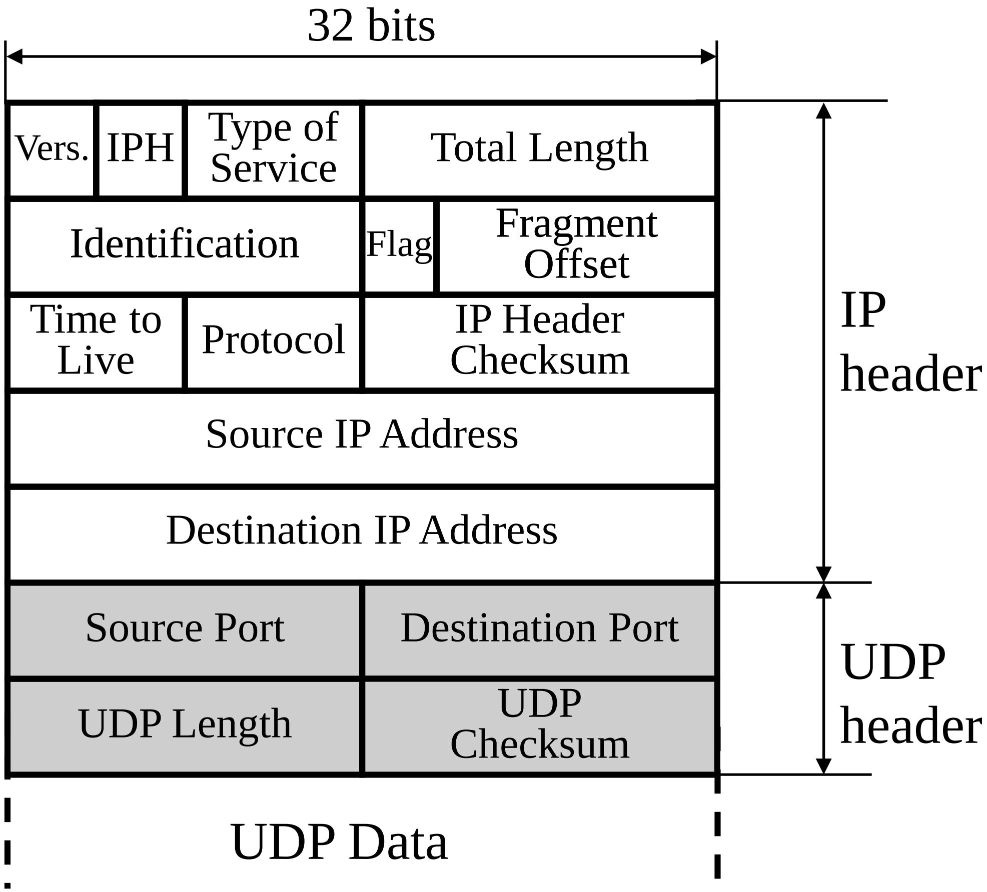
Telecommunications Networking

1. Introduction
2. Local Area Networking
3. Error Control and Retransmission Protocols
4. Architectural Principles of the Internet
5. **Flow Control in Internet: TCP**
6. Routing Algorithms
7. Routing Protocols
8. The principles of ATM
9. Traffic Management in ATM
10. Scheduling
11. Quality of Service
12. Quality of Service routing
13. Peer-to-peer networks

Transport layer protocols

- 2 major transport layer protocols:
  - User Datagram Protocol (UDP)
  - Transmission Control Protocol (TCP)
- UDP: unreliable connectionless
- TCP reliable connection-oriented

UDP Header and Pseudoheader



32 bits

Vers. | IPH | Type of Service | Total Length

Identification | Flag | Fragment Offset

Time to Live | Protocol | IP Header Checksum

Source IP Address

Destination IP Address

Source Port | Destination Port

UDP Length | UDP Checksum

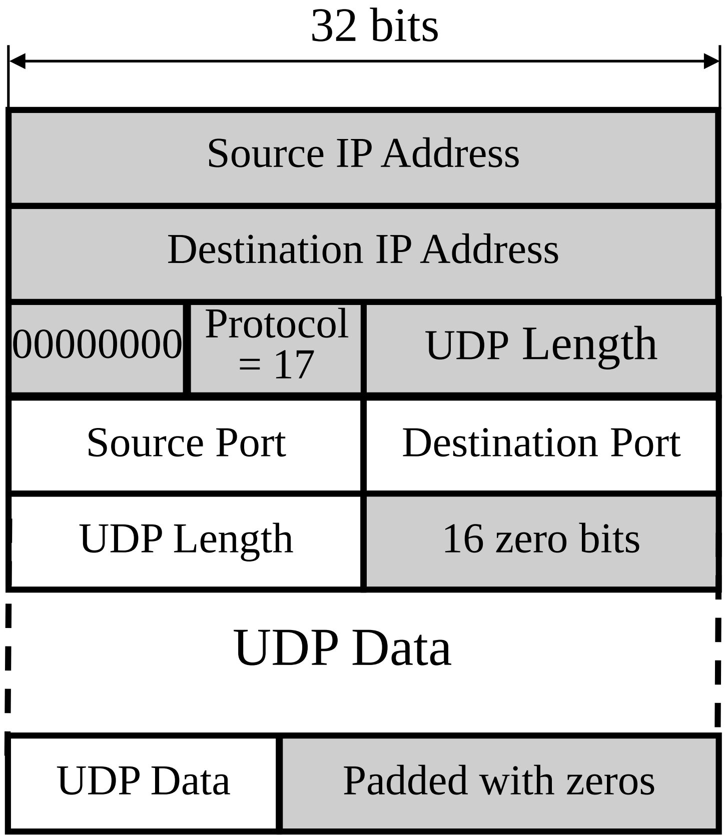
UDP Data

IP header

UDP header

(A)

IP and UDP header



32 bits

Source IP Address

Destination IP Address

00000000 | Protocol = 17 | UDP Length

Source Port | Destination Port

UDP Length | 16 zero bits

UDP Data

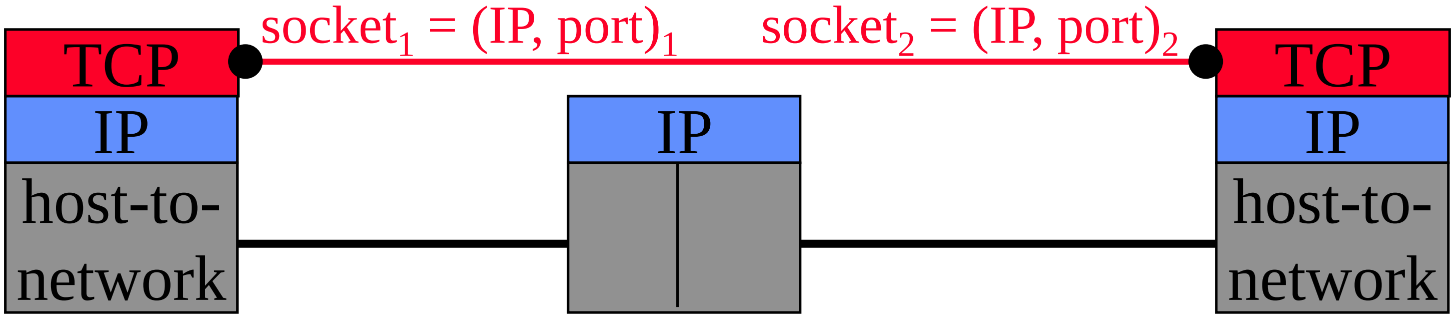
UDP Data | Padded with zeros

(B)

Pseudoheader used in UDP checksum computation

Transmission Control Protocol (TCP)

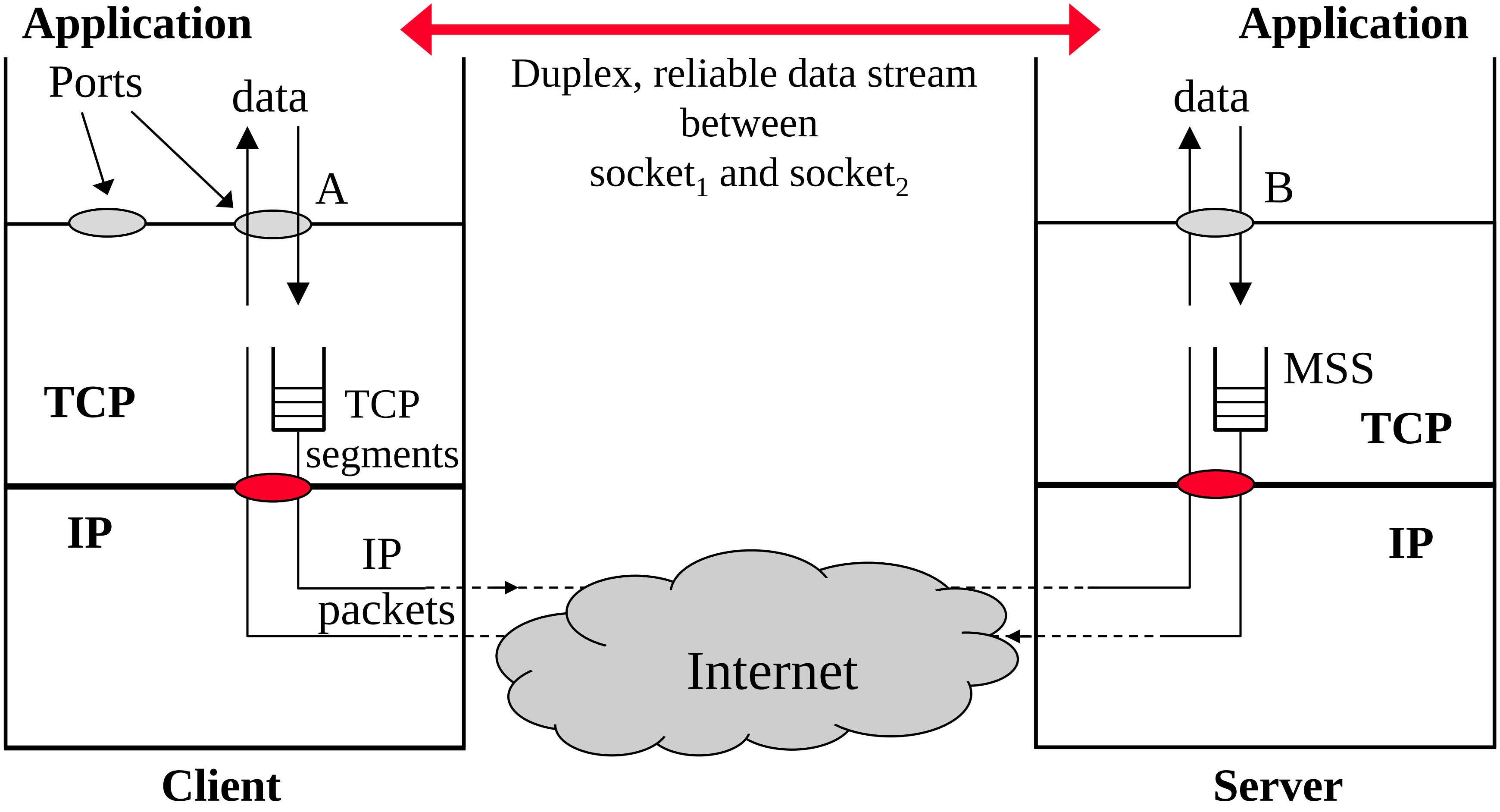
- Connection-oriented over connectionless
- Reliable transport
  - Principles of sliding window
  - Positive acknowledgment with retransmission
- Details:
  - full duplex, ports, connections and endpoints (socket_1,socket_2)


$\text{socket}_1 = (\text{IP, port})_1$
$\text{socket}_2 = (\text{IP, port})_2$
TCP
IP
host-to-network
IP
TCP
IP
host-to-network

TCP basic operation

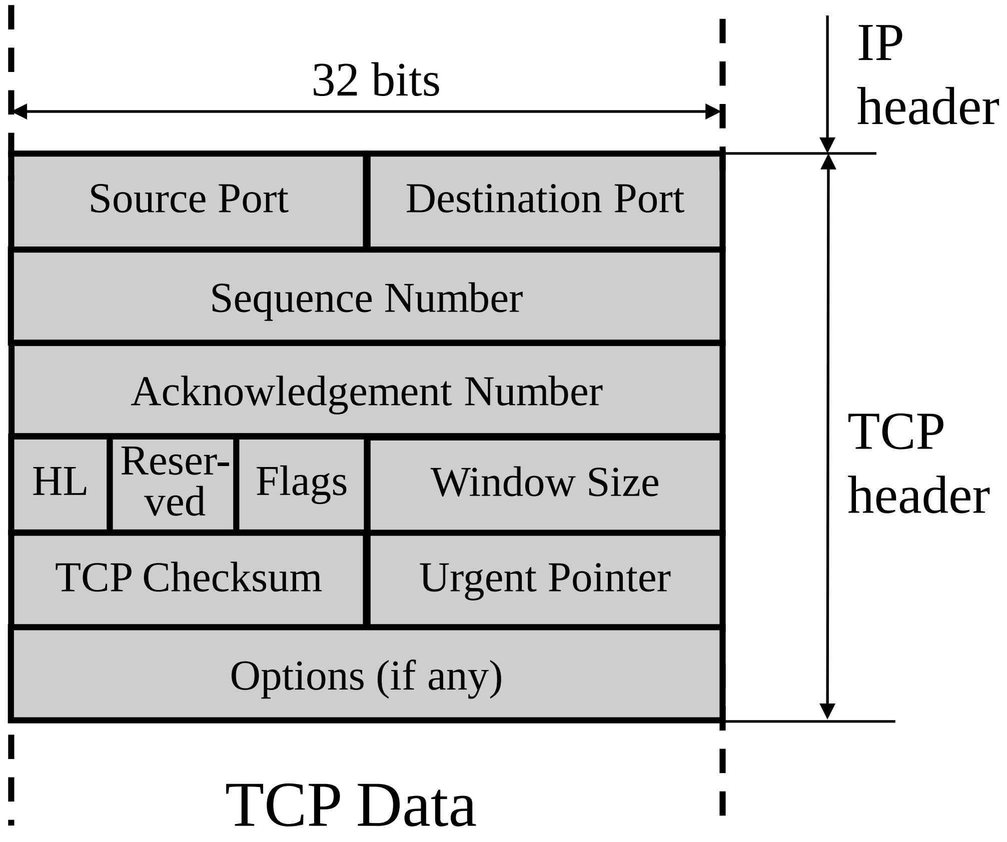
- Data stream grouped in TCP segments
- Timer and time-out threshold
- End-to-end checksum on header and data
- Re-sequencing
- Flow control:
  - Sliding window (imposed by receiver)
  - Slow start (imposed by sender)

TCP viewed by an application



MSS = Maximum segment size

TCP Header



32 bits

IP header

Source Port | Destination Port

Sequence Number

Acknowledgement Number

HL | Reserved | Flags | Window Size

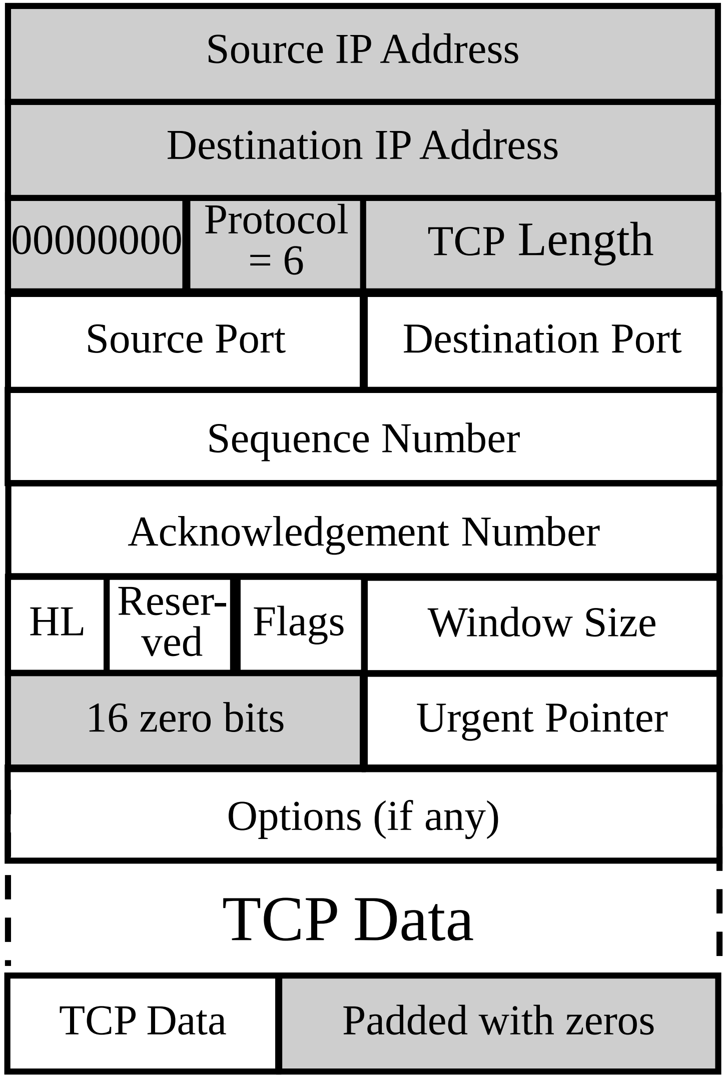
TCP Checksum | Urgent Pointer

Options (if any)

TCP header

TCP Data

(A)



Source IP Address

Destination IP Address

00000000 | Protocol = 6 | TCP Length

Source Port | Destination Port

Sequence Number

Acknowledgement Number

HL | Reserved | Flags | Window Size

16 zero bits | Urgent Pointer

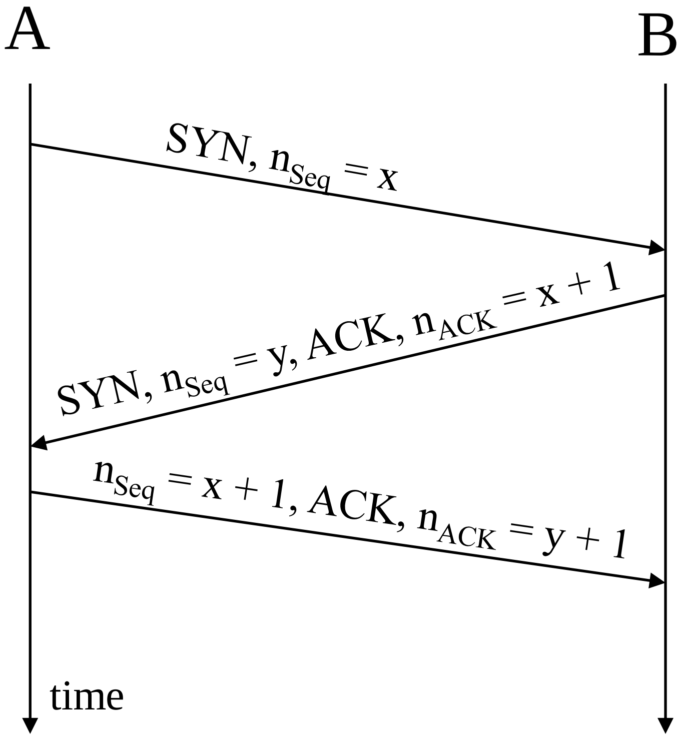
Options (if any)

TCP Data

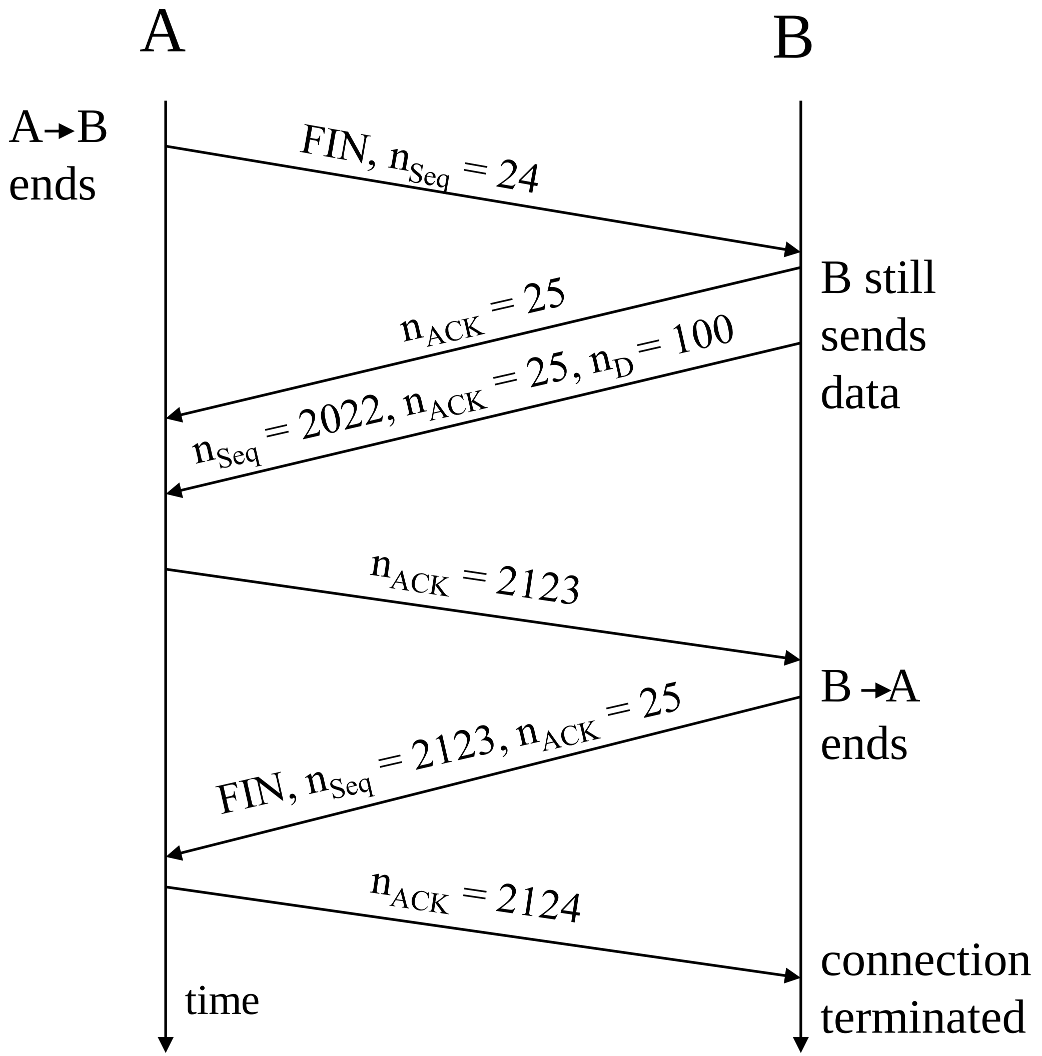
TCP Data | Padded with zeros

(B)

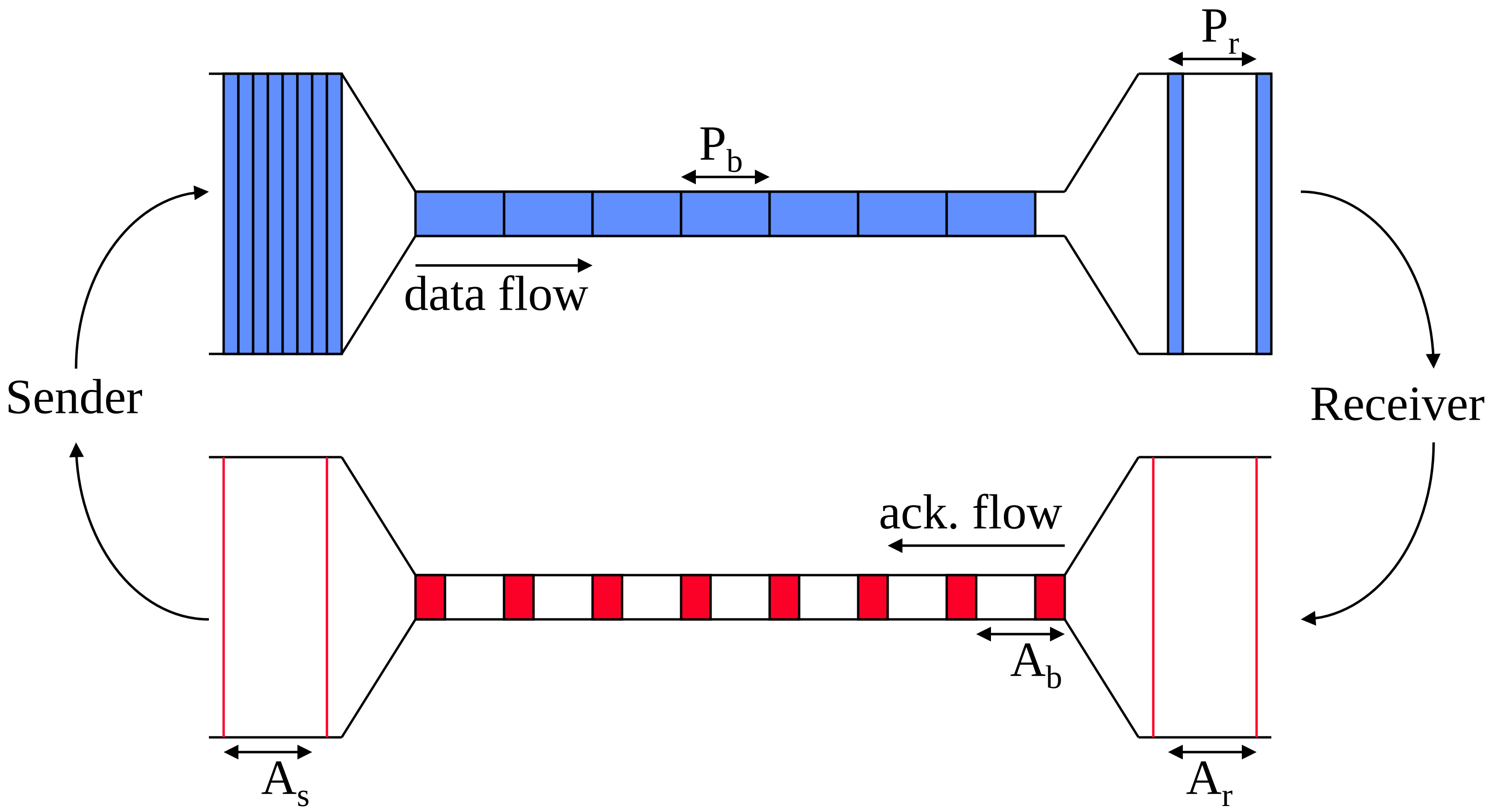
Three-way handshake


A
B
SYN, $n_{Seq} = x$
SYN, $n_{Seq} = y$, ACK, $n_{ACK} = x + 1$
$n_{Seq} = x + 1$, ACK, $n_{ACK} = y + 1$
time

TCP connection termination

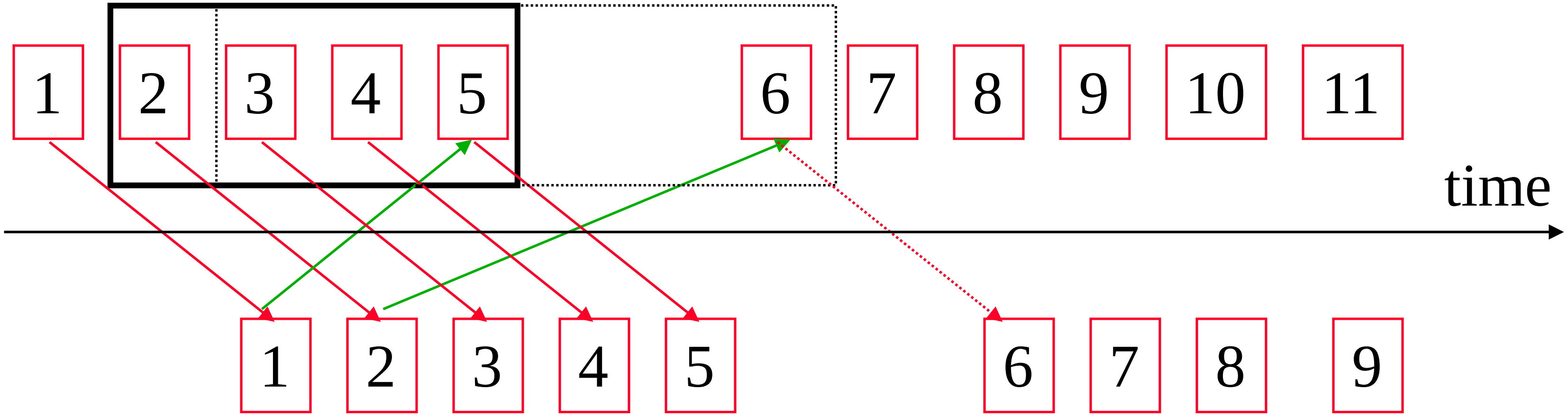


TCP is ‘self-clocking’



Delft University of Technology

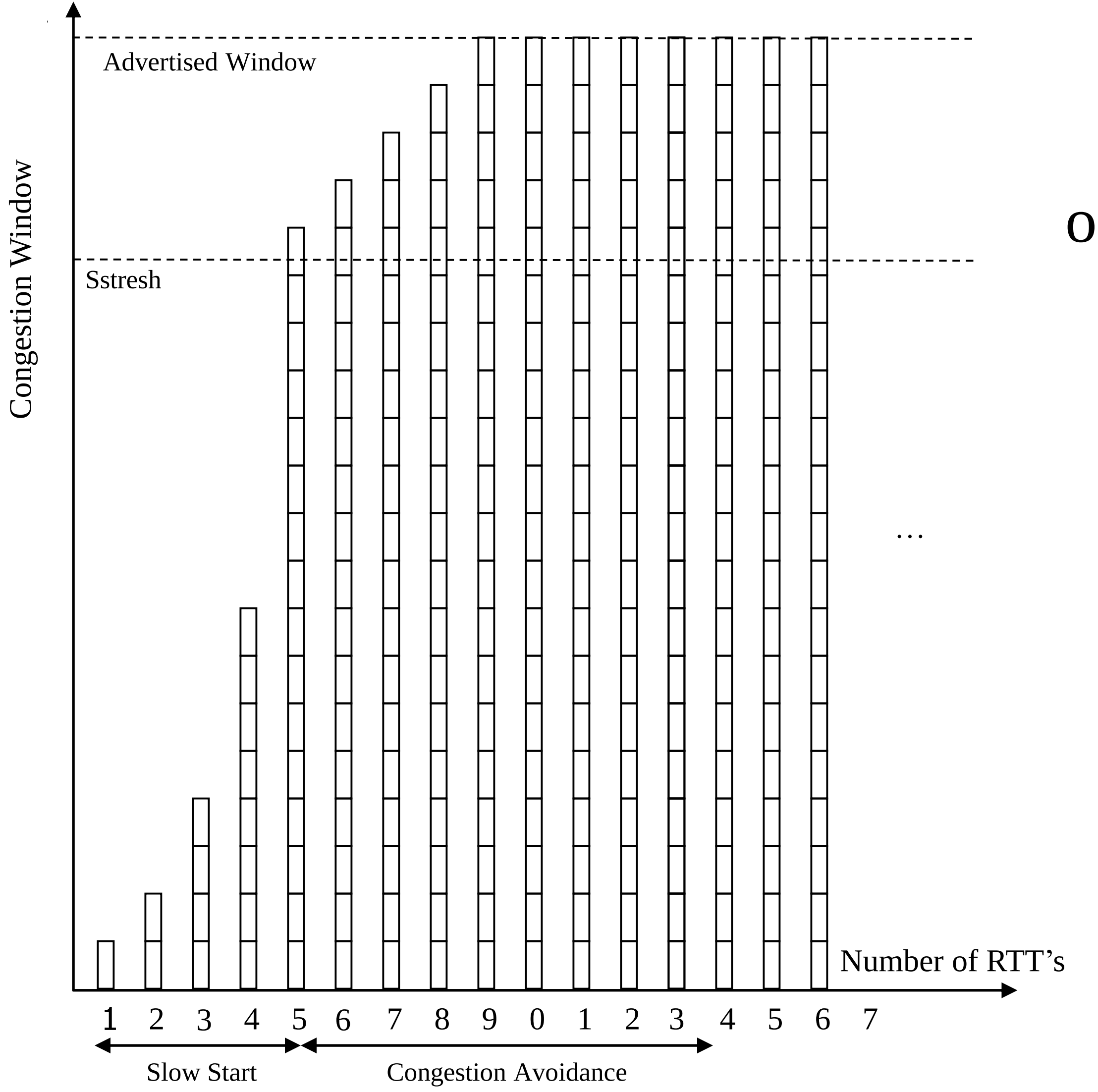
Sliding Window Size



Tuning sliding window impacts number of packets in network

Flow control mechanism: optimal throughput

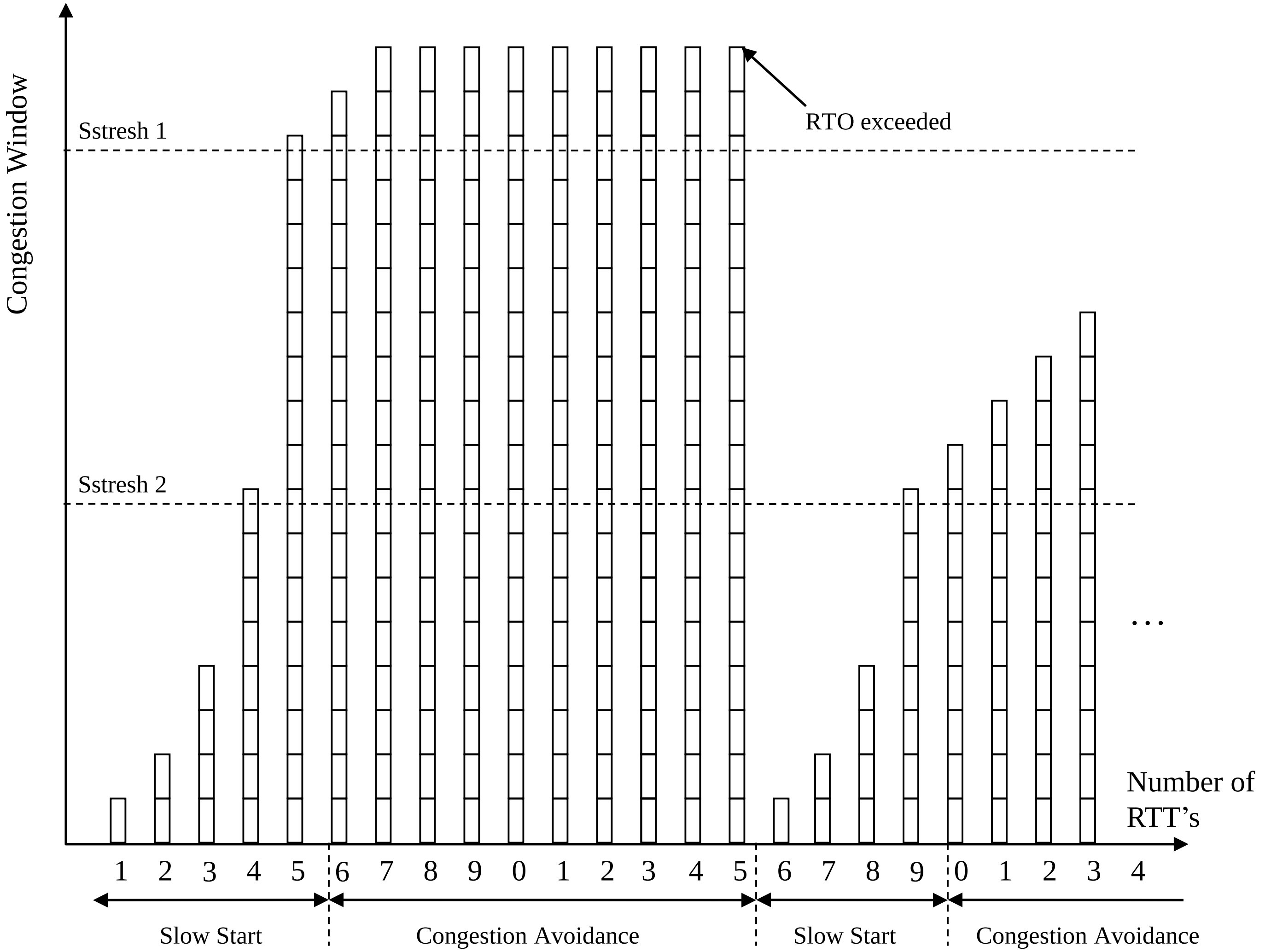
TCP Congestion Window: no loss



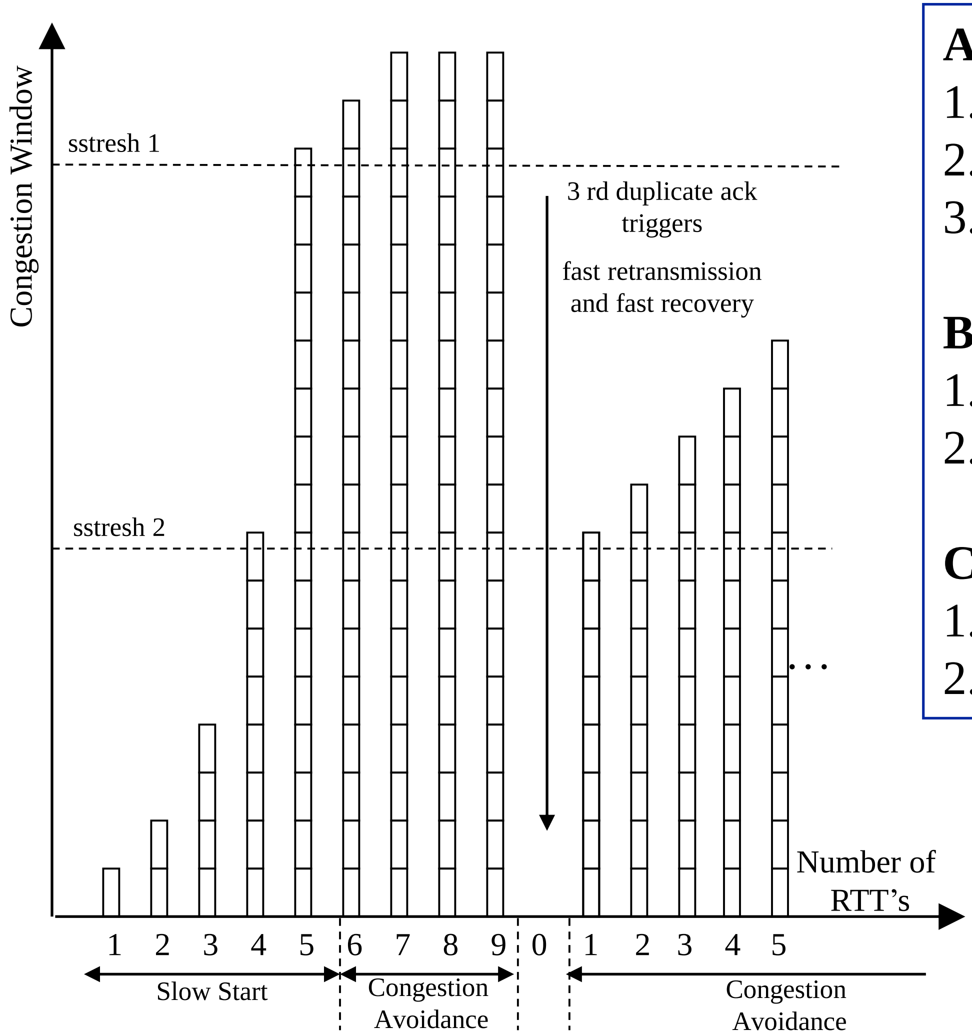
on the arrival of an ack:

```
if (W < sstresh)
    W ++;
else
    $W += \frac{1}{W};$
```

TCP Congestion Window: Time-out and Retransmission



Fast Retransmit and Recovery



A. at receipt of 3rd duplicate ack:

1. sstresh ← W/2
2. retransmit missing segment L
3. W ← sstresh + 3

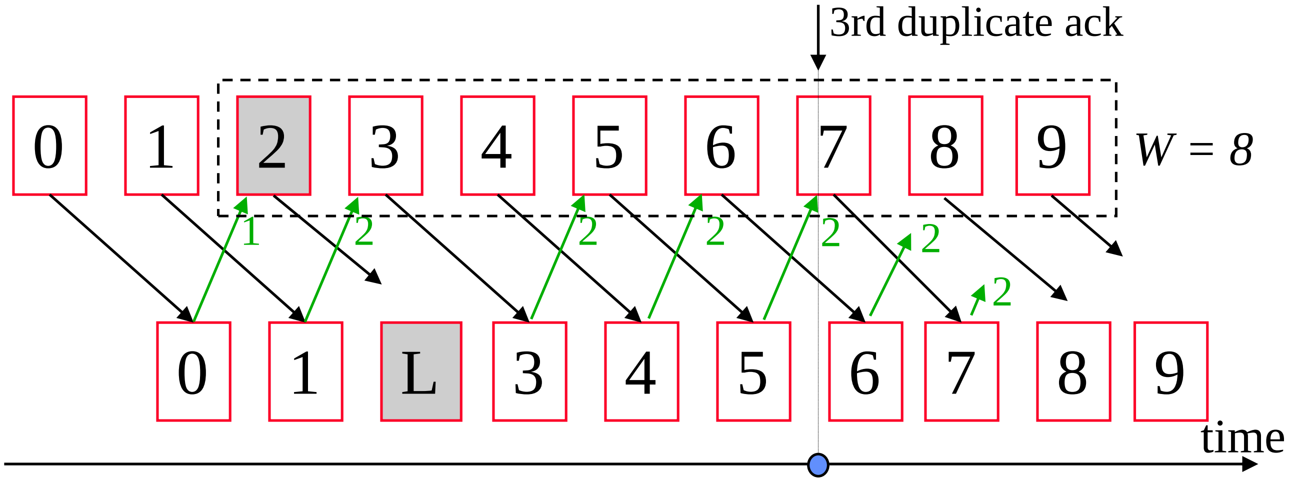
B. at receipt of another duplicate ack:

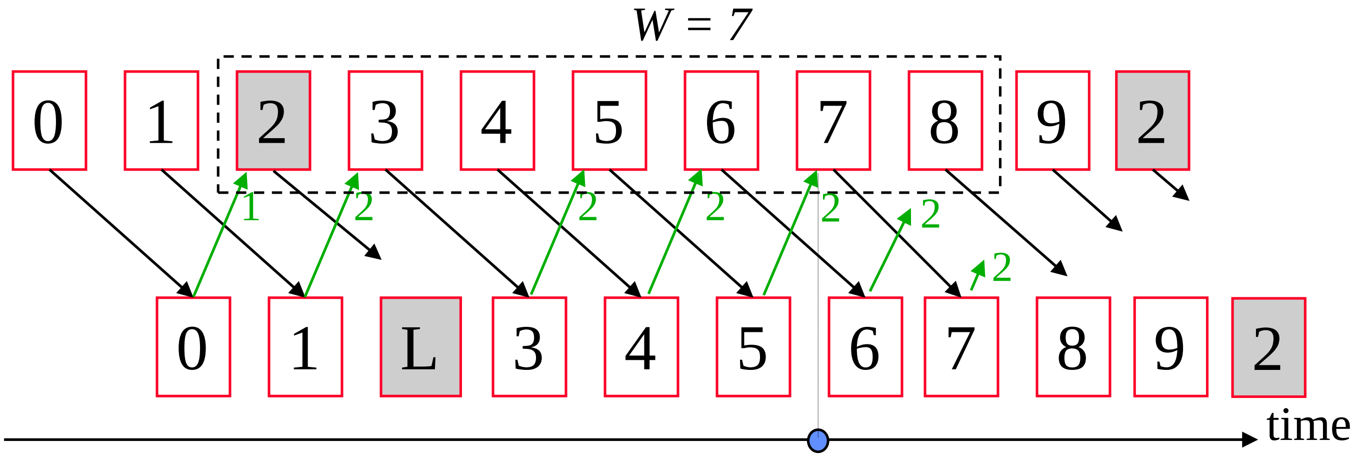
1. W++
2. transmit new, not in transit segment

C. at receipt of ack L

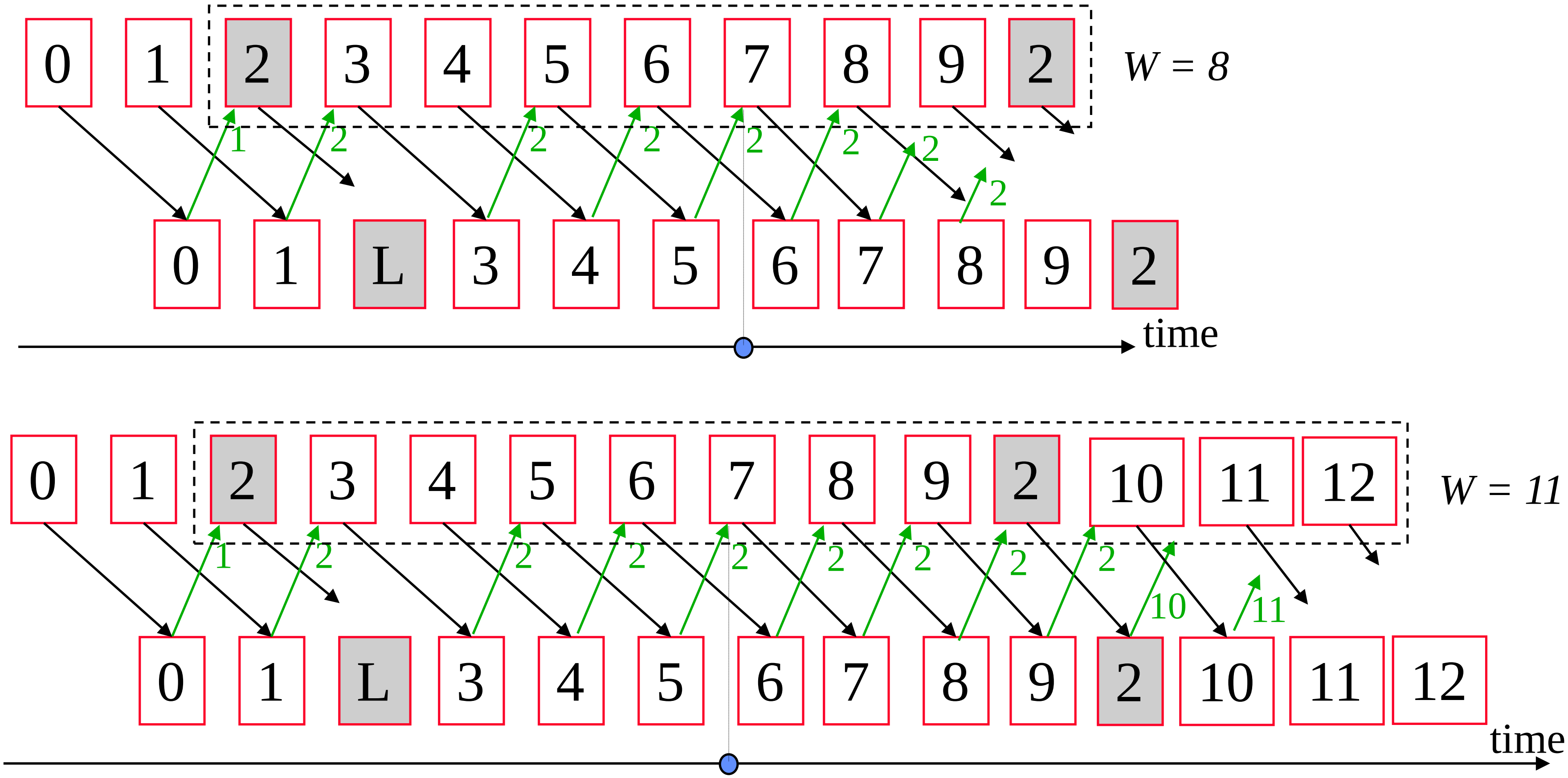
1. W ← sstresh
2. start congestion avoidance phase

Fast Recovery

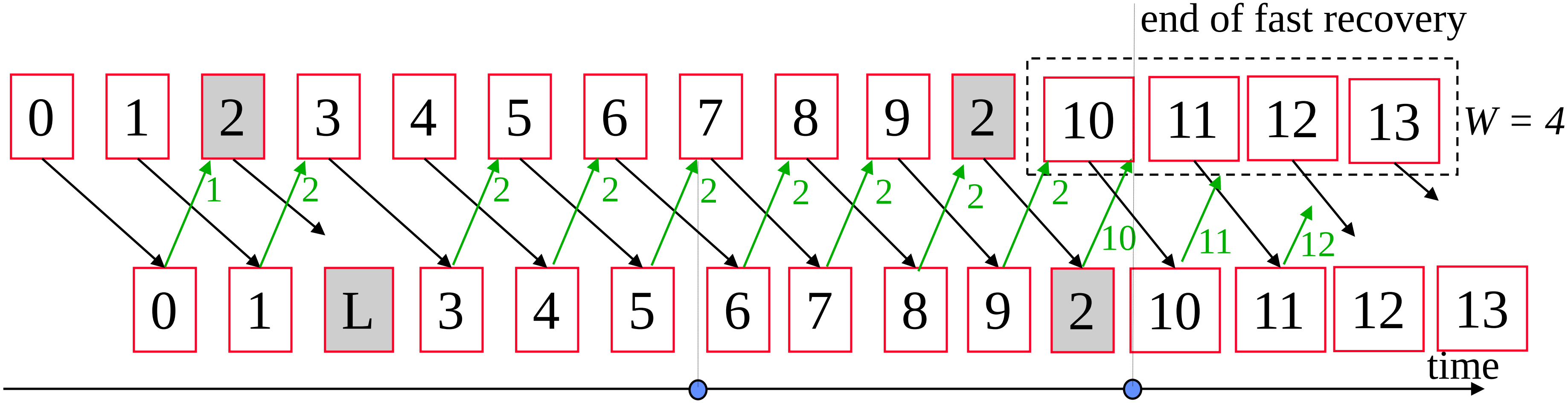




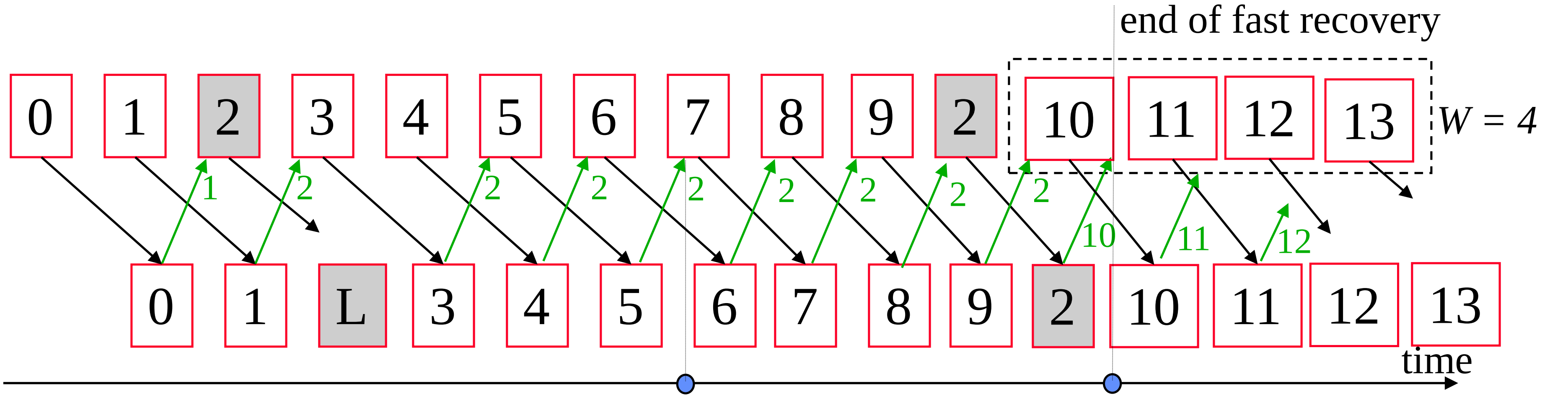
Fast Recovery

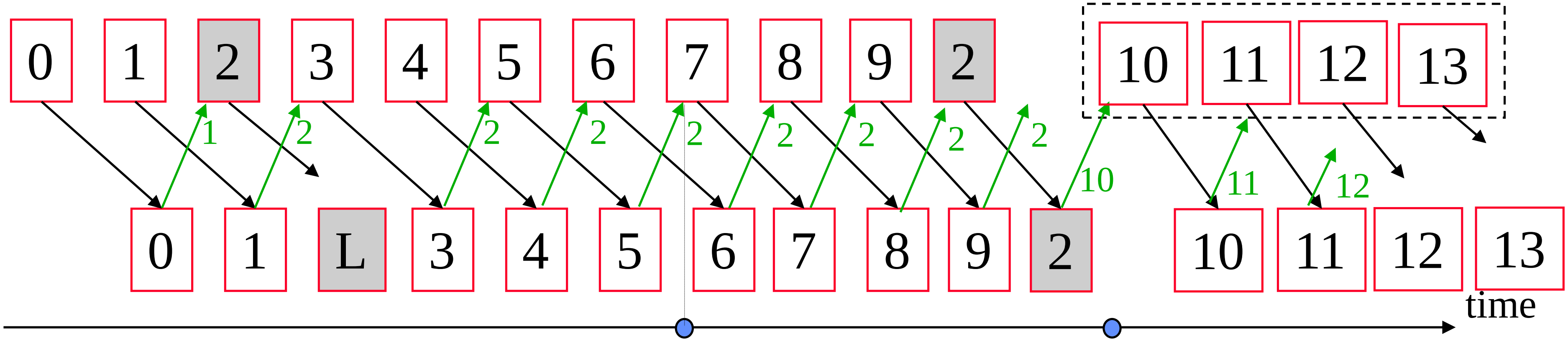


Fast Recovery

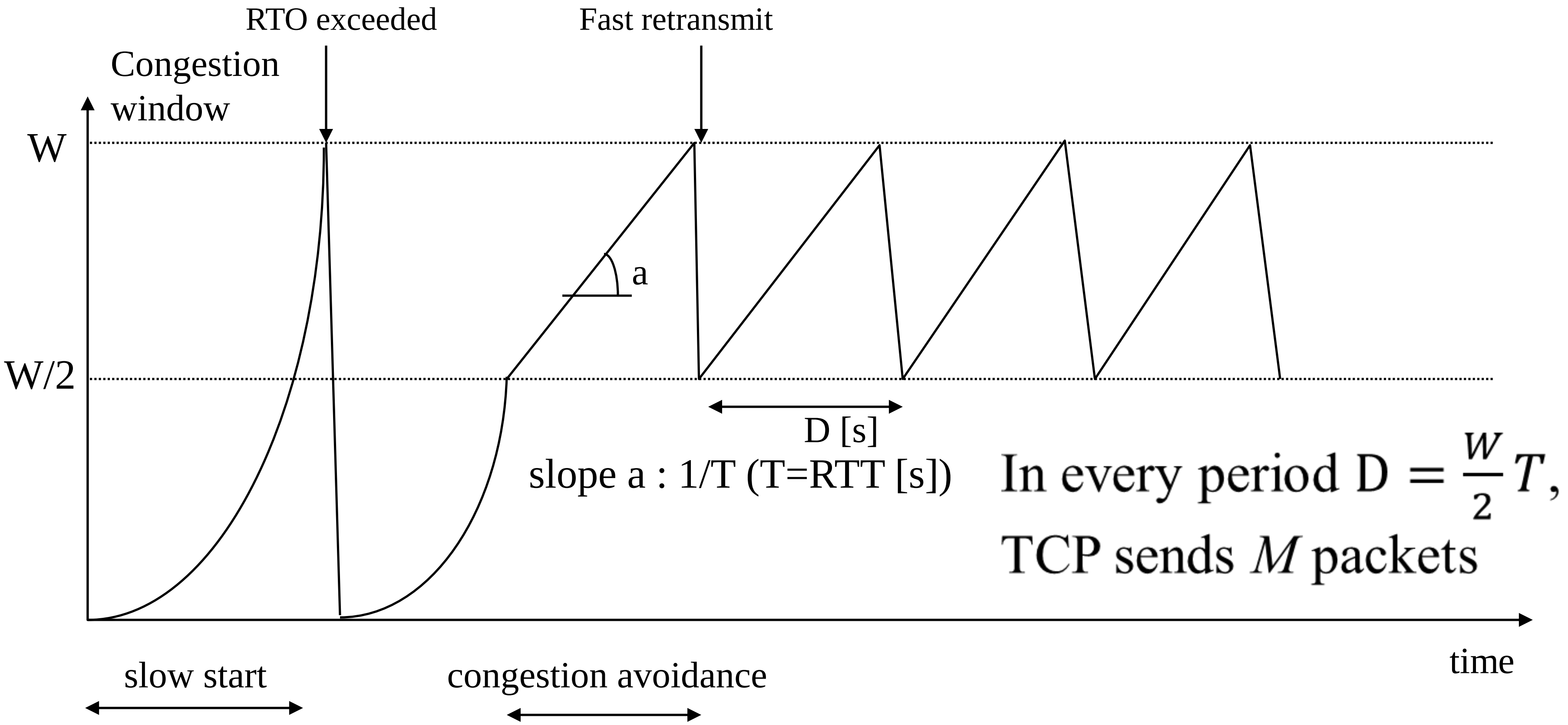


No Fast Recovery





TCP Throughput



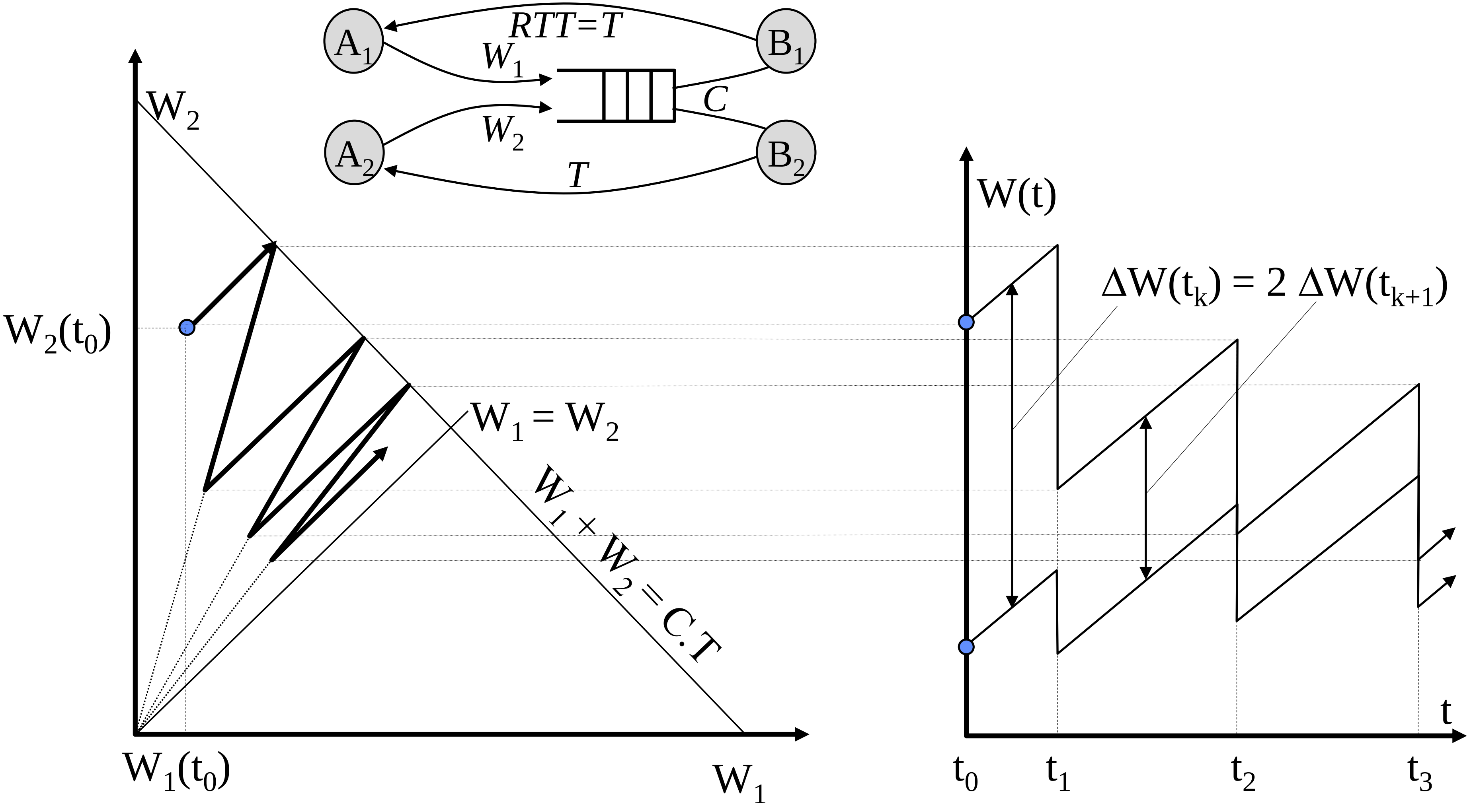
slope a : 1/T (T=RTT [s])

In every period $D = \frac{W}{2}T$, TCP sends M packets

$$M = \frac{D}{T}\frac{W}{2} + \frac{D}{T}\frac{W}{4} = 3/8W^2 \quad \text{(area of a “cycle”)}$$

throughput [segments/s]: $R = \frac{M}{D} = \frac{3W}{4T}$

Additive Increase - Multiplicative Decrease

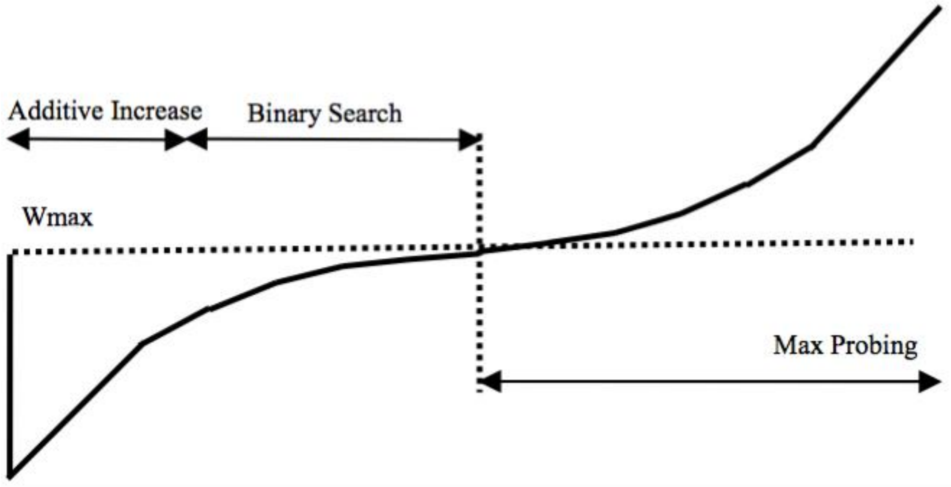


Reality

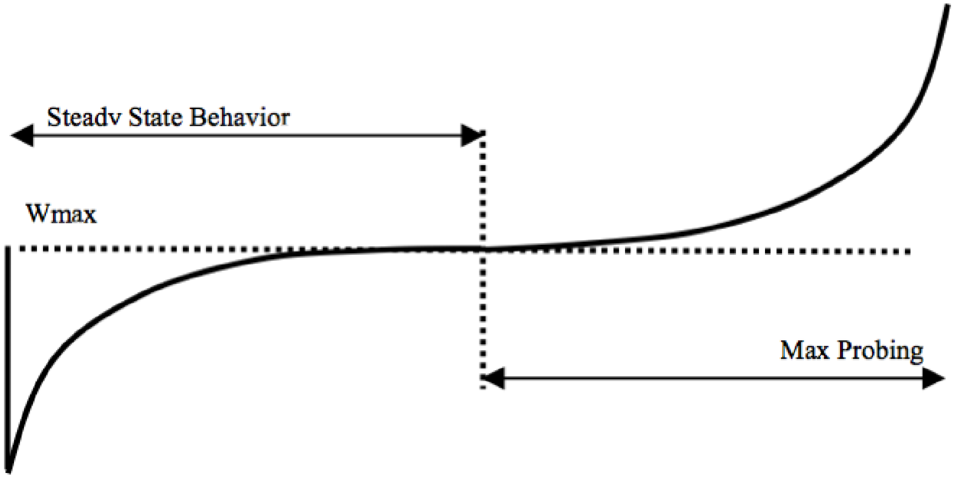
- Optimality varies by distance
- Relies on equal behavior by all hosts, though variation exists,
- Windows uses TCP Reno like
  - Slow-start, fast retransmit + fast recovery
  - Windows 8+ Compound TCP, optimizing time spent in the highest congestion window

BIC + CUBIC TCP

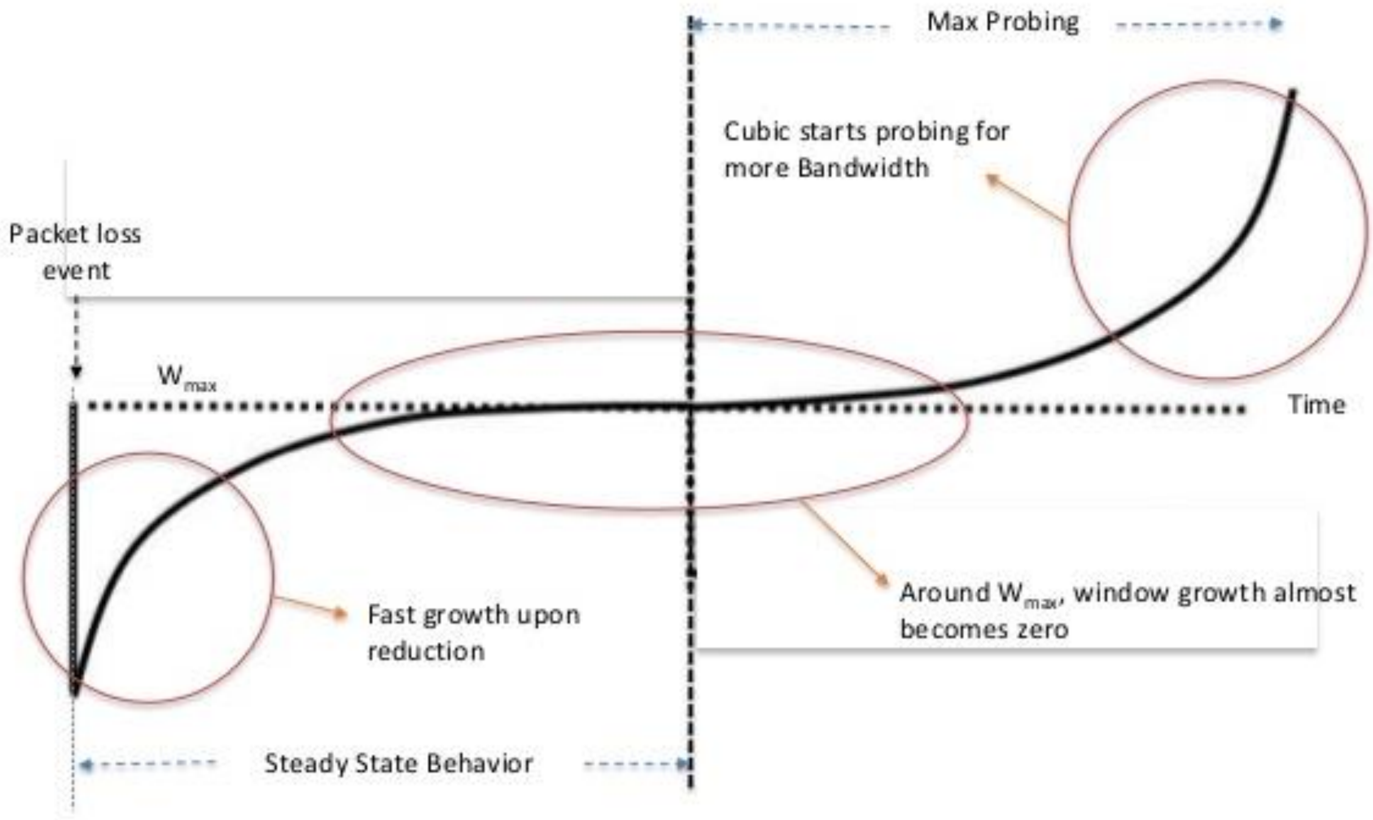
- MAC and Linux use BIC + CUBIC TCP
  - BIC: Binary search of Wmax
  - Decreasing slope



  - CUBIC: cubic function polynomial degree 3 $(ax^3 + bx^3 + cx^3 + d)$



CUBIC TCP


Max Probing
Cubic starts probing for more Bandwidth
Packet loss event
W_{max}
Time
Fast growth upon reduction
Around W_{max}, window growth almost becomes zero
Steady State Behavior

[http://www.slideshare.net/deawooKim/cubic-kdw]

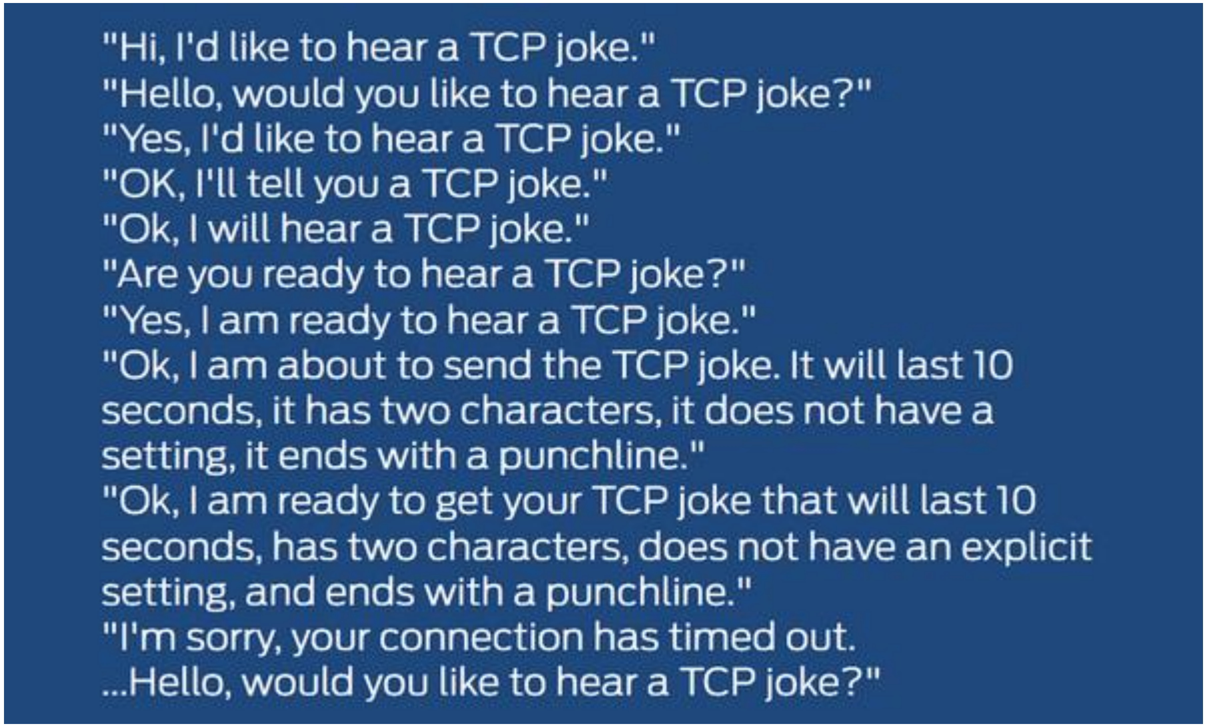
Flow Control Design

- Manage the shared use of the network
- Decentralized solution
- Design flow control:
  - Control parameter (TCP: window)
  - Congestion signal (TCP: segment losses detected via RTO and duplicate ACKs)
  - Control algorithm (TCP: AIMD)
- Efficiency
- Fairness

Questions Ch. 5

- Contrary to TCP, UDP has no mechanism to provide reliable and in-sequence delivery of packets. Can you explain in which cases, if any, UDP is better suited than TCP?
- Explain slow start and congestion avoidance in TCP. Why was this mechanism proposed?
- Explain TCP’s self-clocking property.

TCP joke



"Hi, I'd like to hear a TCP joke."
"Hello, would you like to hear a TCP joke?"
"Yes, I'd like to hear a TCP joke."
"OK, I'll tell you a TCP joke."
"Ok, I will hear a TCP joke."
"Are you ready to hear a TCP joke?"
"Yes, I am ready to hear a TCP joke."
"Ok, I am about to send the TCP joke. It will last 10 seconds, it has two characters, it does not have a setting, it ends with a punchline."
"Ok, I am ready to get your TCP joke that will last 10 seconds, has two characters, does not have an explicit setting, and ends with a punchline."
"I'm sorry, your connection has timed out.
...Hello, would you like to hear a TCP joke?"