urgent: Send this data in ordinary data stream with urgent pointer
- q If TPDU not intended for this connection is received, the “reset” flag is set in the outgoing segment

Implementation Policies (Choices)

- q Send Policy:
 - Too little \Rightarrow More overhead. Too large \Rightarrow Delay
 - Push \Rightarrow Send now, if possible.
- q Delivery Policy:
 - May store or deliver each in-order segment.
 - Urgent \Rightarrow Deliver now, if possible.
- q Accept Policy:
 - May or May not discard out-of-order segments

Implementation Policies (Cont)

q Retransmit Policy:

First only

Retransmit all

Retransmit individual

(maintain separate timer for each segment)

q Ack Policy:

Immediate (no piggybacking)

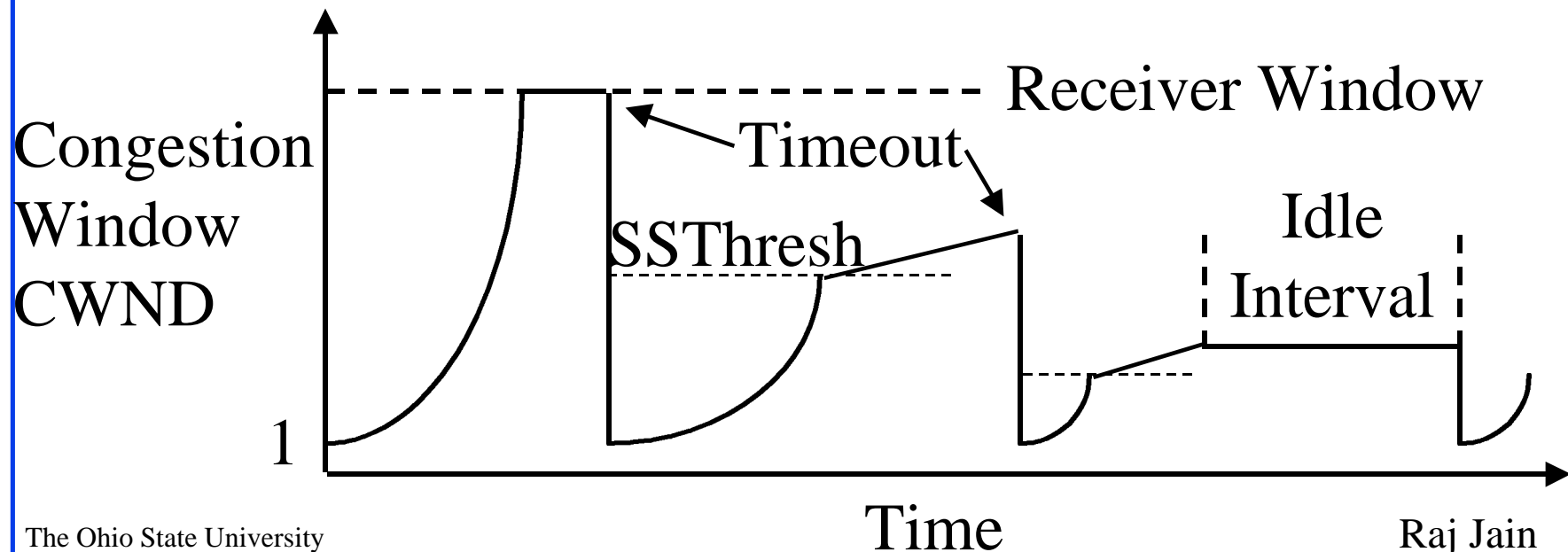
Cumulative (wait for outgoing data or timeout)

Slow Start Flow Control

- q Window = Flow Control Avoids receiver overrun
- q Need congestion control to avoid network overrun
- q The sender maintains two windows:
 - Credits from the receiver
 - Congestion window from the network
 - Congestion window is always less than the receiver window
- q Starts with a congestion window (CWND) of 1 segment (one max segment size)
 - ⇒ Do not disturb existing connections too much.
- q Increase CWND by 1 every time an ack is received

Slow Start (Cont)

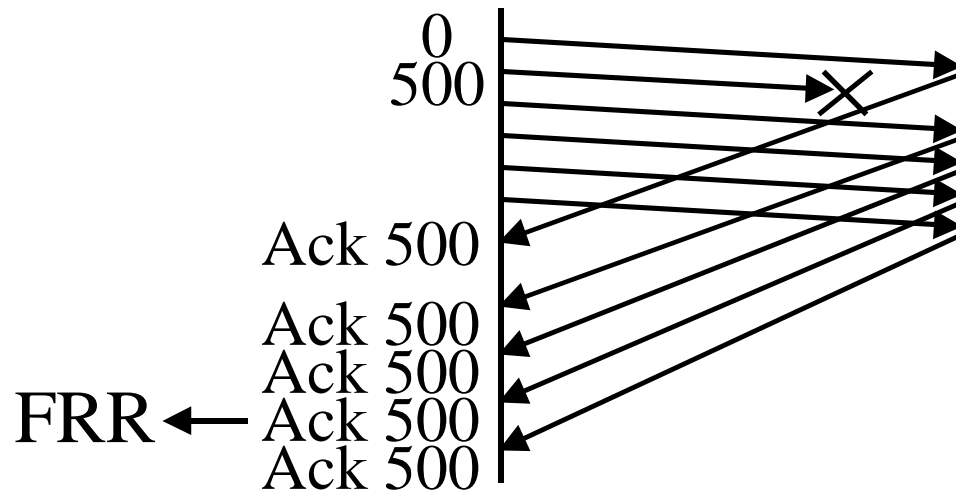
- q If packets lost, remember slow start threshold (SSThresh) to $CWND/2$
Set $CWND$ to 1
Increment by 1 per ack until SSThresh
Increment by $1/CWND$ per ack afterwards



Slow Start (Cont)

- q At the beginning, $SSThresh = \text{Receiver window}$
- q After a long idle period (exceeding one round-trip time), reset the congestion window to one.
- q Exponential growth phase is also known as “Slow start” phase
- q The linear growth phase is known as “congestion avoidance phase”

Fast Retransmit and Recovery (FRR)



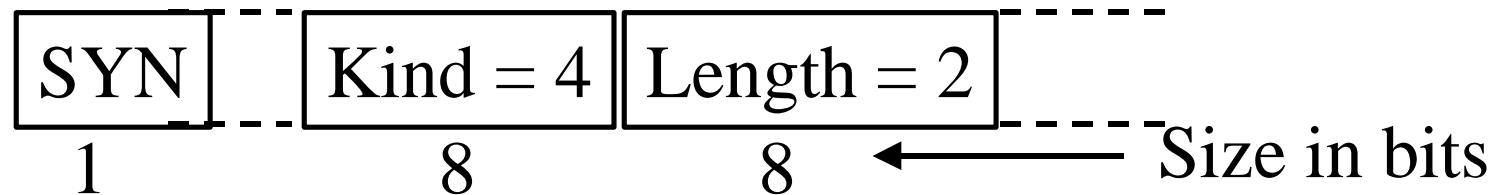
- q If 3 duplicate acks are received for the same packet, assume that the next packet has been lost. Retransmit it right away. Retransmit only one packet.
- q Helps if a single packet is lost.
Does not help if multiple packets lost.
- q Ref: Stevens, Internet draft

FRR (Cont)

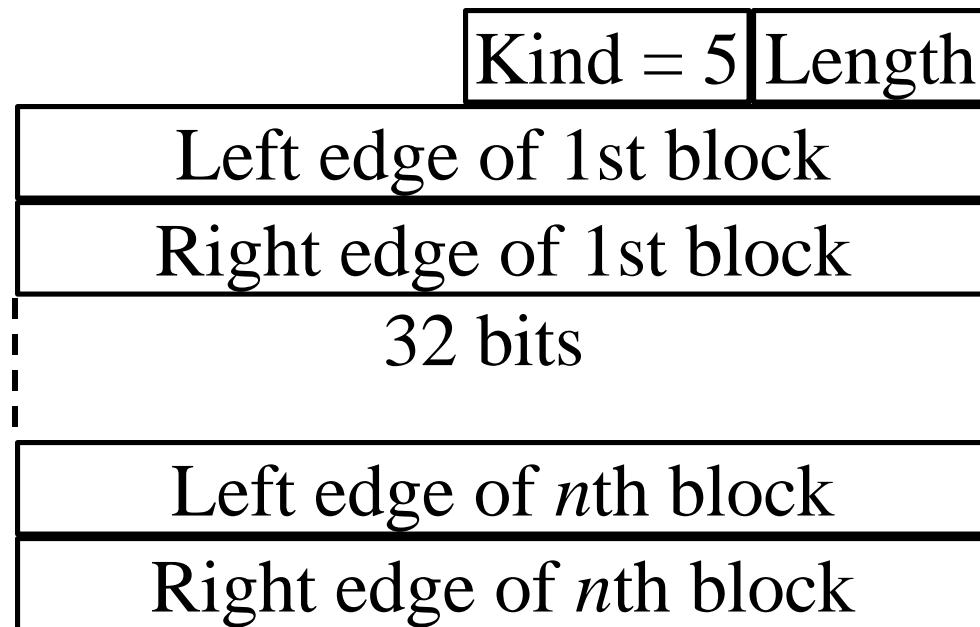
- q Upon receiving the third duplicate Ack:
 - q Set SS_{thresh} to $1/2$ of current CW_{ND}
 - q Retransmit the missing segment
 - q Set CW_{ND} to $SS_{thresh}+3$
- q For each successive duplicate Ack:
 - q Increment CW_{ND} by 1 MSS
 - q New packets are transmitted if allowed by CW_{ND}
- q Upon receiving the next (non-duplicate) Ack:
 - q Set CW_{ND} to $SS_{thresh} \Rightarrow$ Enter linear growth phase
- q Receiver caches out-of-order data.

Selective Ack (SACK)

- q Initial Negotiation: Sender to receiver: “sack permitted”



- q Selective Ack: Variable length. Receiver to sender

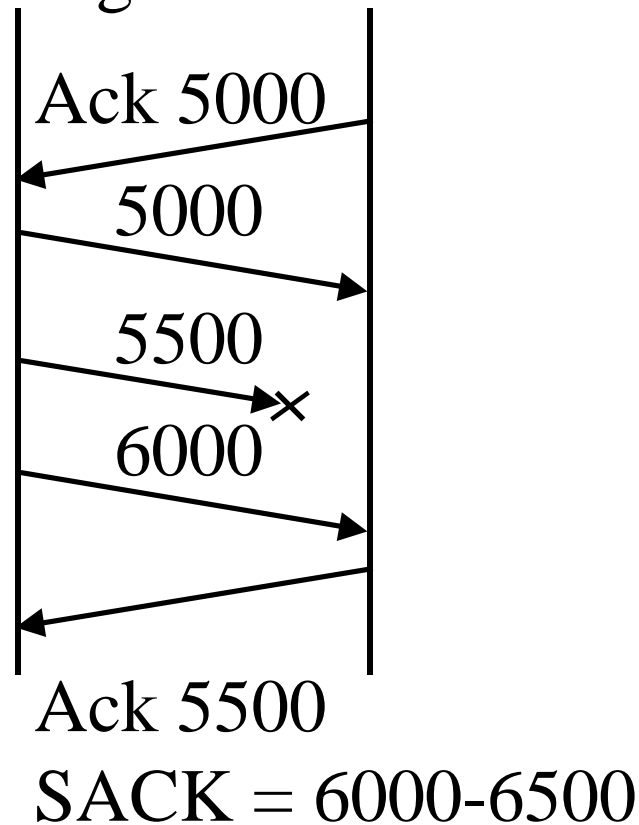


SACK (Cont)

- q Left edge = 1st sequence number in this block
- q Right edge = sequence number immediately after the last sequence number in this block
- q Ack field meaning is same as before.
It is the next byte the receiver is expecting.
- q When missing segments are received, ack field is advanced.
- q Receiver can send SACK only if sender has “sack permitted” option in the SYN segment of the connection.
- q Option Length = $8*n+2$ byte for n blocks.
40 Bytes max options \Rightarrow Max n = 4

SACK (Cont)

- q Data receiver can discard SACKed (queued) data
Sender must not discard data until acked.
- q Example: 500 byte segments



Window Scaling Option

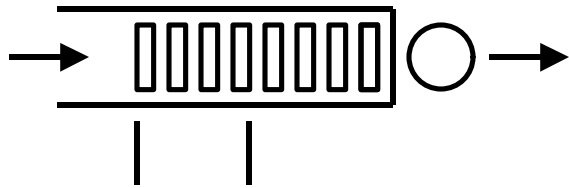
- q Long Fat Pipe Networks (LFN): Satellite links
Pronounced elephant(t)
 - q Need very large window sizes.
 - q Normally, Max window = $2^{16} = 64$ KBytes
 - q Window scale option: Window = $W \times 2^S$
- | | | |
|----------|------------|-------|
| Kind = 3 | Length = 3 | Scale |
|----------|------------|-------|
- q Max window = $2^{16} \times 2^{255}$
 - q Option sent only in SYN an SYN
+ Ack segments.



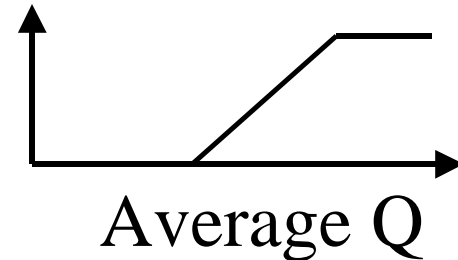
q RFC 1323
The Ohio State University

Raj Jain

Random Early Drop (RED)



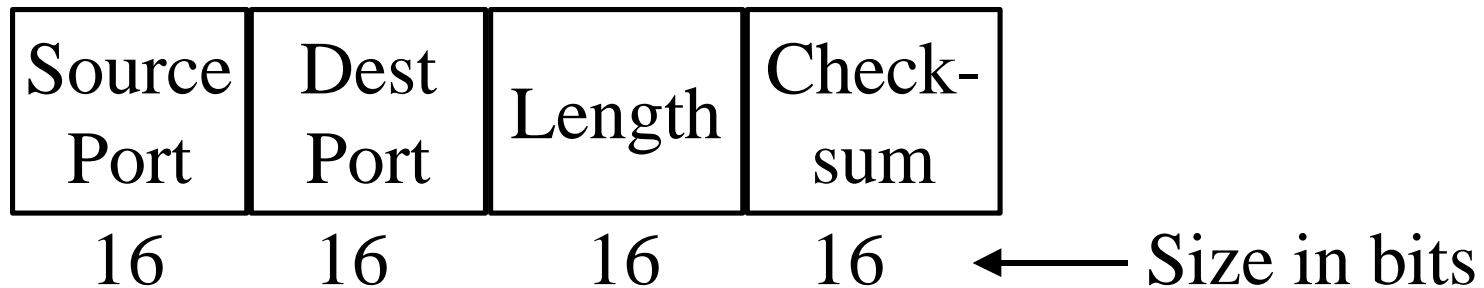
Probability
of Drop



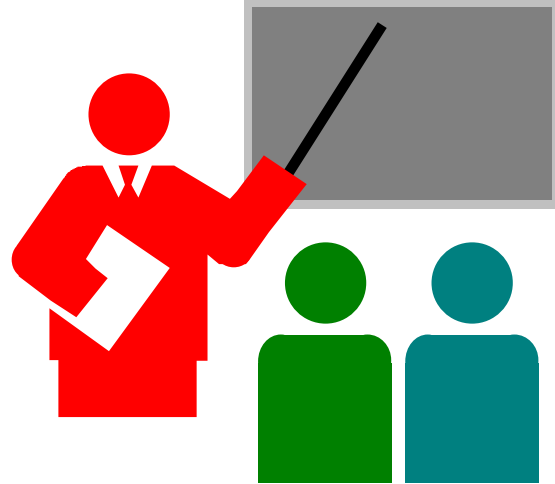
- q Routers compute average queue size using an exponential weighted average
- q If the average queue size is more than a high-threshold, drop all arriving packets
- q If the average queue size is between the low and high threshold, drop the arriving packet with a probability $p = \text{fn}(\text{avg } q, \# \text{ of packets since the last dropped packet})$
- q High-rate sources are more likely to be dropped

User Datagram Protocol (UDP)

- q Connectionless end-to-end service
- q No flow control. No error recovery (no acks)
- q Provides port addressing
- q Error detection (Checksum) optional. Applies to pseudo-header (same as TCP) and UDP segment. If not used, it is set to zero.
- q Used by network management



Summary



- q TCP provides reliable full-duplex connections.
- q TCP Streams, credit flow control, 3-way handshake
- q Slow-start, Fast retransmit/recovery, SACK, Scaling
- q UDP is connectionless and simple. No flow/error control.