UNIT-IV

IPv4 ADDRESSES

An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a device (for example, a computer or a router) to the Internet.

Topics discussed in this section:

Address Space
Notations
Classful Addressing
Classless Addressing
Network Address Translation (NAT)

Note

An IPv4 address is 32 bits long.

Note

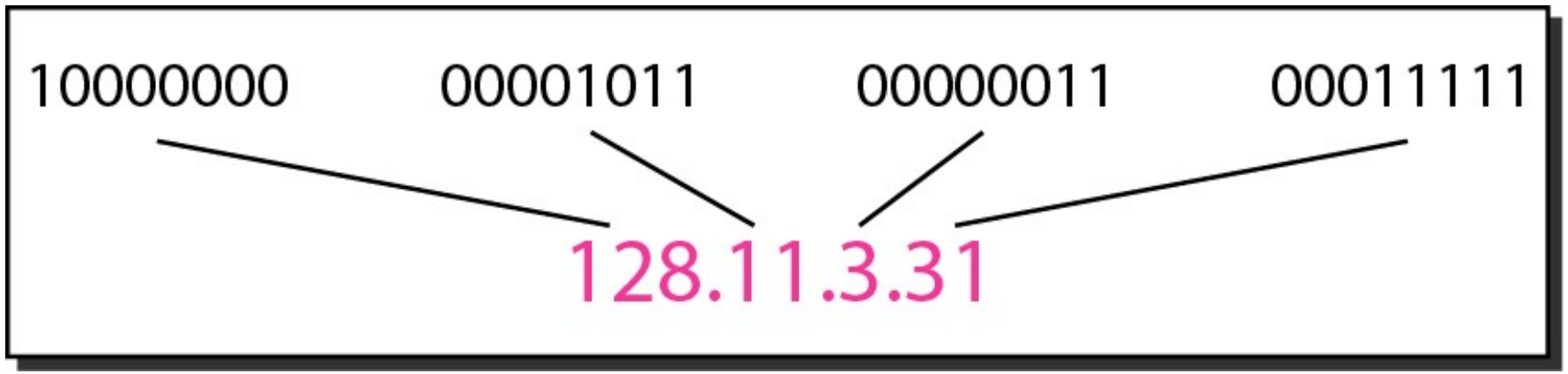
The IPv4 addresses are unique and universal.

Note

The address space of IPv4 is 2^{32} or 4,294,967,296.

Address space: is the total no of addresses used by the protocol

Figure 19.1 *Dotted-decimal notation and binary notation for an IPv4 address*



Example 19.3

Find the error, if any, in the following IPv4 addresses.

a. 111.56.045.78

b. 221.34.7.8.20

c. 75.45.301.14

d. 11100010.23.14.67

Solution

a. There must be no leading zero (045).

b. There can be no more than four numbers.

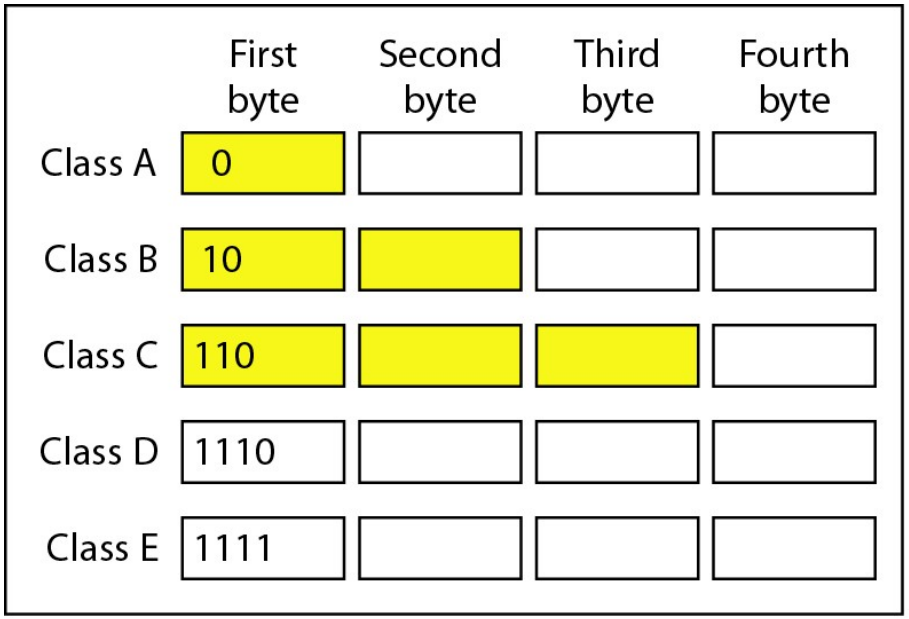
c. Each number needs to be less than or equal to 255.

d. A mixture of binary notation and dotted-decimal notation is not allowed.

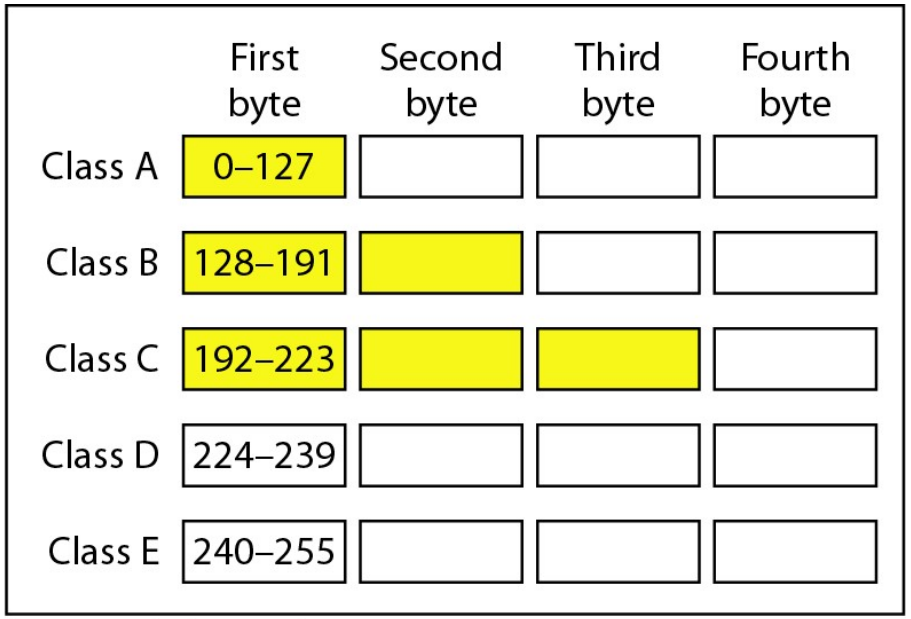
Note

In classful addressing, the address space is divided into five classes: A, B, C, D, and E.

Figure 19.2 *Finding the classes in binary and dotted-decimal notation*



a. Binary notation



b. Dotted-decimal notation

Example 19.4

Find the class of each address.

a. 00000001 00001011 00001011 11101111

b. 11000001 10000011 00011011 11111111

c. 14.23.120.8

d. 252.5.15.111

Solution

a. The first bit is 0. This is a class A address.

b. The first 2 bits are 1; the third bit is 0. This is a class C address.

c. The first byte is 14; the class is A.

d. The first byte is 252; the class is E.

Table 19.1 *Number of blocks and block size in classful IPv4 addressing*

Class	Number of Blocks	Block Size	Application
A	128	16,777,216	Unicast
B	16,384	65,536	Unicast
C	2,097,152	256	Unicast
D	1	268,435,456	Multicast
E	1	268,435,456	Reserved

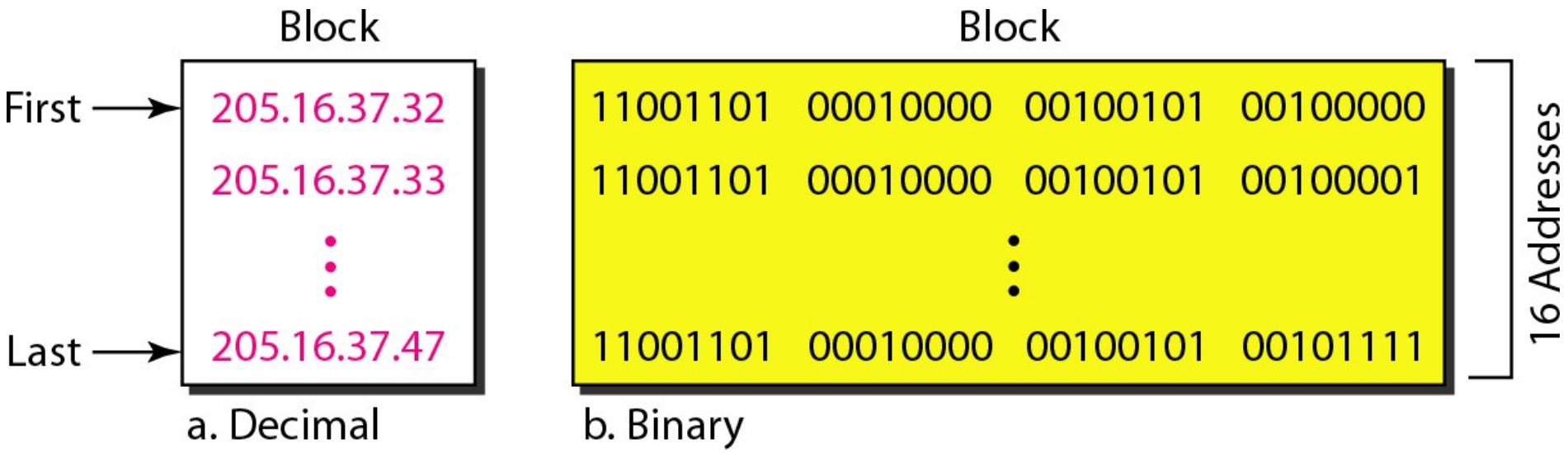
Note

In classful addressing, a large part of the available addresses were wasted.

Note

Classful addressing, which is almost obsolete, is replaced with classless addressing.

Classless Addresses : *A block of 16 addresses granted to a small organization*



Note

In IPv4 addressing, a block of addresses can be defined as x.y.z.t /*n* in which x.y.z.t defines one of the addresses and the /*n* defines the mask.

Table 19.2 *Default masks for classful addressing*

Class	Binary	Dotted-Decimal	CIDR
A	11111111 00000000 00000000 00000000	255.0.0.0	/8
B	11111111 11111111 00000000 00000000	255.255.0.0	/16
C	11111111 11111111 11111111 00000000	255.255.255.0	/24

Note

The first address in the block can be found by setting the rightmost $32 - n$ bits to 0s.

Example 19.6

A block of addresses is granted to a small organization. We know that one of the addresses is 205.16.37.39/28. What is the first address in the block?

Solution

The binary representation of the given address is

11001101 00010000 00100101 00100111

If we set 32−28 rightmost bits to 0, we get

11001101 00010000 00100101 00100000

or

205.16.37.32.

.

Note

The last address in the block can be found by setting the rightmost 32 − n bits to 1s.

Example 19.7

Find the last address for the block in Example 19.6.

Solution

The binary representation of the given address is

11001101 00010000 00100101 00100111

If we set 32 − 28 rightmost bits to 1, we get

11001101 00010000 00100101 00101111

or

205.16.37.47

This is actually the block shown in Figure 19.3.

Note

The number of addresses in the block can be found by using the formula 2^{32-n}.

Figure 19.4 *A network configuration for the block 205.16.37.32/28*

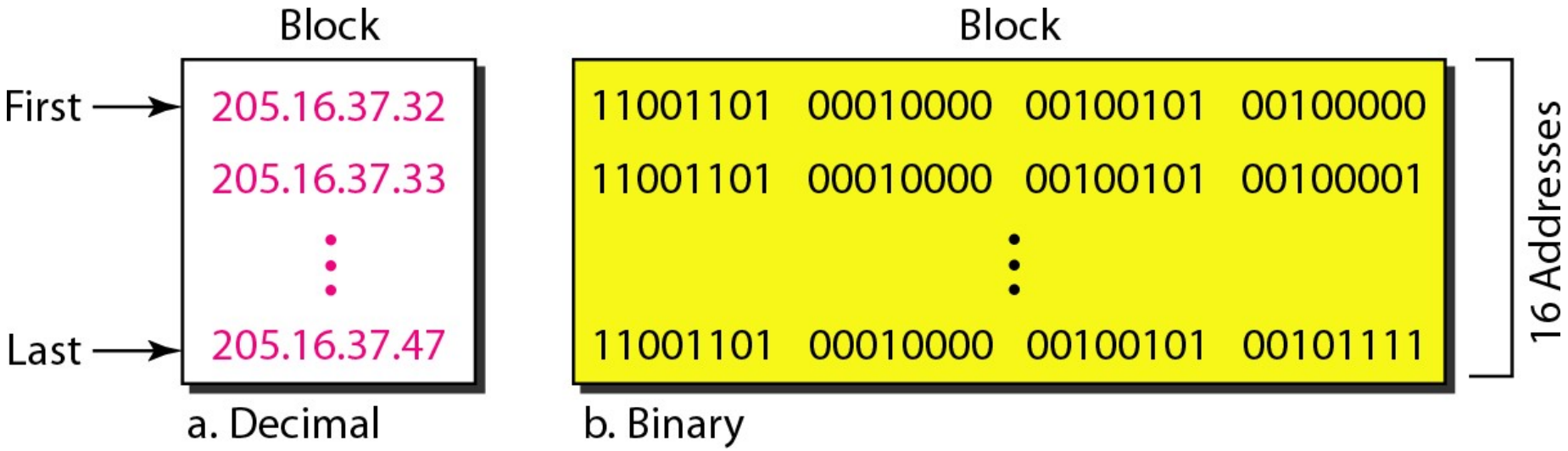
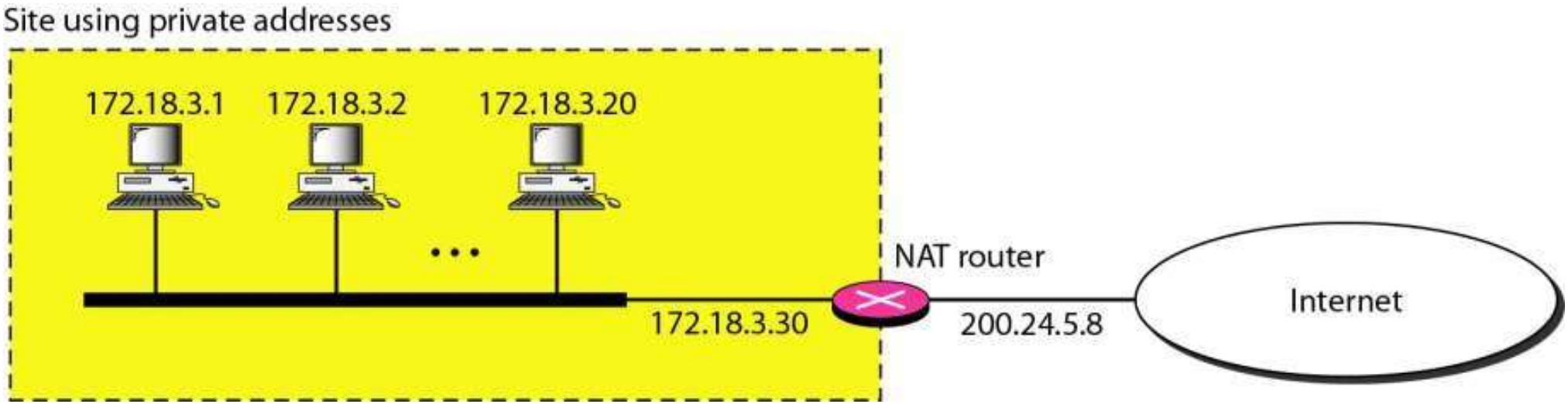


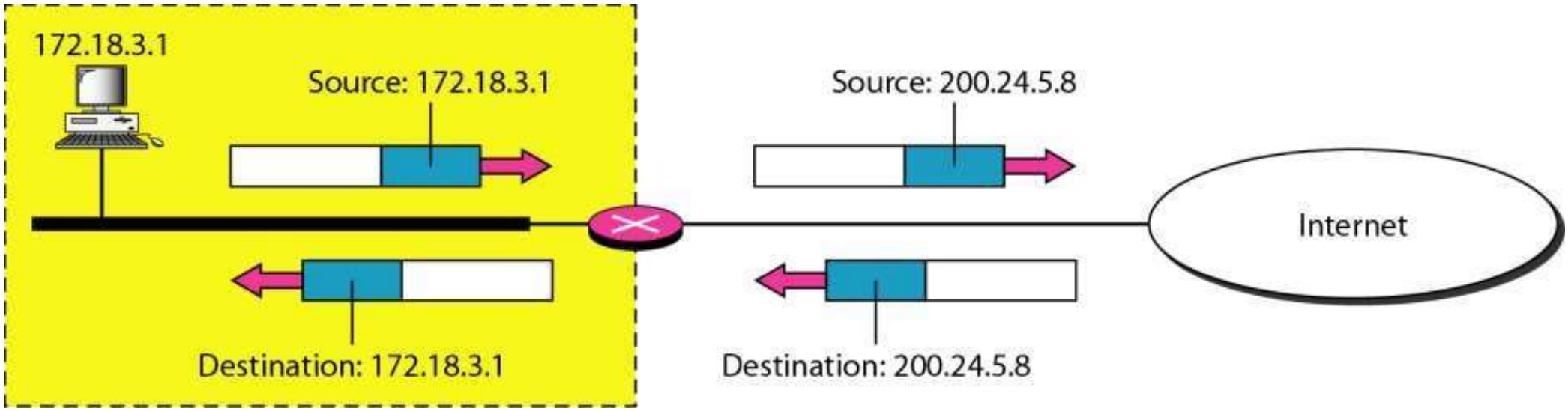
Table 19.3 *Addresses for private networks*

Range			*Total*
10.0.0.0	to	10.255.255.255	2^{24}
172.16.0.0	to	172.31.255.255	2^{20}
192.168.0.0	to	192.168.255.255	2^{16}

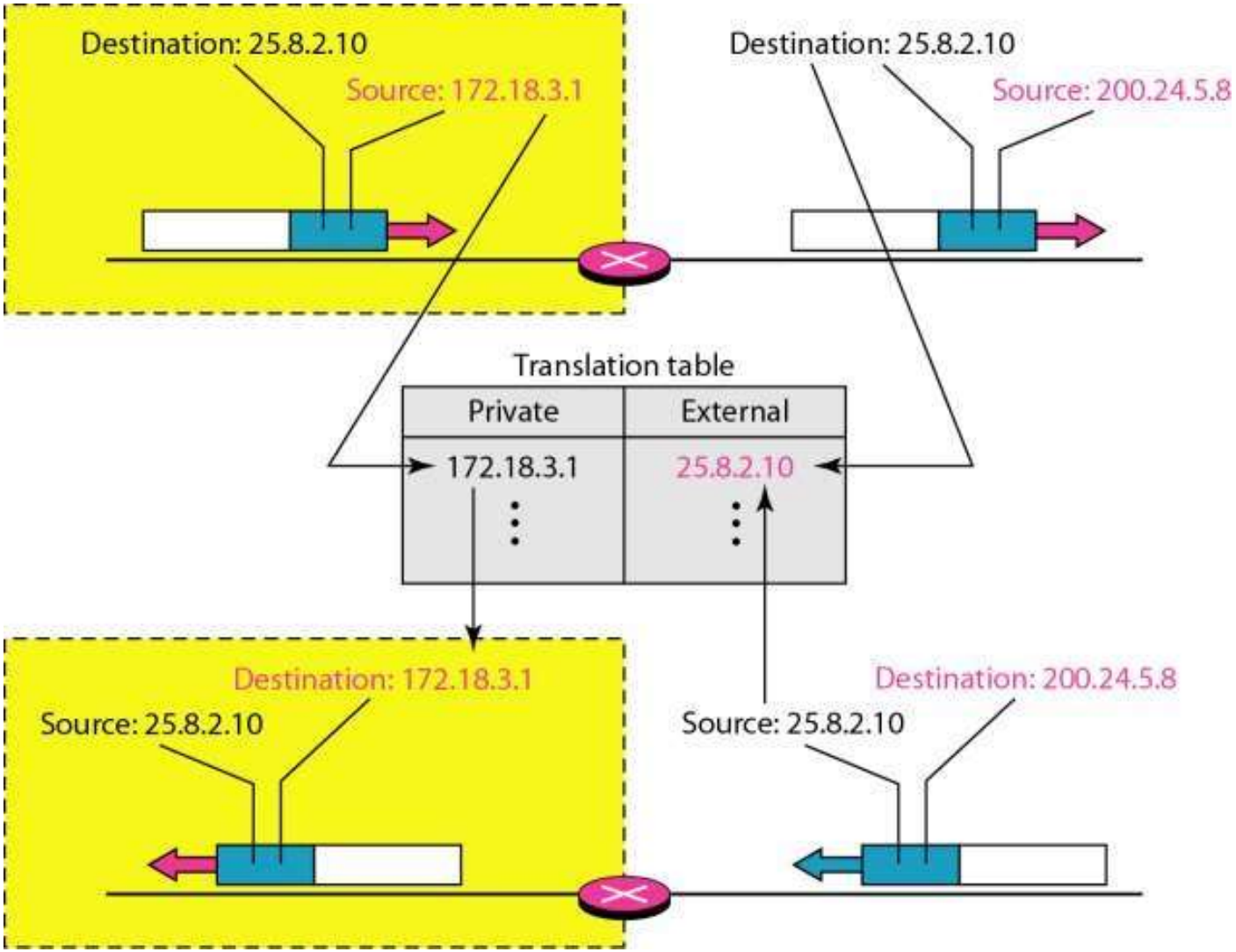
A NAT implementation



Addresses in a NAT



NAT address translation



IPv6 ADDRESSES

Despite all short-term solutions, address depletion is still a long-term problem for the Internet. This and other problems in the IP protocol itself have been the motivation for IPv6.

Topics discussed in this section:

Structure
Address Space

Note

An IPv6 address is 128 bits long.

Figure 19.14 *IPv6 address in binary and hexadecimal colon notation*

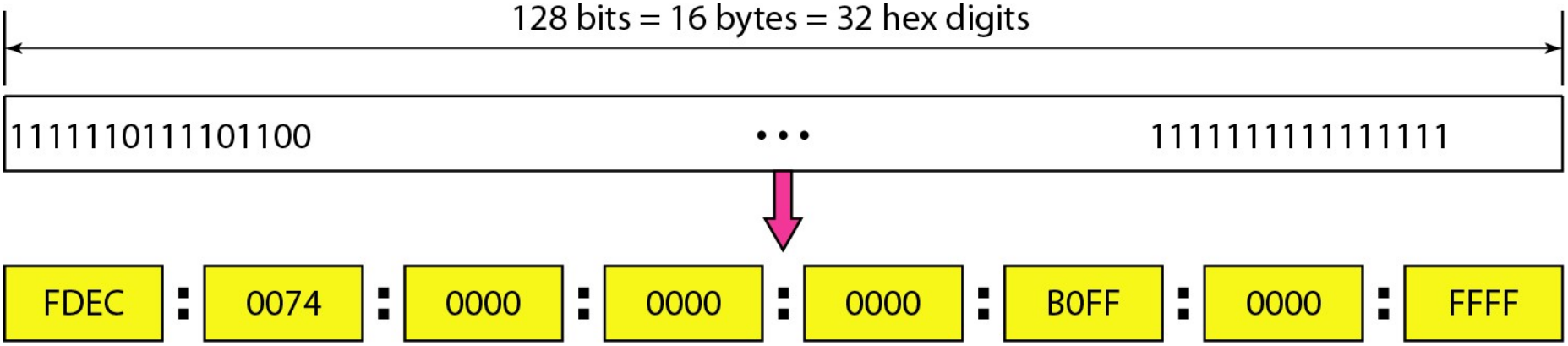
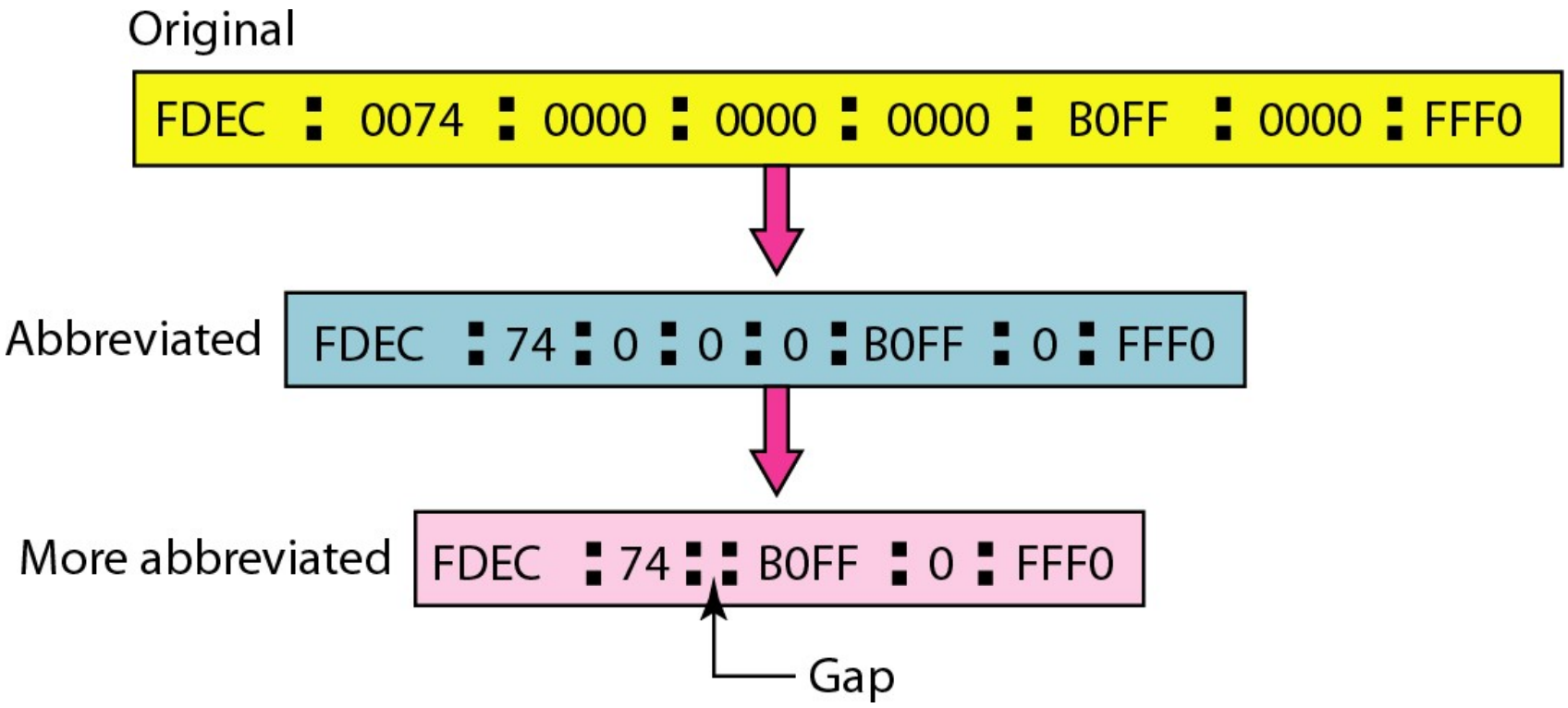
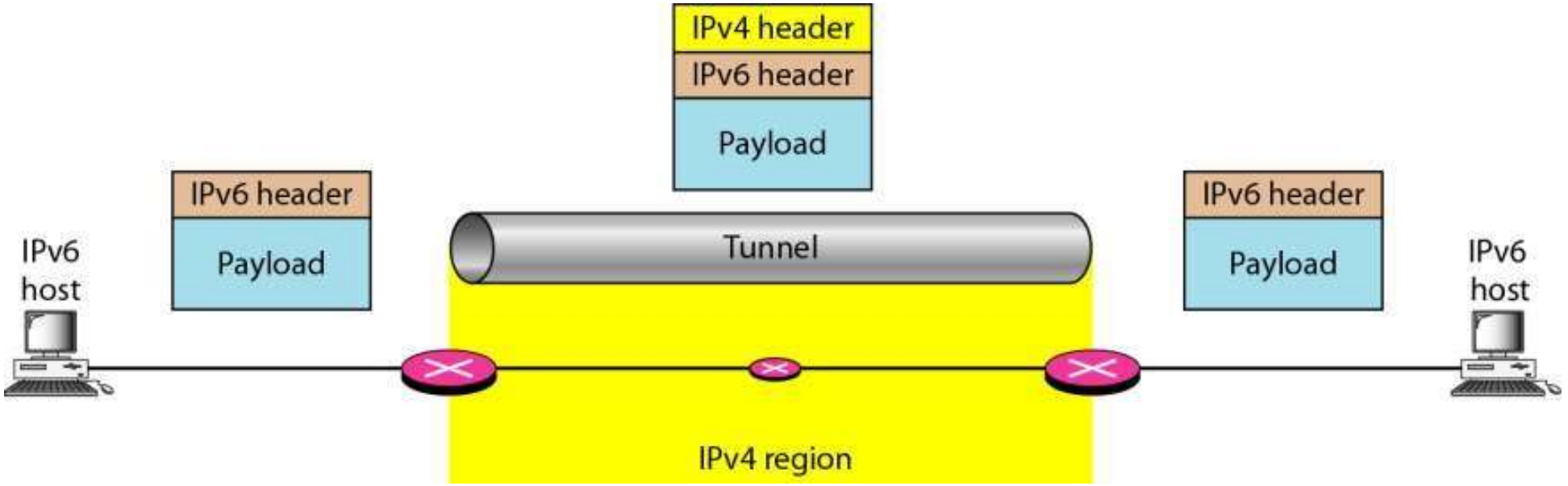


Figure 19.15 *Abbreviated IPv6 addresses*



Network Layer: Internet Protocol

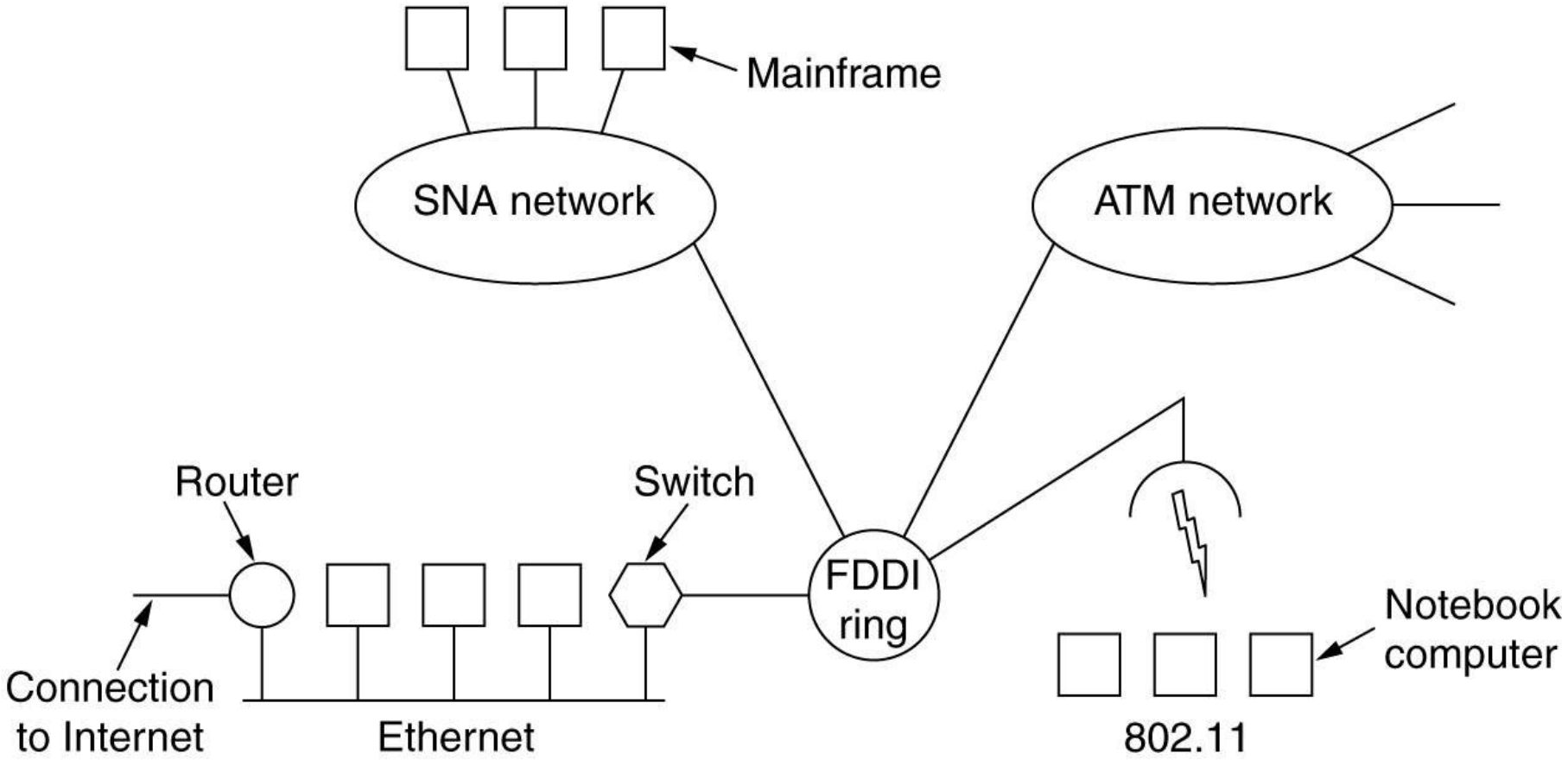
Tunneling


IPv4 header
IPv6 header
Payload
IPv6 header
Payload
Tunnel
IPv6 header
Payload
IPv6 host
IPv6 host
IPv4 region

Internetworking

- How Networks Differ
- How Networks Can Be Connected
- Concatenated Virtual Circuits
- Connectionless Internetworking
- Tunneling
- Internetwork Routing
- Fragmentation

Connecting Networks



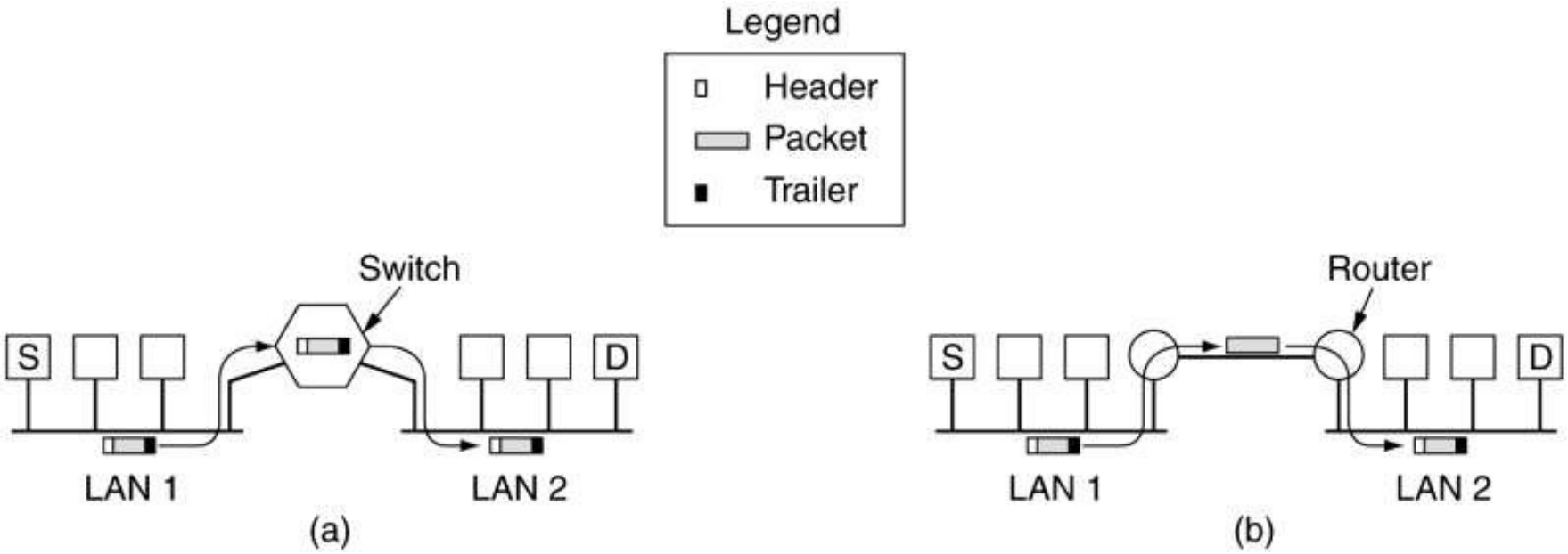
A collection of interconnected networks.

How Networks Differ

Item	Some Possibilities
Service offered	Connection oriented versus connectionless
Protocols	IP, IPX, SNA, ATM, MPLS, AppleTalk, etc.
Addressing	Flat (802) versus hierarchical (IP)
Multicasting	Present or absent (also broadcasting)
Packet size	Every network has its own maximum
Quality of service	Present or absent; many different kinds
Error handling	Reliable, ordered, and unordered delivery
Flow control	Sliding window, rate control, other, or none
Congestion control	Leaky bucket, token bucket, RED, choke packets, etc.
Security	Privacy rules, encryption, etc.
Parameters	Different timeouts, flow specifications, etc.
Accounting	By connect time, by packet, by byte, or not at all

Some of the many ways networks can differ.

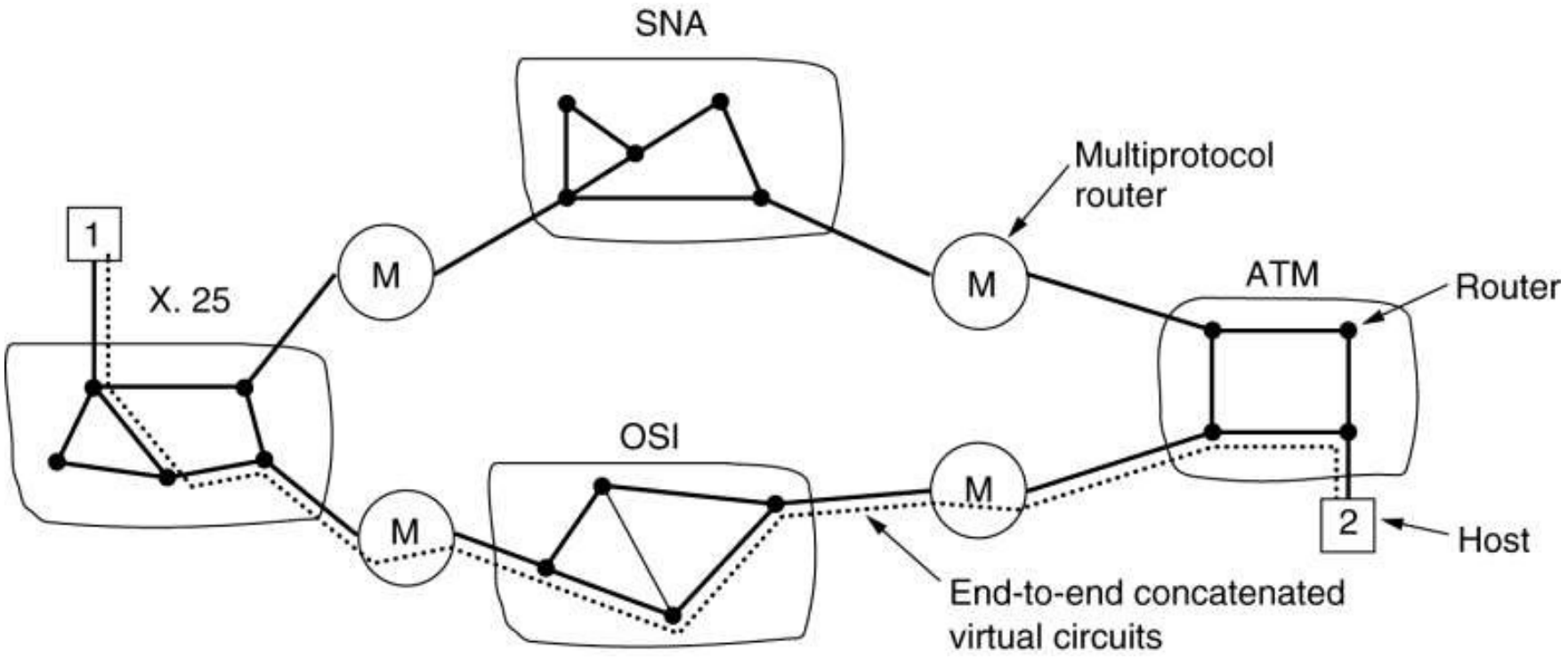
How Networks Can Be Connected



(a) Two Ethernets connected by a switch.

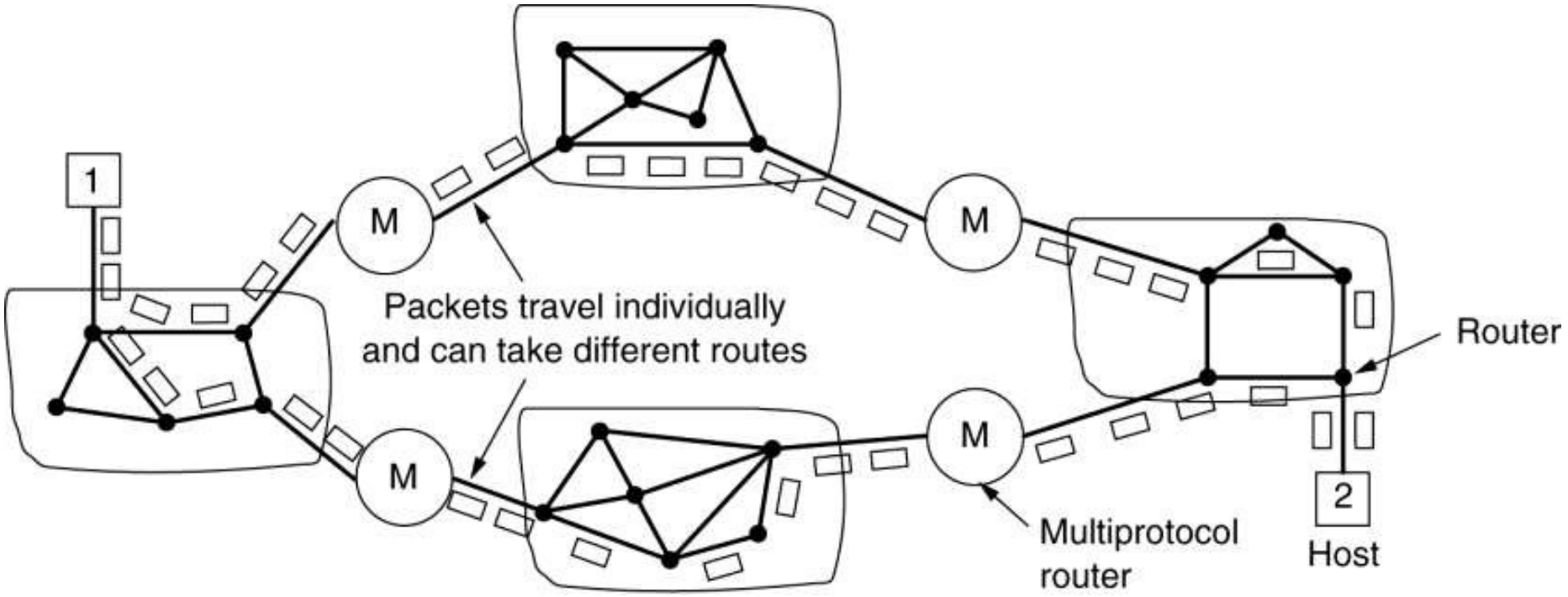
(b) Two Ethernets connected by routers.

Concatenated Virtual Circuits



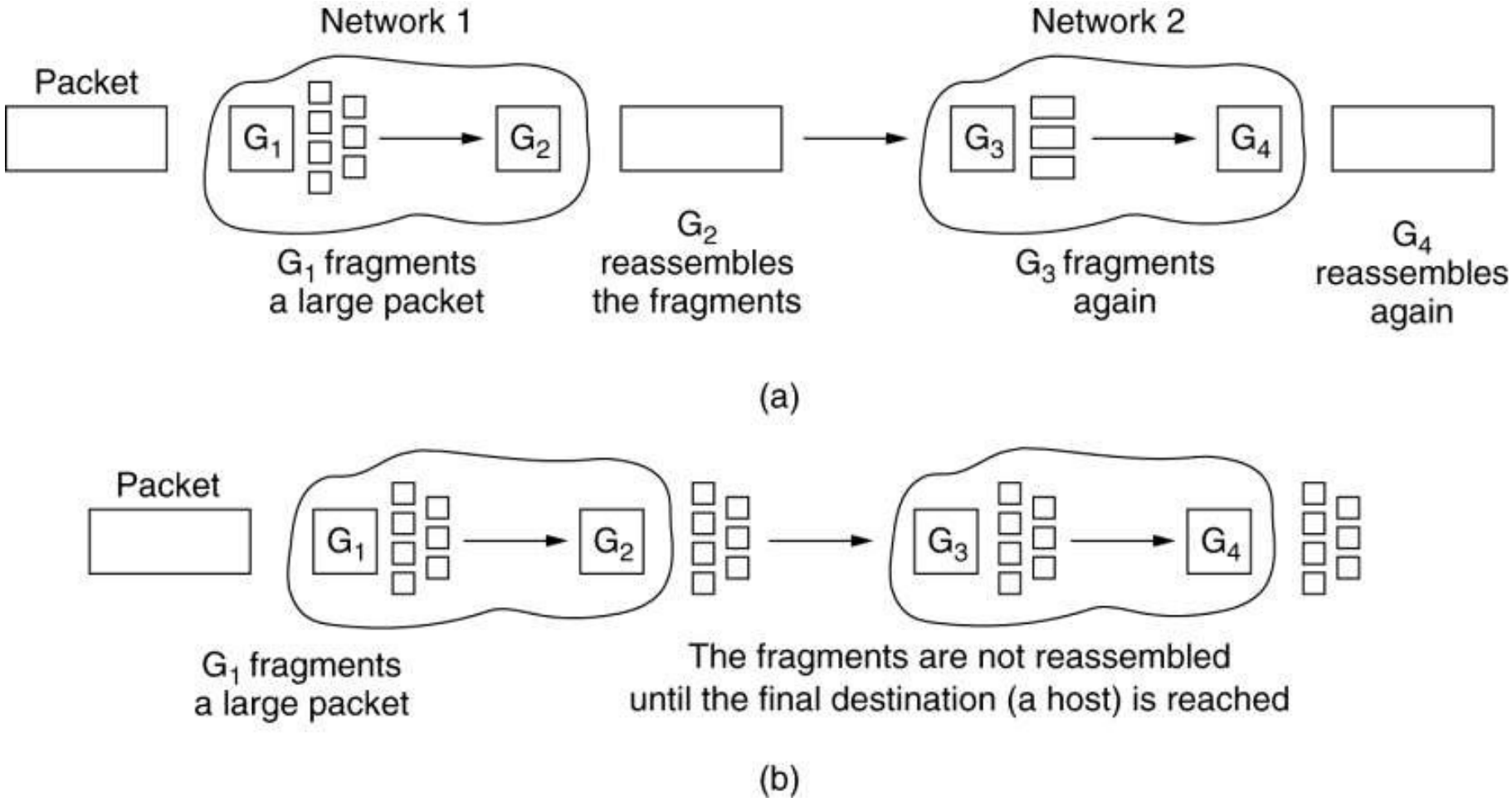
Internetworking using concatenated virtual circuits.

Connectionless Internetworking



A connectionless internet.

Fragmentation



(a) Transparent fragmentation. (b) Nontransparent fragmentation.

Network Layer: Address Mapping,

ADDRESS MAPPING

The delivery of a packet to a host or a router requires two levels of addressing: logical and physical. We need to be able to map a logical address to its corresponding physical address and vice versa. This can be done by using either static or dynamic mapping.

Topics discussed in this section:

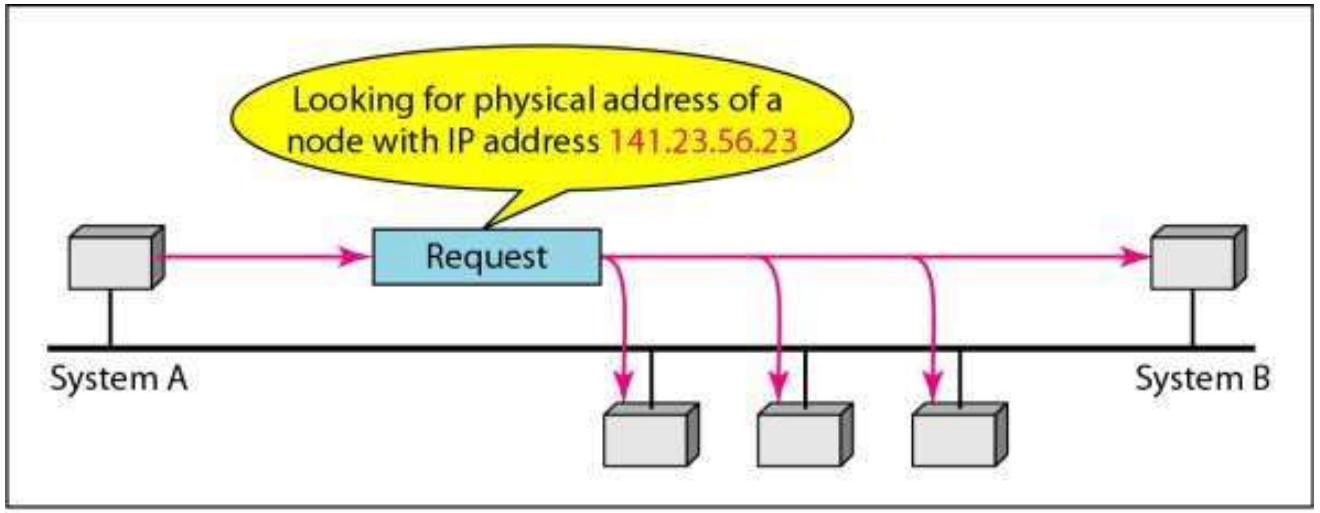
Mapping Logical to Physical Address

Mapping Physical to Logical Address

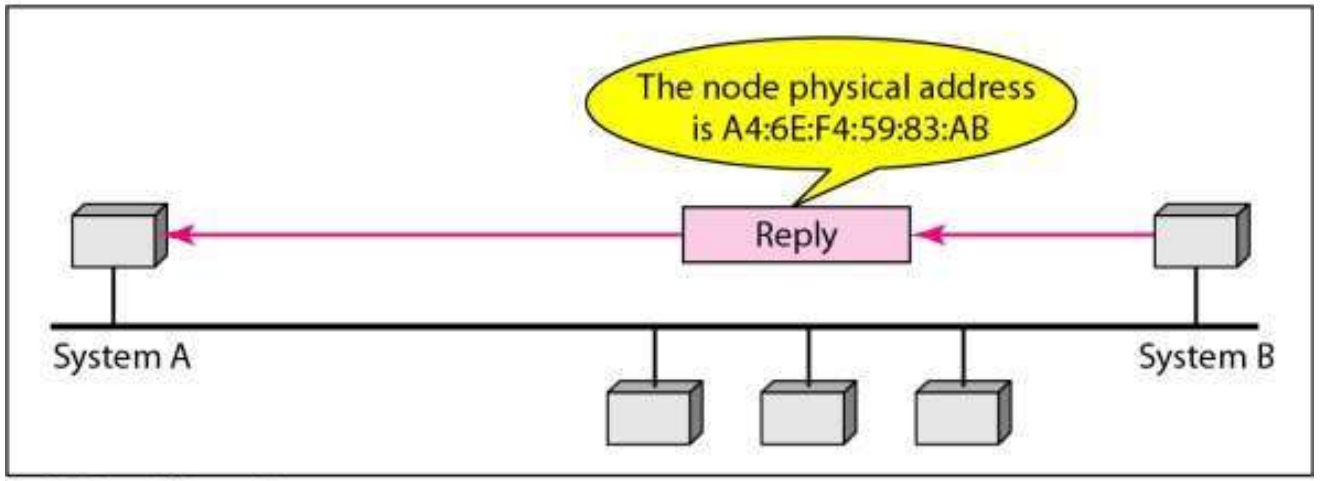
Note

An ARP request is broadcast; an ARP reply is unicast.

Figure 21.1 *ARP operation*



a. ARP request is broadcast



b. ARP reply is unicast

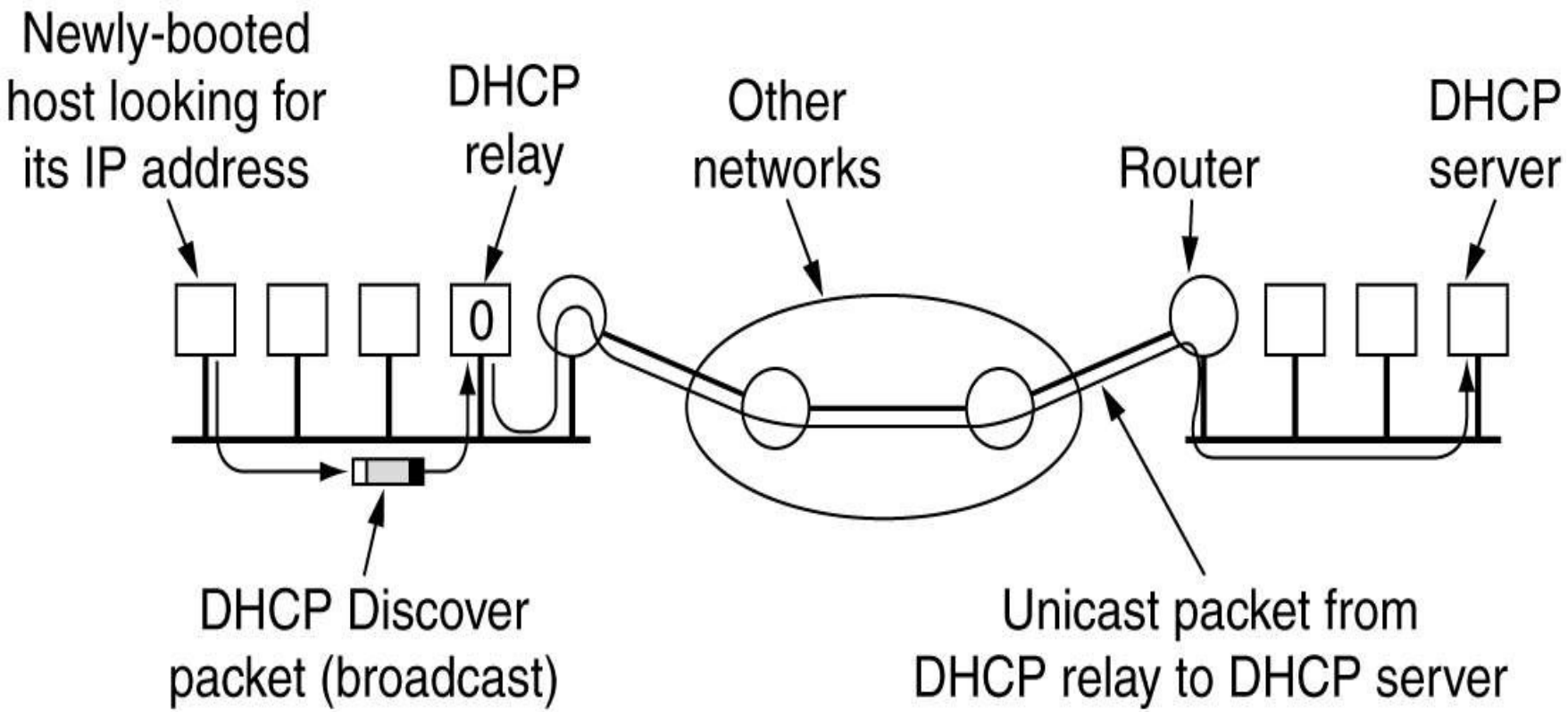
Reverse Address Resolution Protocol (RARP)

TO find logical address for a machine that knows only its physical address

Note

DHCP provides static and dynamic address allocation that can be manual or automatic.

Dynamic Host Configuration Protocol



Operation of DHCP.

ICMP

The IP protocol has no error-reporting or error-correcting mechanism. The IP protocol also lacks a mechanism for host and management queries. The Internet Control Message Protocol (ICMP) has been designed to compensate for the above two deficiencies. It is a companion to the IP protocol.

Topics discussed in this section:

Types of Messages
Message Format
Error Reporting and Query
Debugging Tools

Internet Control Message Protocol

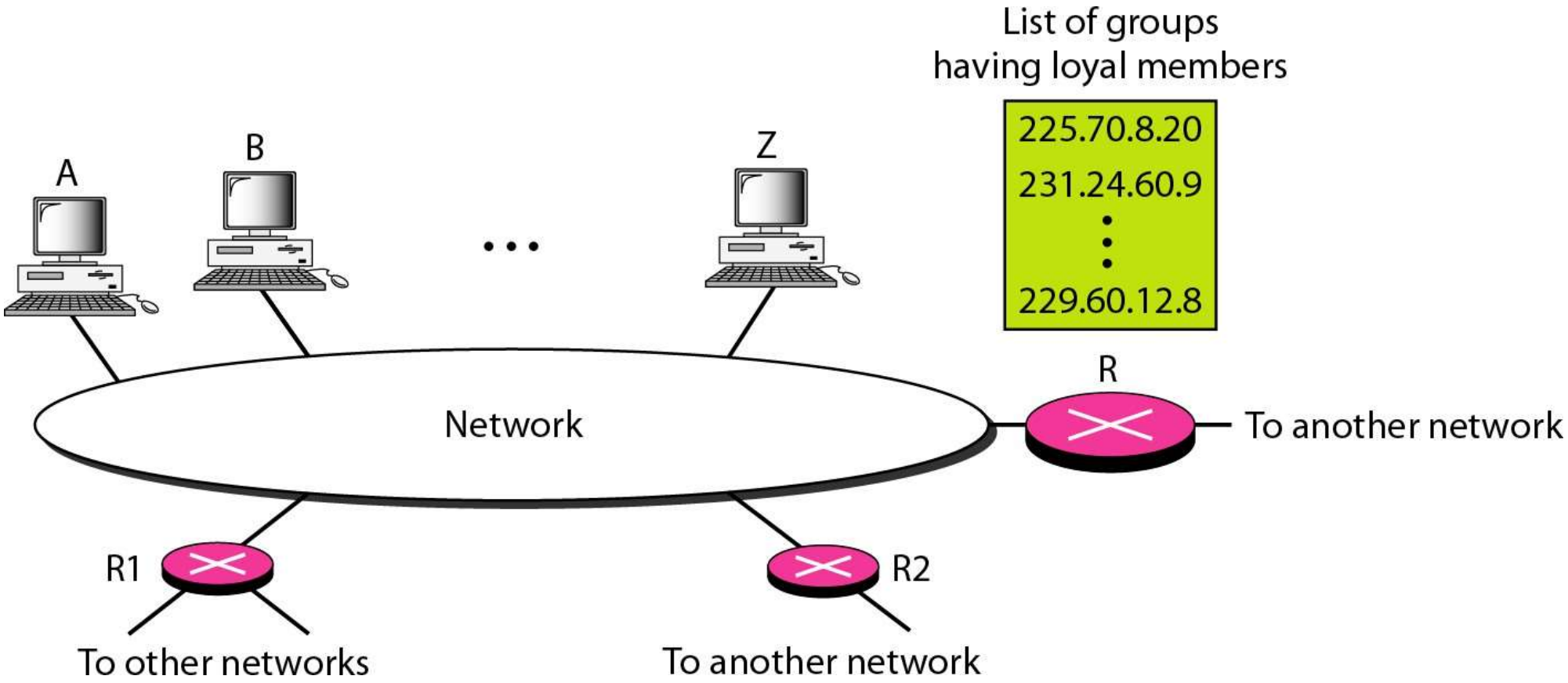
Message type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo request	Ask a machine if it is alive
Echo reply	Yes, I am alive
Timestamp request	Same as Echo request, but with timestamp
Timestamp reply	Same as Echo reply, but with timestamp

The principal ICMP message types.

IGMP

The IP protocol can be involved in two types of communication: unicasting and multicasting. The Internet Group Management Protocol (IGMP) is one of the necessary, but not sufficient, protocols that is involved in multicasting. IGMP is a companion to the IP protocol.

IGMP operation

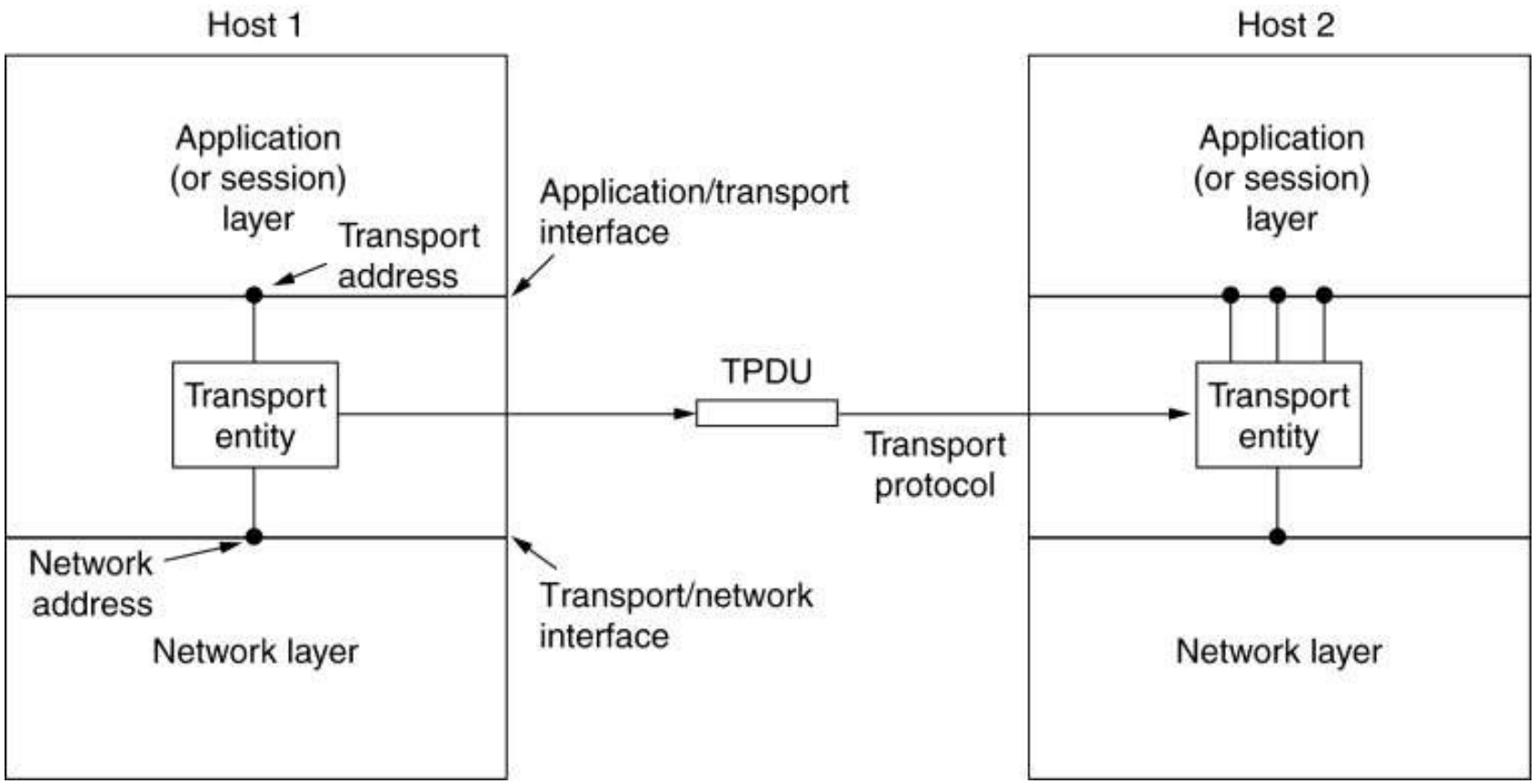


The Transport Layer

The Transport Service

- Services Provided to the Upper Layers
- Transport Service Primitives

Services Provided to the Upper Layers



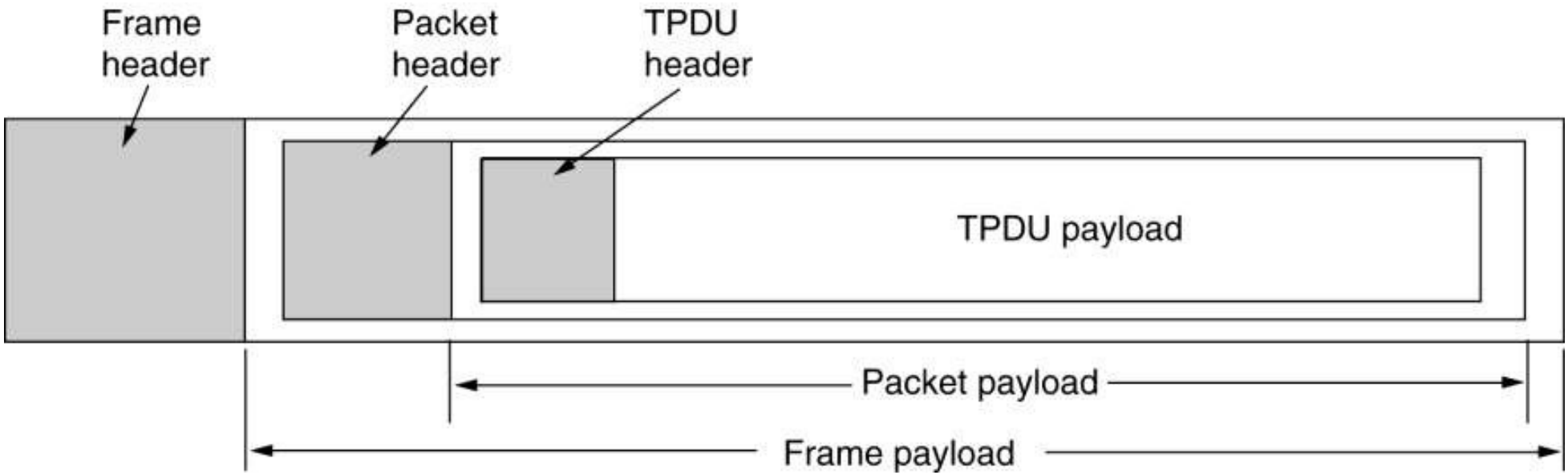
The network, transport, and application layers.

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

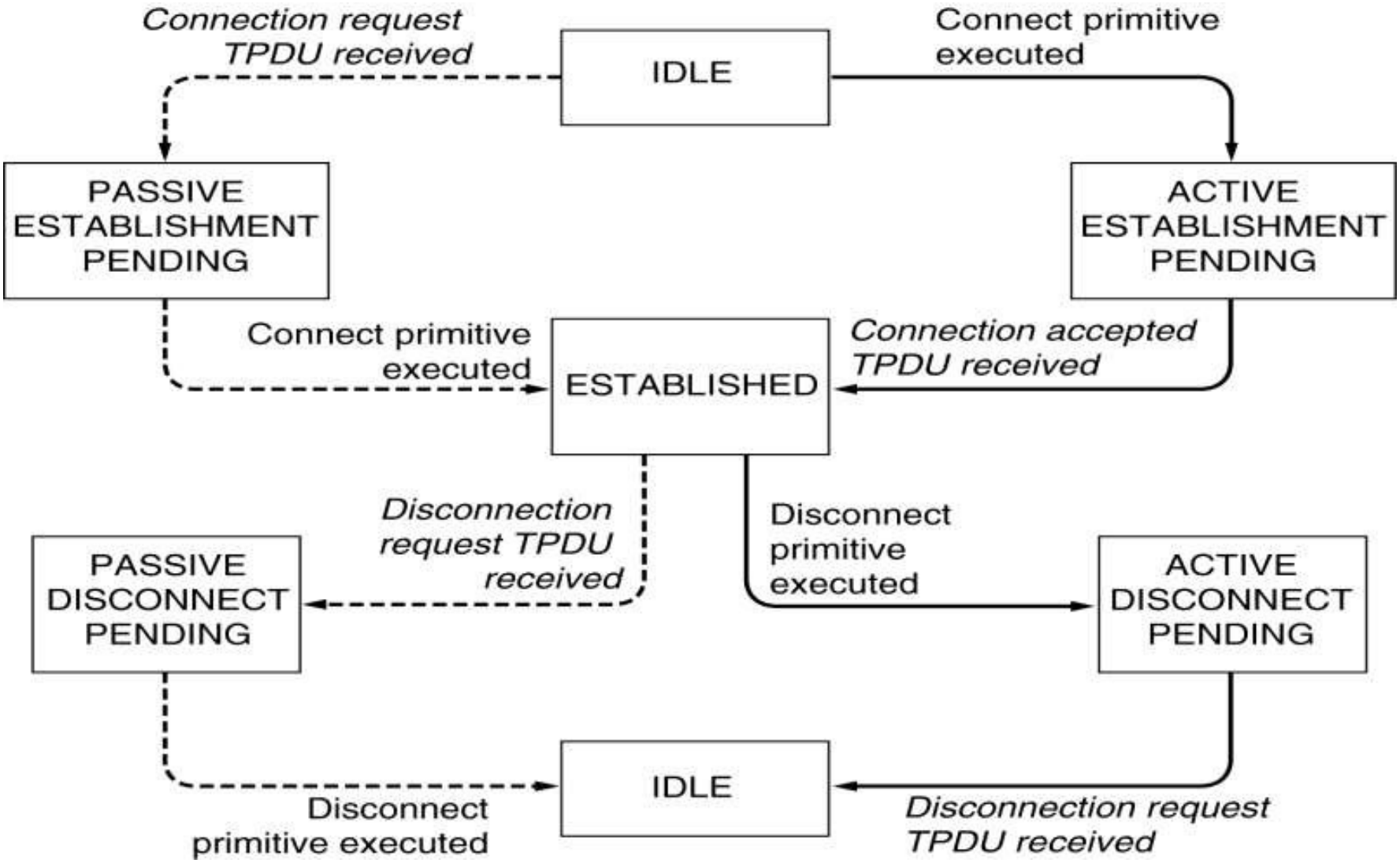
The primitives for a simple transport service.

Transport Service Primitives (2)



The nesting of TPDUs, packets, and frames.

Transport Service Primitives (3)

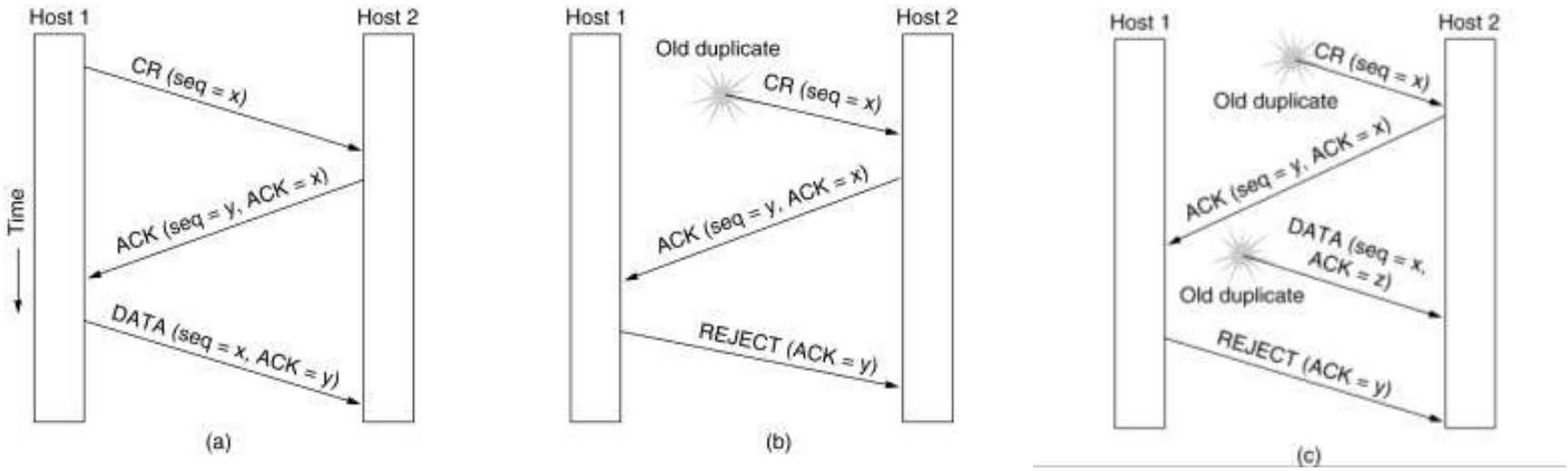


A state diagram for a simple connection management scheme. Transitions labeled in italics are caused by packet arrivals. The **solid lines** show the **client's state sequence.** The **dashed lines** show the **server's state sequence.**

Elements of Transport Protocols

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

Connection Establishment



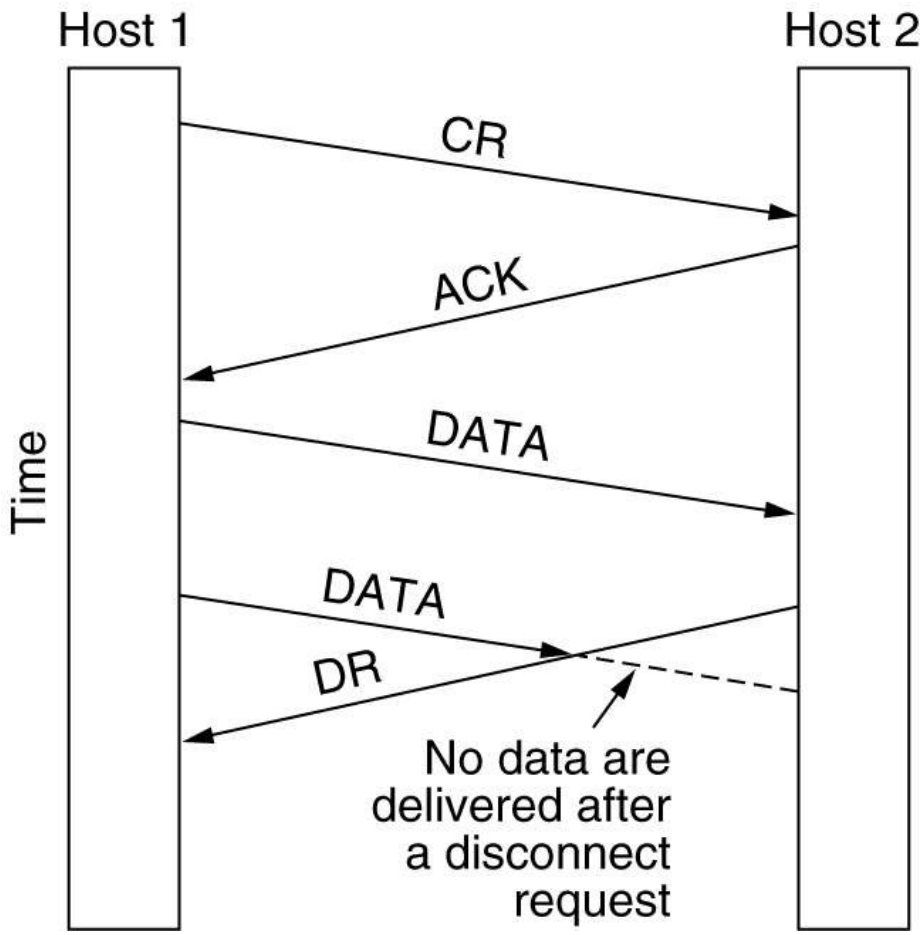
Three protocol scenarios for establishing a connection using a three-way handshake. CR denotes CONNECTION REQUEST.

(a) Normal operation,

(b) Old CONNECTION REQUEST appearing out of nowhere.

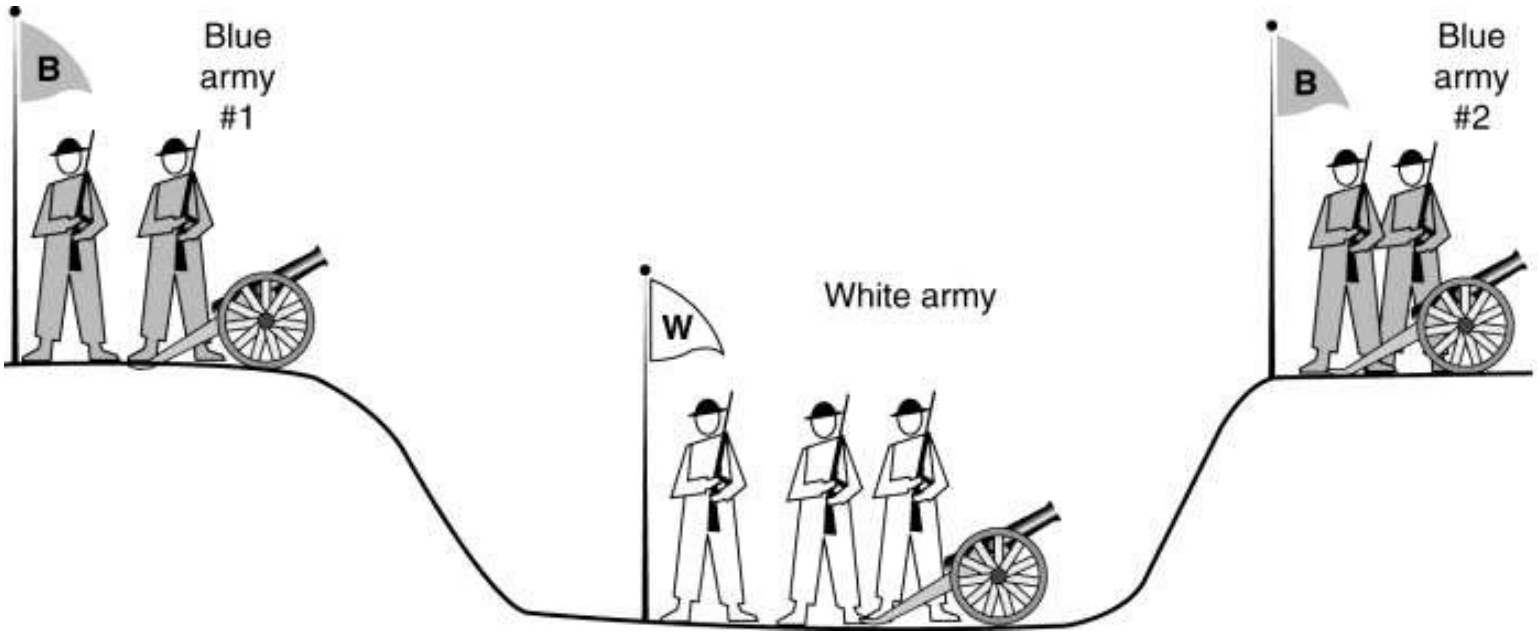
(c) Duplicate CONNECTION REQUEST and duplicate ACK.

Connection Release



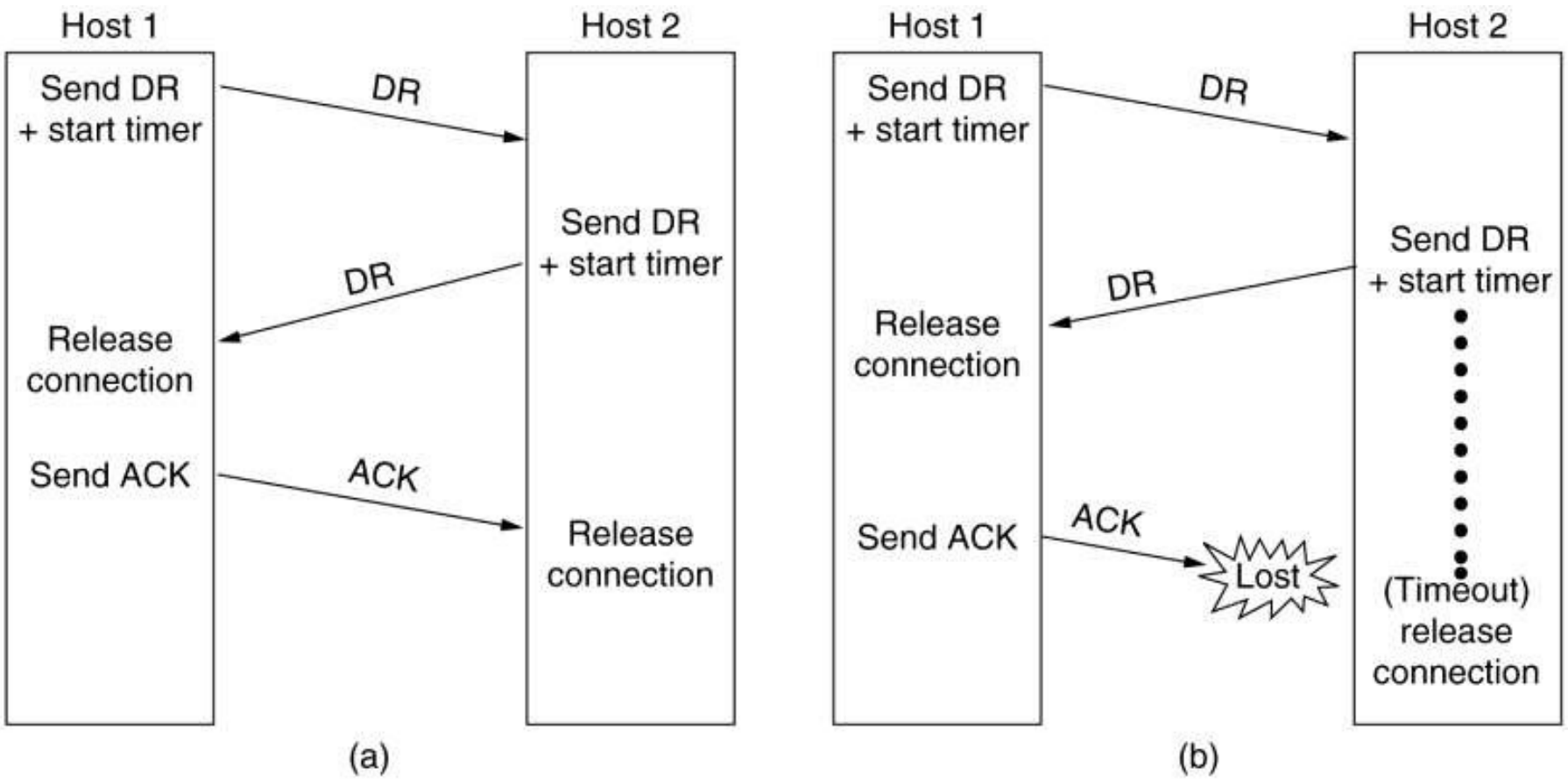
Abrupt disconnection with loss of data.

Connection Release (2)



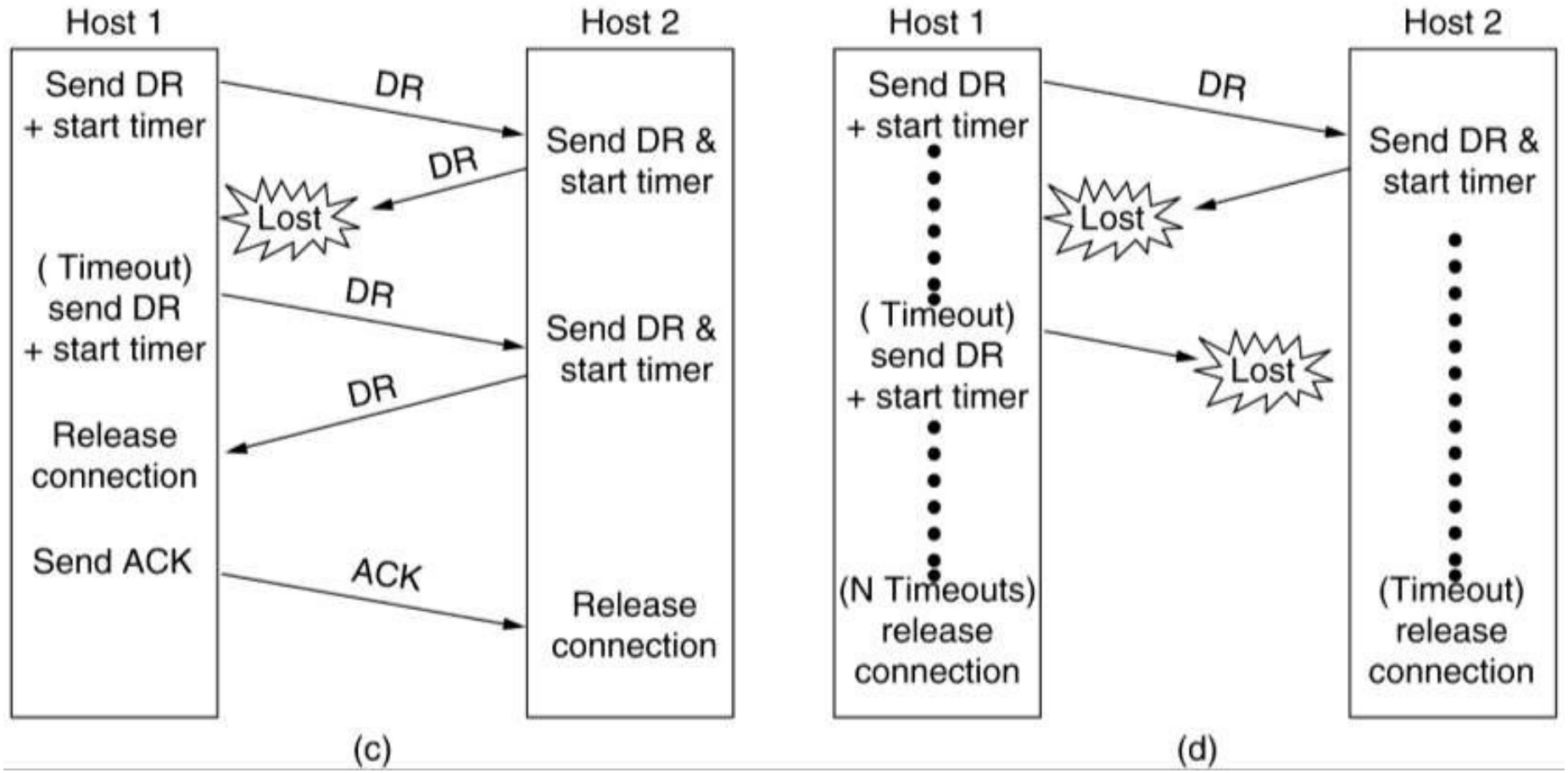
The two-army problem.

Connection Release (3)



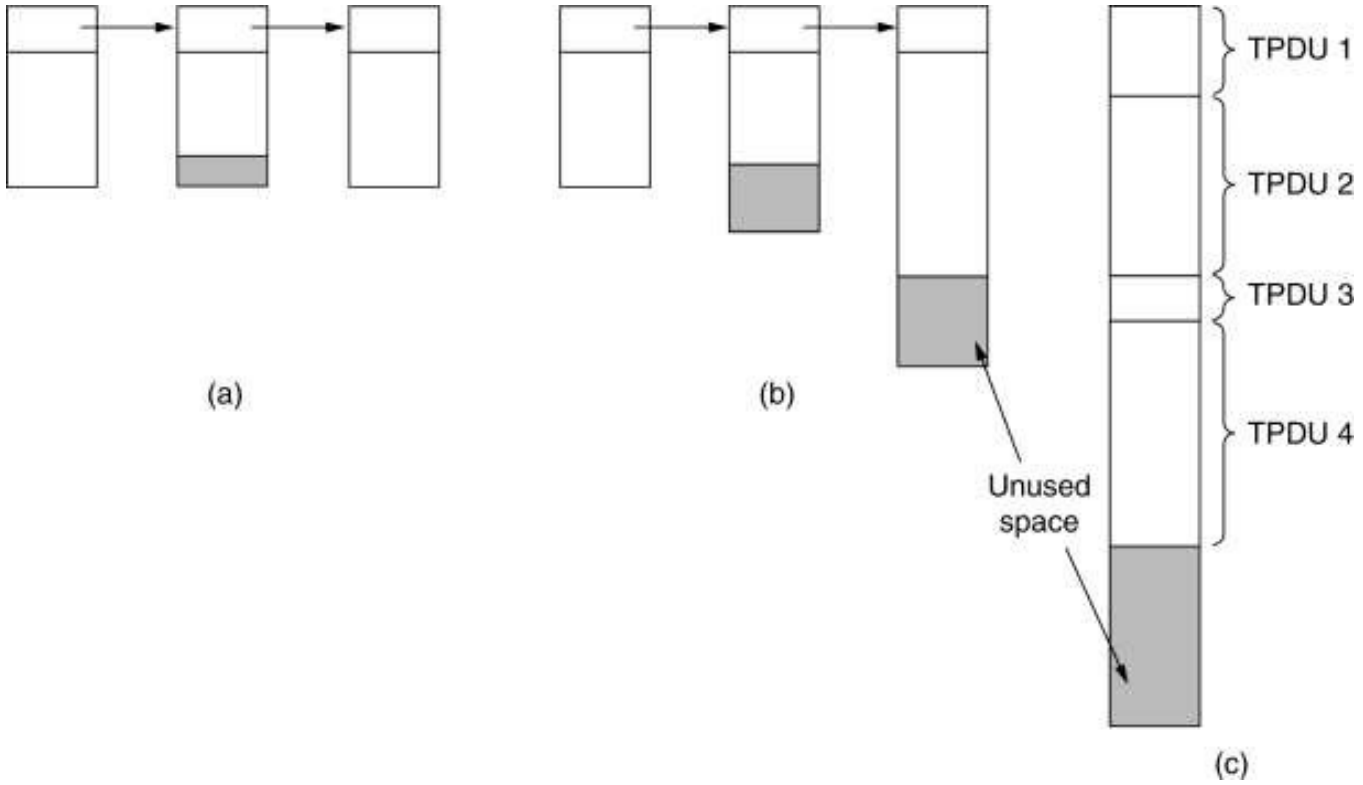
Four protocol scenarios for releasing a connection. (a) Normal case of a three-way handshake. (b) final ACK lost.

Connection Release (4)



(c) Response lost. (d) Response lost and subsequent DRs lost.

Flow Control and Buffering



(a) Chained fixed-size buffers. (b) Chained variable-sized buffers.
(c) One large circular buffer per connection.

Crash Recovery

Strategy used by sending host	Strategy used by receiving host: First ACK, then write			Strategy used by receiving host: First write, then ACK		
	AC(W)	AWC	C(AW)	C(WA)	W AC	WC(A)
Always retransmit	OK	DUP	OK	OK	DUP	DUP
Never retransmit	LOST	OK	LOST	LOST	OK	OK
Retransmit in S0	OK	DUP	LOST	LOST	DUP	OK
Retransmit in S1	LOST	OK	OK	OK	OK	DUP

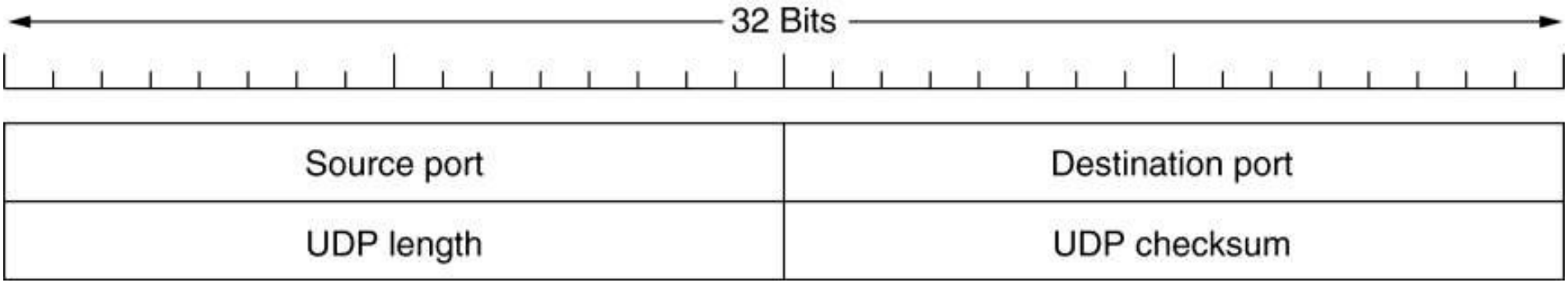
OK = Protocol functions correctly
DUP = Protocol generates a duplicate message
LOST = Protocol loses a message

Different combinations of client and server strategy.

The Internet Transport Protocols: UDP

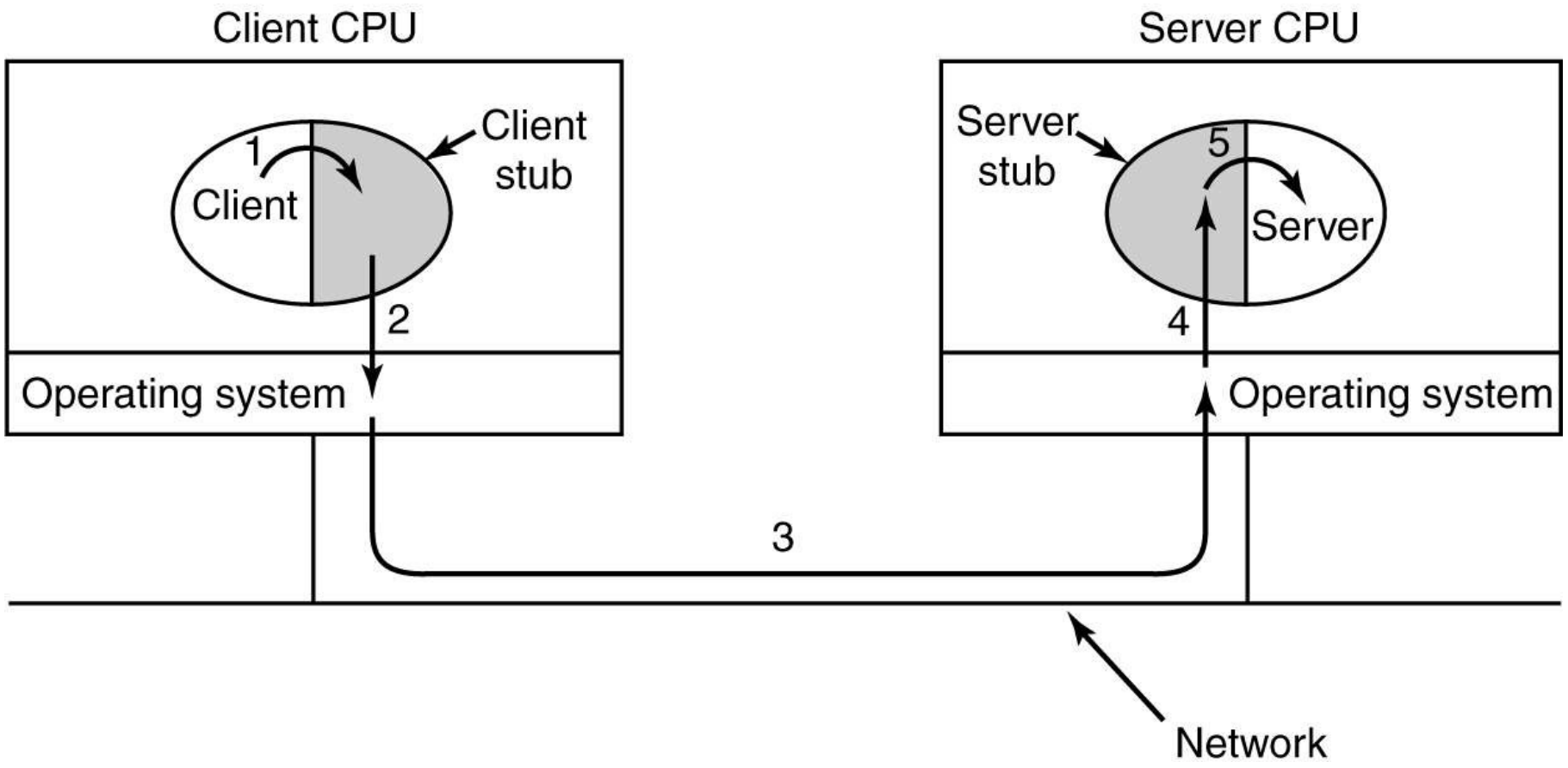
- Introduction to UDP
- Remote Procedure Call
- The Real-Time Transport Protocol

Introduction to UDP



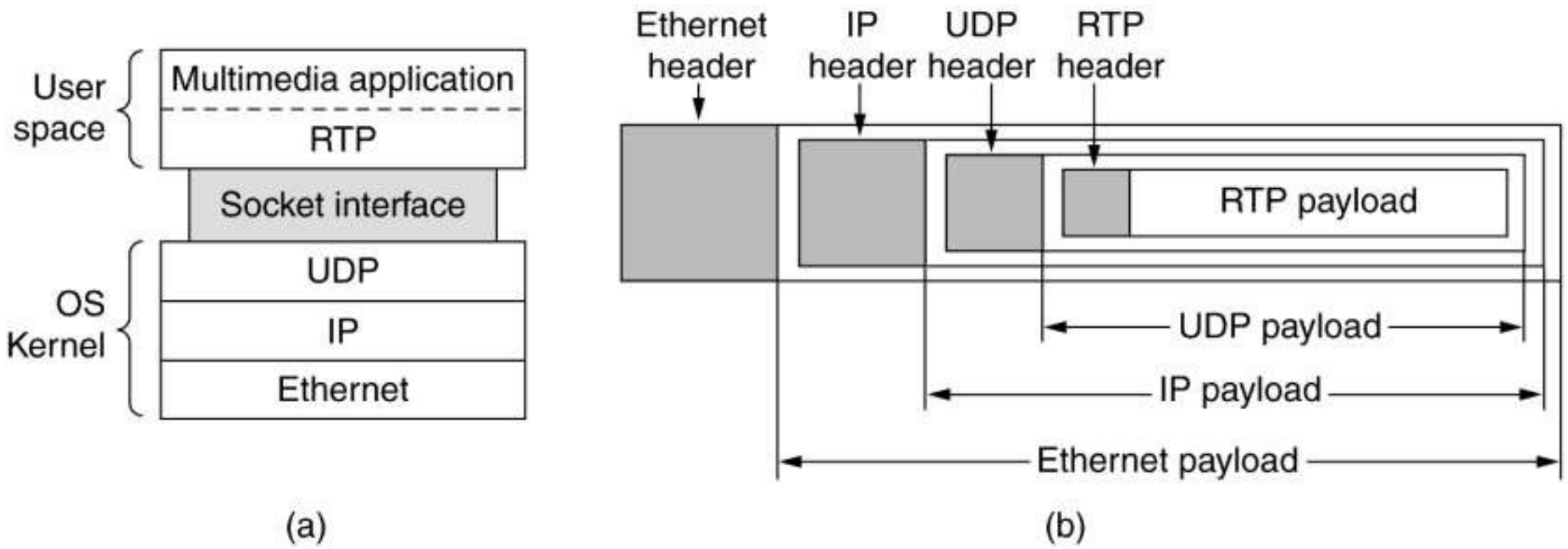
The UDP header.

Remote Procedure Call



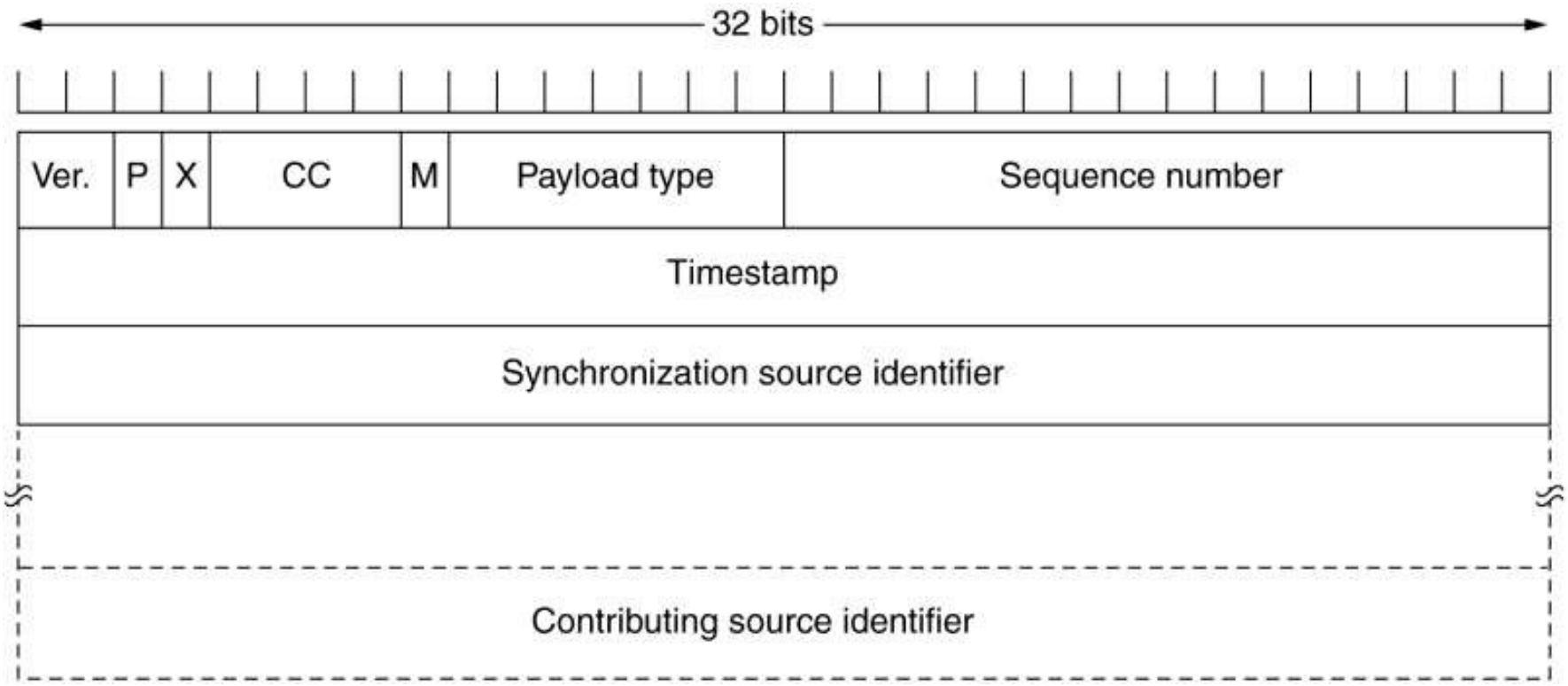
Steps in making a remote procedure call. The stubs are shaded.

The Real-Time Transport Protocol



(a) The position of RTP in the protocol stack. (b) Packet nesting.

The Real-Time Transport Protocol (2)



The RTP header.

The Internet Transport Protocols: TCP

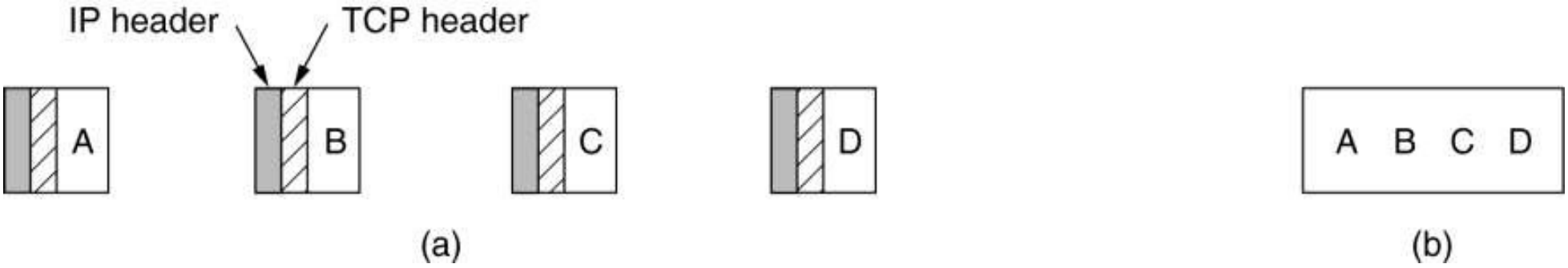
- Introduction to TCP
- The TCP Service Model
- The TCP Protocol
- The TCP Segment Header
- TCP Connection Establishment
- TCP Connection Release
- TCP Connection Management Modeling
- TCP Transmission Policy
- TCP Congestion Control
- TCP Timer Management
- Wireless TCP and UDP
- Transactional TCP

The TCP Service Model

Port	Protocol	Use
21	FTP	File transfer
23	Telnet	Remote login
25	SMTP	E-mail
69	TFTP	Trivial File Transfer Protocol
79	Finger	Lookup info about a user
80	HTTP	World Wide Web
110	POP-3	Remote e-mail access
119	NNTP	USENET news

Some assigned ports.

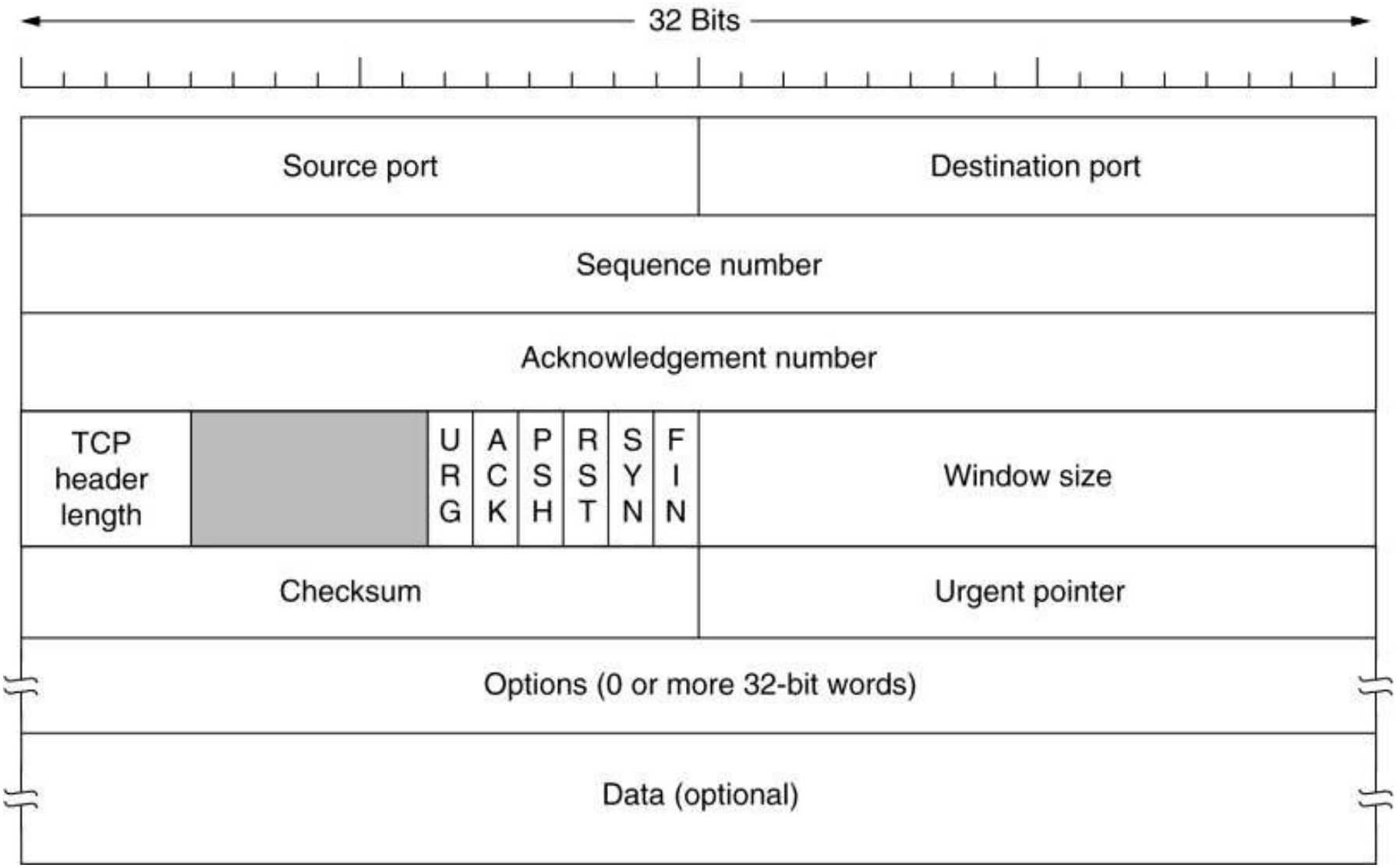
The TCP Service Model (2)



(a) Four 512-byte segments sent as separate IP datagrams.

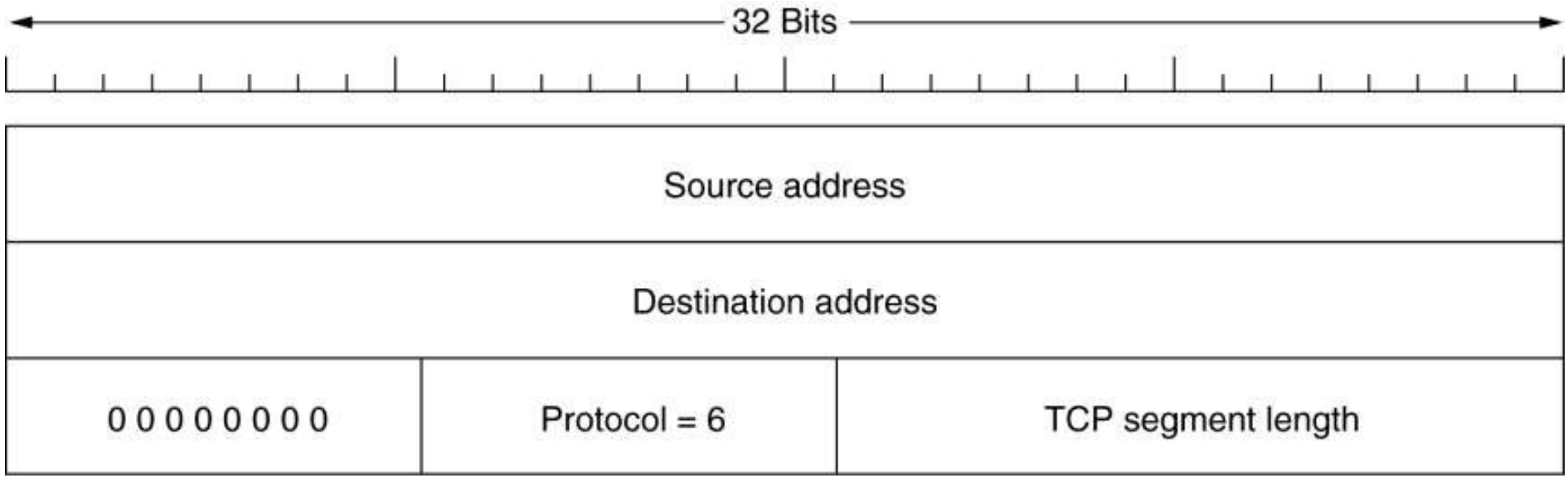
(b) The 2048 bytes of data delivered to the application in a single READ CALL.

The TCP Segment Header



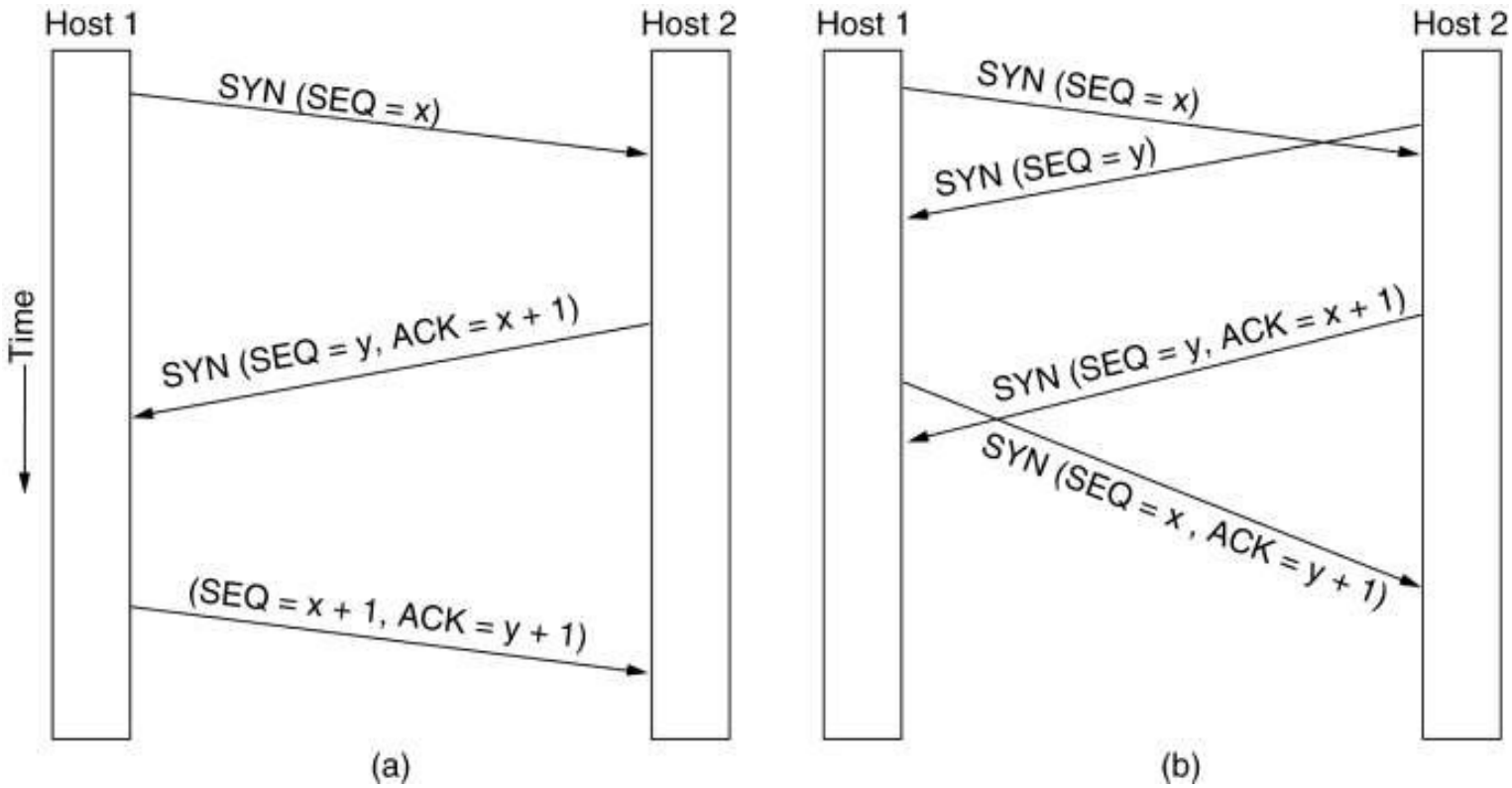
TCP Header.

The TCP Segment Header (2)



The pseudoheader included in the TCP checksum.

TCP Connection Establishment



(a) TCP connection establishment in the normal case.

(b) Call collision.

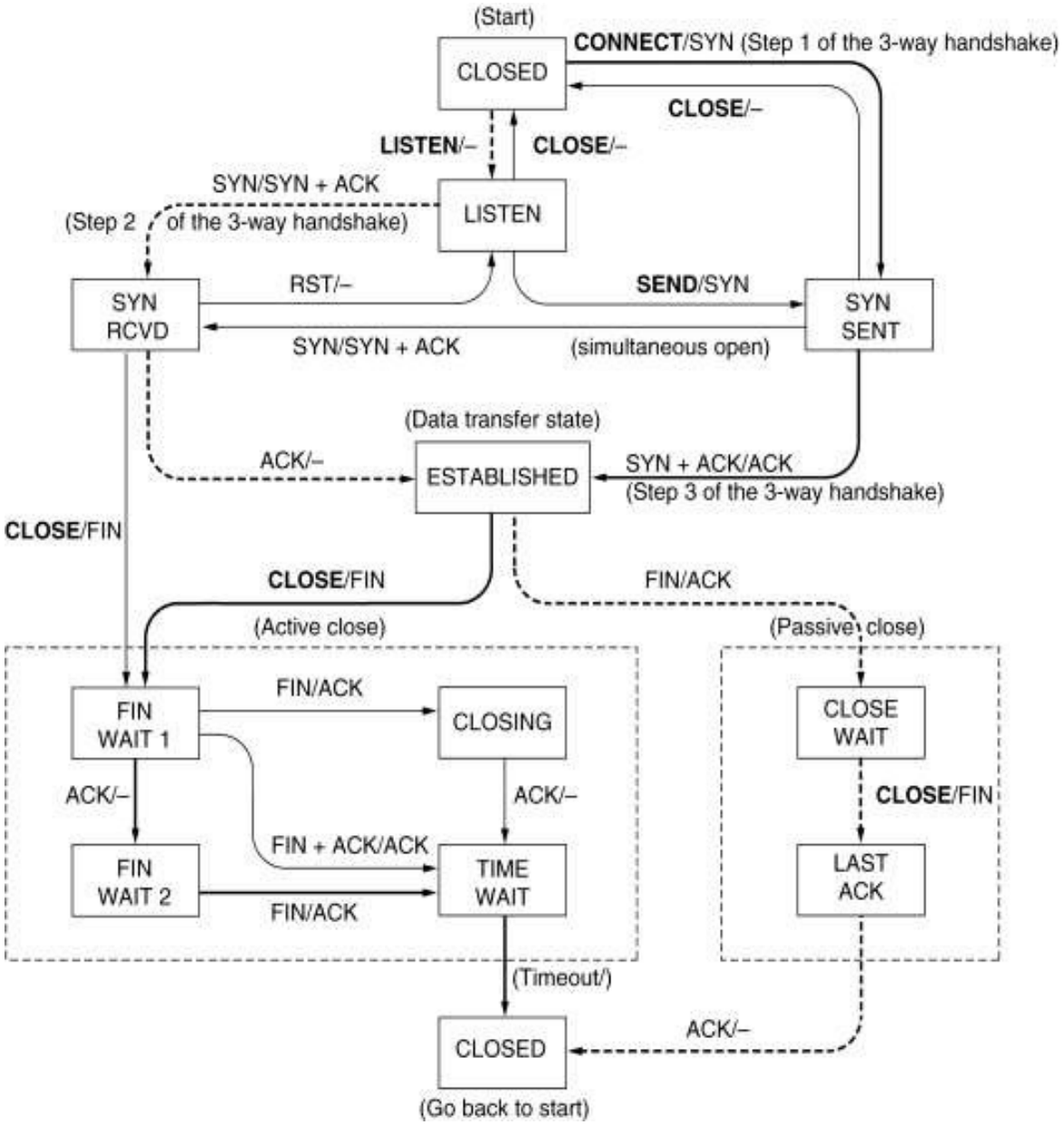
TCP Connection Management Modeling

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

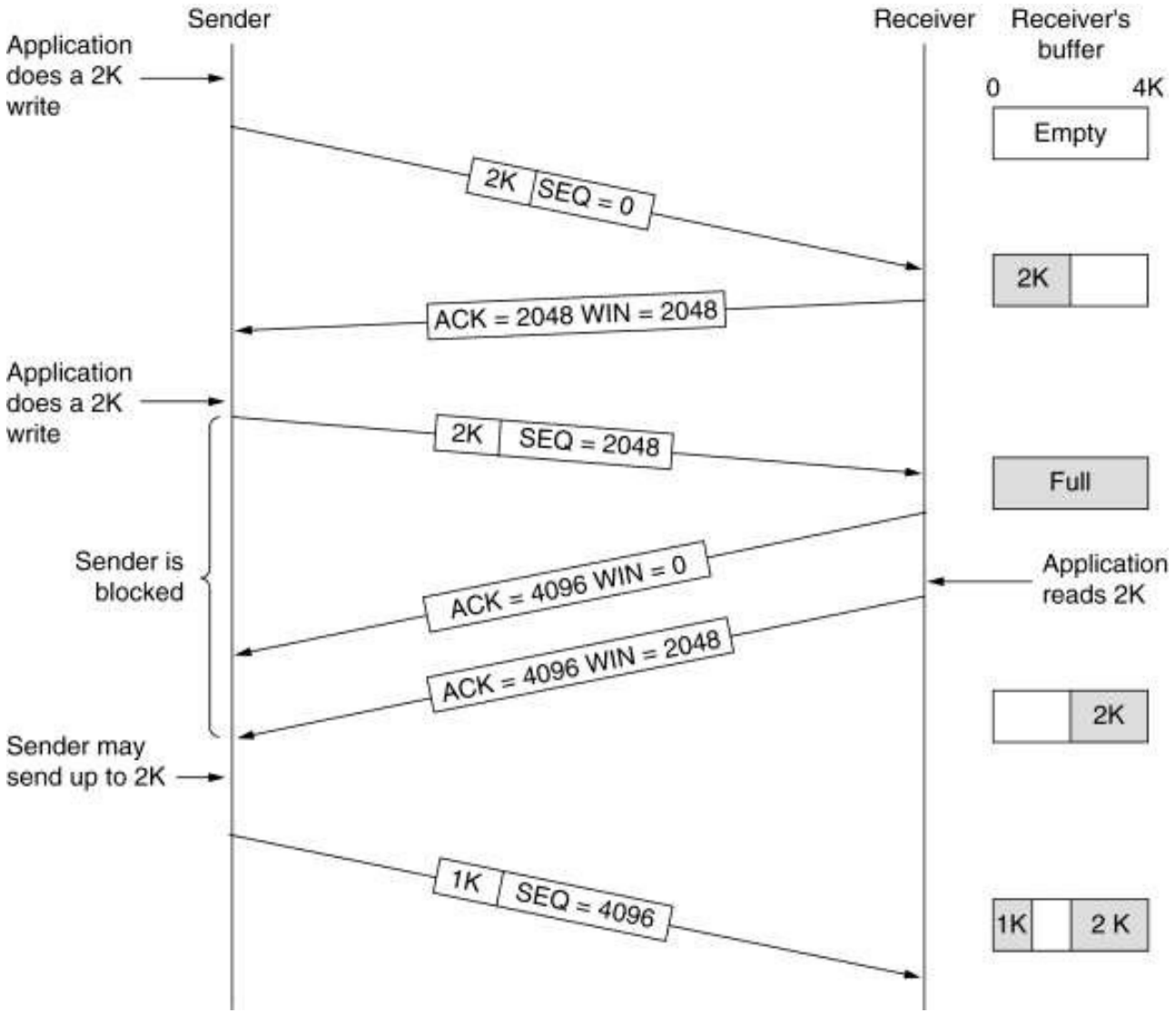
The states used in the TCP connection management finite state machine.

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.

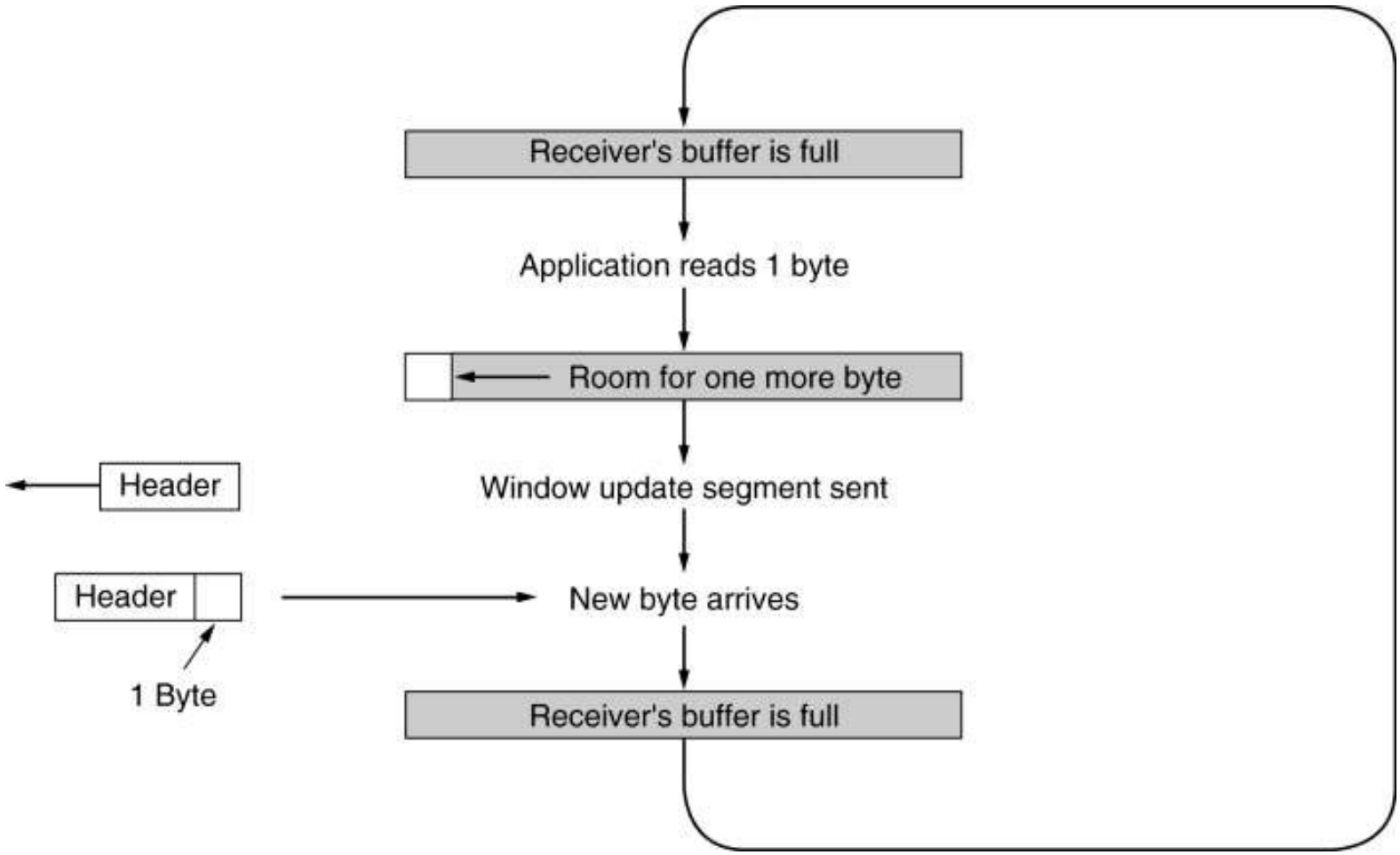


TCP Transmission Policy



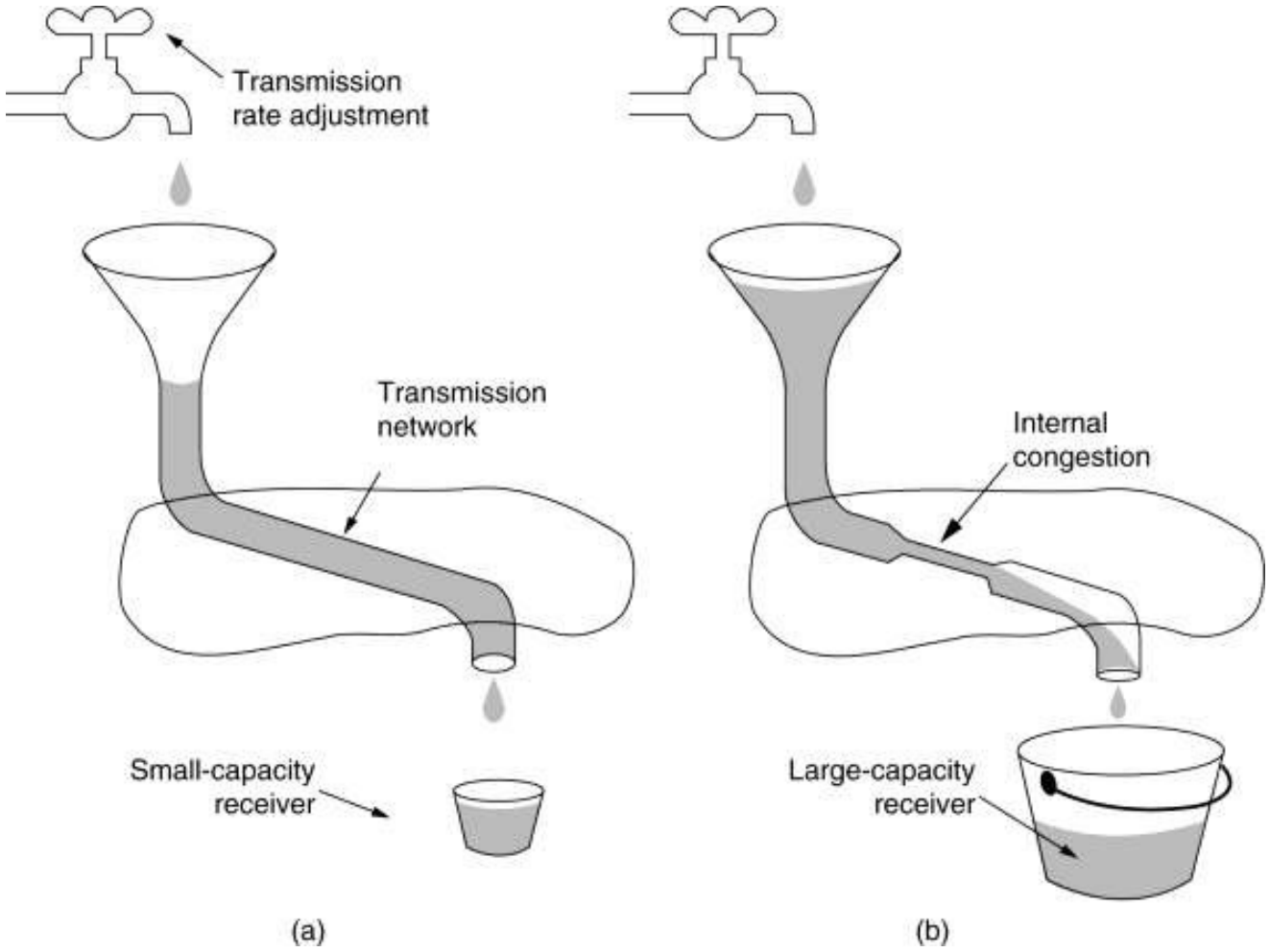
Window management in TCP.

TCP Transmission Policy (2)



Silly window syndrome.

TCP Congestion Control



(a) A fast network feeding a low capacity receiver.

(b) A slow network feeding a high-capacity receiver.