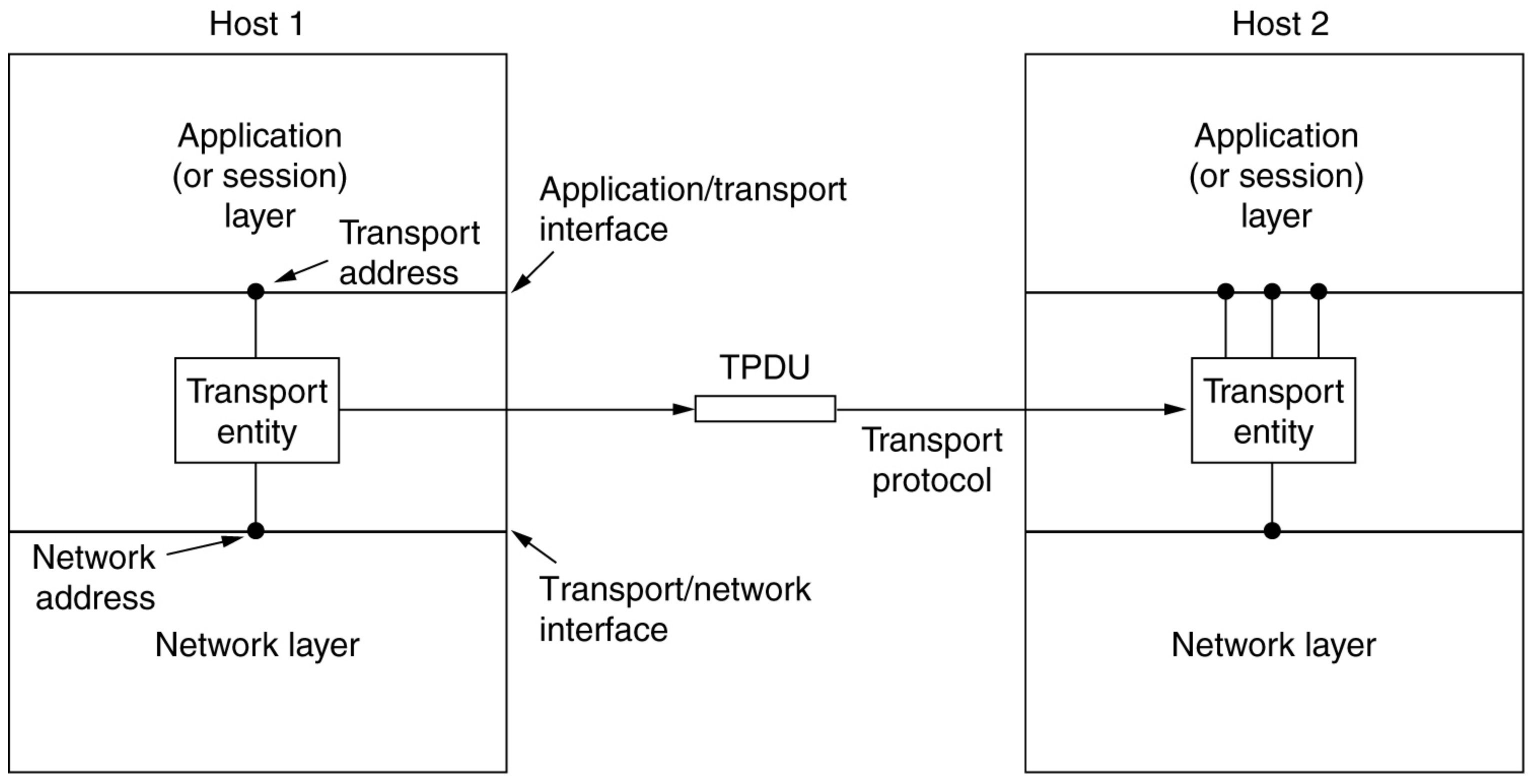
Chapter 6

The Transport Layer

The Transport Service

- Services Provided to the Upper Layers
- Transport Service Primitives
- Berkeley Sockets
- An Example of Socket Programming:
  - An Internet File Server

Services Provided to the Upper Layers



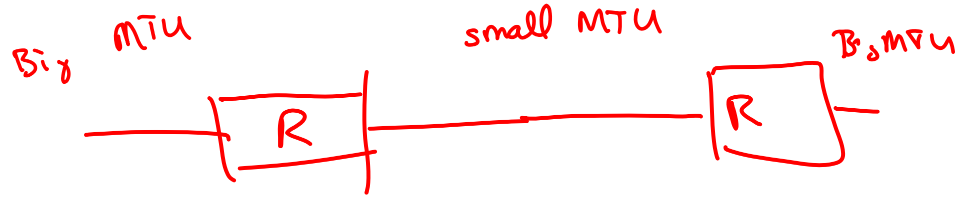
The network, transport, and application layers.

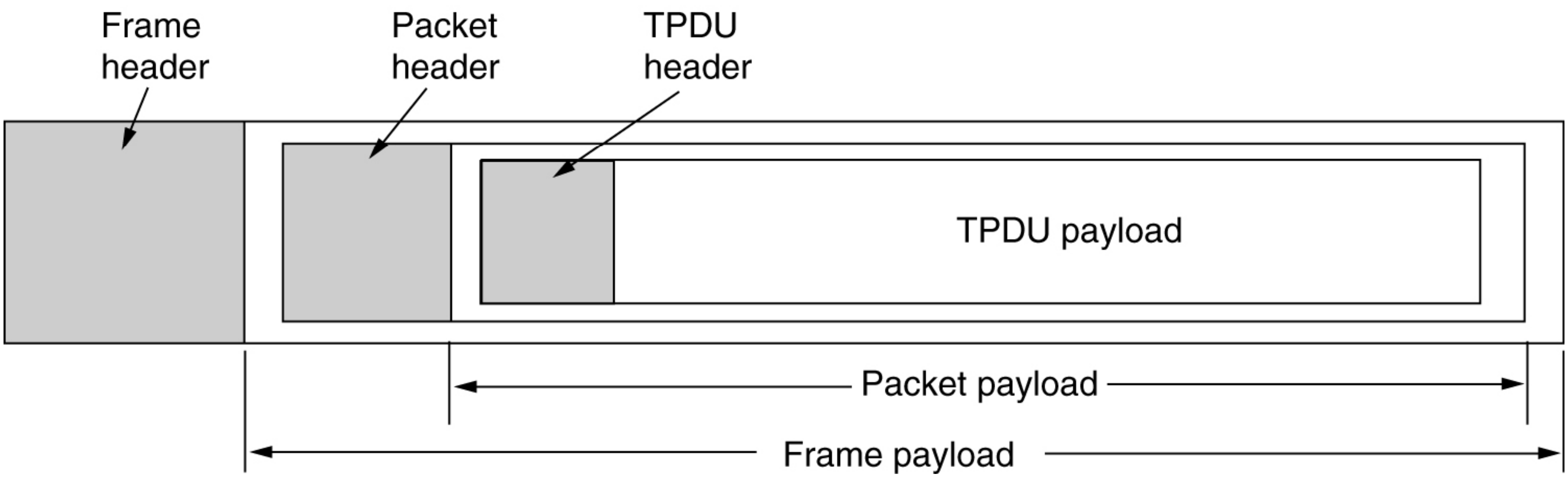
Transport Service Primitives

Primitive	Packet sent	Meaning
LISTEN	(none)	Block until some process tries to connect
CONNECT	CONNECTION REQ.	Actively attempt to establish a connection
SEND	DATA	Send information
RECEIVE	(none)	Block until a DATA packet arrives
DISCONNECT	DISCONNECTION REQ.	This side wants to release the connection

The primitives for a simple transport service.

Transport Service Primitives (2)





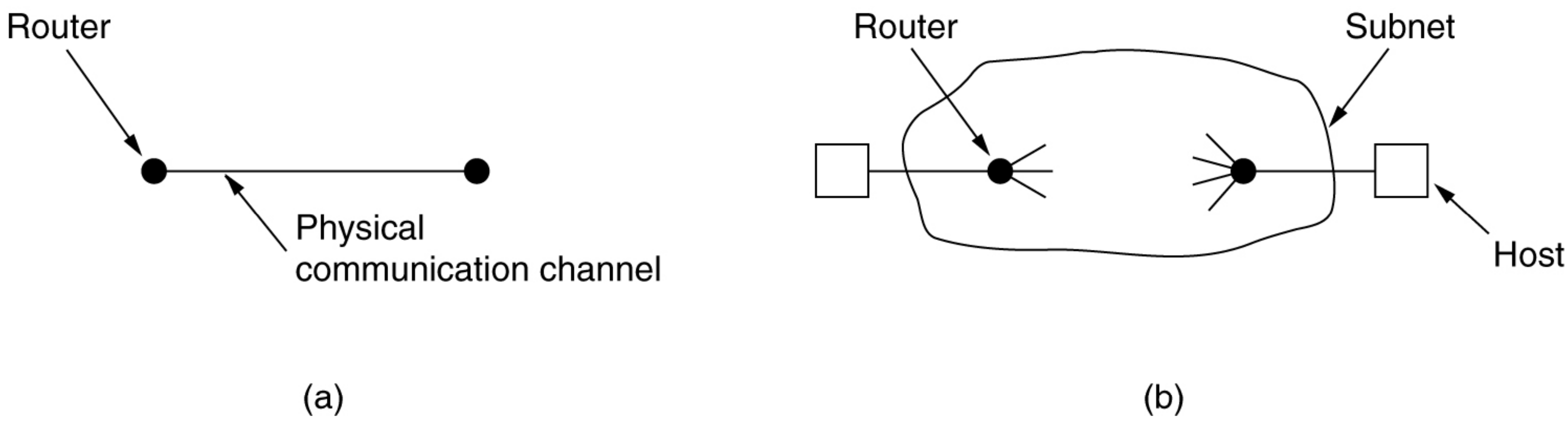
The nesting of TPDUs, packets, and frames.

Elements of Transport Protocols

- The environment in which Transport Operates

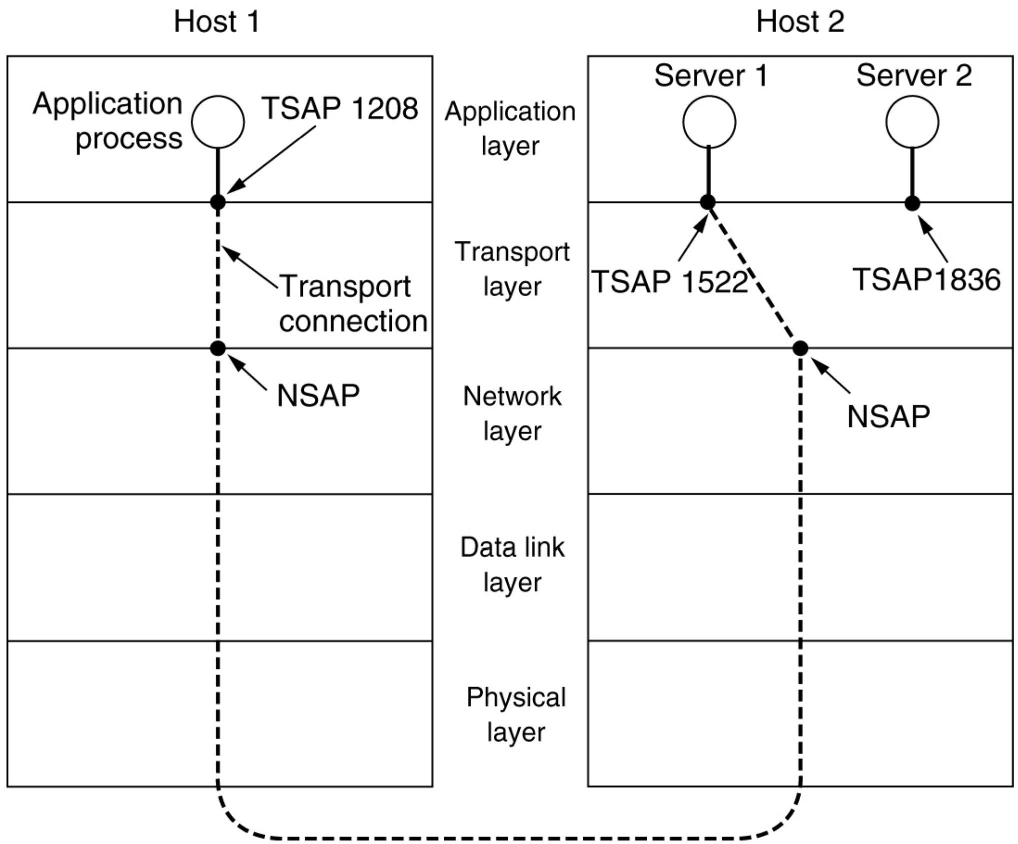
- Addressing
- Connection Establishment
- Connection Release
- Flow Control and Buffering
- Multiplexing
- Crash Recovery

Transport Protocol



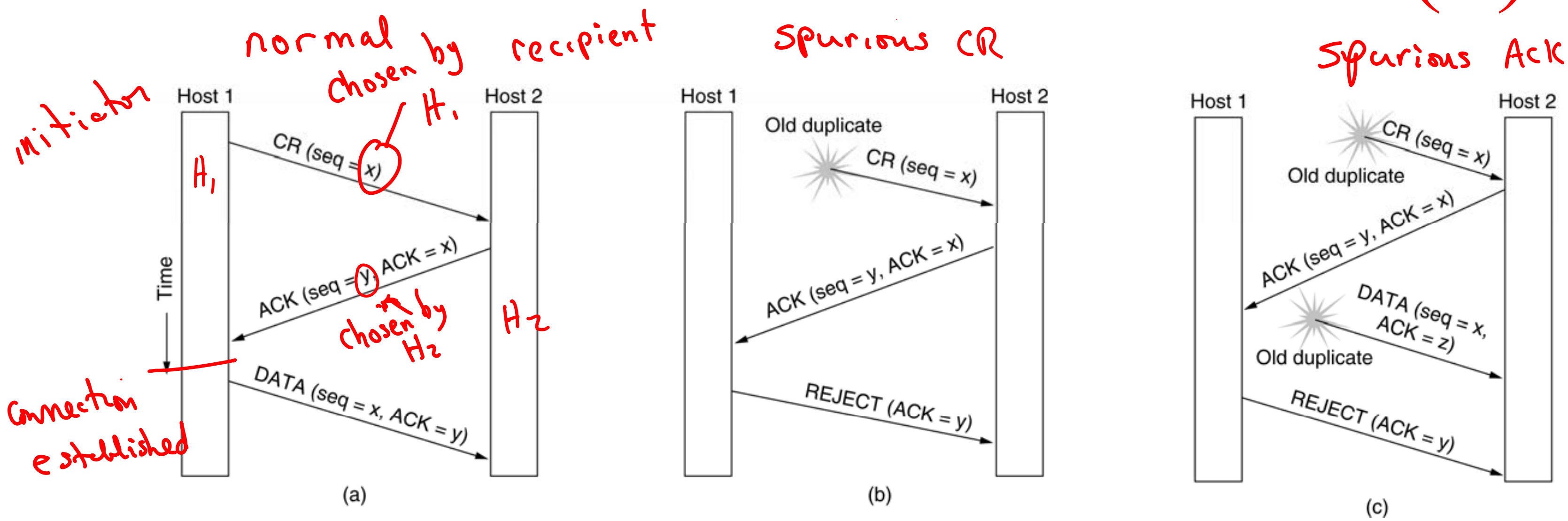
(a) Environment of the data link layer.
(b) Environment of the transport layer.

Addressing



TSAPs, NSAPs and transport connections.

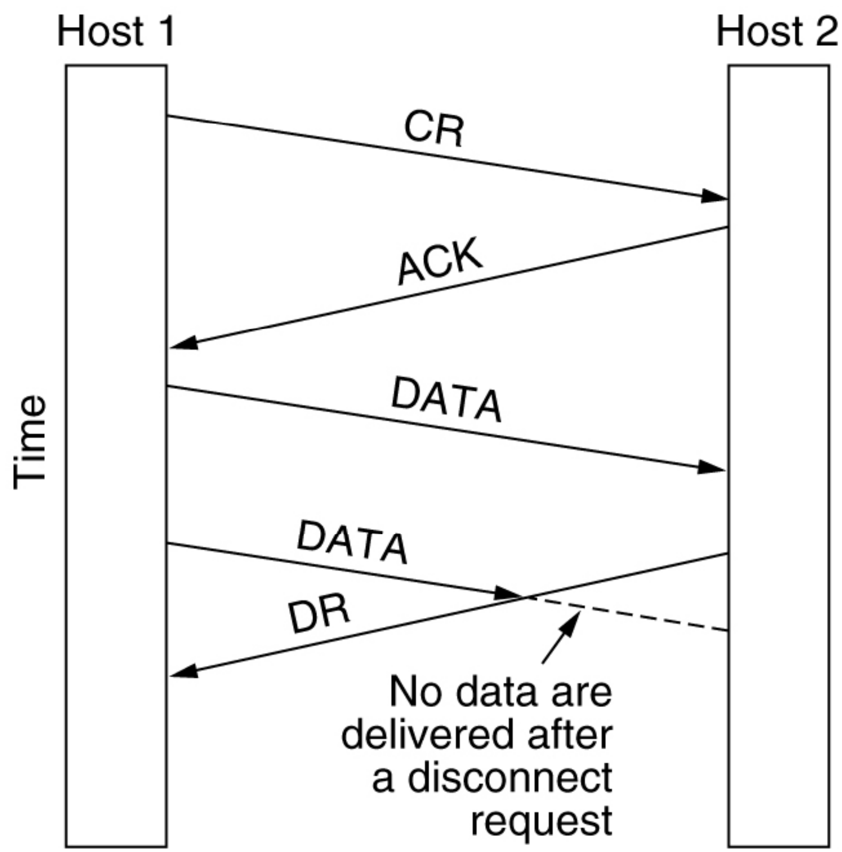
Connection Establishment (3)



Three protocol scenarios for establishing a connection using a three-way handshake. CR denotes CONNECTION REQUEST.
(a) Normal operation,
(b) Old CONNECTION REQUEST appearing out of nowhere.
(c) Duplicate CONNECTION REQUEST and duplicate ACK.

Connection Release

Start here 11/5

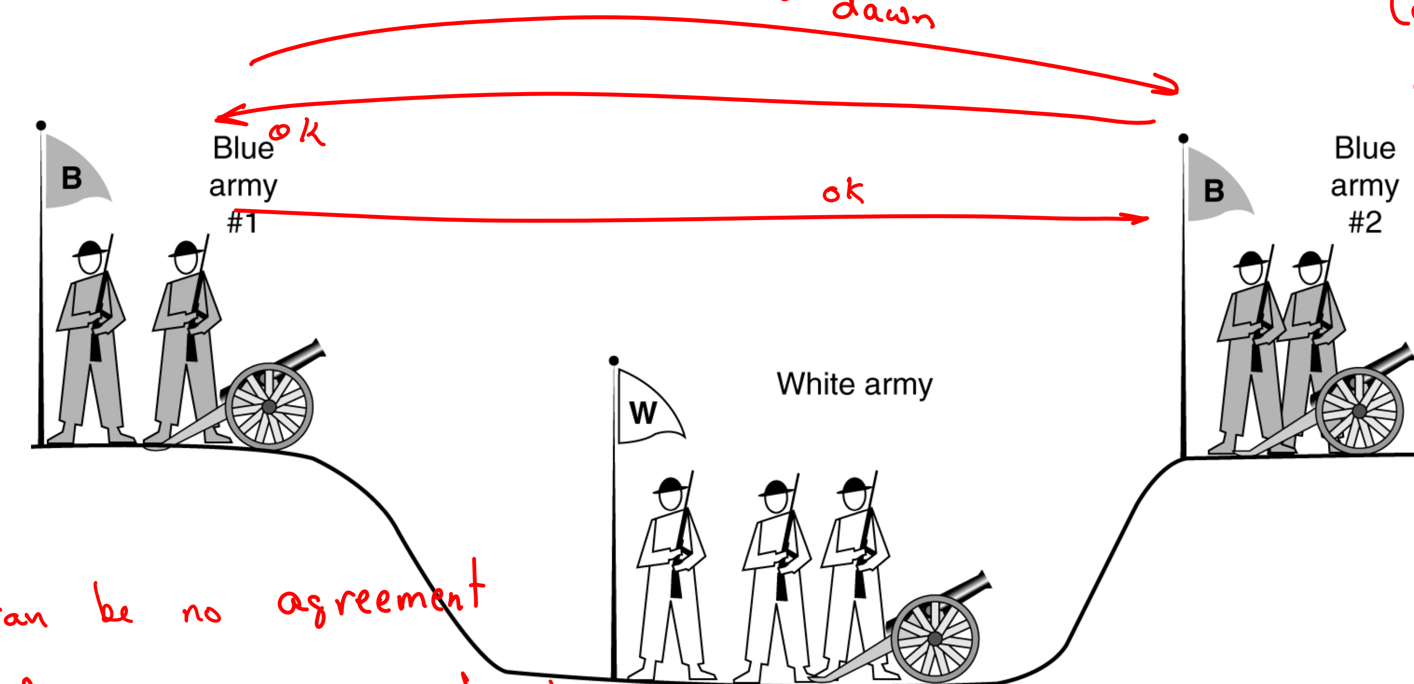


Abrupt disconnection with loss of data.

~~Connection Release (2)~~

1. Connection Release.

2. Distributed Commit of database transactions.



There can be no agreement protocol if messages may be lost.

Proof idea:

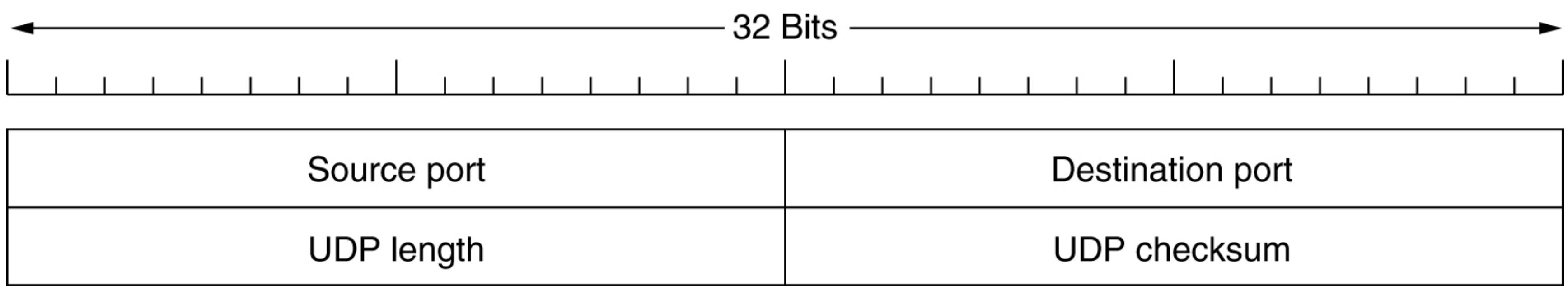
message is essential if its receipt affects a party's behavior.

The two-army problem.

The Internet Transport Protocols: UDP

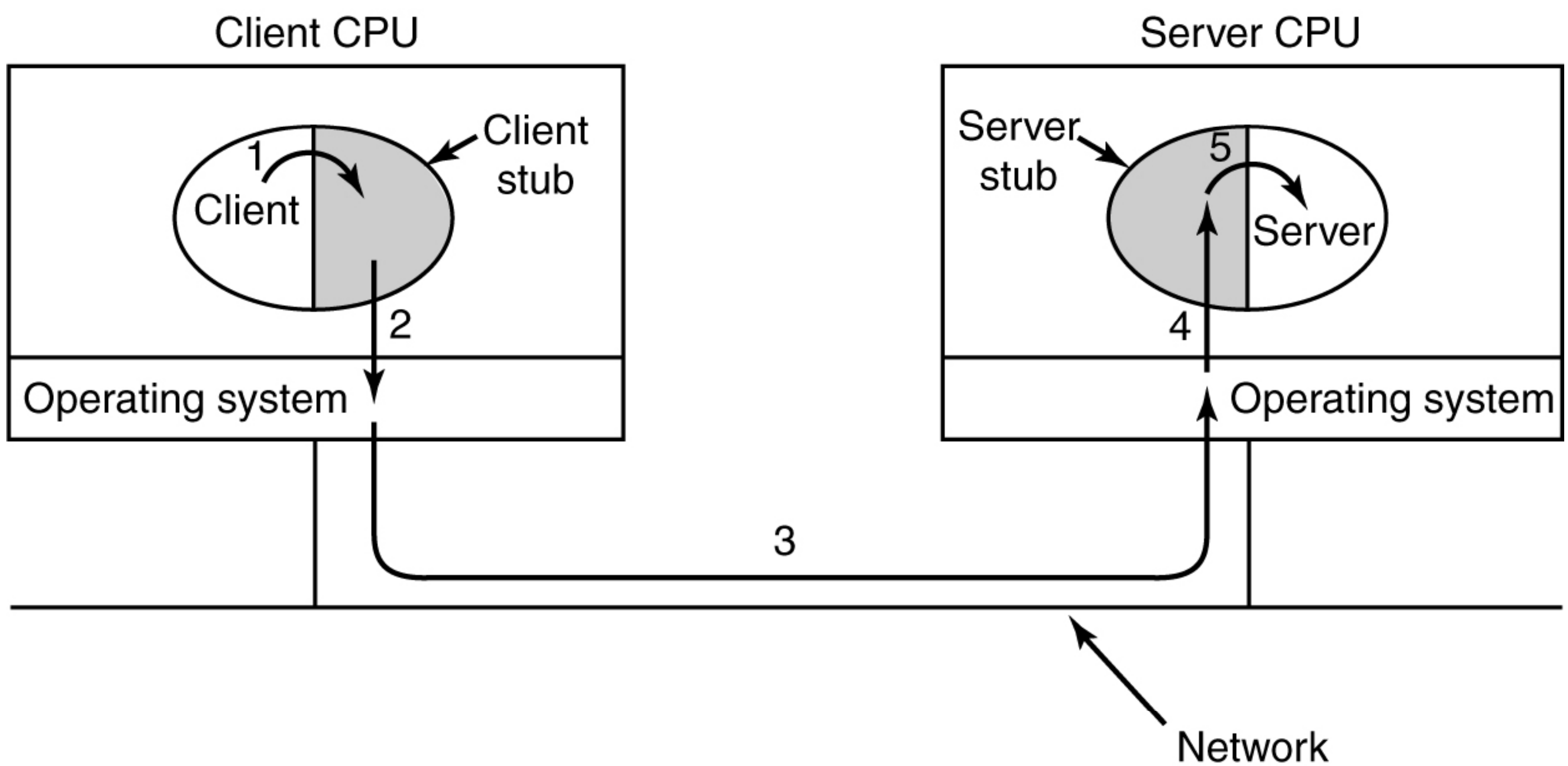
- Introduction to UDP
- Remote Procedure Call
- The Real-Time Transport Protocol

Introduction to UDP



The UDP header.

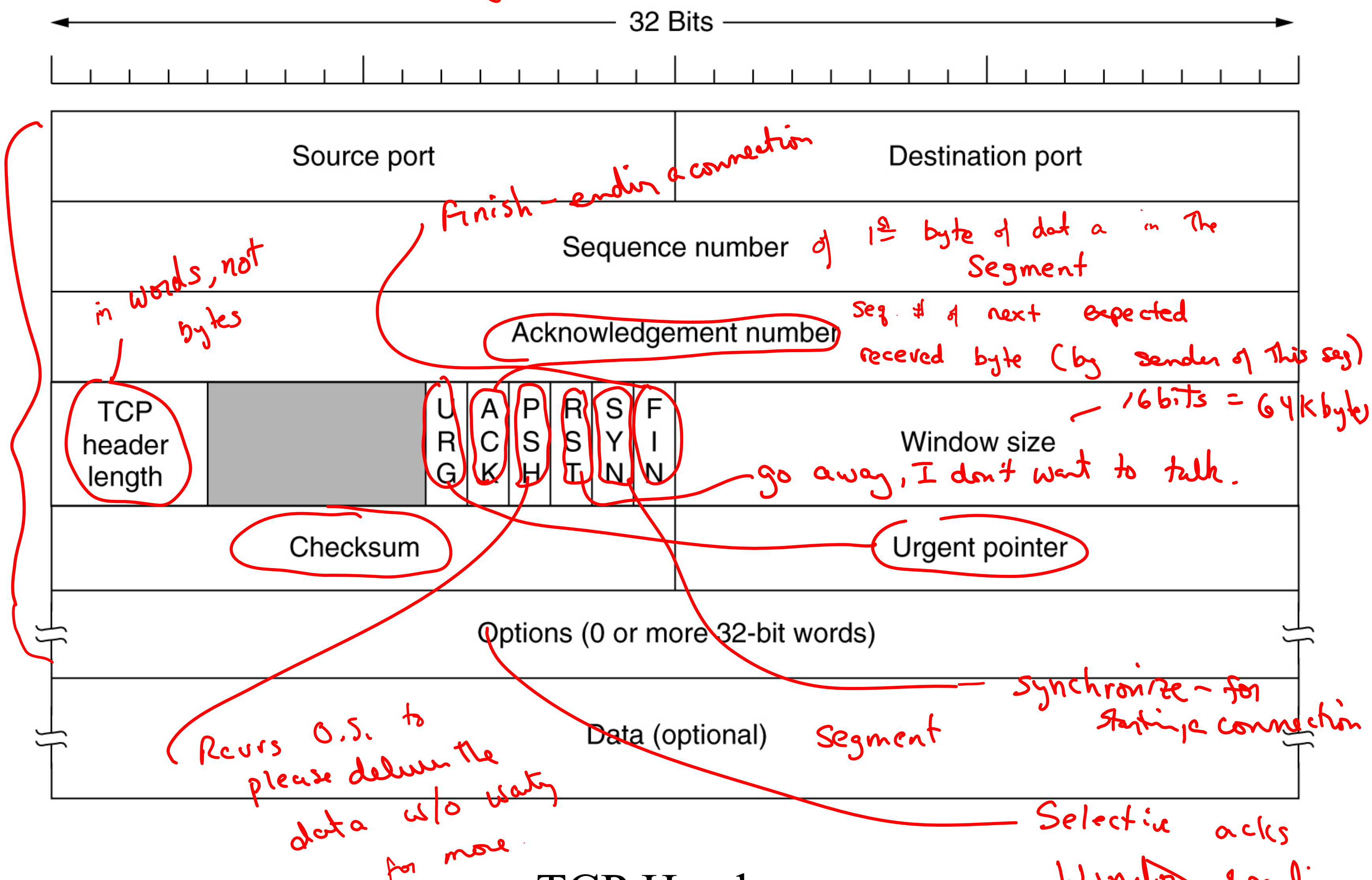
Remote Procedure Call



Steps in making a remote procedure call. The stubs are shaded.

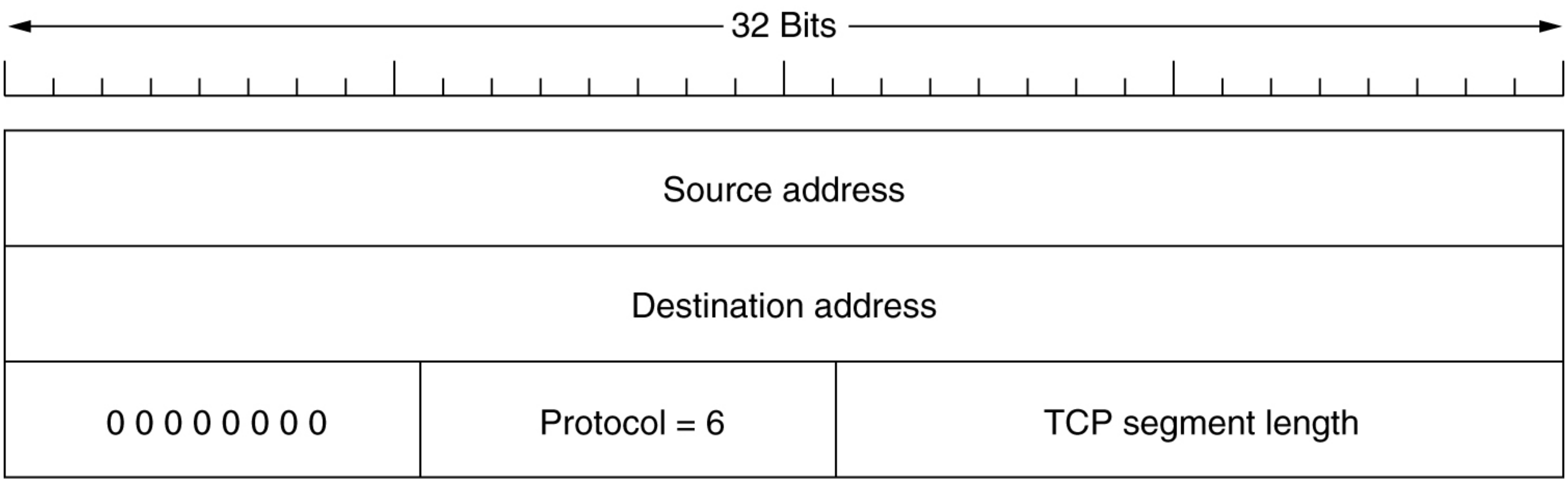
The TCP Segment Header

TCP connection identified by (<src ip, src port>, <dest ip, dest port>)



TCP Header.

The TCP Segment Header (2)

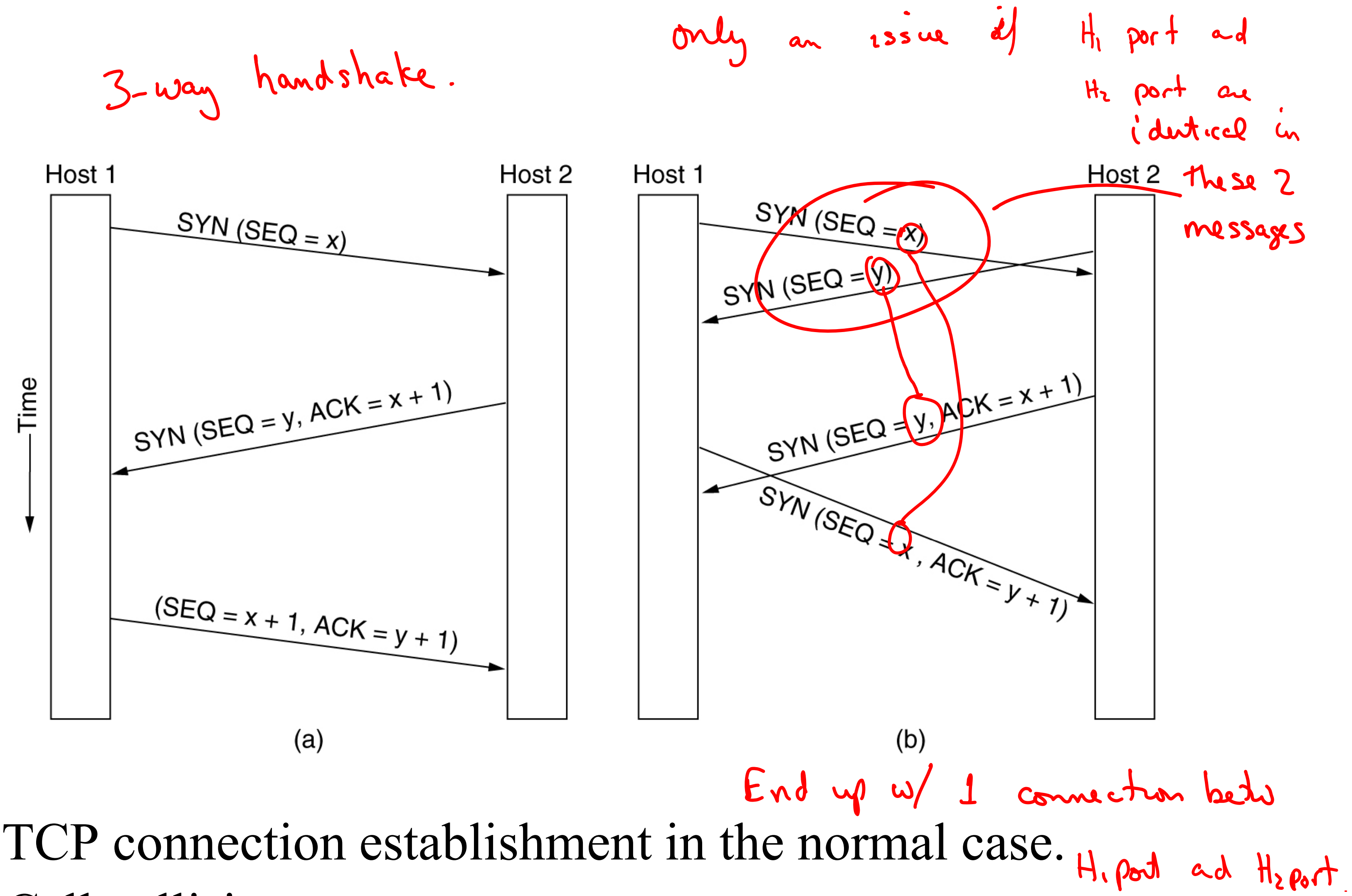


violates layering

IP

The pseudoheader included in the TCP checksum.

TCP Connection Establishment



(a) TCP connection establishment in the normal case.
(b) Call collision.

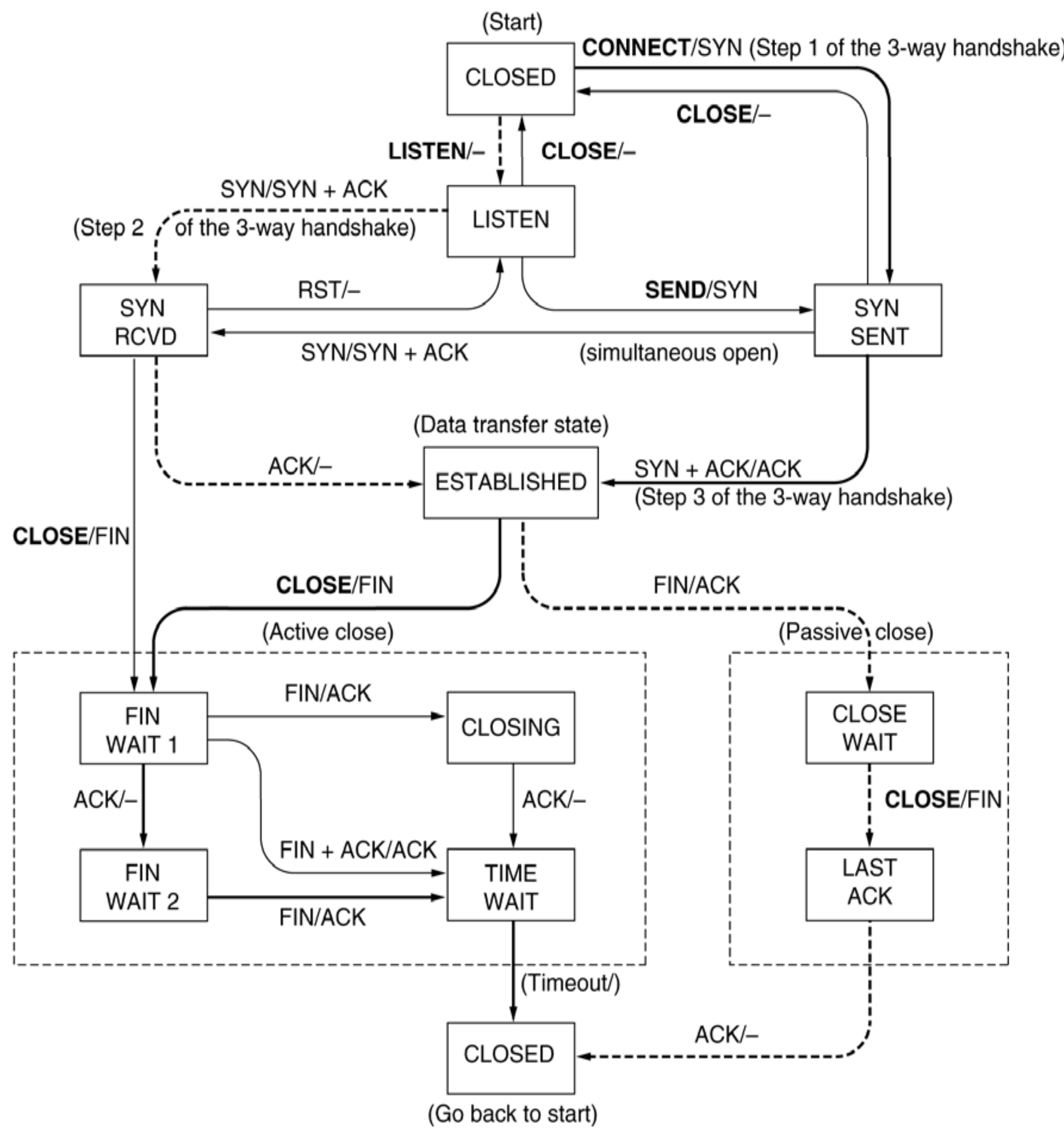
TCP Connection Management Modeling

State	Description
CLOSED	No connection is active or pending
LISTEN	The server is waiting for an incoming call
SYN RCVD	A connection request has arrived; wait for ACK
SYN SENT	The application has started to open a connection
ESTABLISHED	The normal data transfer state
FIN WAIT 1	The application has said it is finished
FIN WAIT 2	The other side has agreed to release
TIMED WAIT	Wait for all packets to die off
CLOSING	Both sides have tried to close simultaneously
CLOSE WAIT	The other side has initiated a release
LAST ACK	Wait for all packets to die off

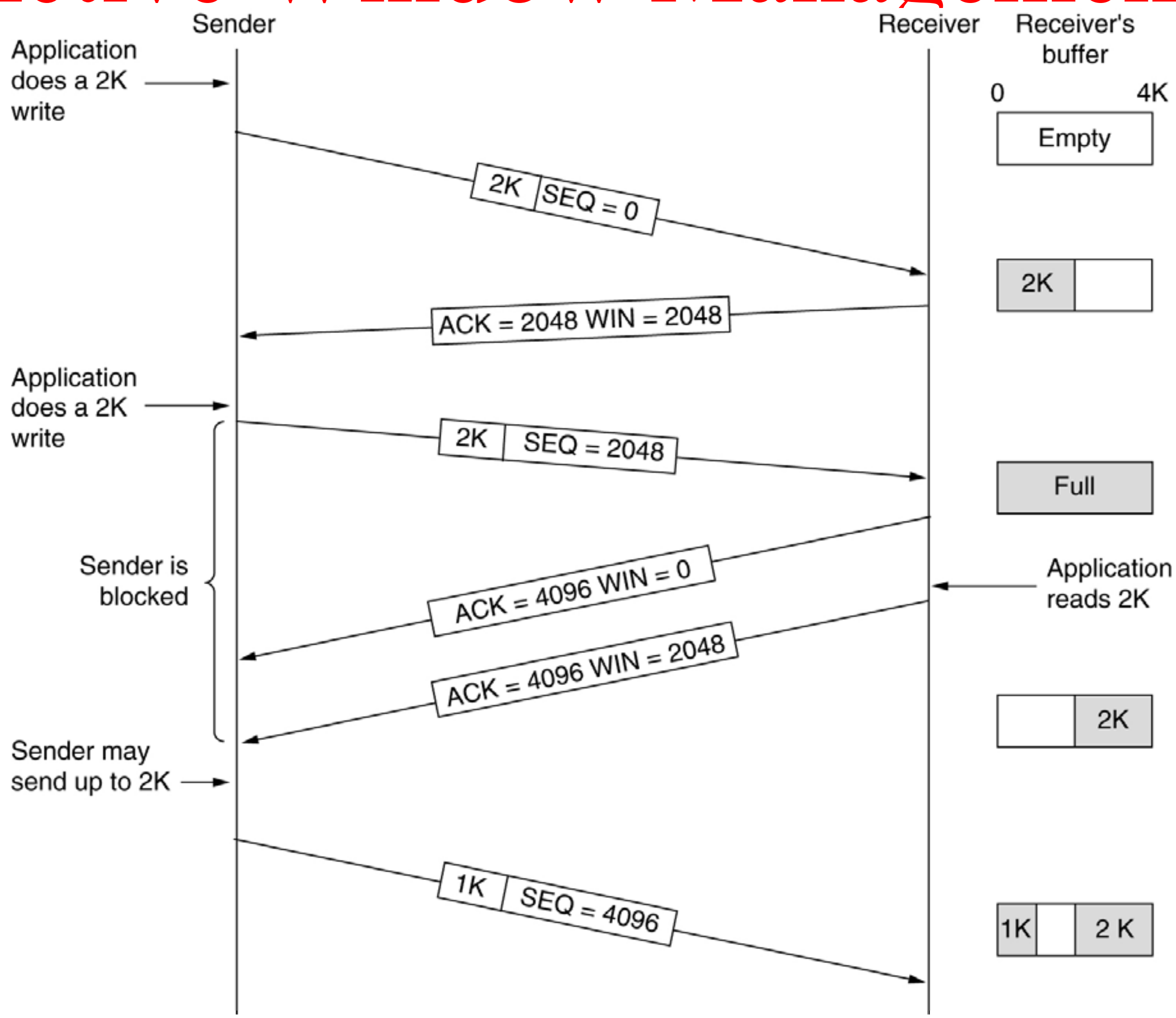
The states used in the TCP connection management finite state machine.

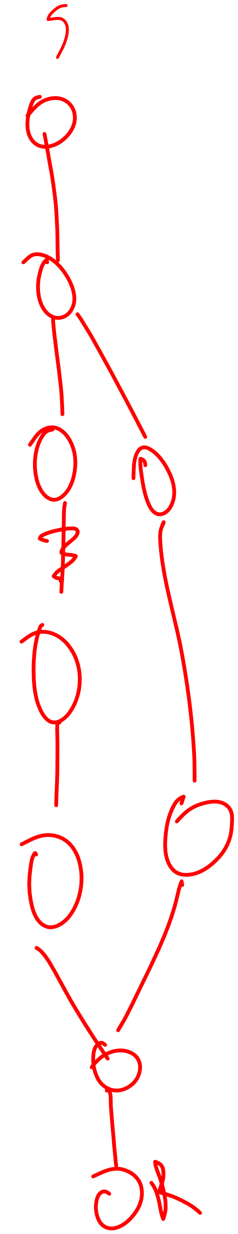
TCP Connection Management Modeling (2)

TCP connection management finite state machine. The heavy solid line is the normal path for a client. The heavy dashed line is the normal path for a server. The light lines are unusual events. Each transition is labeled by the event causing it and the action resulting from it, separated by a slash.



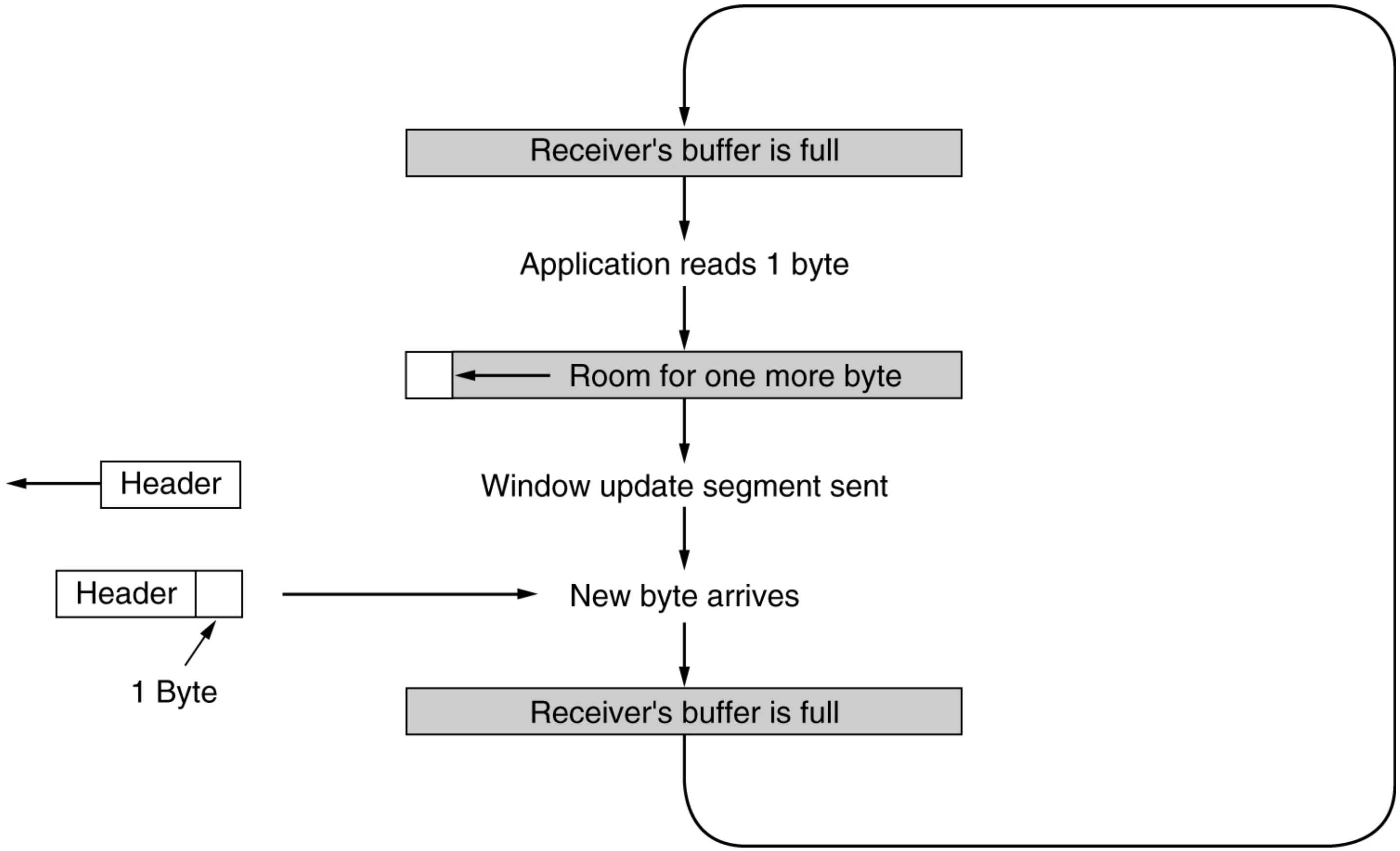
TCP Transmission Policy
Active Window Management





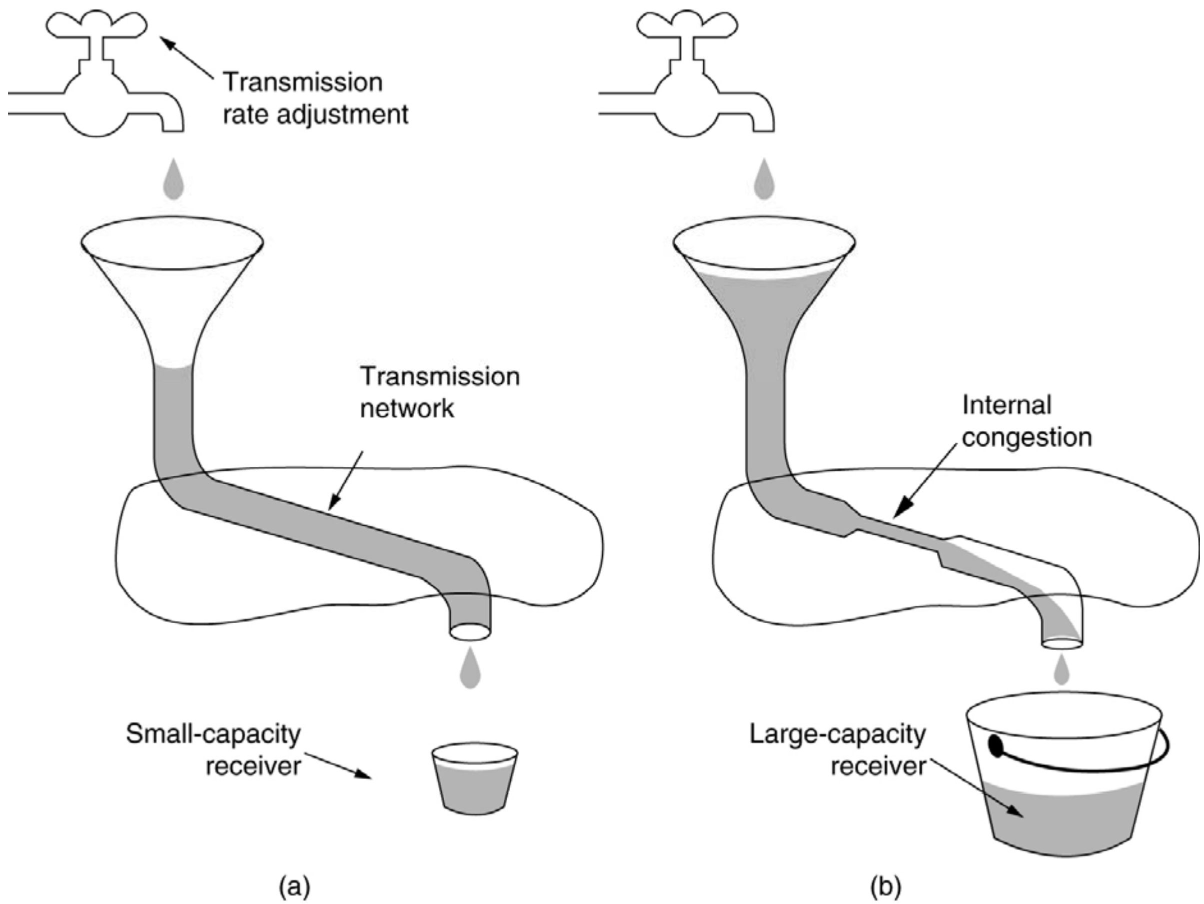
Window management in TCP.

TCP Transmission Policy (2)



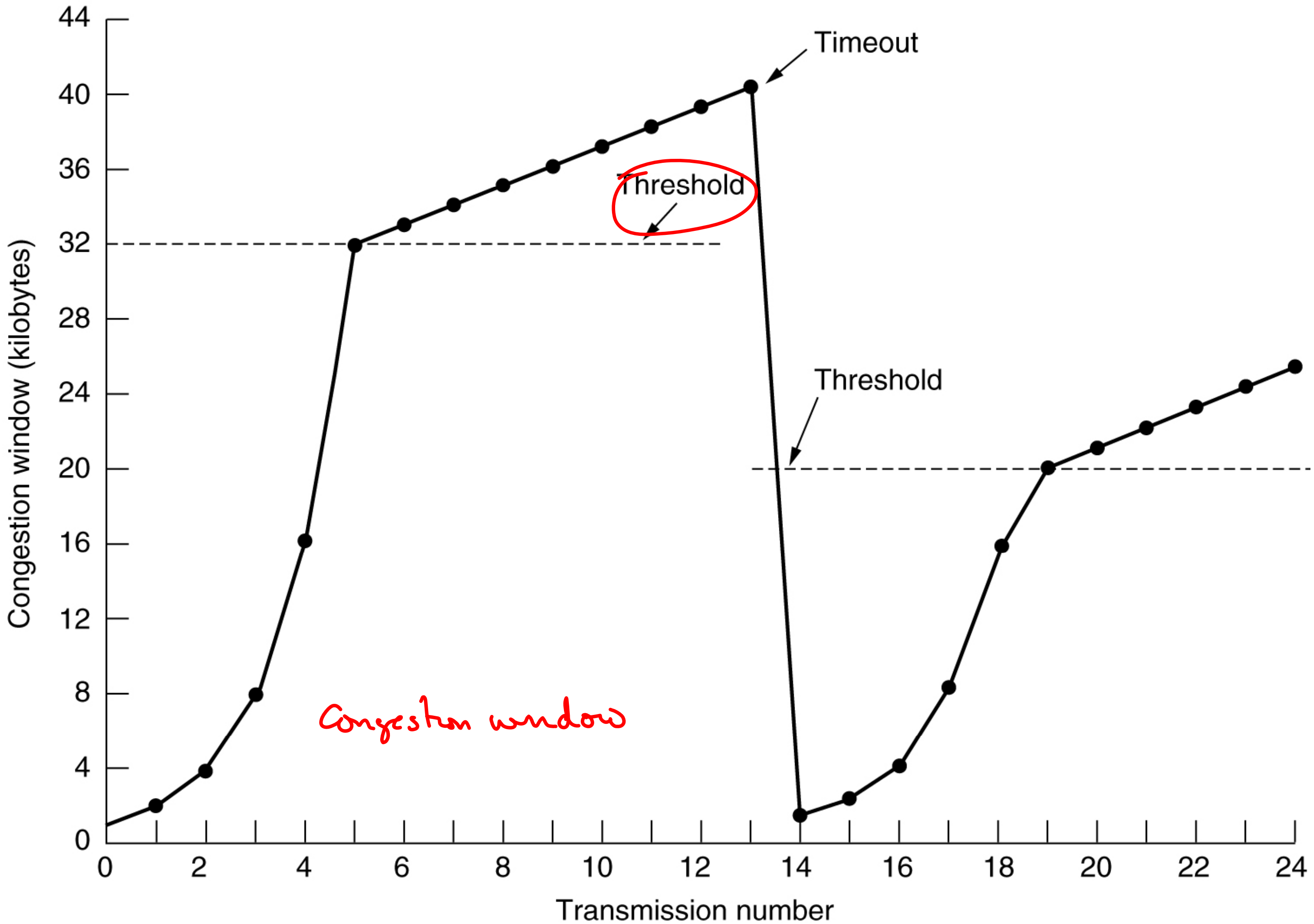
Solving the silly window syndrome
Nagle's algorithm for transmission

TCP Congestion Control



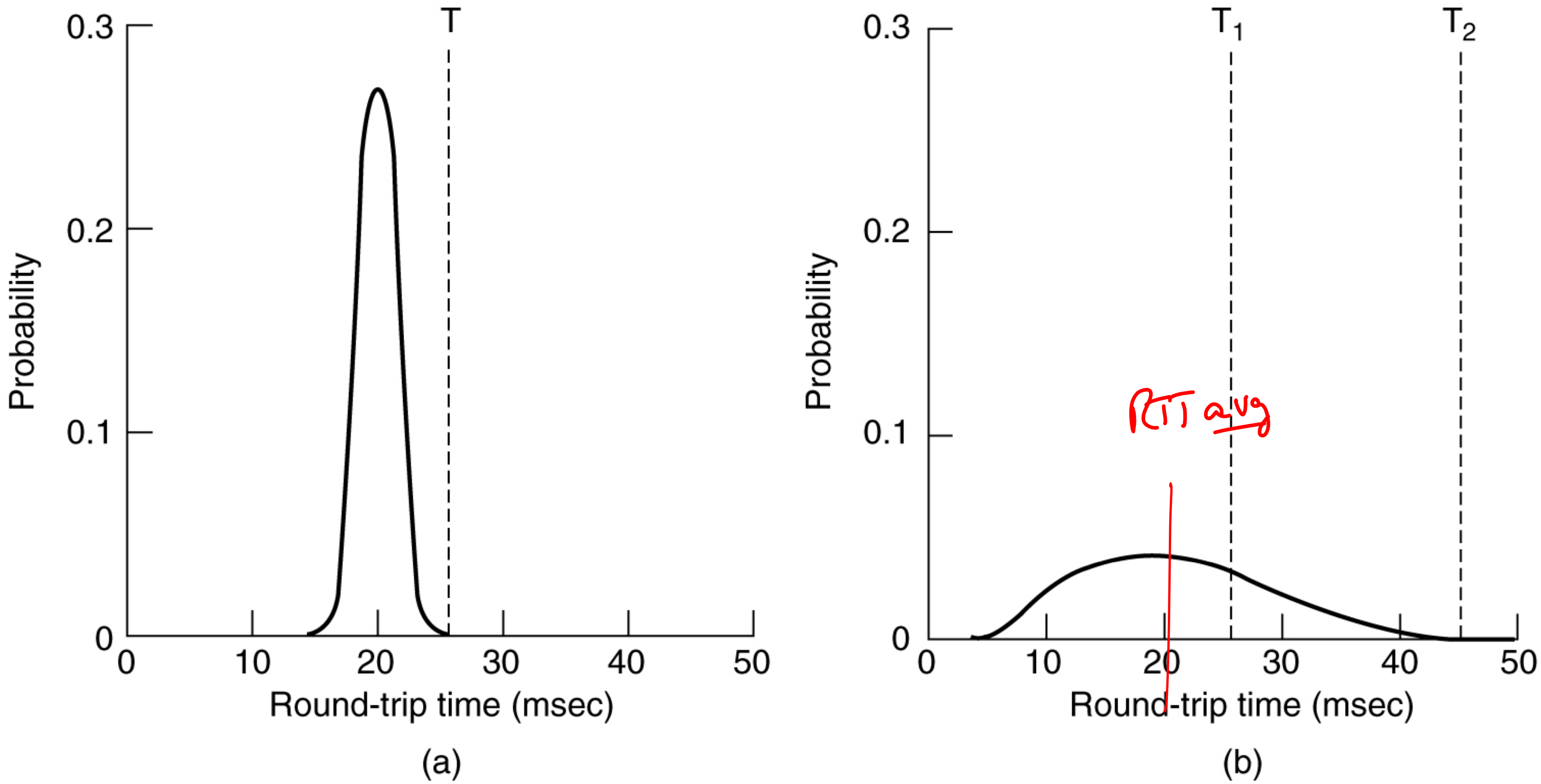
(a) A fast network feeding a low capacity receiver.
(b) A slow network feeding a high-capacity receiver.
Outstanding data must be limited by both the network AND the receiver

TCP Congestion Control (2)



An example of the Internet congestion algorithm
TCP Slow-start (Slow wrt to what since it is exponential!?)

TCP Timer Management



(a) Probability density of ACK arrival times in the data link layer.

(b) Probability density of ACK arrival times for TCP.

Round-trip and Variance Estimation

$0 \le a \le 1.0$ $a_{typical} = 7/8$

(a) RTT = a(RTT) + (1-a)M (exponential smoothing)

(b) Dev = a(Dev) + (1-a) |RTT-M|

(c) Retransmission Timeout = RTT + 4*Dev

  (a) Used to use RTO = 2*RTT

(d) What about retransmitted segments?

  (a) Ignore them in the RTT calculation

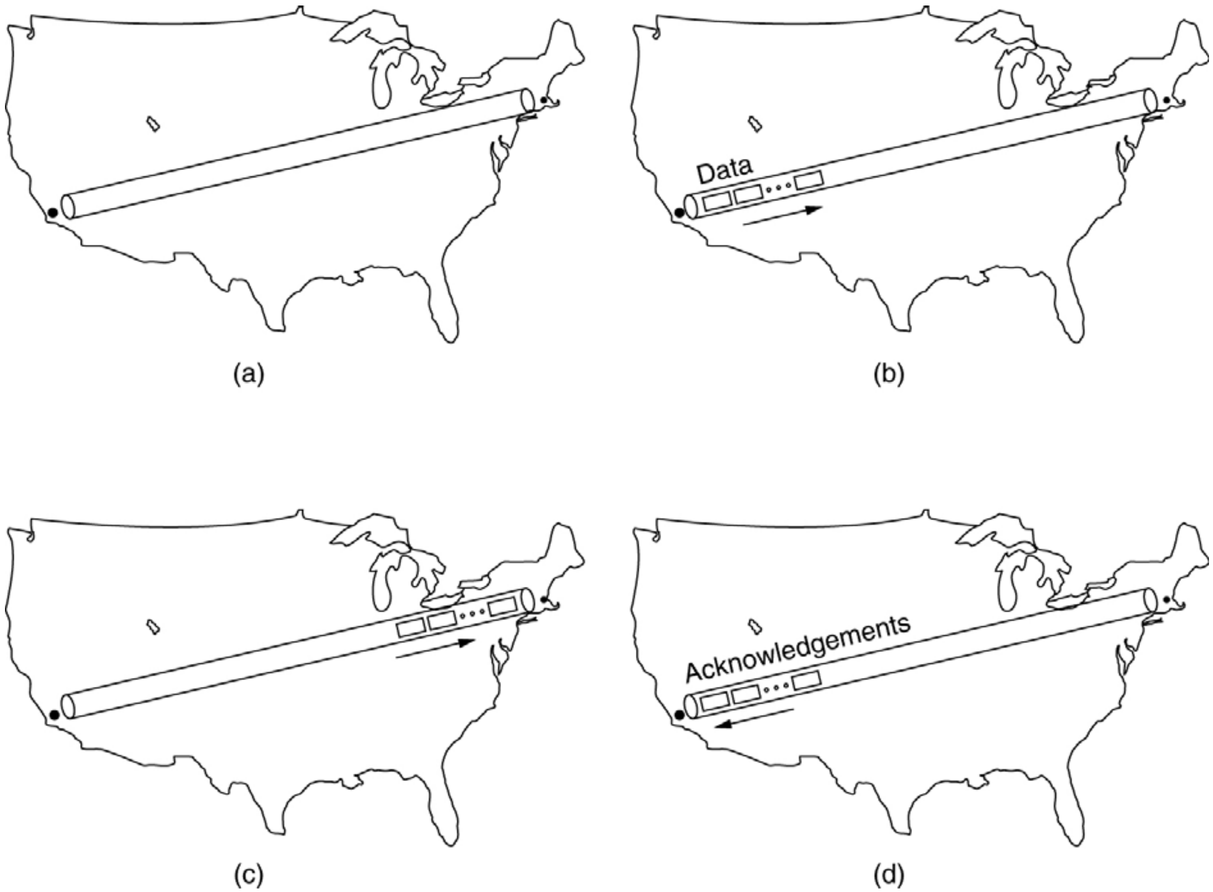
Wireless TCP

- Missing acks can mean two things with opposite required responses
  - In wired networks: congestion; response: slow down!
  - In wireless networks: dropped packets due to noise; try again, real soon!
- Approaches for wireless
  - Split TCP connection in two (not common)
  - Acks and retransmission at the link layer

Performance Issues

- Performance Problems in Computer Networks
- Network Performance Measurement
- System Design for Better Performance
- Fast TPDU Processing
- Protocols for Gigabit Networks

Performance Problems in Computer Networks



The state of transmitting one megabit from San Diego to Boston
(a) At t = 0, (b) After 500 µsec, (c) After 20 msec, (d) after 40 msec.

Network Performance Measurement

The basic loop for improving network performance.

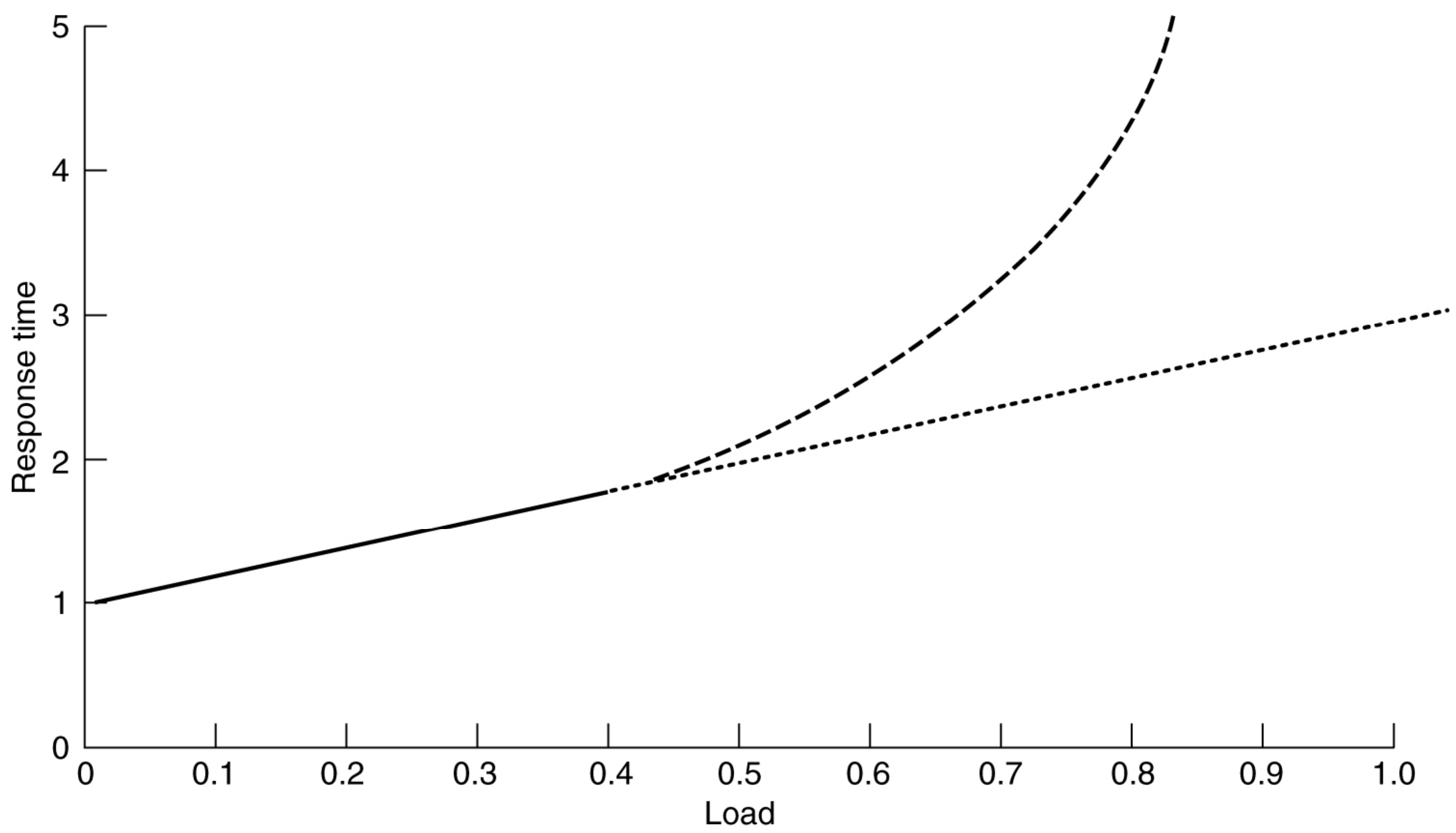
1. Measure relevant network parameters, performance.
2. Try to understand what is going on.
3. Change one parameter.

System Design for Better Performance

Rules:

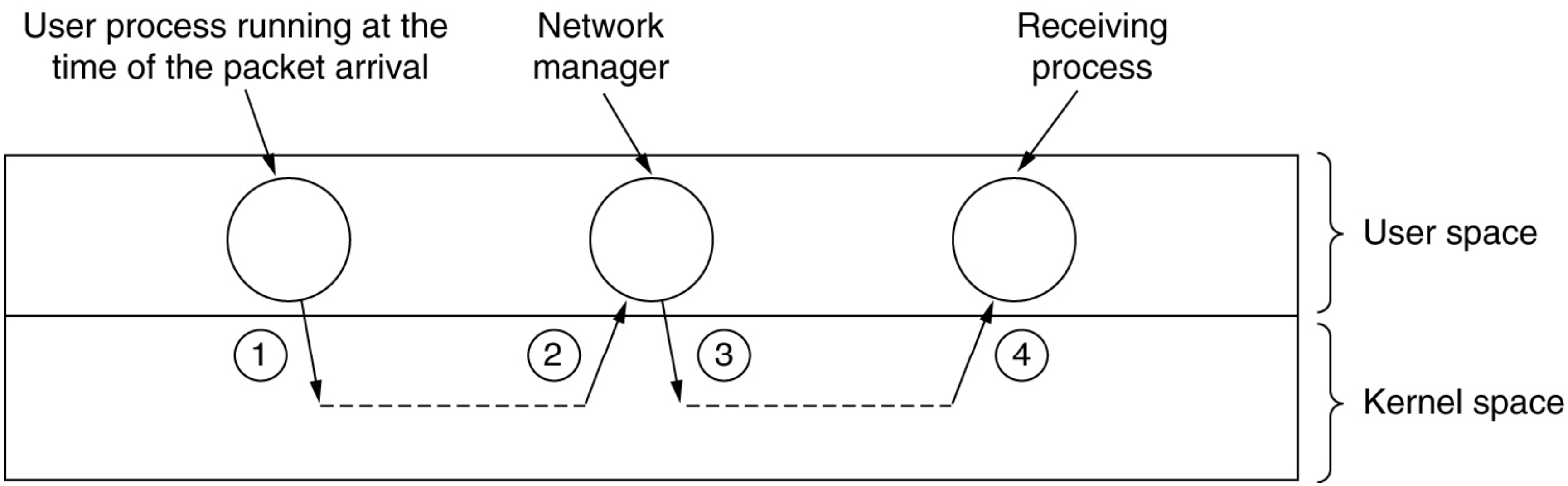
1. CPU speed is more important than network speed.
2. Reduce packet count to reduce software overhead.
3. Minimize context switches.
4. Minimize copying.
5. You can buy more bandwidth but not lower delay.
6. Avoiding congestion is better than recovering from it.
7. Avoid timeouts.

System Design for Better Performance (2)



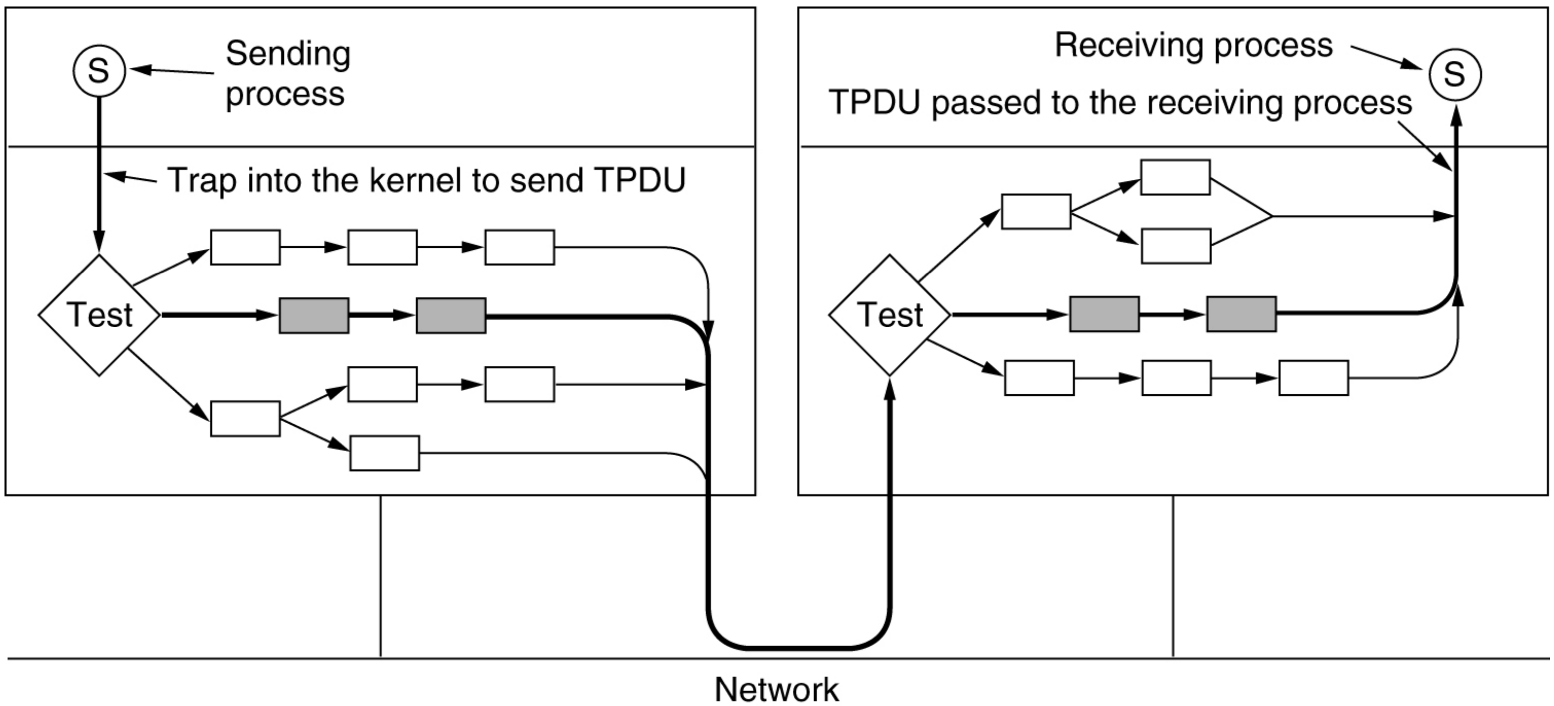
Response as a function of load.

System Design for Better Performance (3)



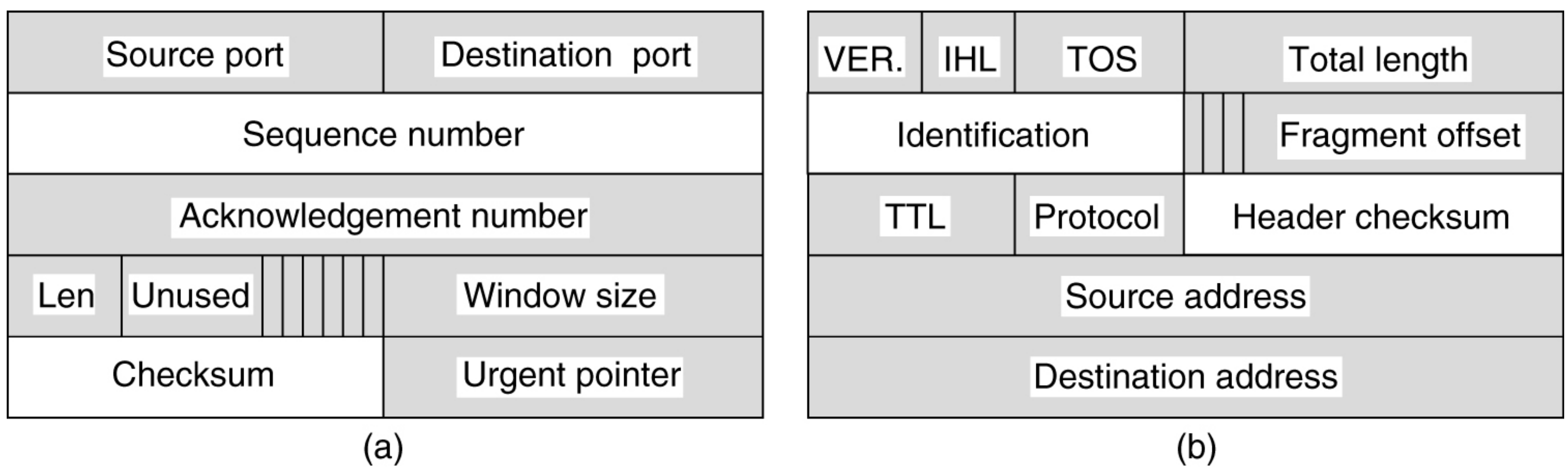
Four context switches to handle one packet with a user-space network manager.

Fast TPDU Processing



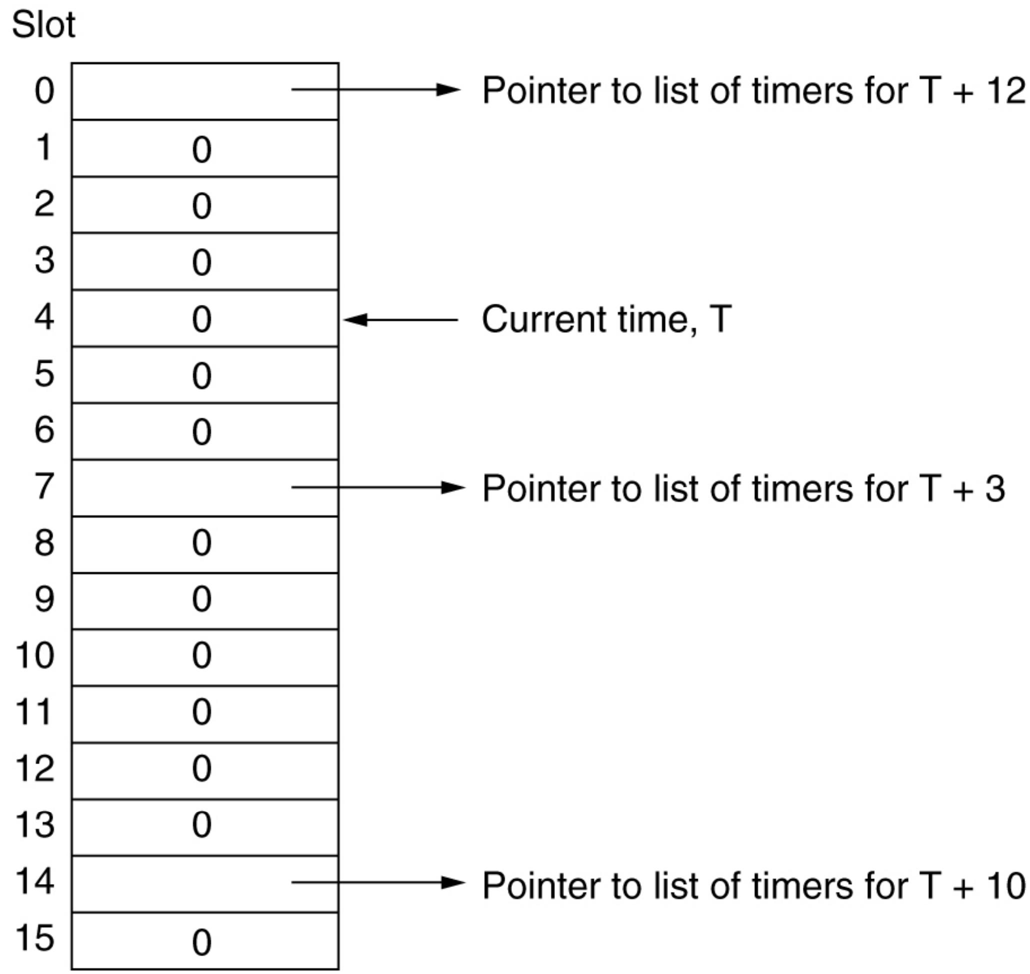
The fast path from sender to receiver is shown with a heavy line.
The processing steps on this path are shaded.

Fast TPDU Processing (2)



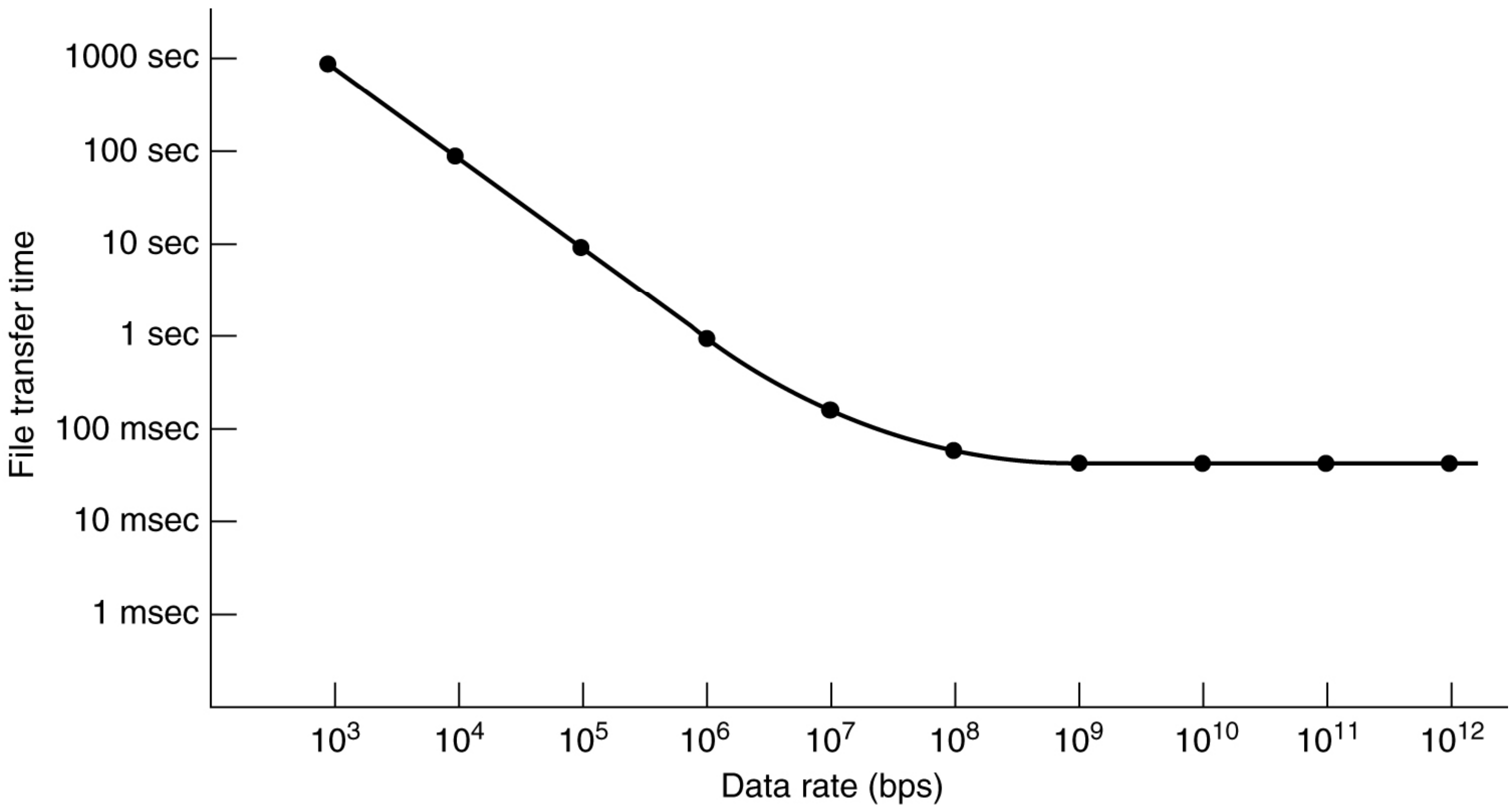
(a) TCP header. (b) IP header. In both cases, the shaded fields are taken from the prototype without change.

Fast TPDU Processing (3)



A timing wheel.

Protocols for Gigabit Networks



Time to transfer and acknowledge a 1-megabit file over a 4000-km line.