Exploring Networks

Midterm Review

Dominik Kovacs

October 1, 2021

Midterm: Thursday 7th during class time on canvas

Study Guide

1. What is a host? Are there more hosts on the internet or non-host network nodes? How do you know?
2. What is a protocol?
3. Bandwidth – what does it define, why is it something of a misnomer?
4. Compare and contrast a statistical multiplexed (packet switched) vs. a statically multiplexed network. How are they different? What are the advantages and disadvantages of either? What are two classes of static multiplexing?
5. How long does it take a packet of length 1000 bytes to propagate over a link distance of 2500 km, propagation speed (s) of 2.5x10^8m/s, and transmission rate of 2mbps?
6. What is an ISP? What different types of ISPs exist?
7. What are Internet Exchange Points (IXPs) and Points of Presences (PoPs)?
8. How does the Traceroute command work?
9. What are two advantages of having a layered network? What about two disadvantages? What are the layers in the Internet model? In the OSI 7 layer model?

Study Guide

10. Suppose users share a 2Mbps link. Also suppose each user transmits continuously at 1Mbps when transmitting, but each user transmits only 20% of the time.
    a. When circuit switching is used, how many users can be supported?
    b. When packet switching is used, how many users can be supported?
    c. What is the probability a given user is transmitting at any time?
11. What are the advantages of a P2P model over a client server model? Disadvantages?
12. What is a socket?
13. What does TCP provide that UDP does not? Which is more impactful on the network?
14. In general, what does an HTTP message between 200 and 299 mean?
15. What are the advantages and disadvantages of persistent vs. non-persistent HTTP? Which would I want to use for a large, one-time file transfer?
16. What is the difference between IMAP and POP?
17. What roles may a Super Node play in a P2P network?
18. Give a high level overview of the operation of SMTP from client and server perspectives.

Study Guide

19. Give a high level overview of how DNS works. Make sure to include the concept of a Root and TLD domain server. Why do root servers exist?
20. Explain, at a high level, the code flow in C or Python of a socket connection and sending/receiving data. What types of sockets can be created and used?
21. Explain the differences between stop-and-wait, go-back-N, and selective repeat reliable transfer mechanisms.
22. What is the difference between flow control and congestion control? Why do both exist?
23. Understand how a TCP 3-way handshake works and what initialization is performed.
24. Suppose you have a 1Gbps link and two hosts separated on this link at 100ms RTT. How much data (in Bytes) will need to be "in-flight" to achieve full utilization using TCP as a transport?
25. Explain TCP's congestion control operation. What are AIMD, slow start, congestion avoidance? How does a sender respond by adjusting its congestion window and slow start threshold based on inferred loss events?
26. Explain at a high level the operation of fast retransmit and fast recovery in TCP. Why are they useful compared to the alternative?

Chapter 1: Computer Networks and the Internet

Protocol

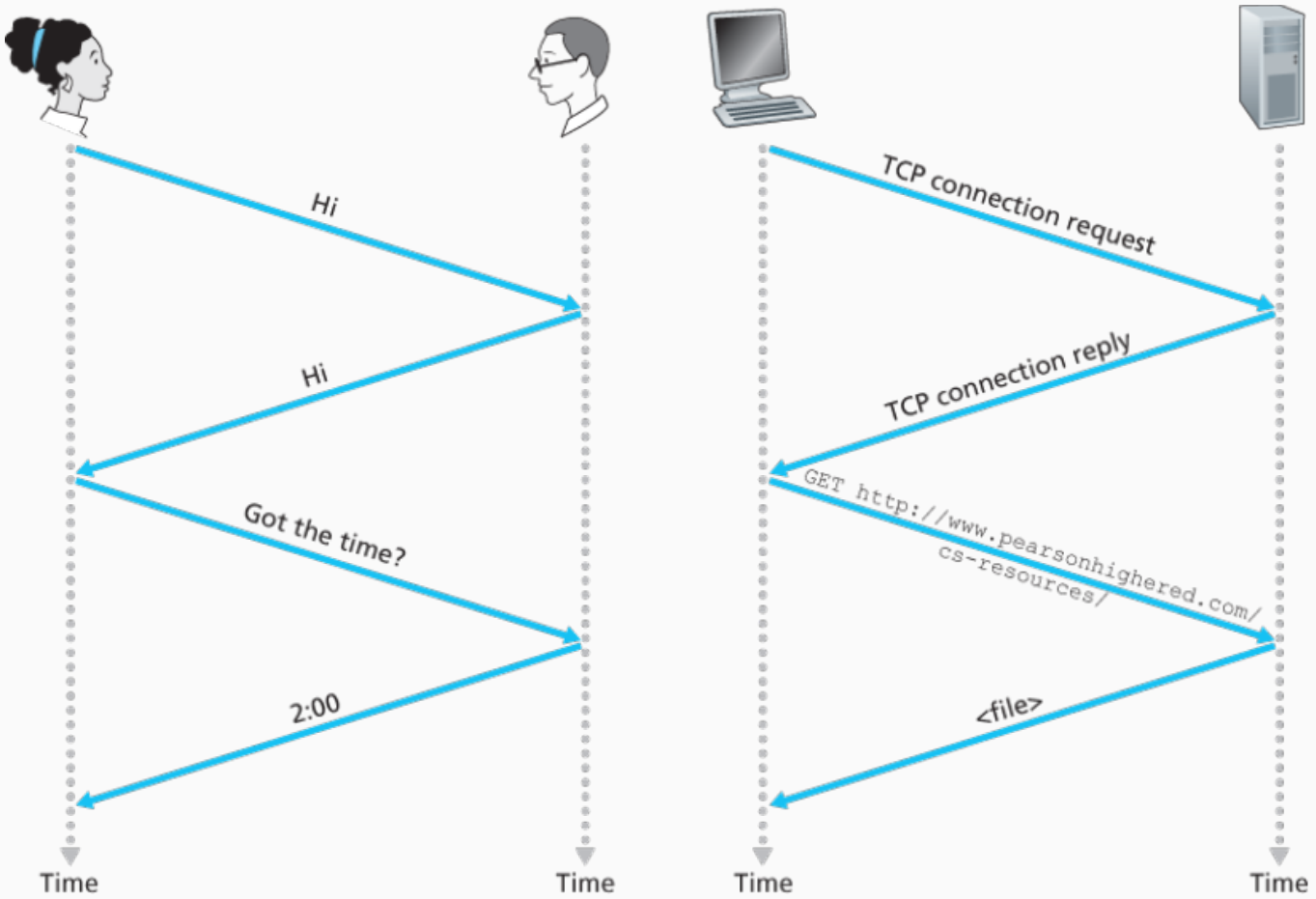


Figure 1.2 A human protocol and a computer network protocol

Set of rules (order, format, actions) for transmitting data

Packet Switching vs. Circuit-Switching

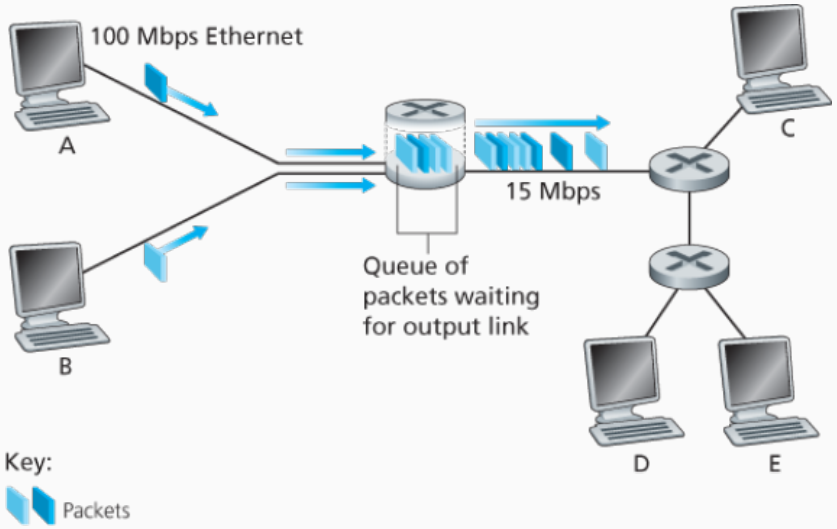


Figure 1.12 Packet switching

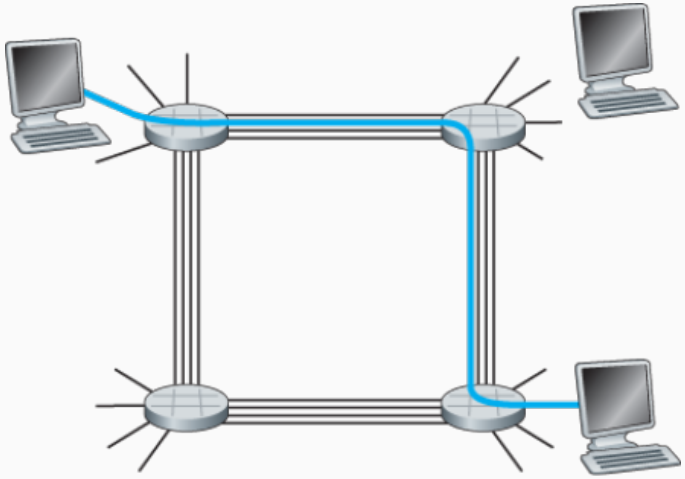


Figure 1.13 A simple circuit-switched network

	Packet Switching	Circuit Switching
Setup	**None**	Required
Path	**Dynamic**	Static
Bandwidth	Variable	**Fixed**
Utilization	**Optimal**	May be less

End-to-end Packet Delay

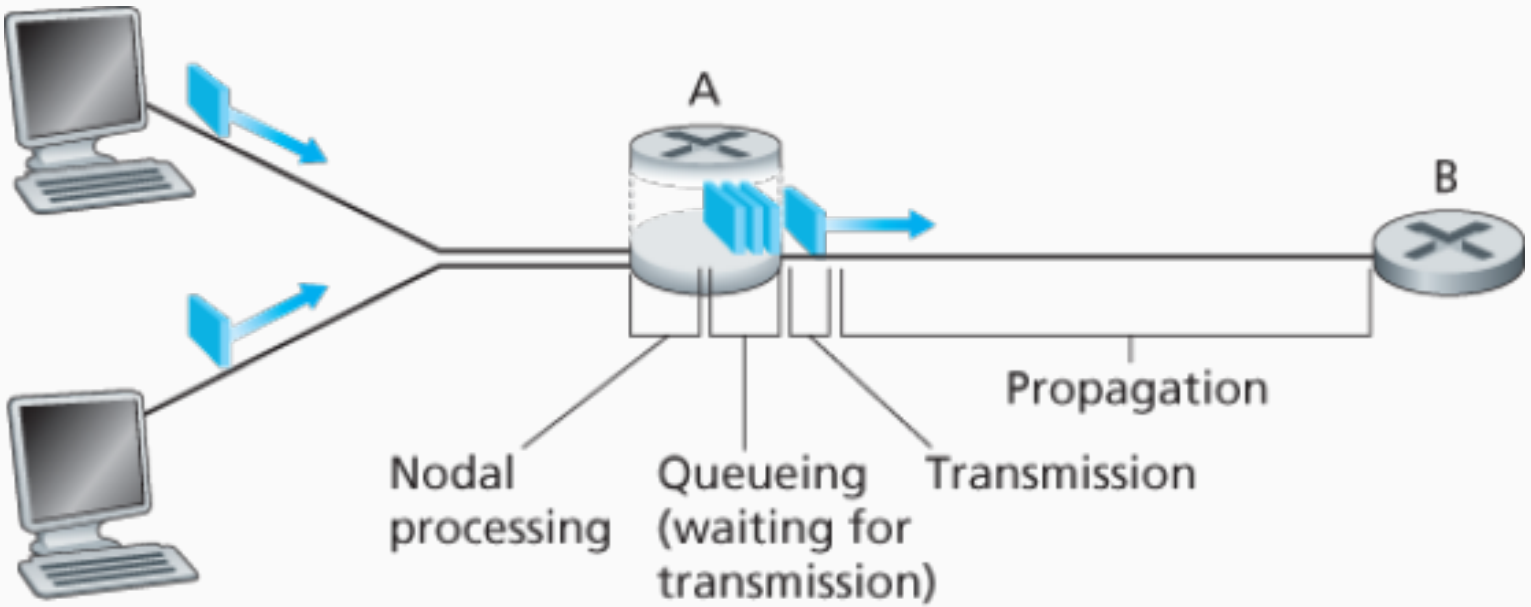


Figure 1.16 The nodal delay at router A

OSI Model

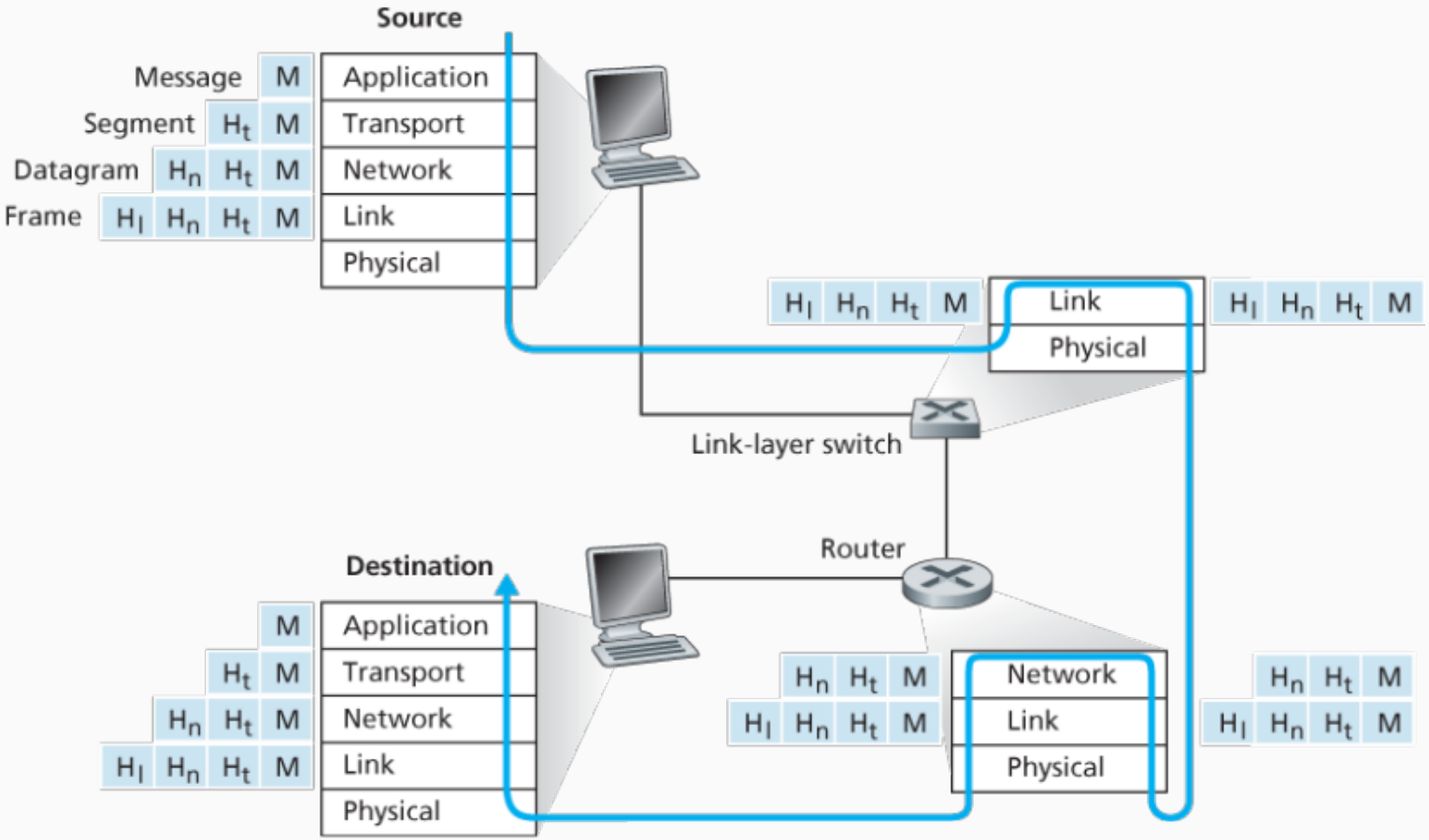


Figure 1.24 Hosts, routers, and link-layer switches; each contains a different set of layers, reflecting their differences in functionality

Chapter 2: Application Layer

Architecture Types: Client-server and peer-to-peer

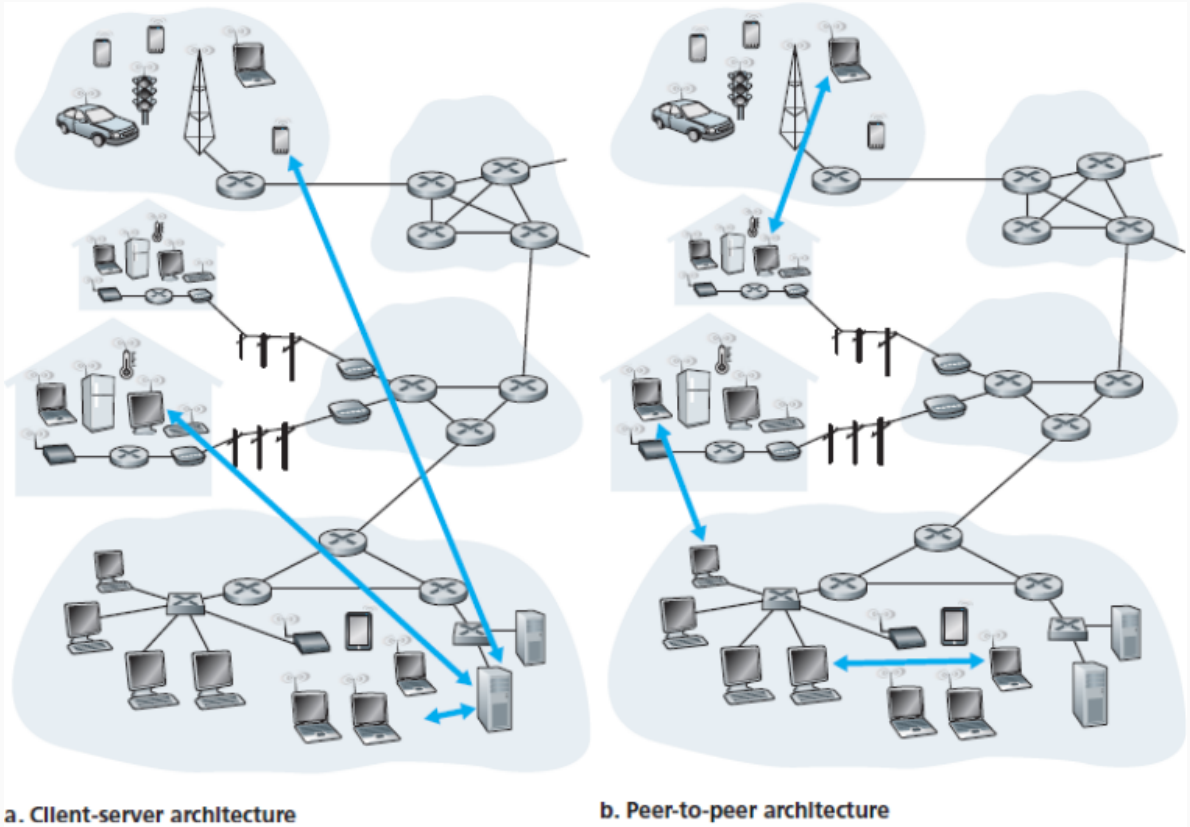


Figure 2.2 (a) Client-server architecture; (b) P2P architecture

	Client-Server	Peer-to-Peer
Discovery	**Trivial**	Via supernodes
Synchronization	**High**	Low
Reliability	Low	**High**
Scaling	Limited	**Flexible**

Socket API

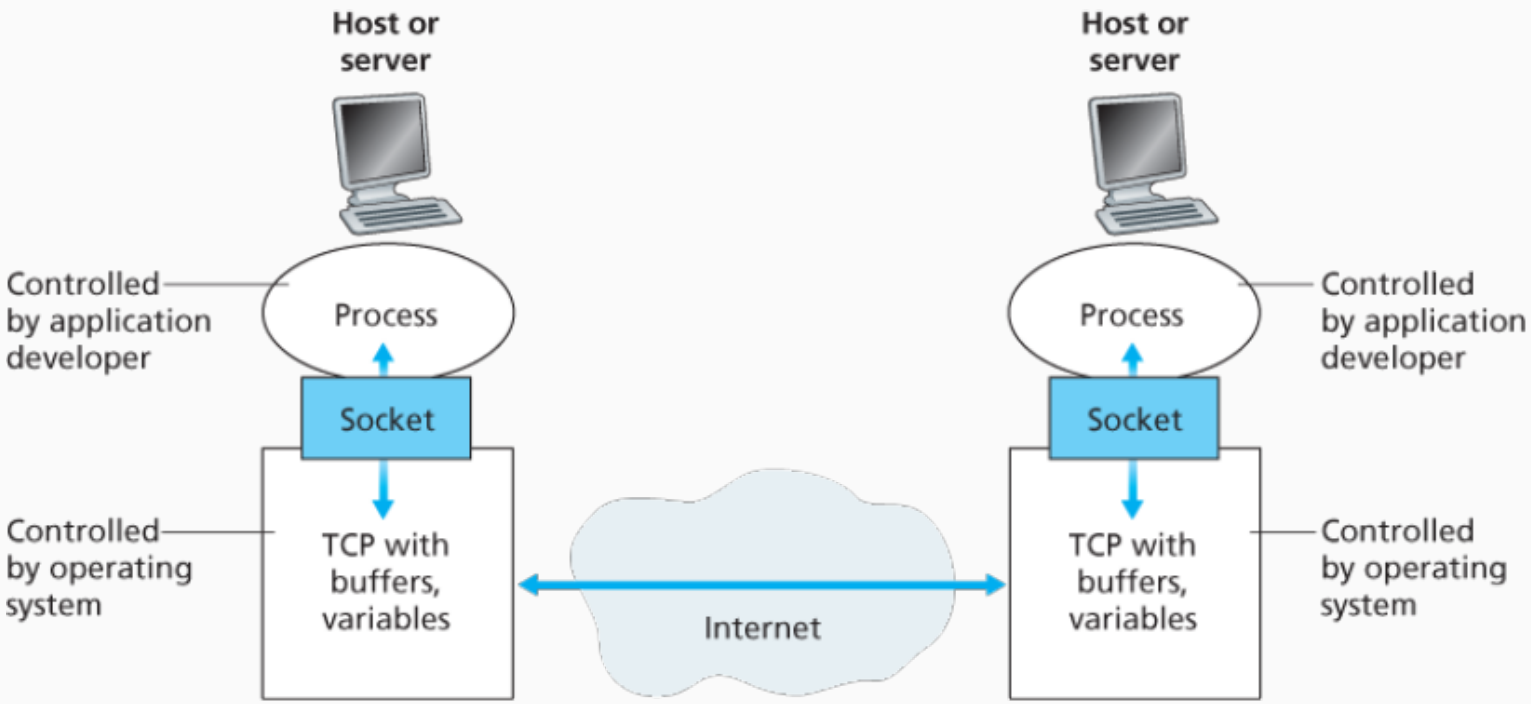
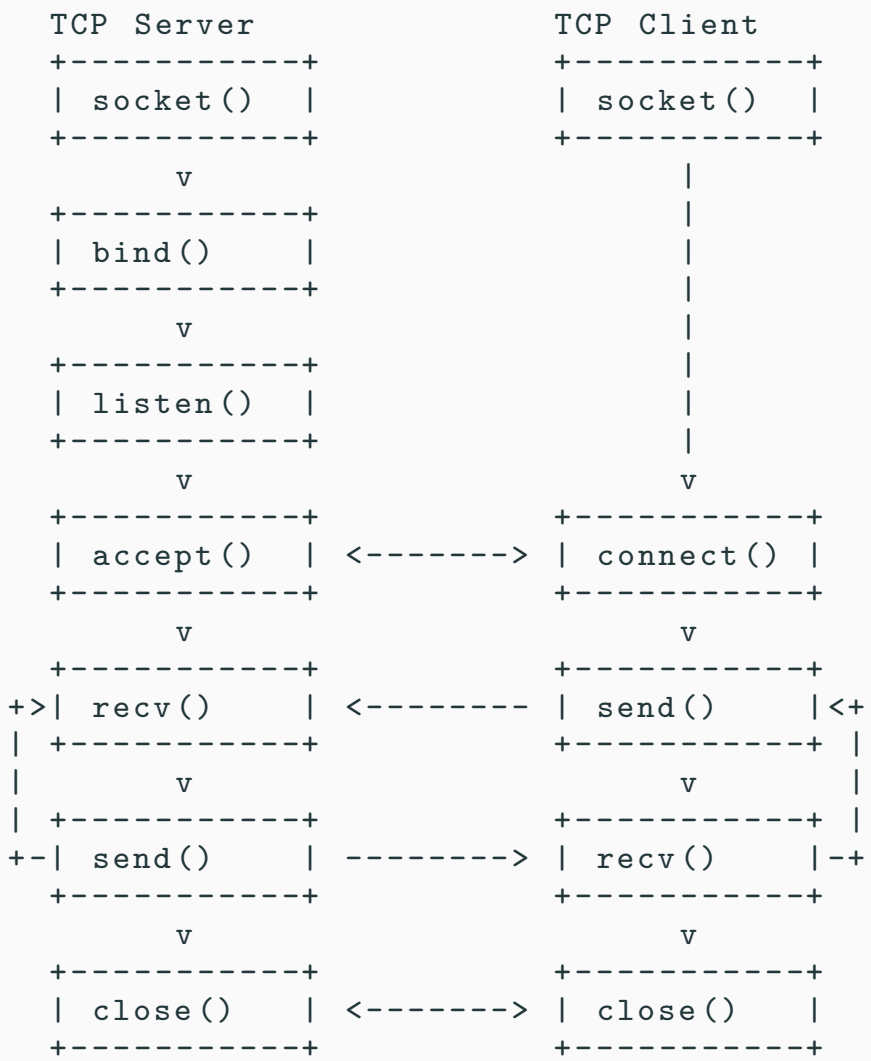
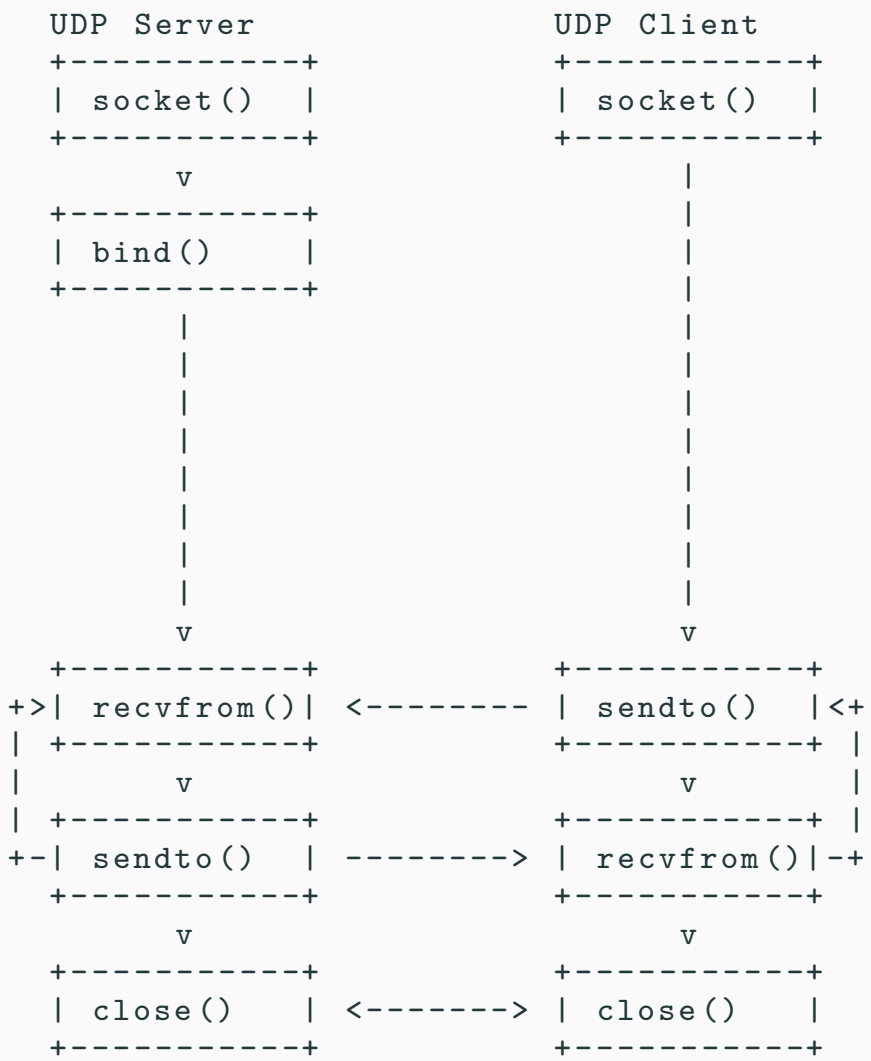


Figure 2.3 Application processes, sockets, and underlying transport protocol

TCP Server/Client Flow Diagram

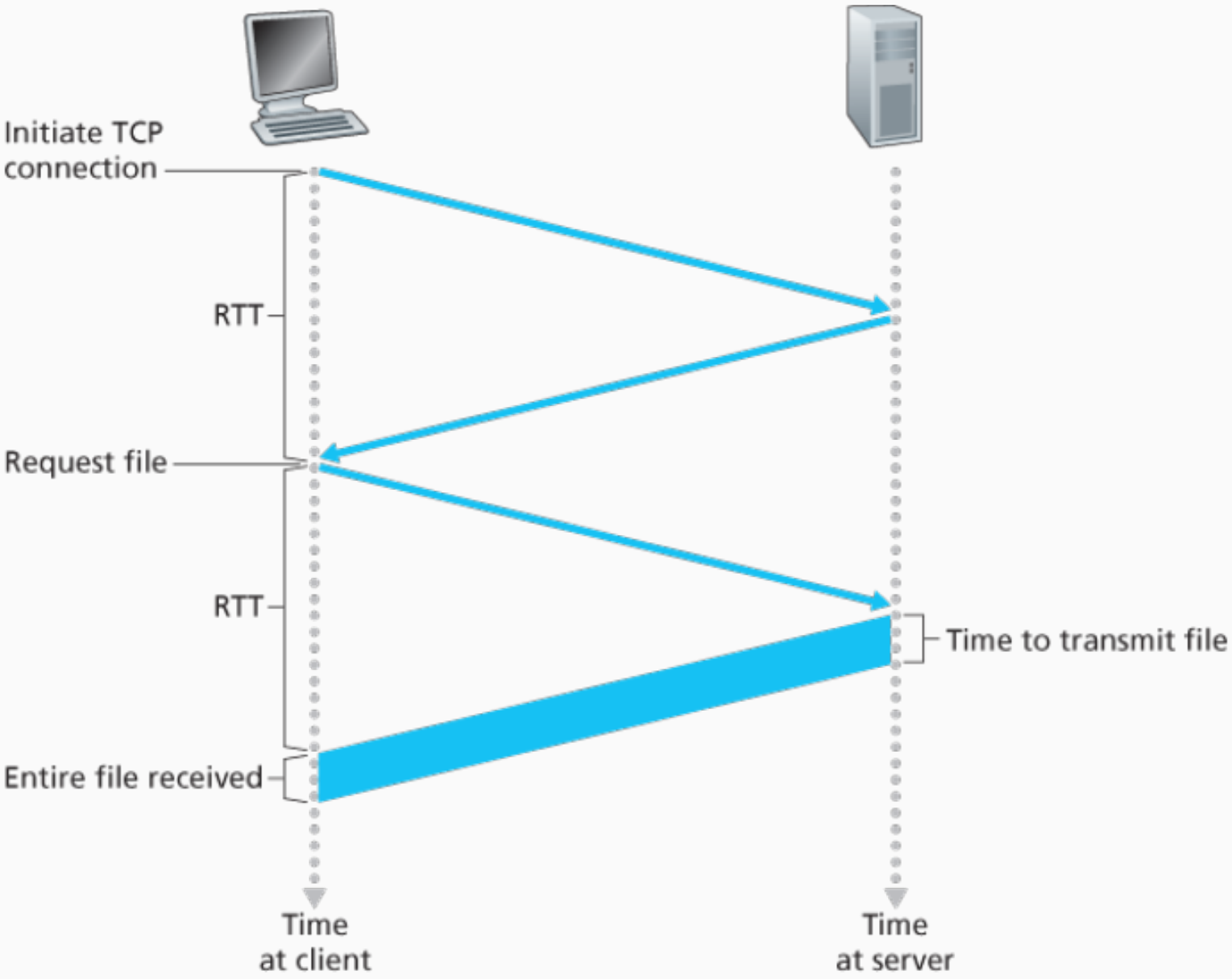


UDP Server/Client Flow Diagram



Hypertext Transfer Protocol (HTTP)

HTTP Latency



HTTP Request Format

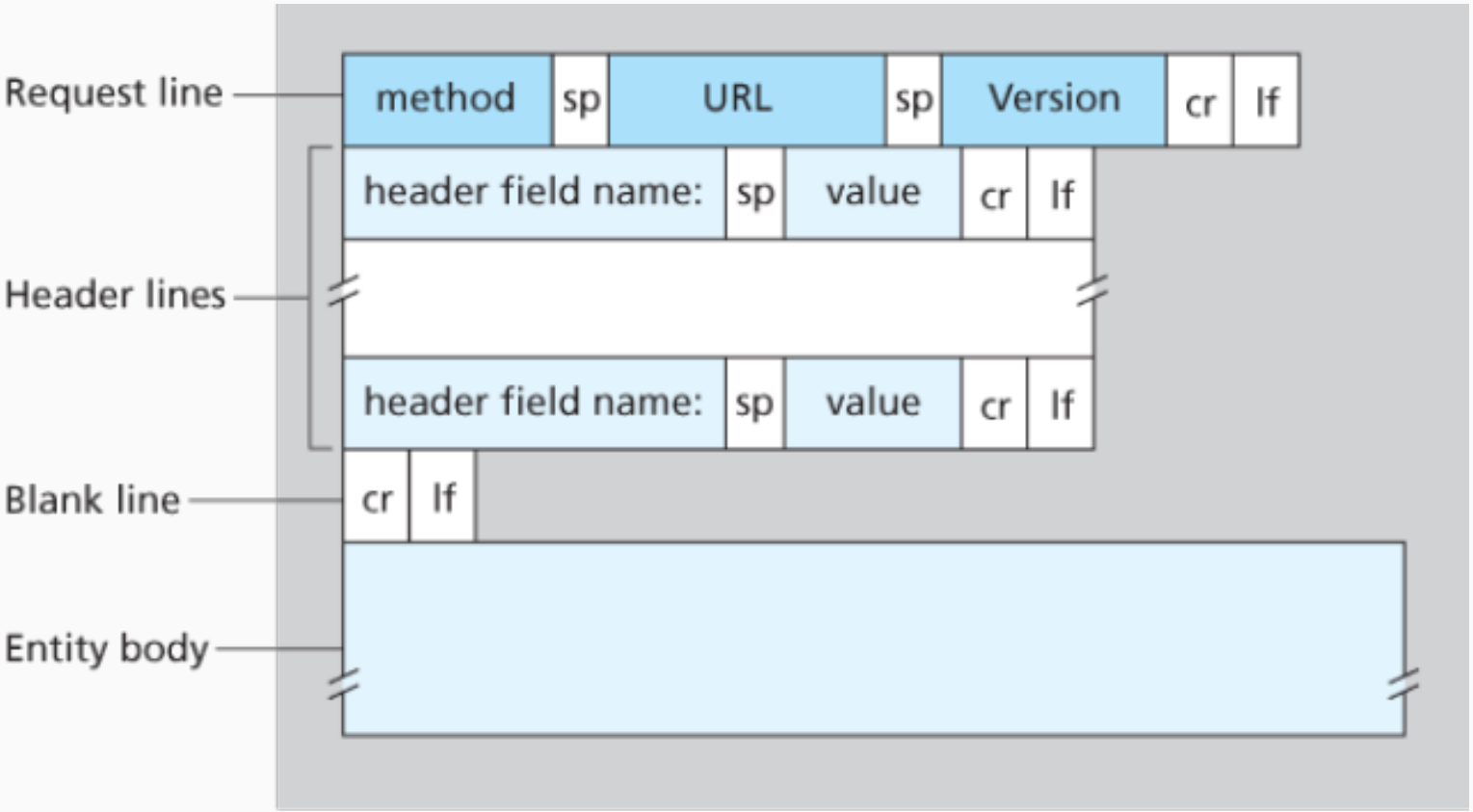


Figure 2.8 General format of an HTTP request message

HTTP Response Format

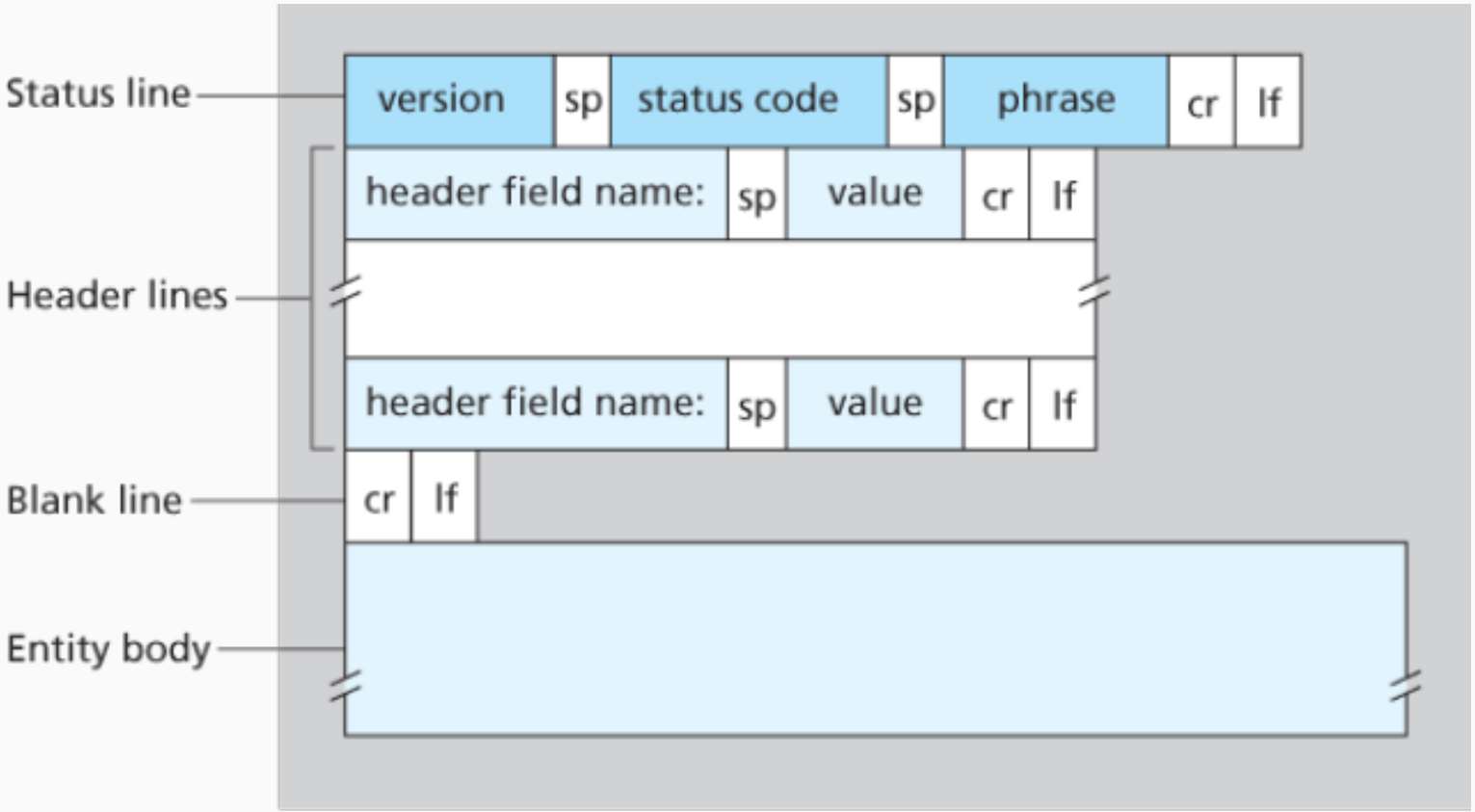


Figure 2.9 General format of an HTTP response message

Maintain State with Stateless HTTP

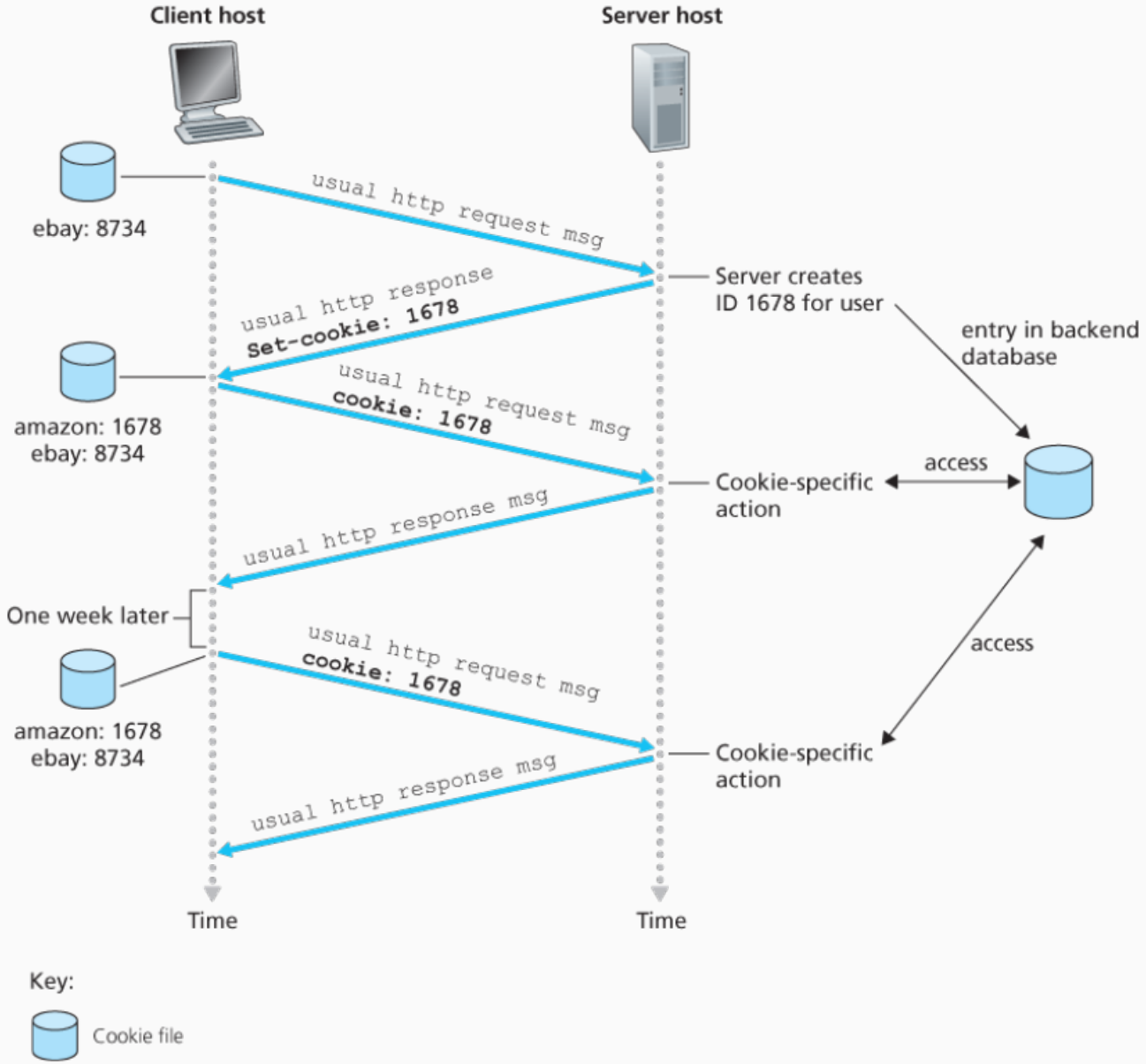


Figure 2.10 Keeping user state with cookies

SMTP

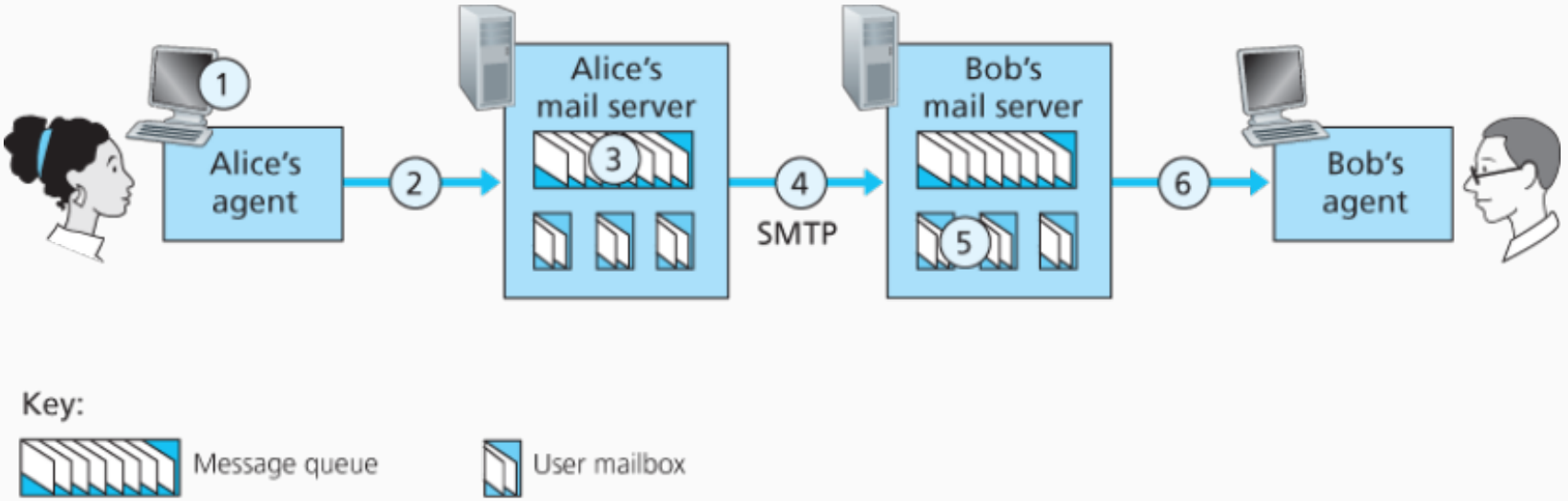
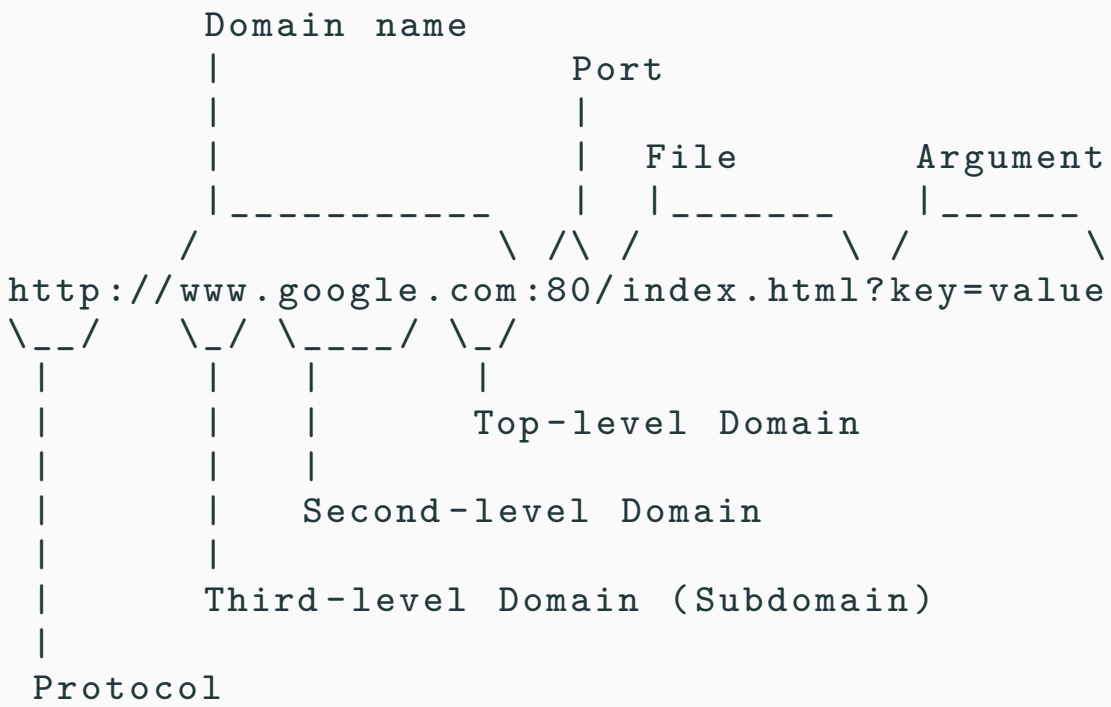
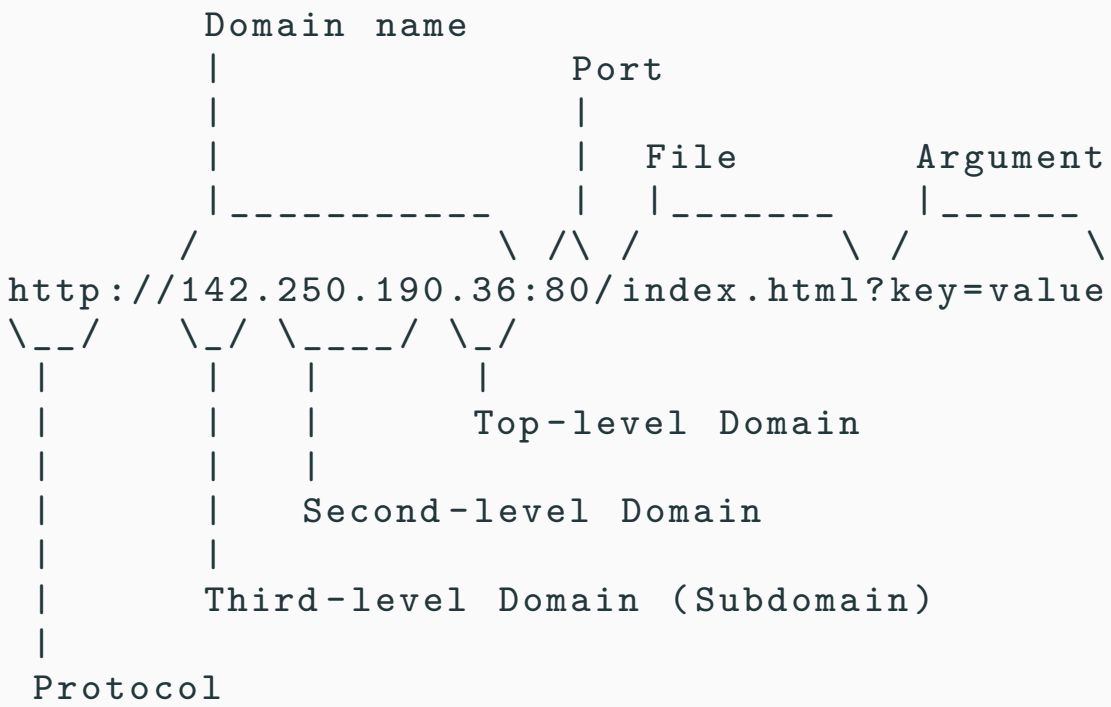


Figure 2.15 Alice sends a message to Bob

Domain Names



Domain Names



DNS Hierarchy

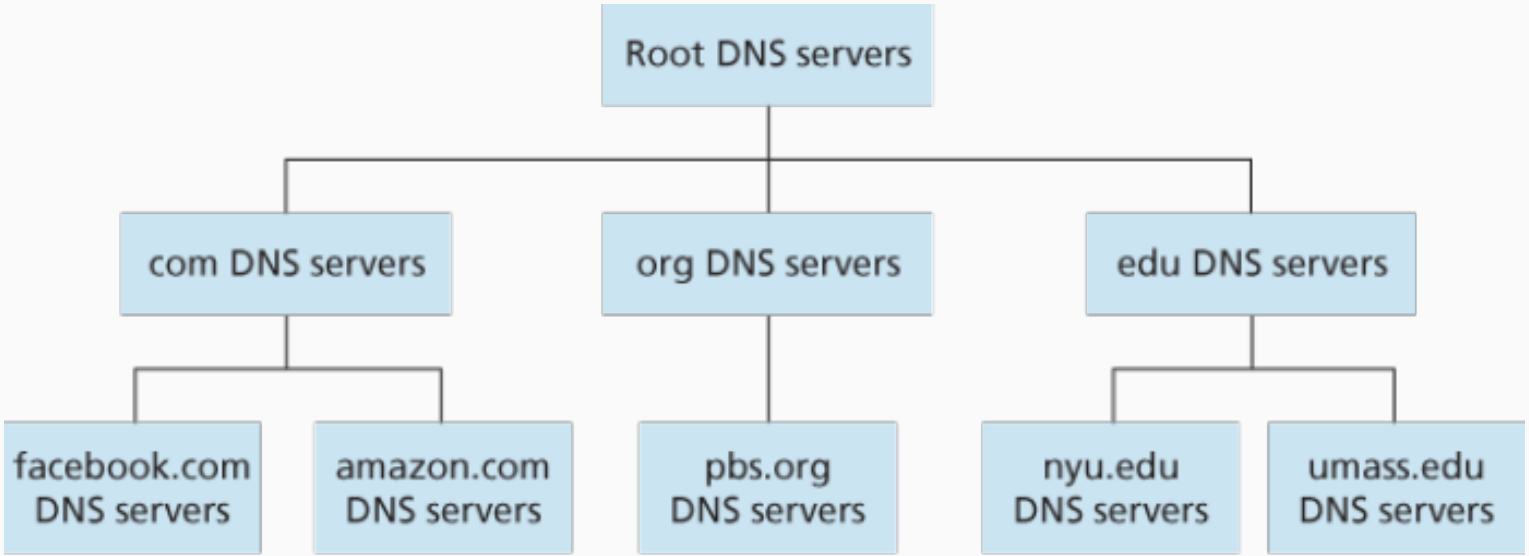
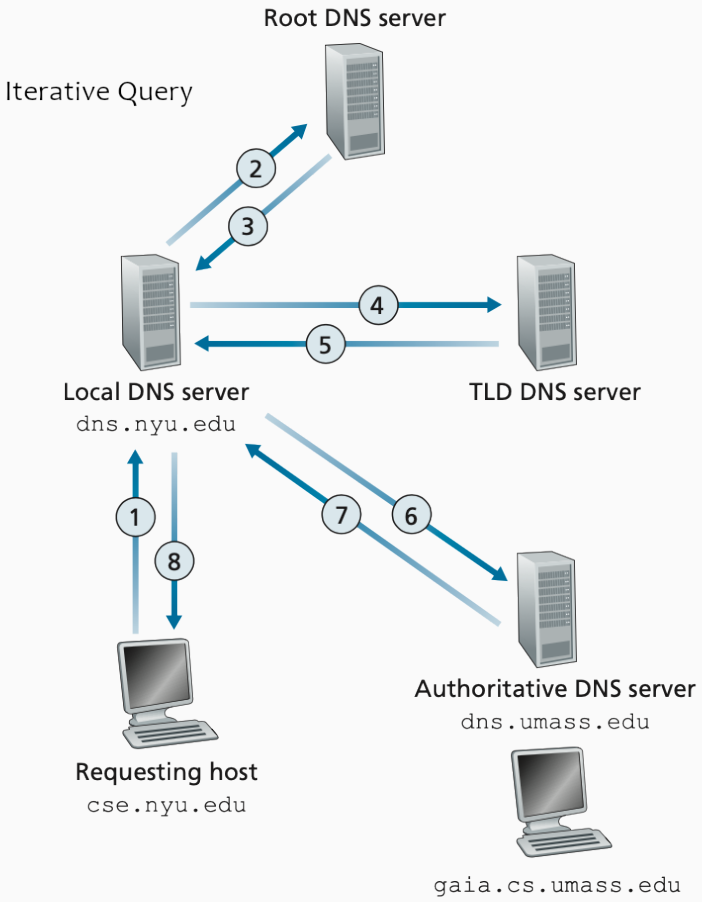
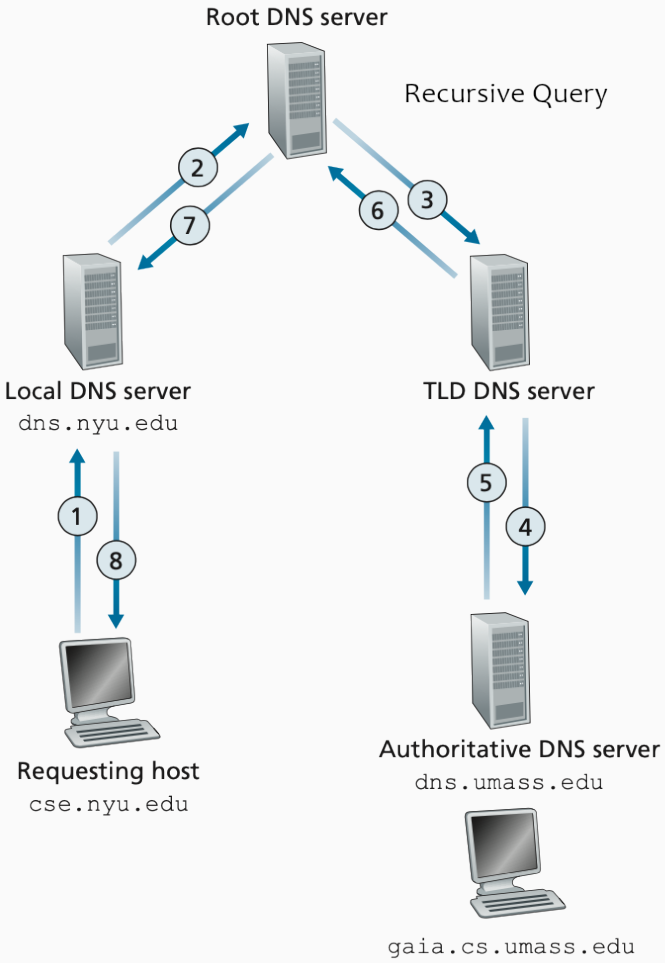


Figure 2.17 Portion of the hierarchy of DNS servers

DNS Queries


Root DNS server
Iterative Query
2
3
4
5
Local DNS server
dns.nyu.edu
TLD DNS server
1
8
7
6
Authoritative DNS server
dns.umass.edu
Requesting host
cse.nyu.edu
gaia.cs.umass.edu


Root DNS server
Recursive Query
2
7
6
3
Local DNS server
dns.nyu.edu
TLD DNS server
1
8
5
4
Requesting host
cse.nyu.edu
Authoritative DNS server
dns.umass.edu
gaia.cs.umass.edu

Chapter 3: Transport Layer

Multiplexing and Demultiplexing

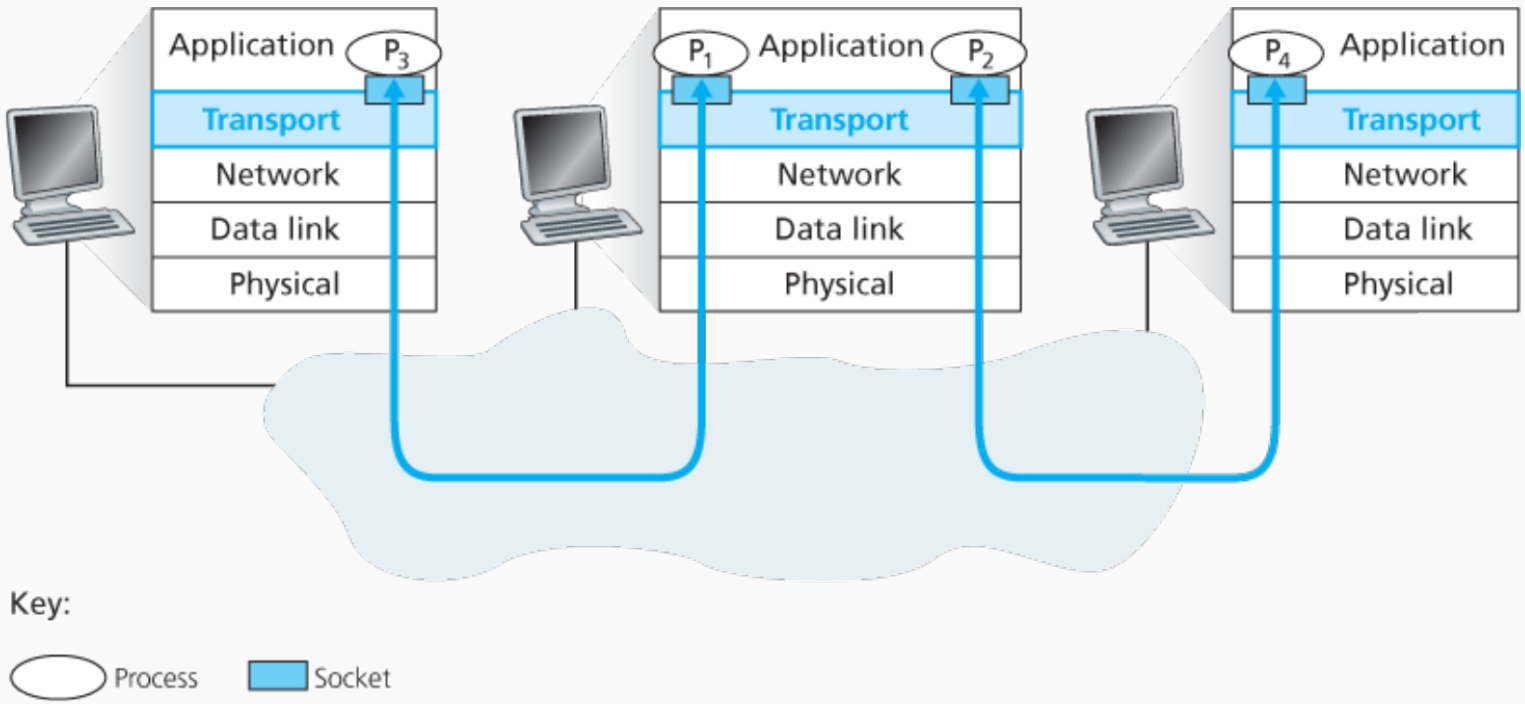


Figure 3.2 Transport-layer multiplexing and demultiplexing

Demultiplexing Example: HTTP Server

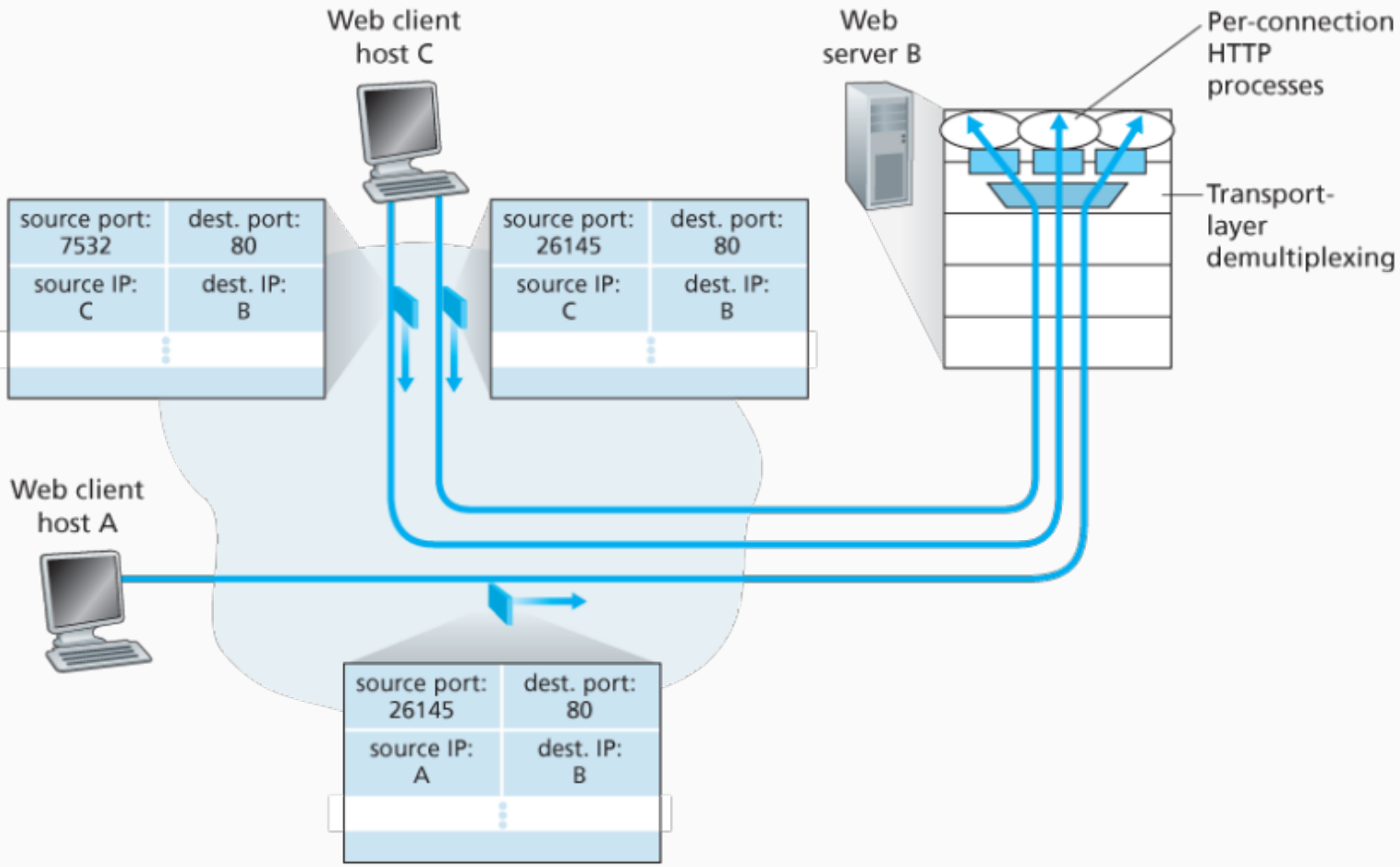


Figure 3.5 Two clients, using the same destination port number (80) to communicate with the same Web server application

Reliable Data Transfer (RDT)

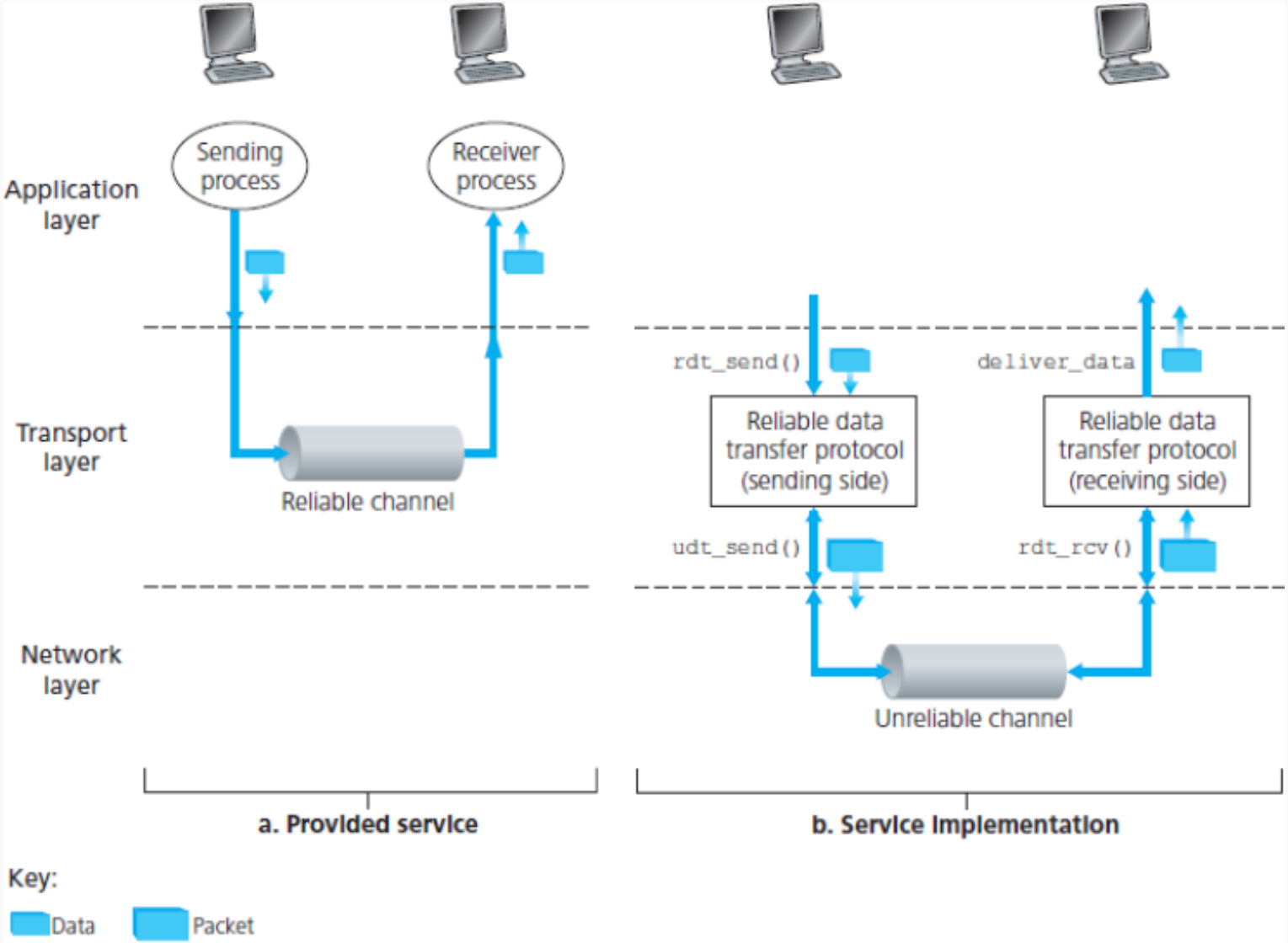
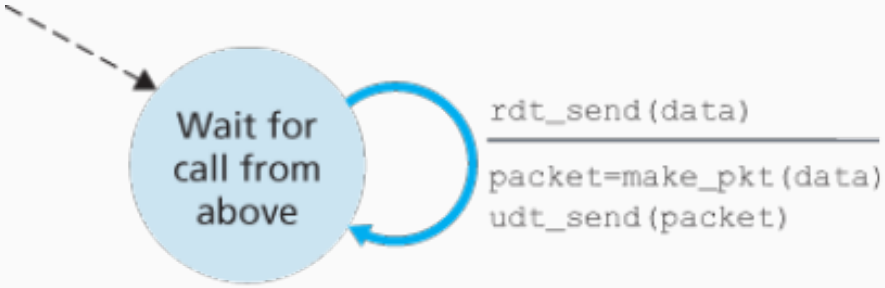
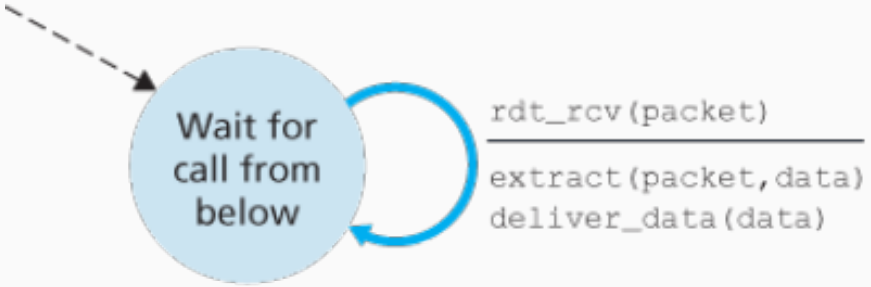


Figure 3.8 Reliable data transfer: Service model and service implementation

RDT over reliable channel: rdt1.0



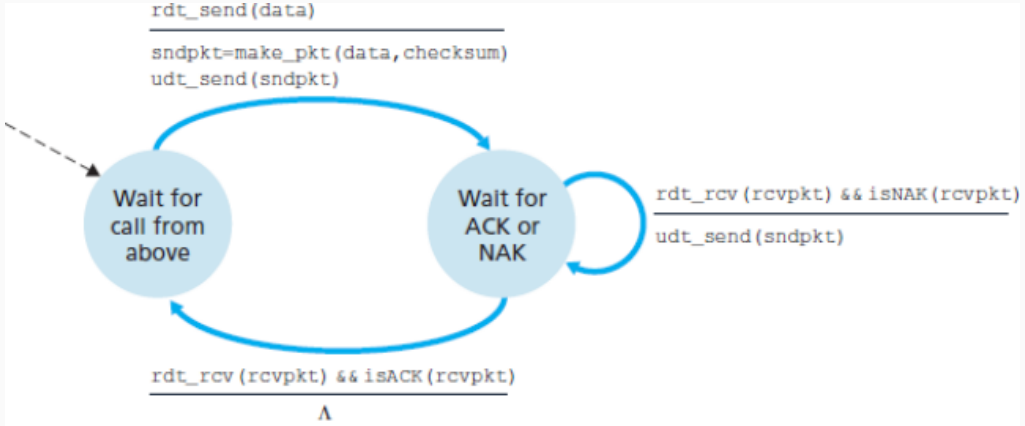
a. rdt1.0: sending side



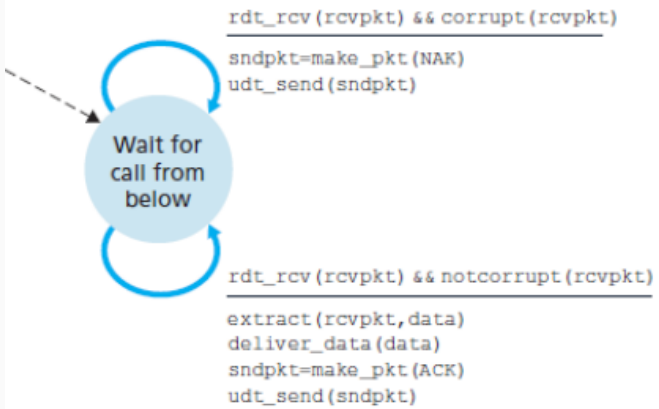
b. rdt1.0: receiving side

Figure 3.9 *rdt1.0* – A protocol for a completely reliable channel

RDT over unreliable channel with bit errors: rdt2.0



a. rdt2.0: sending side

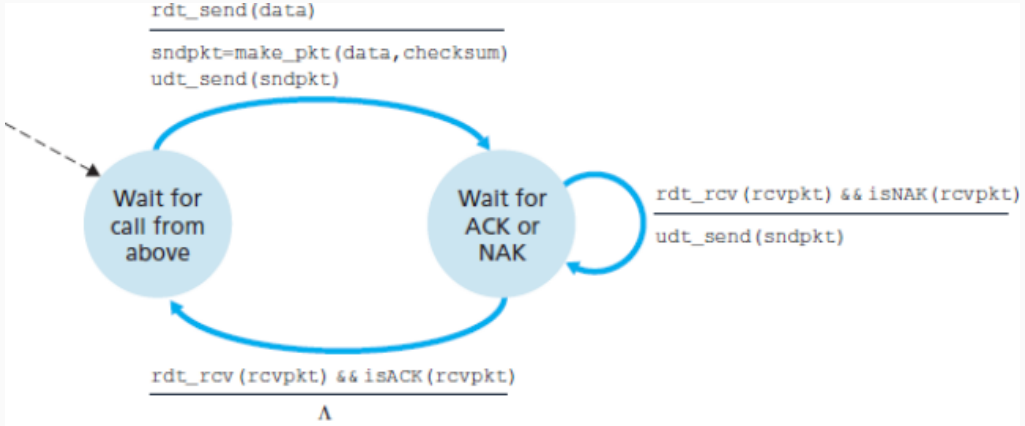


b. rdt2.0: receiving side

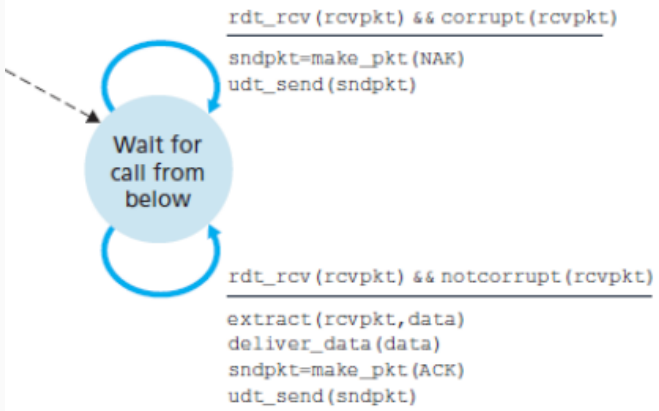
Figure 3.10 *rdt2.0* – A protocol for a channel with bit errors

Problem?

RDT over unreliable channel with bit errors: rdt2.0



a. rdt2.0: sending side



b. rdt2.0: receiving side

Figure 3.10 *rdt2.0* – A protocol for a channel with bit errors

Problem: Corrupted ACKs cause duplicates!

RDT2.1: Added sequence numbers

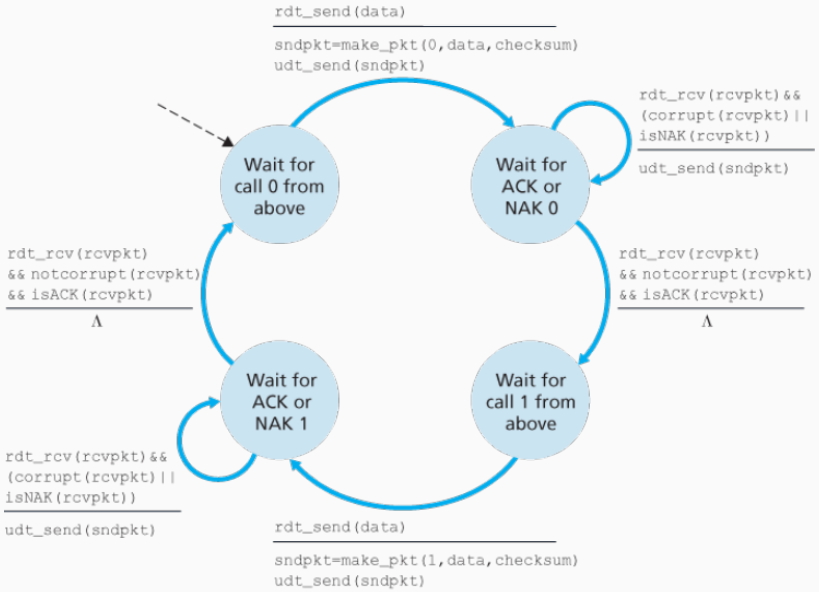


Figure 3.11 *rdt2.1* sender

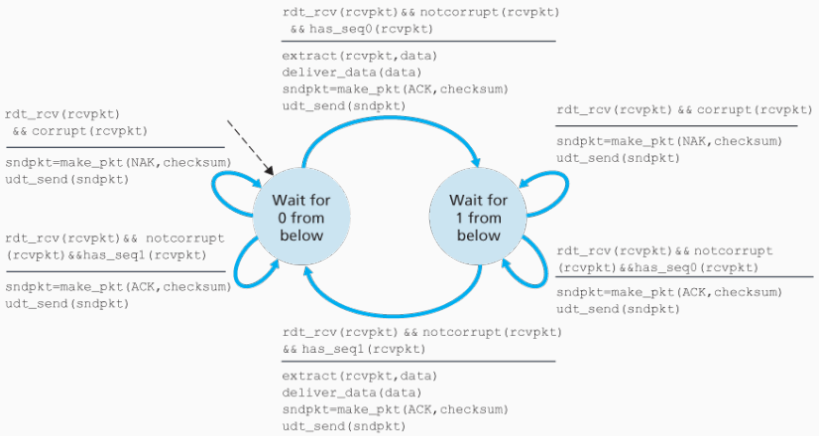


Figure 3.12 *rdt2.1* receiver

Sequence numbers help detecting duplicates

RDT2.2: NACK-free version

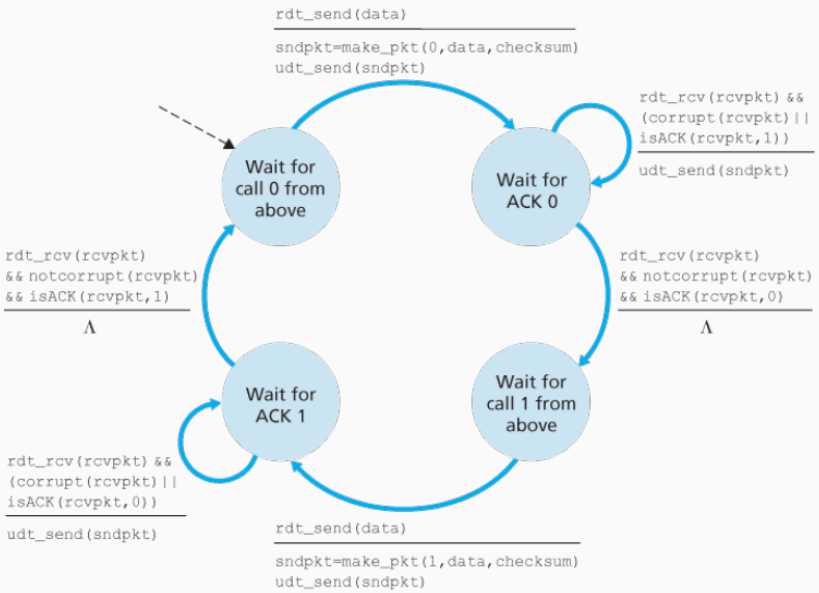


Figure 3.13 *rdt2.2* sender

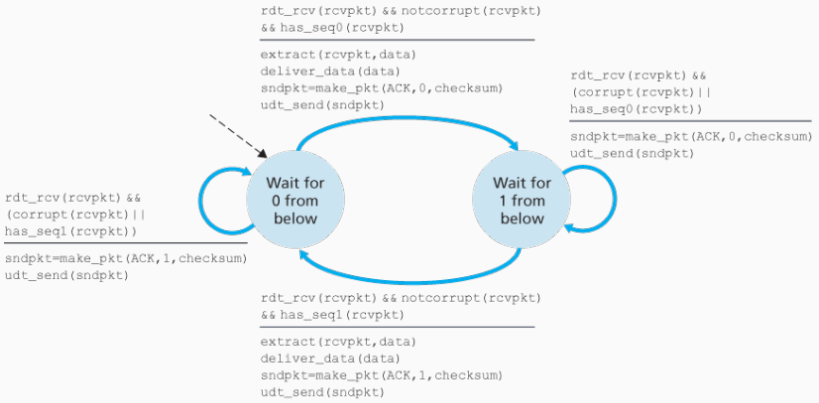


Figure 3.14 *rdt2.2* receiver

Instead of NACK, ACK the last one

Sequence numbers help detecting NACKs

RDT3.0: Added timeout

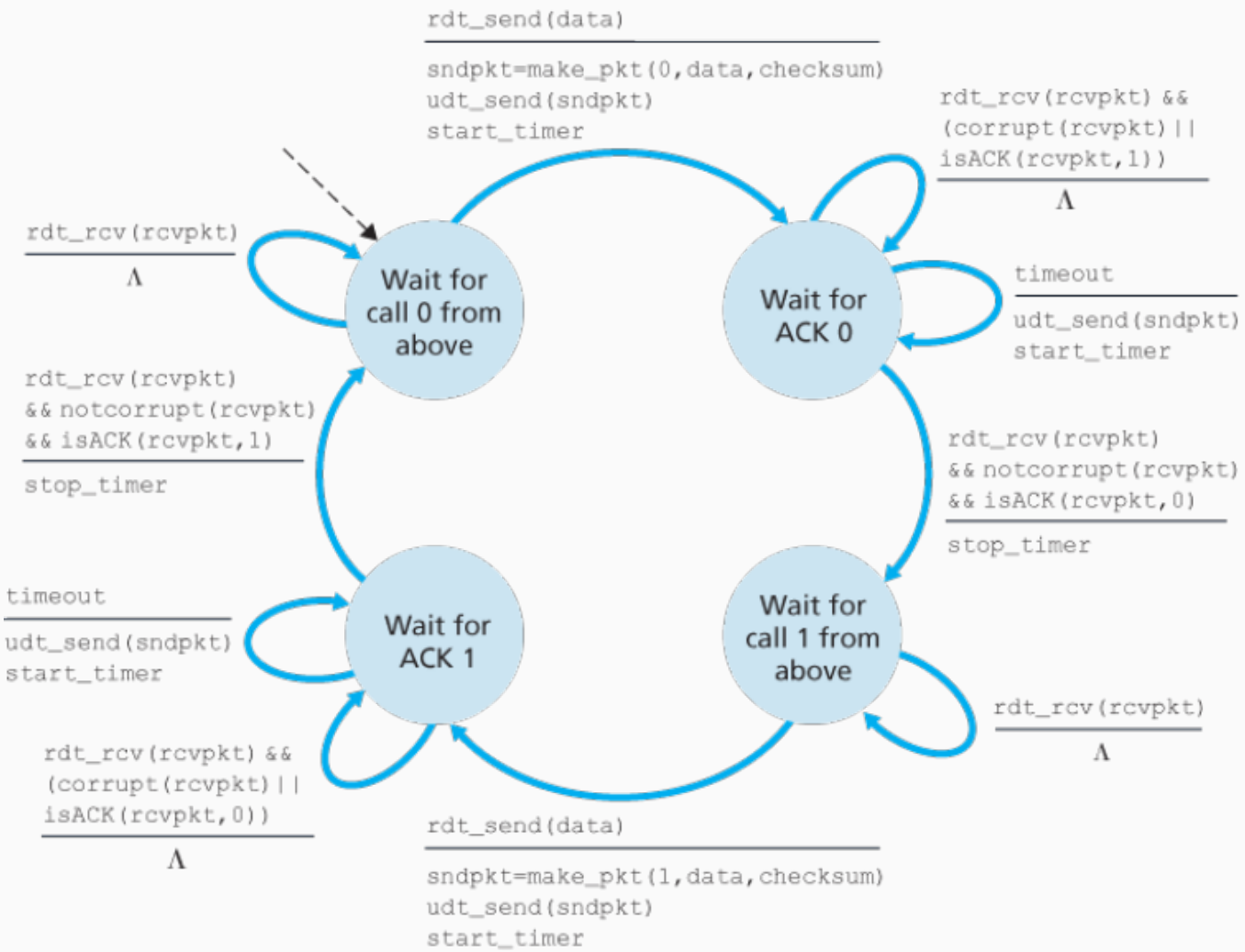
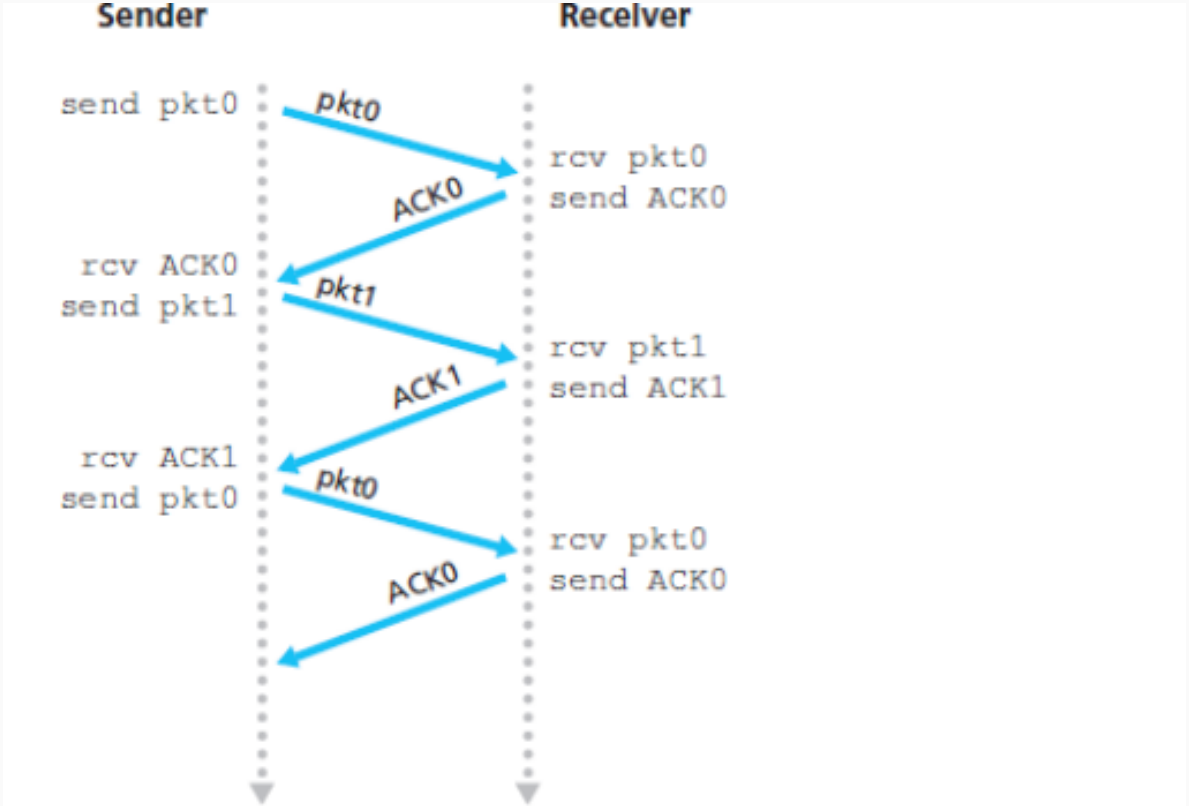


Figure 3.15 *rdt3.0* **sender**

RDT3.0: No Loss



a. Operation with no loss

Figure 3.16 Operation of *rdt3.0*, the alternating-bit protocol

RDT3.0: Lost Packet

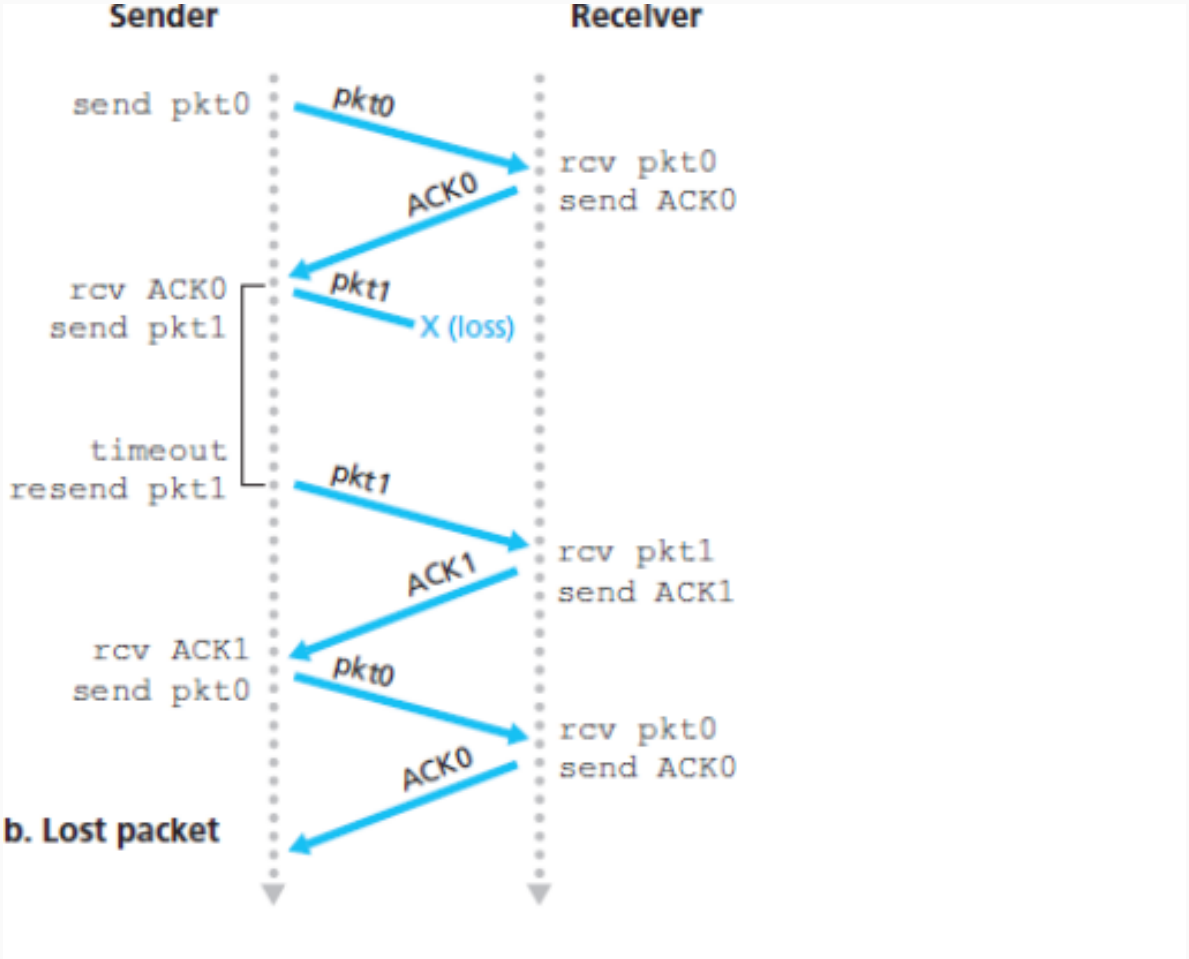
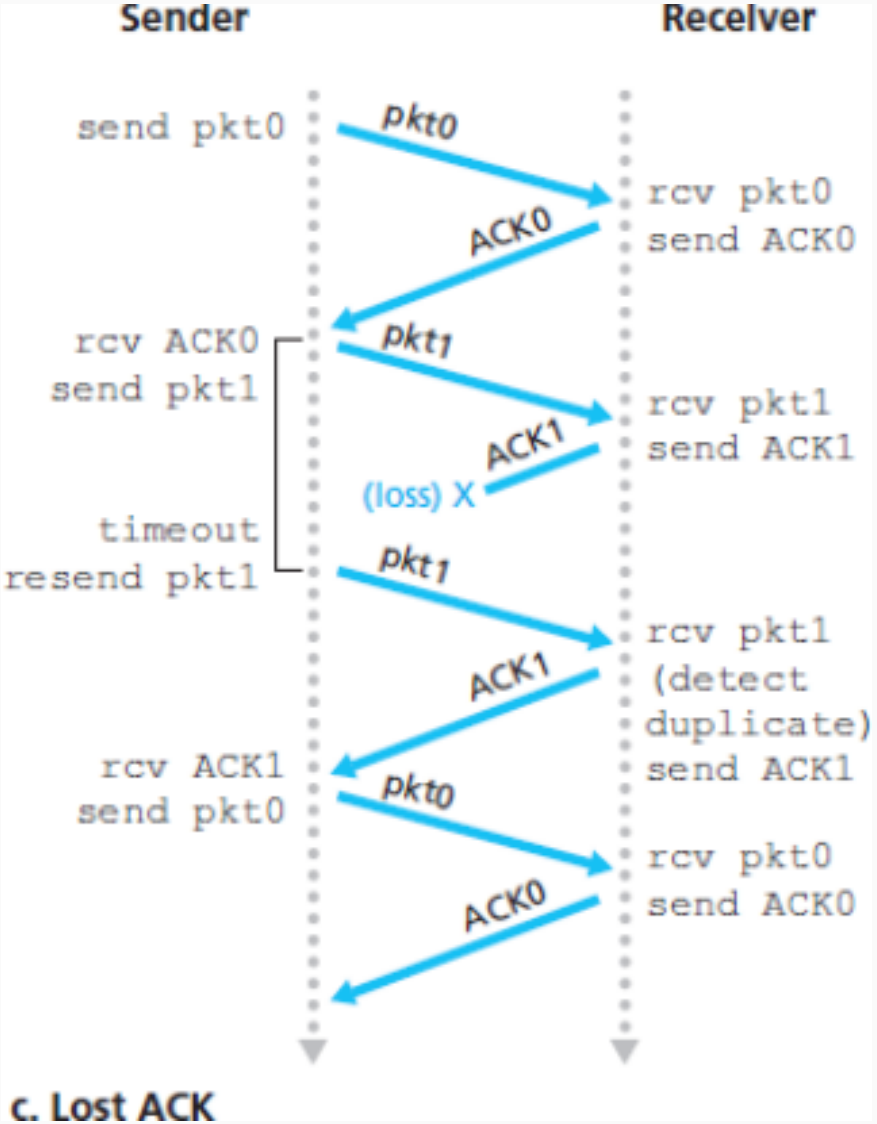
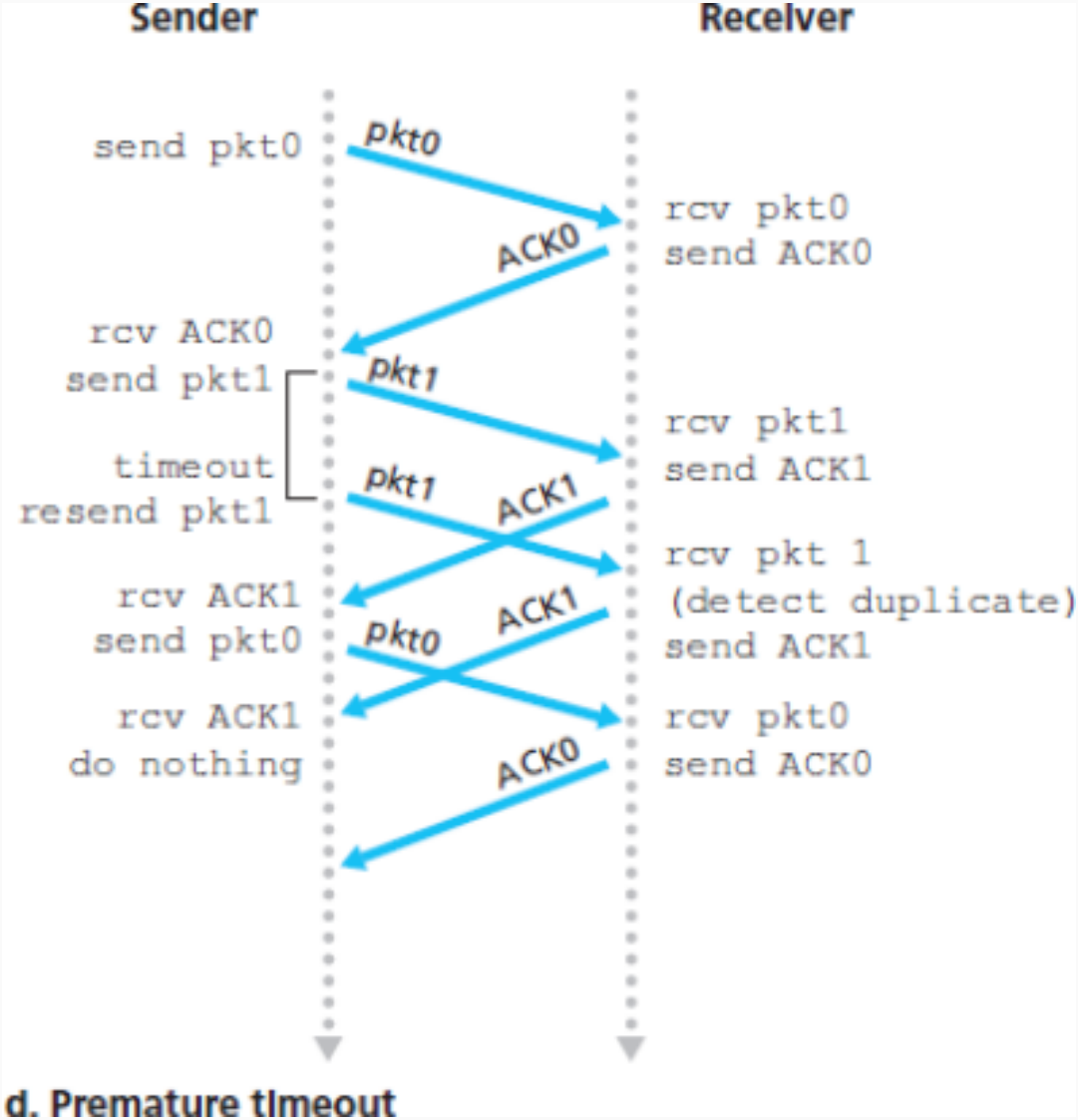


Figure 3.16 Operation of *rdt3.0*, the alternating-bit protocol

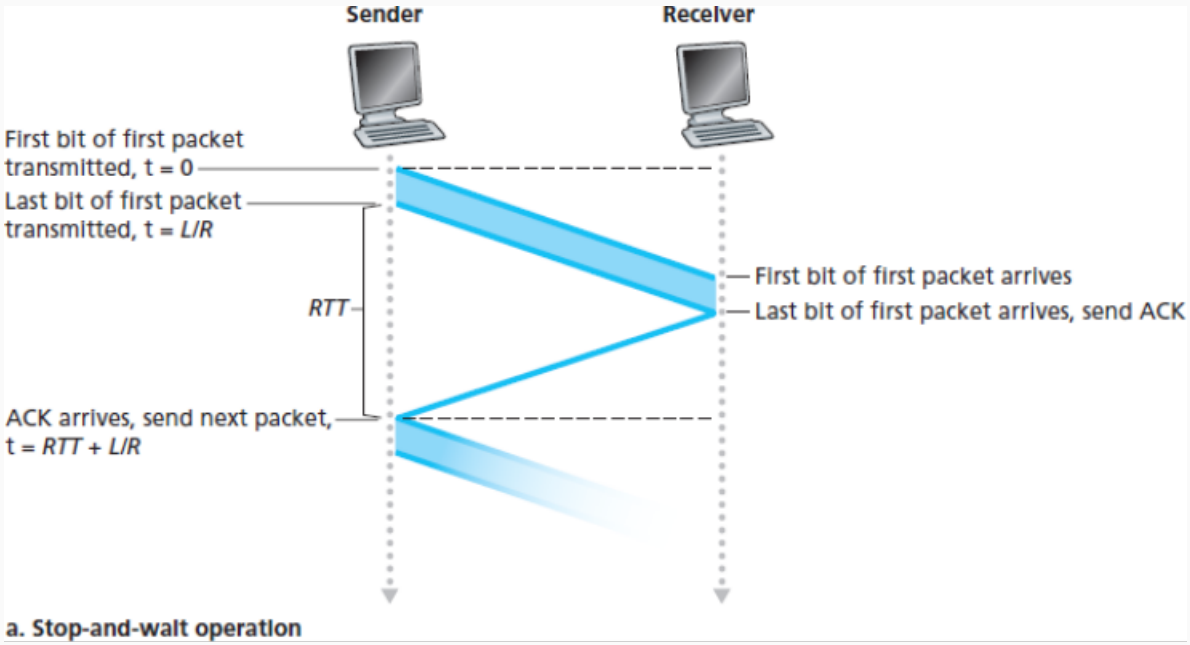
RDT3.0: Lost ACK



RDT3.0: Premature Timeout

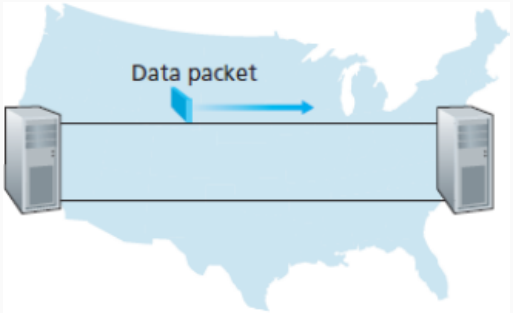


Stop-and-wait



a. Stop-and-wait operation

Figure 3.18 Stop-and-wait and pipelined sending



a. A stop-and-wait protocol in operation

Stop-and-wait

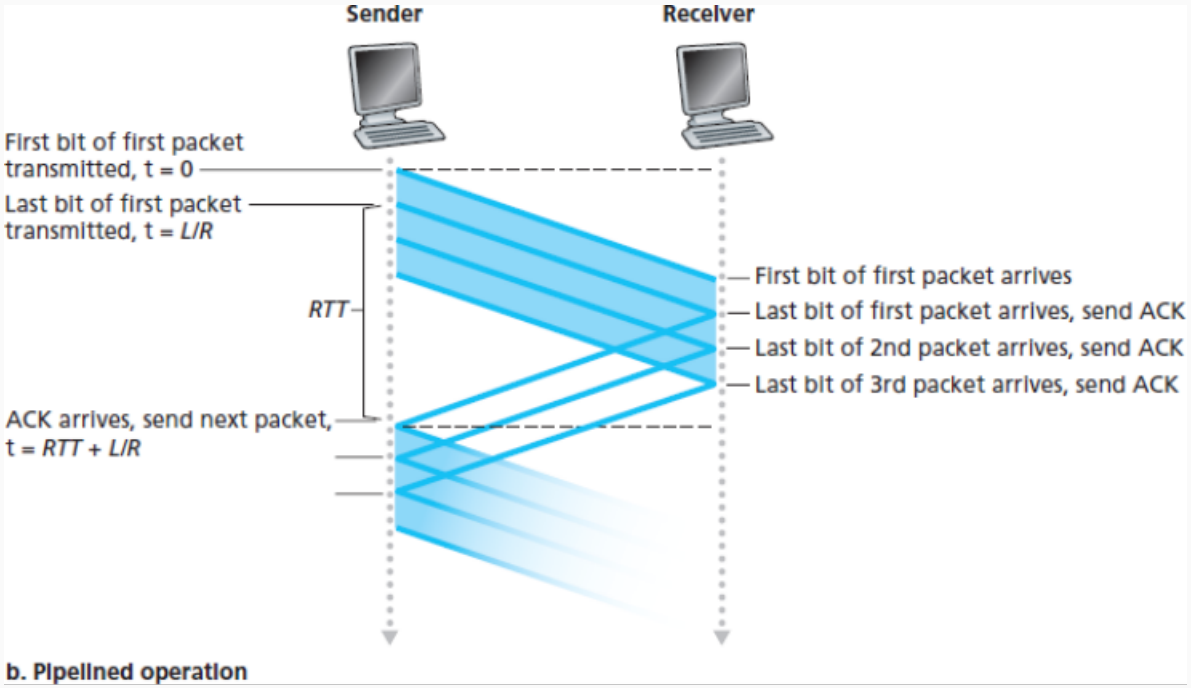
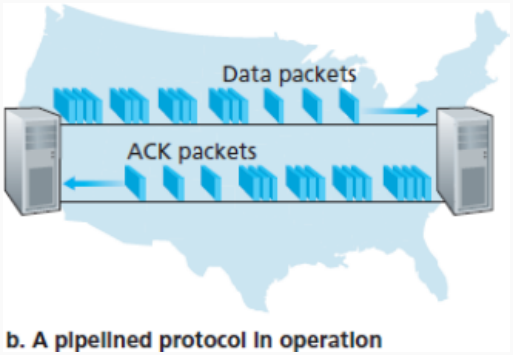


Figure 3.18 Stop-and-wait and pipelined sending



Go-Back-N

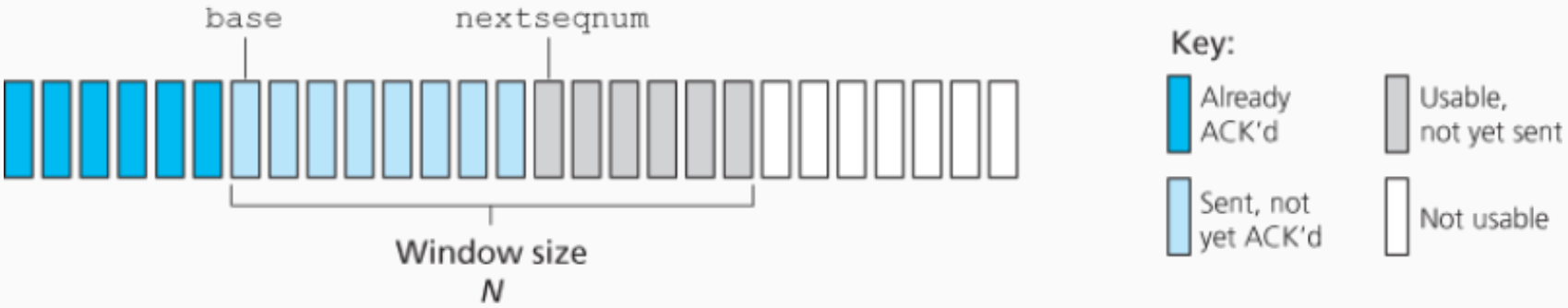


Figure 3.19 Sender's view of sequence numbers in Go-Back-N

Go-Back-N

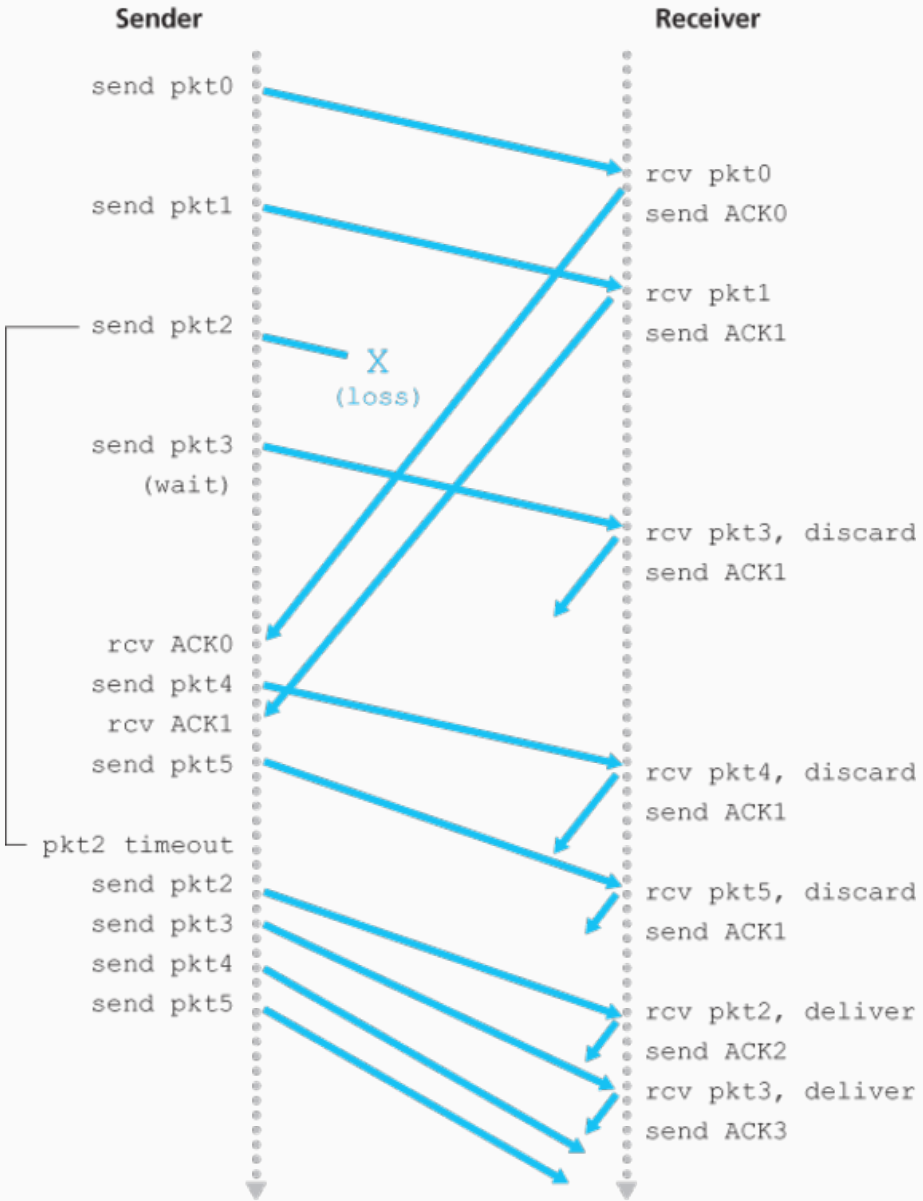


Figure 3.22 Go-Back-N in operation

Selective-Repeat

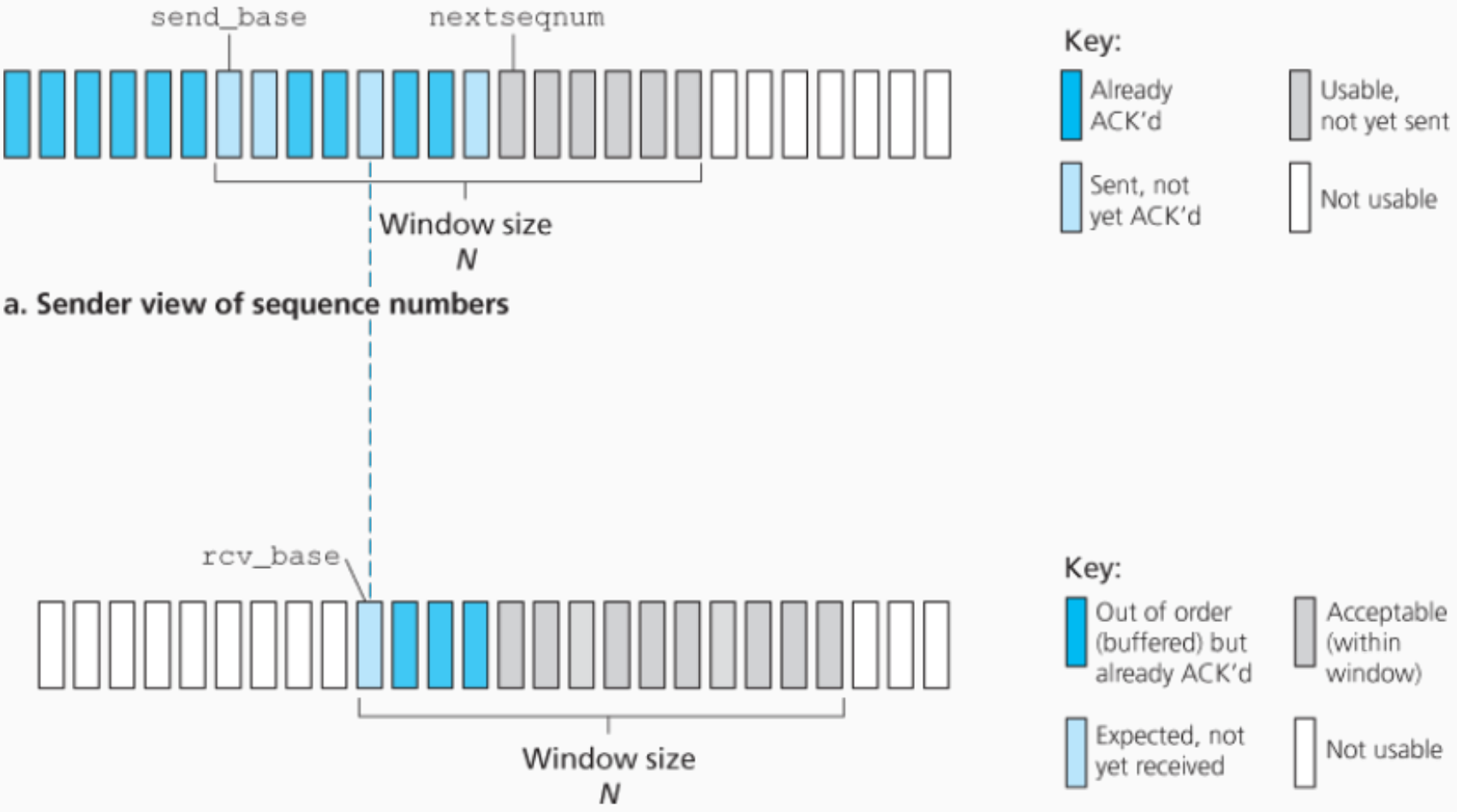


Figure 3.23 Selective-repeat (SR) sender and receiver views of sequence-number space

Selective-Repeat

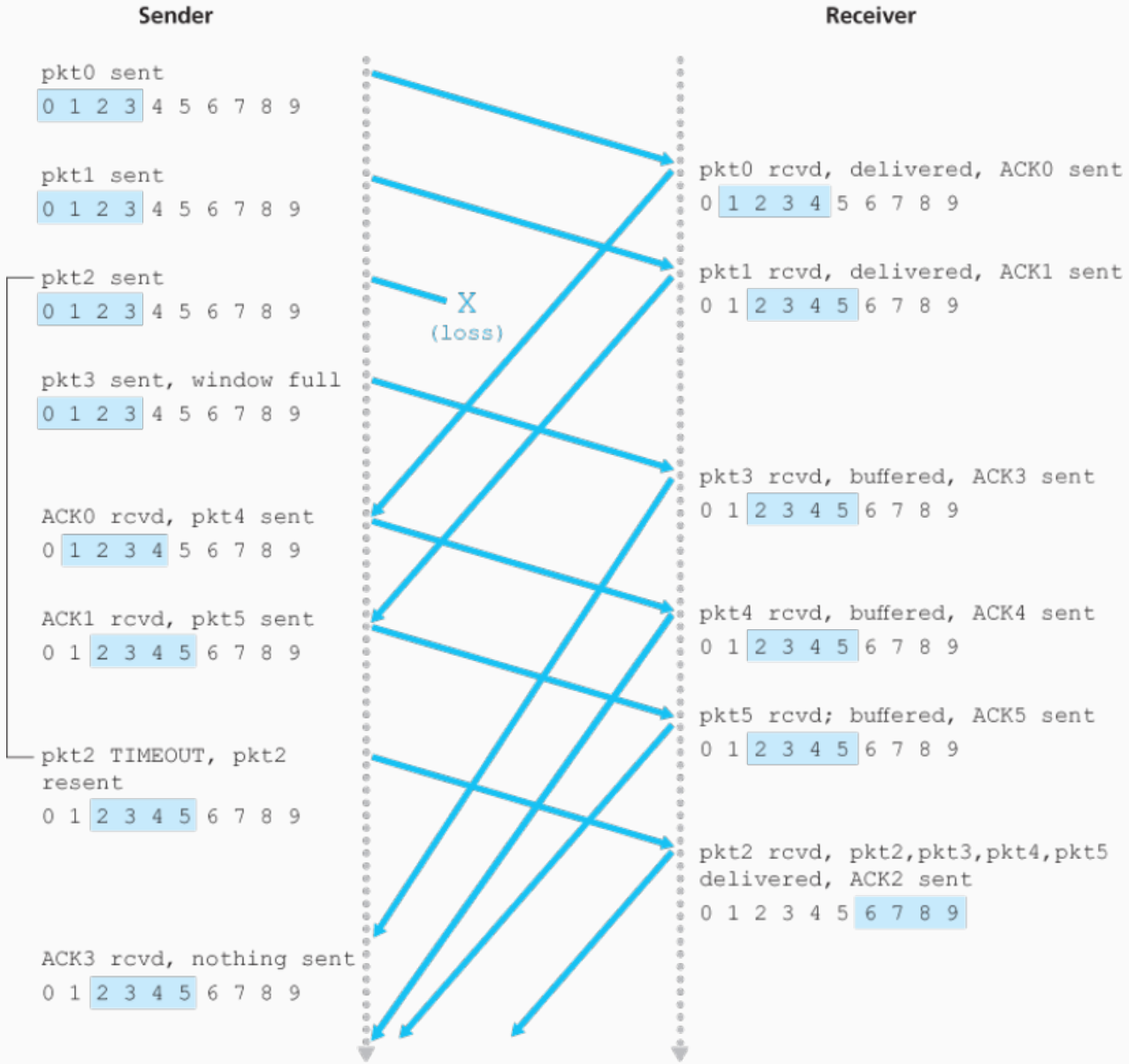


Figure 3.26 SR operation

Reliable Data Transfer Summary

Properties	Stop-and-Wait	Go-Back-N	Selective Repeat
Sender window size	1	N	N
Receiver window size	1	1	N
Minimum sequence number	2	$N + 1$	2N
Type of ACK	Individual	Cumulative	Individual
Supported order	-	In-order only	Out-of-order as well
Number of retransmissions	1	N	1

Reliable Data Transfer Mechanisms

Mechanism	Use, Comments
Checksum	Detect *some* bit errors
Timer	Detect loss
Sequence number	Detect loss, duplicate
ACK	Individual or cumulative
NACK	
Window, pipelining	Increase utilization

TCP Send and Receive Buffers

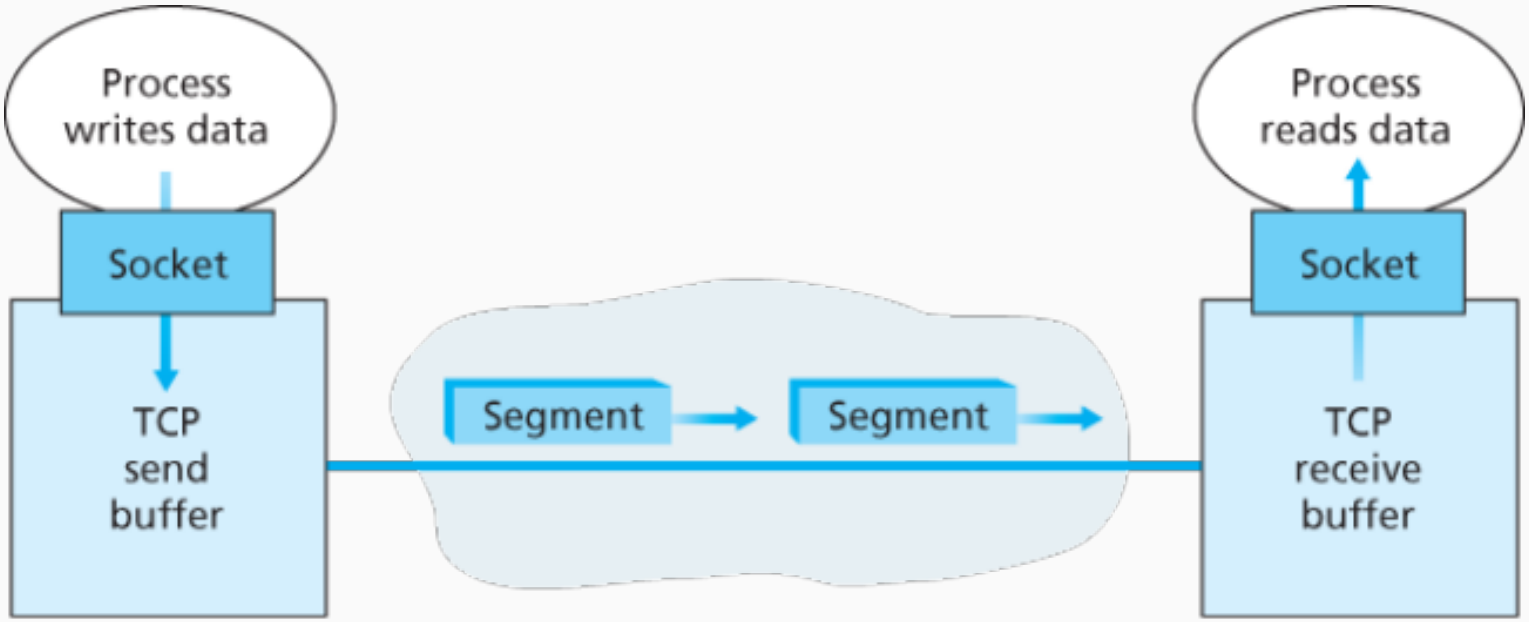


Figure 3.28 TCP send and receive buffers

TCP Format

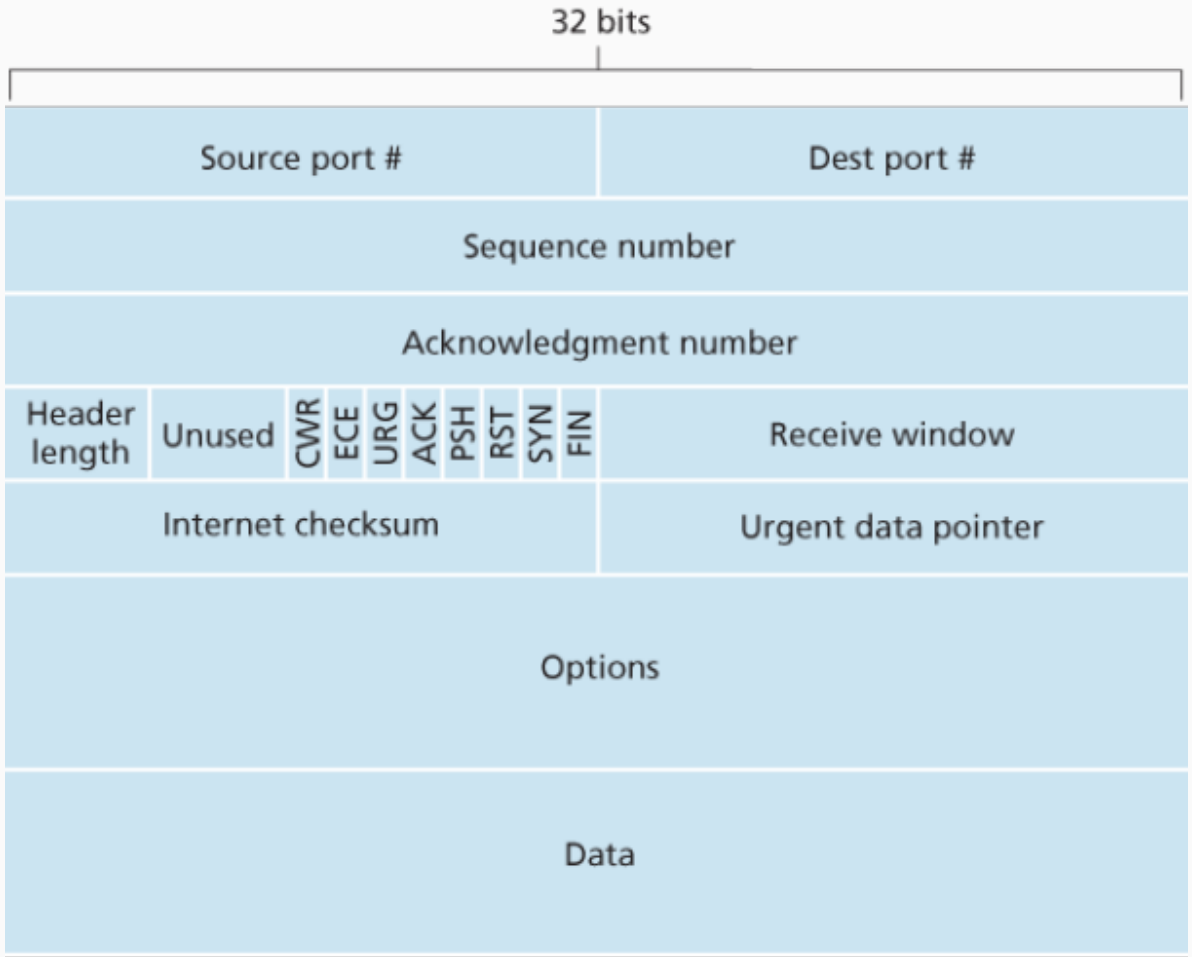


Figure 3.29 TCP segment structure

TCP Sequence Numbers

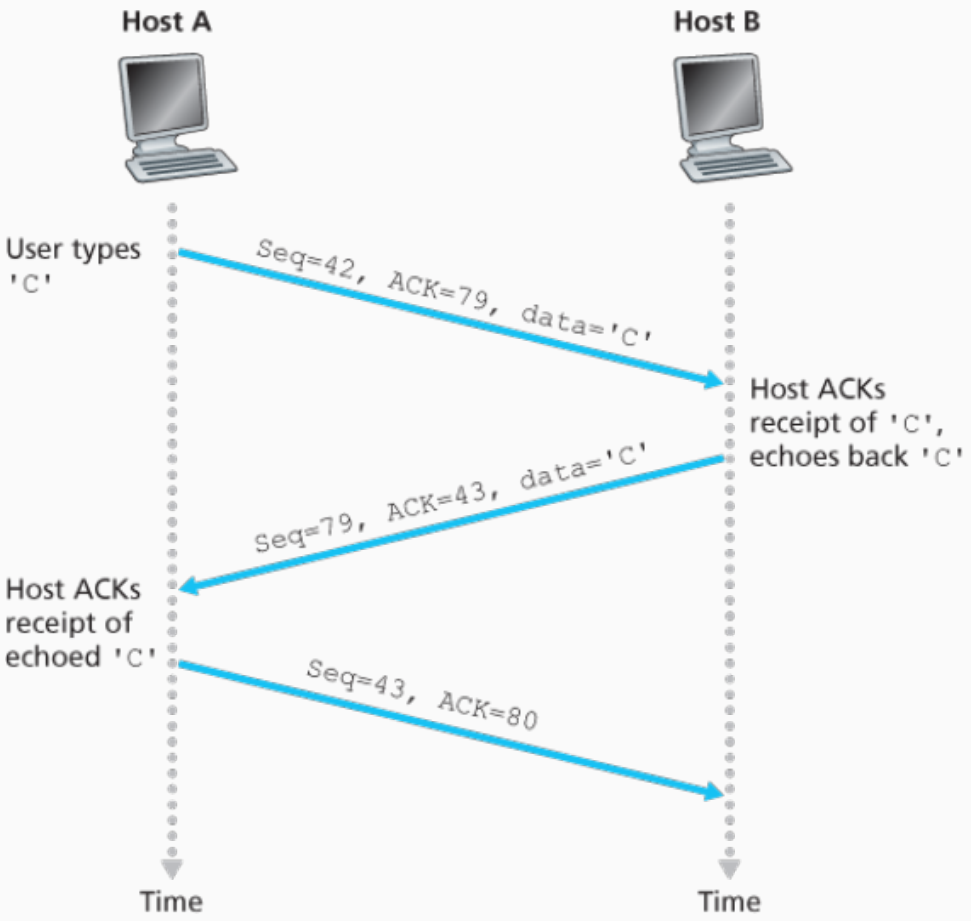
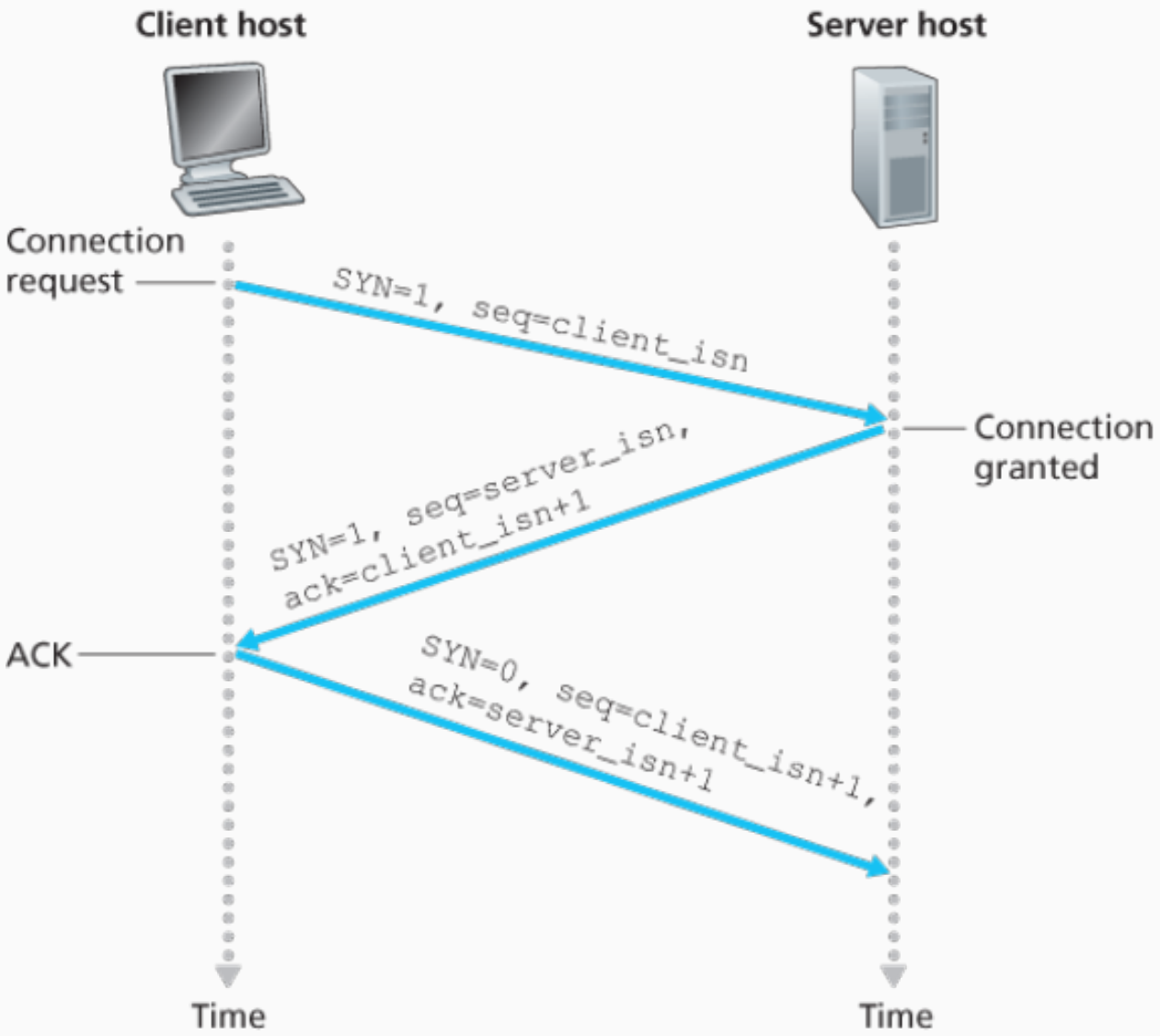


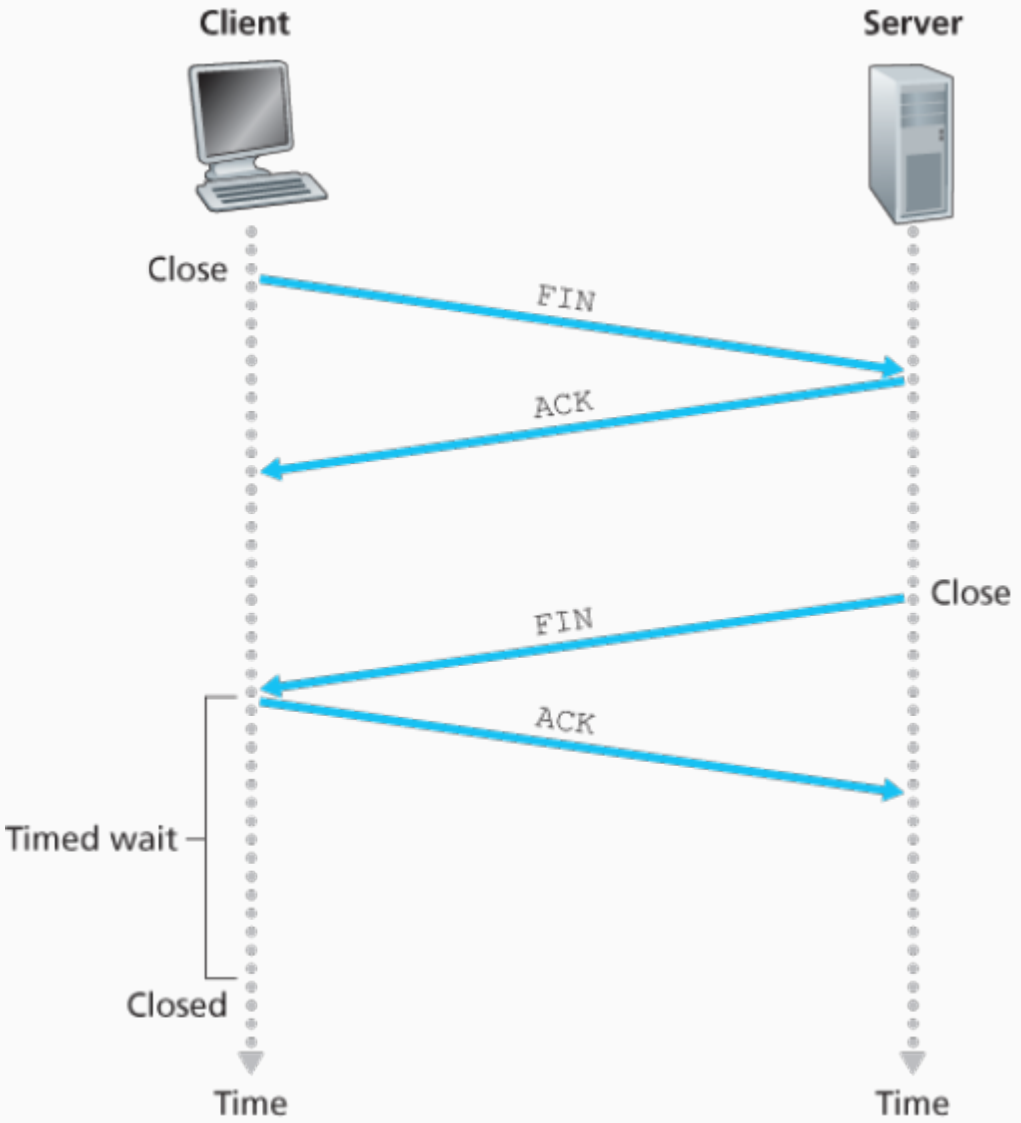
Figure 3.31 Sequence and acknowledgment numbers for a simple Telnet application

ACK = received Seq + data length

TCP Handshake



TCP Connection Termination



TCP Retransmission Timeout

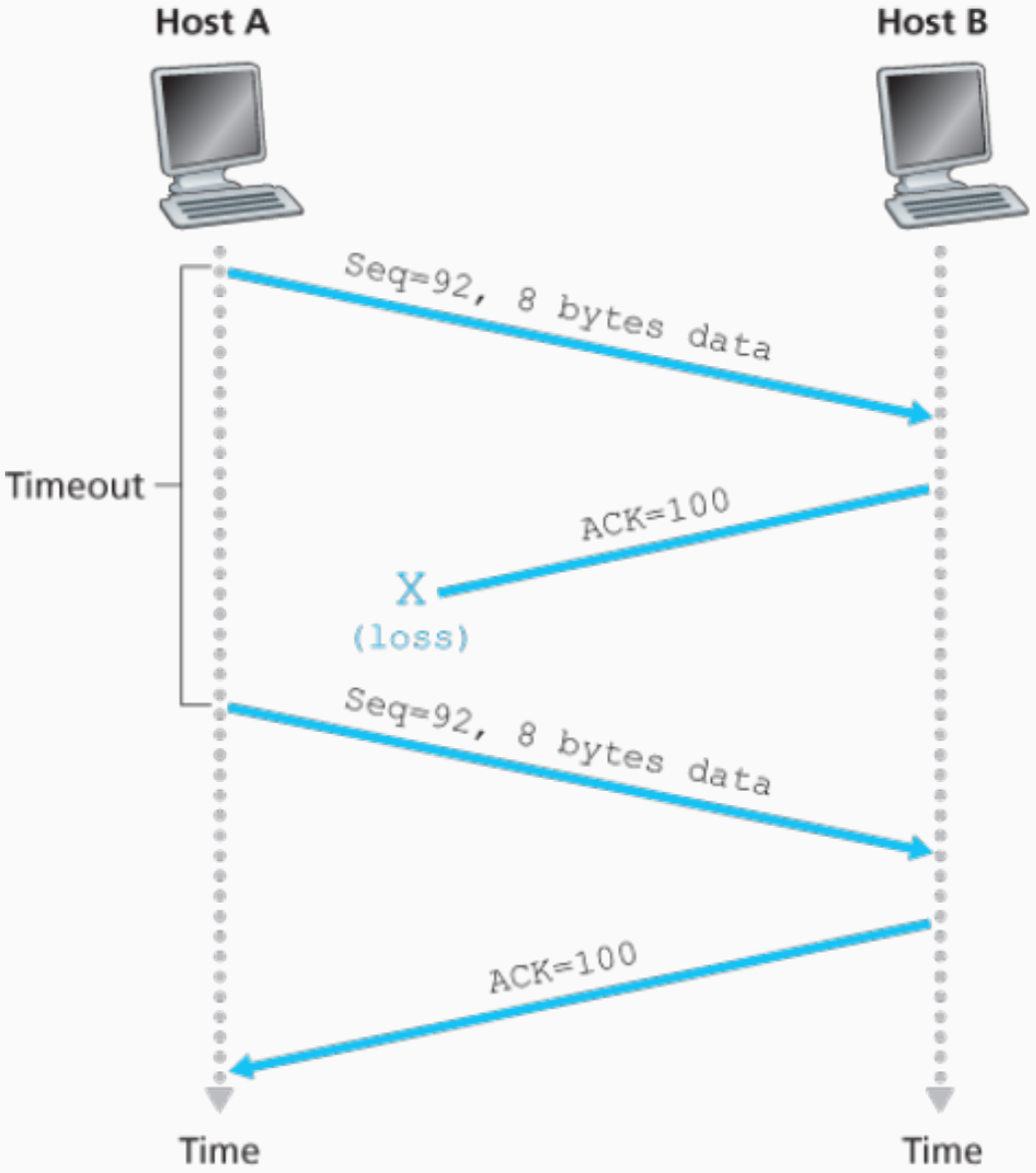


Figure 3.34 Retransmission due to a lost acknowledgment

TCP Cumulative Acknowledgment

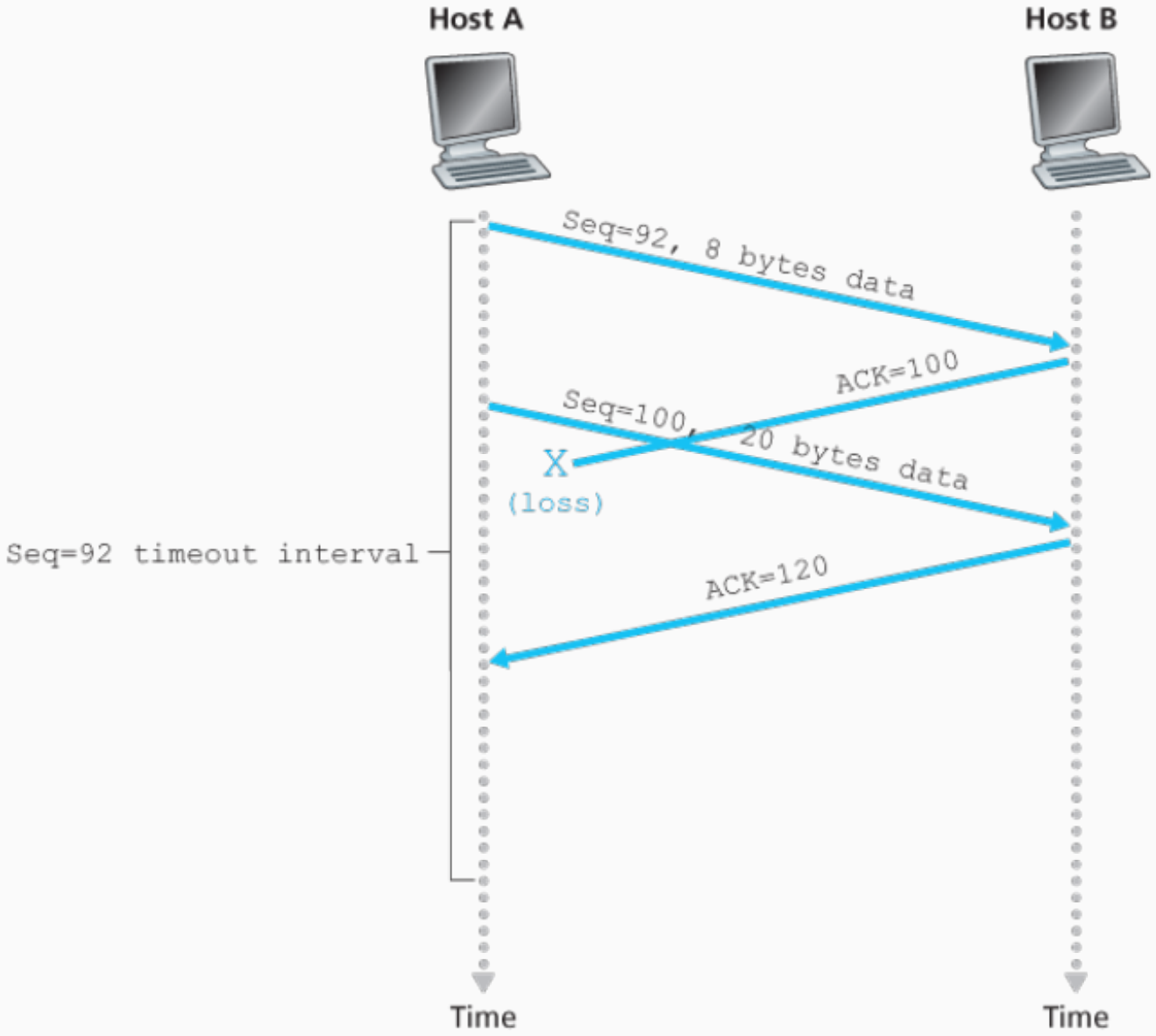


Figure 3.36 A cumulative acknowledgment avoids retransmission of the first segment

TCP Flow Control

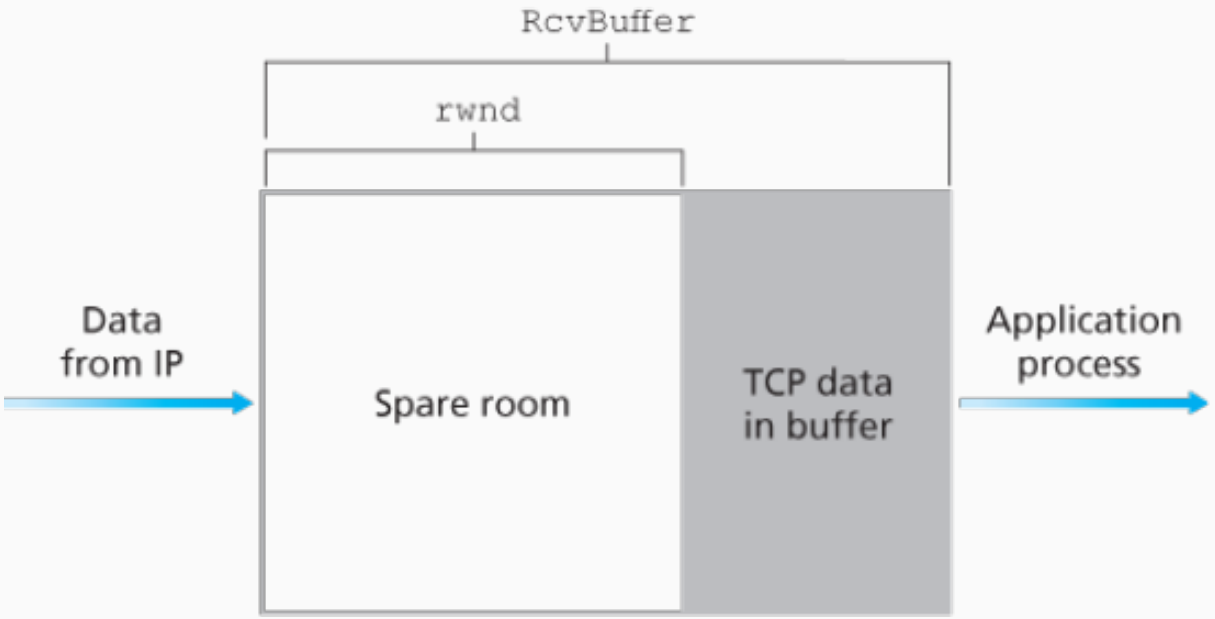


Figure 3.38 The receive window *(rwnd)* and the receive buffer *(RcvBuffer)*

TCP Congestion Control

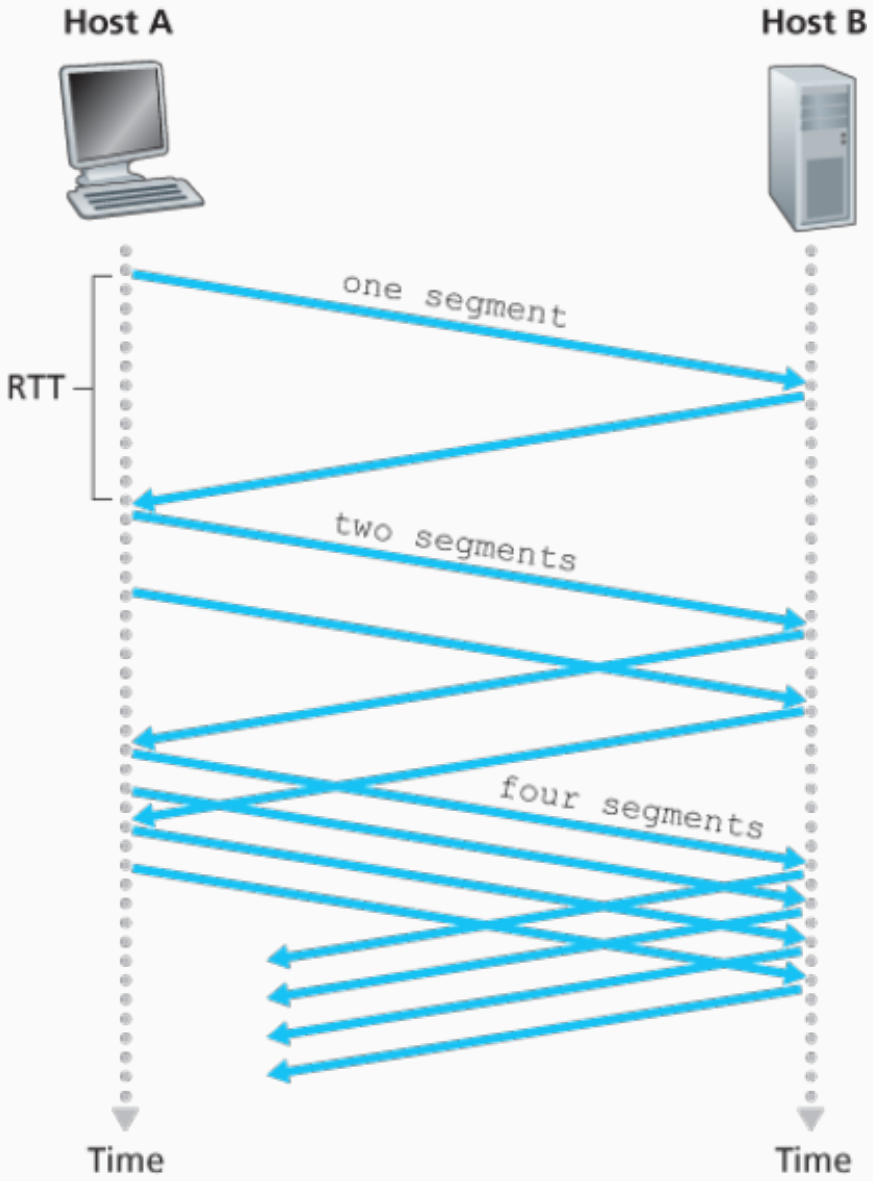


Figure 3.50 TCP slow start

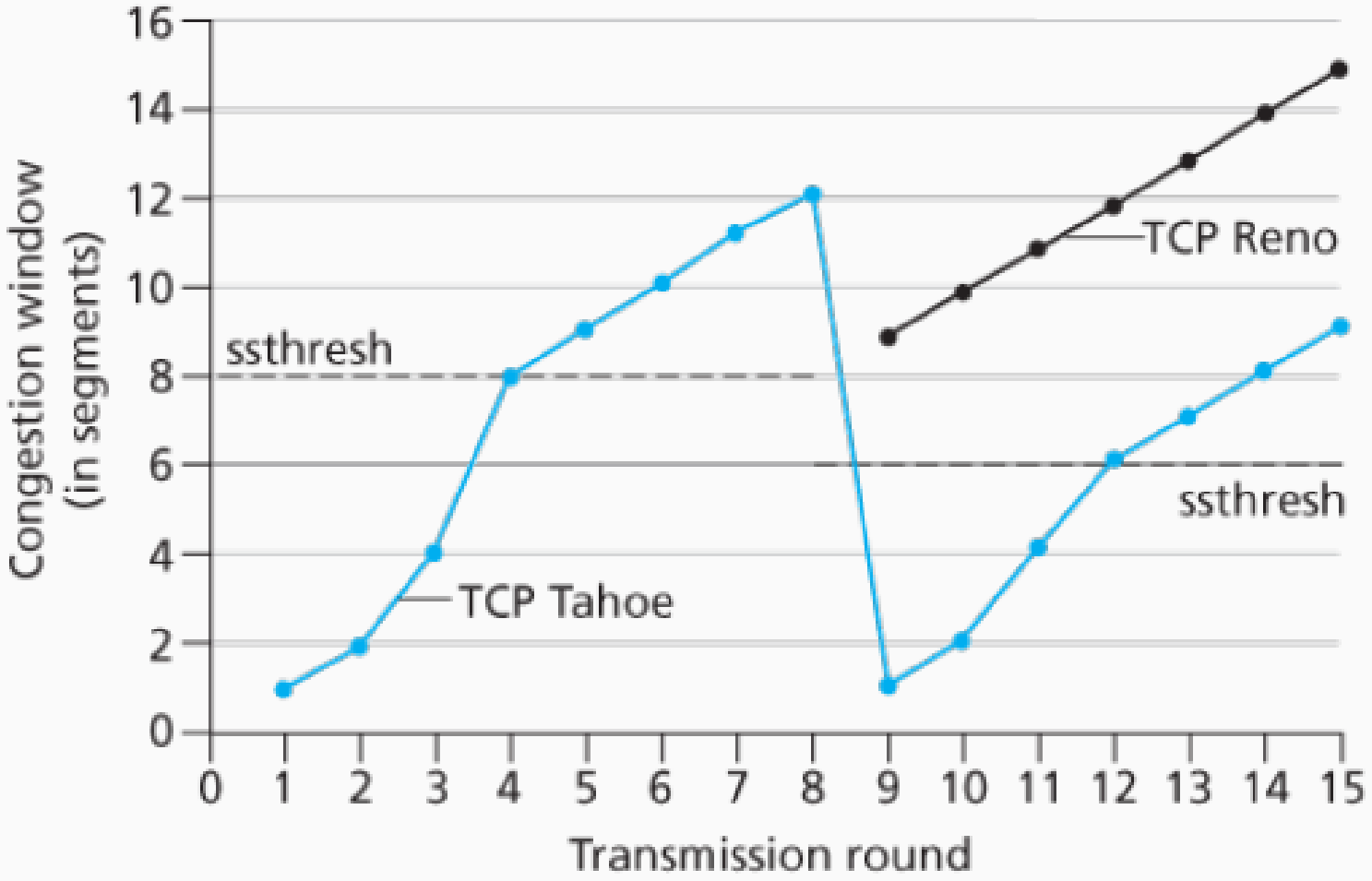


Figure 3.52 Evolution of TCP's congestion window (Tahoe and Reno)