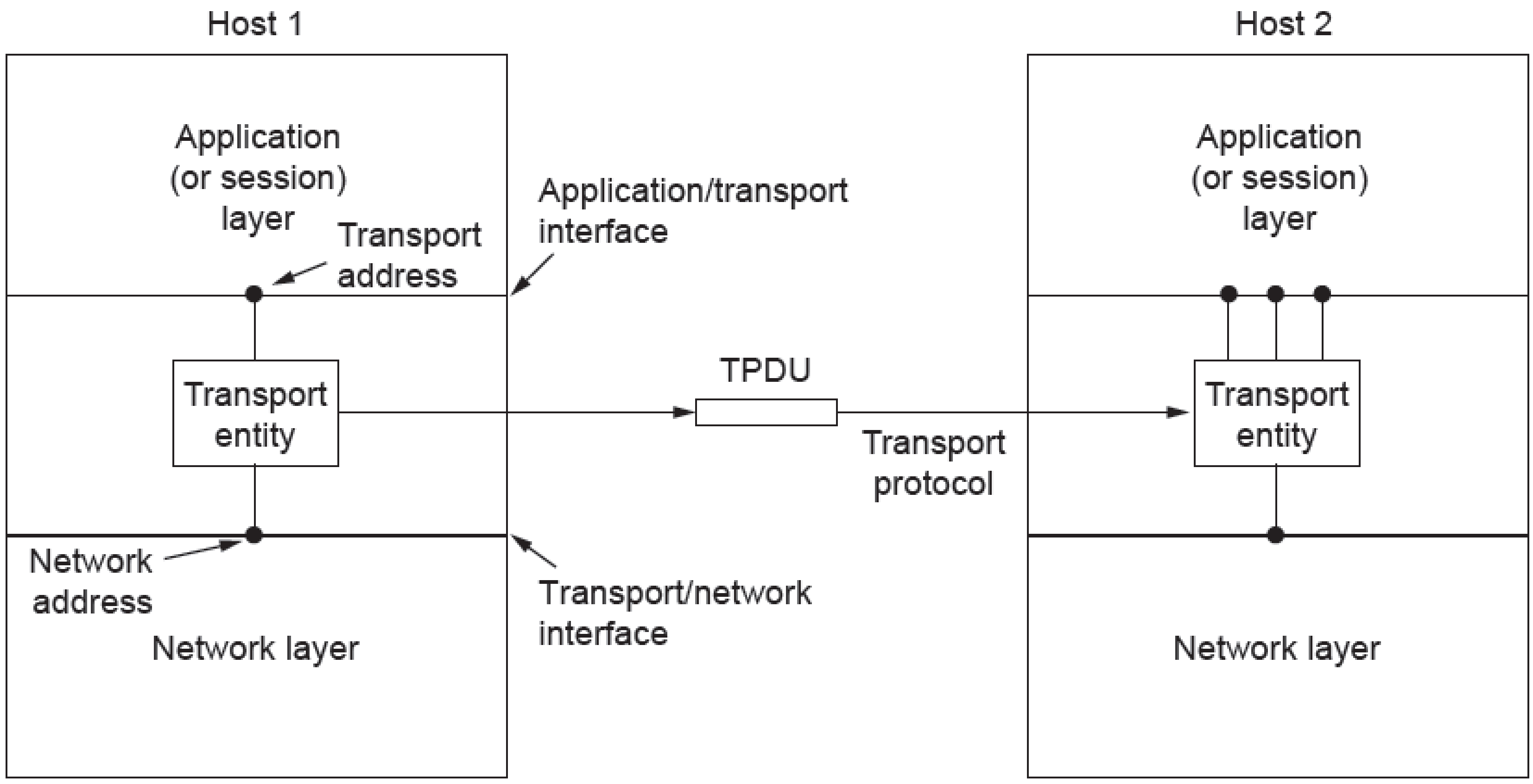
The Transport Layer

Chapter 6

Transport Layer

- It is the heart of the whole protocol hierarchy.
- Its task is to provide reliable, cost-effective data transport from the source machine to the destination machine, independently of the physical network or networks currently in use.
- It provides service to the application layer.
- Transport layer makes use of the services provided by the network layer.
- The hardware and/or software within the transport layer that does the work is called the transport entity.
- It provides both connectionless and connection oriented service.
- Connections have three phases: establishment, data transfer, and release.

Services Provided to the Upper Layers



The relationship of network, transport, and application layers

Transport Service Primitives

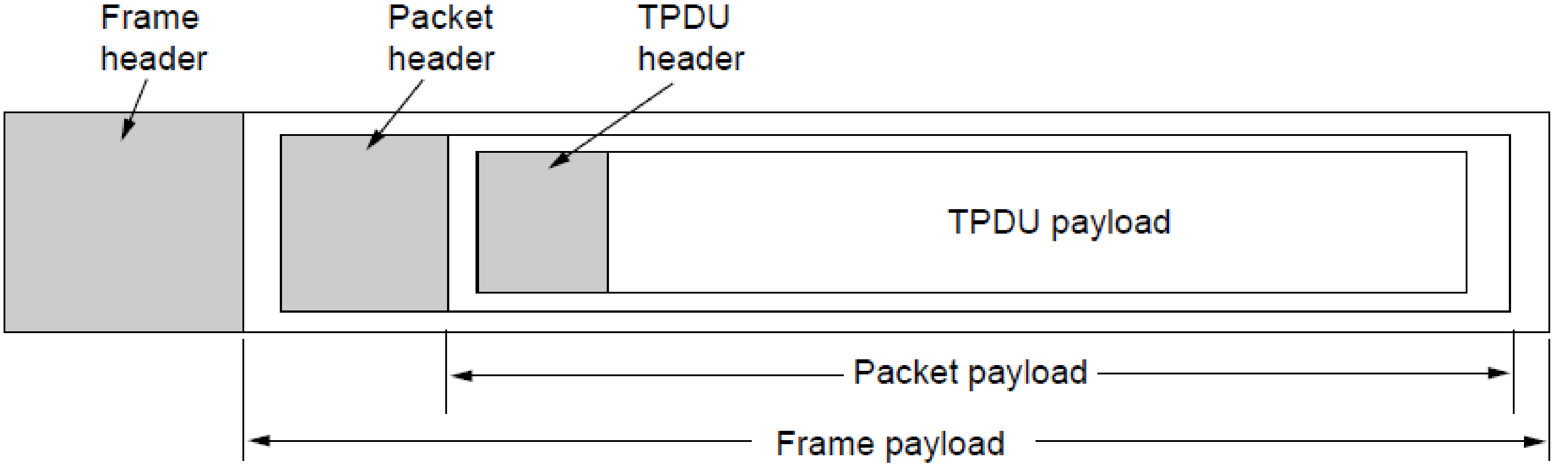
Primitive	Packet sent	Meaning
LISTEN	(none)	Block until some process tries to connect
CONNECT	CONNECTION REQ.	Actively attempt to establish a connection
SEND	DATA	Send information
RECEIVE	(none)	Block until a DATA packet arrives
DISCONNECT	DISCONNECTION REQ.	This side wants to release the connection

The primitives for a simple transport service

Transport Service Primitives

- To see how these primitives might be used, consider an application with a server and a number of remote clients. To start with, the server executes a LISTEN primitive, typically by calling a library procedure that makes a system call to block the server until a client turns up. When a client wants to talk to the server, it executes a CONNECT primitive. The transport entity carries out this primitive by blocking the caller and sending a packet to the server. Encapsulated in the payload of this packet is a transport layer message for the server's transport entity.

Transport Service Primitives

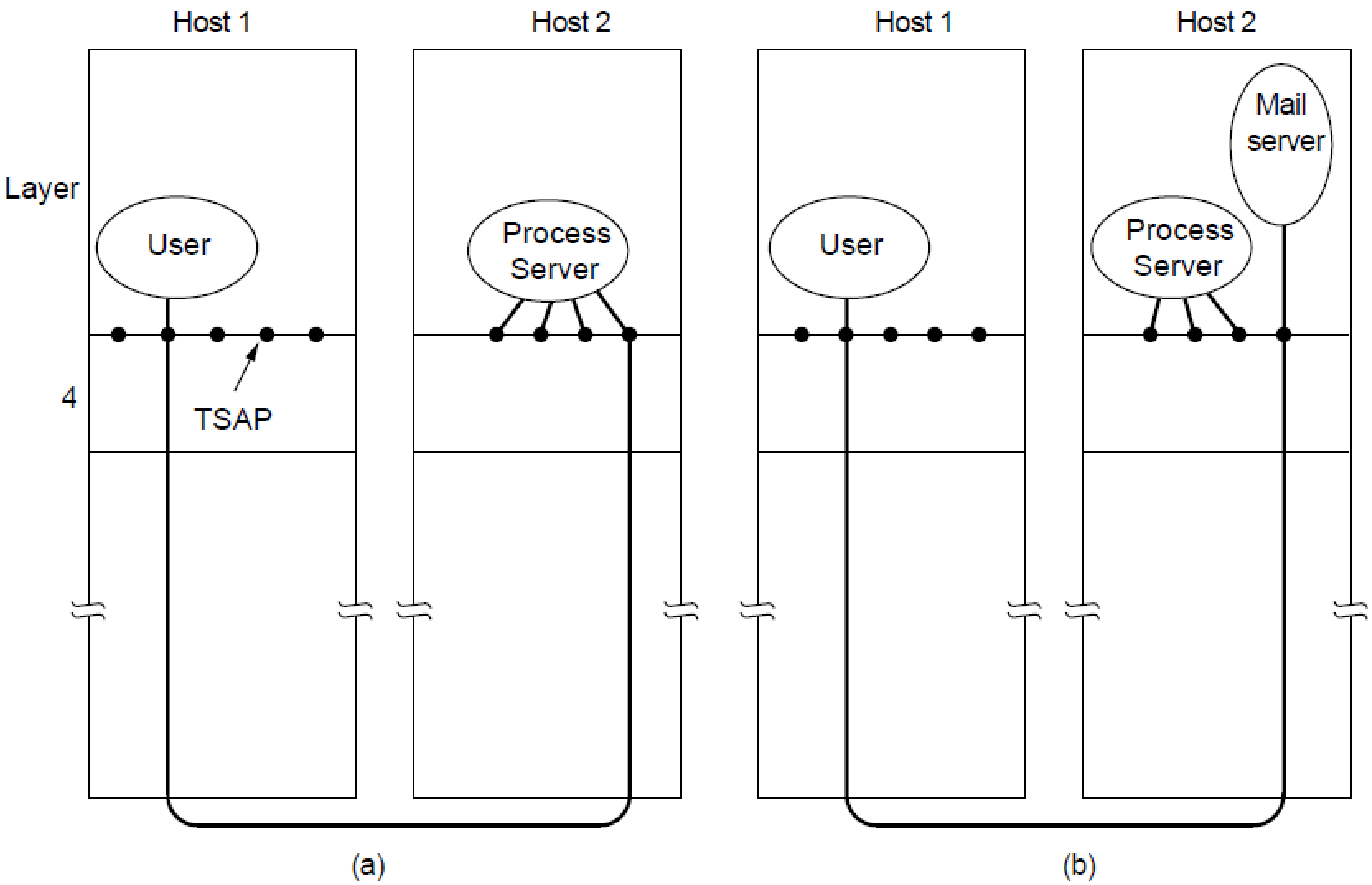


Nesting of TPDUs, packets, and frames.

Elements of Transport Protocols

- Addressing
- Connection establishment
- Connection release
- Error control and flow control
- Multiplexing
- Crash recovery

Addressing



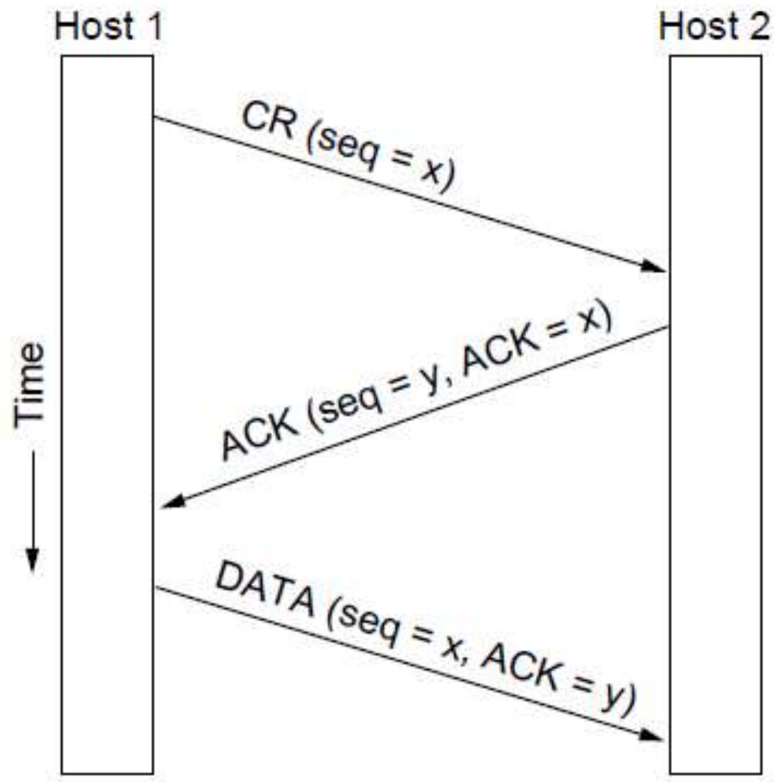
How a user process in host 1 establishes a connection with a mail server in host 2 via a process server.

Connection Establishment

Techniques for restricting packet lifetime

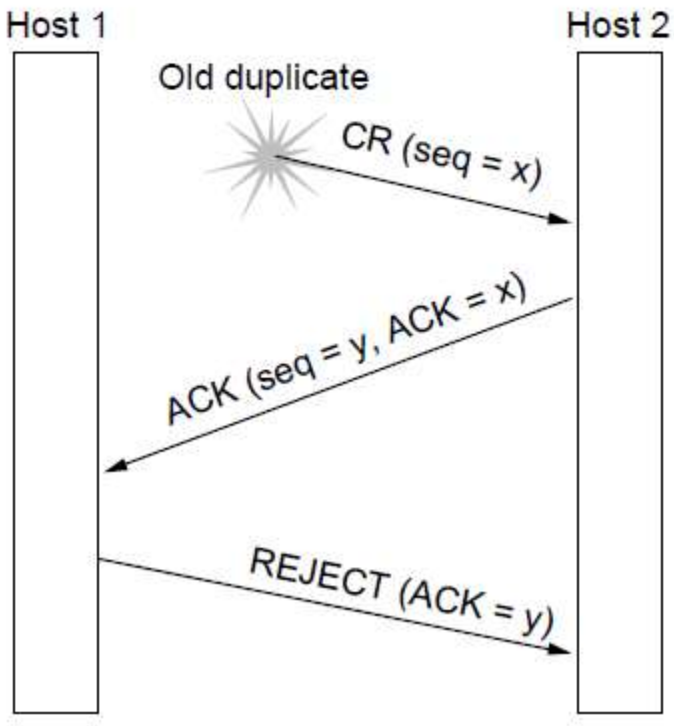
- Restricted network design.
- Putting a hop counter in each packet.
- Timestamping each packet.

Connection Establishment



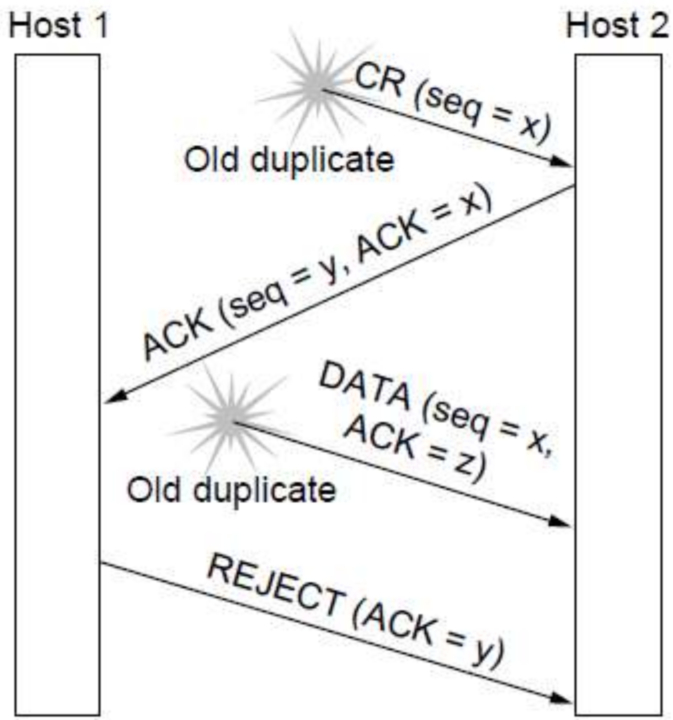
Three protocol scenarios for establishing a connection using a three-way handshake. CR denotes CONNECTION REQUEST. Normal operation.

Connection Establishment



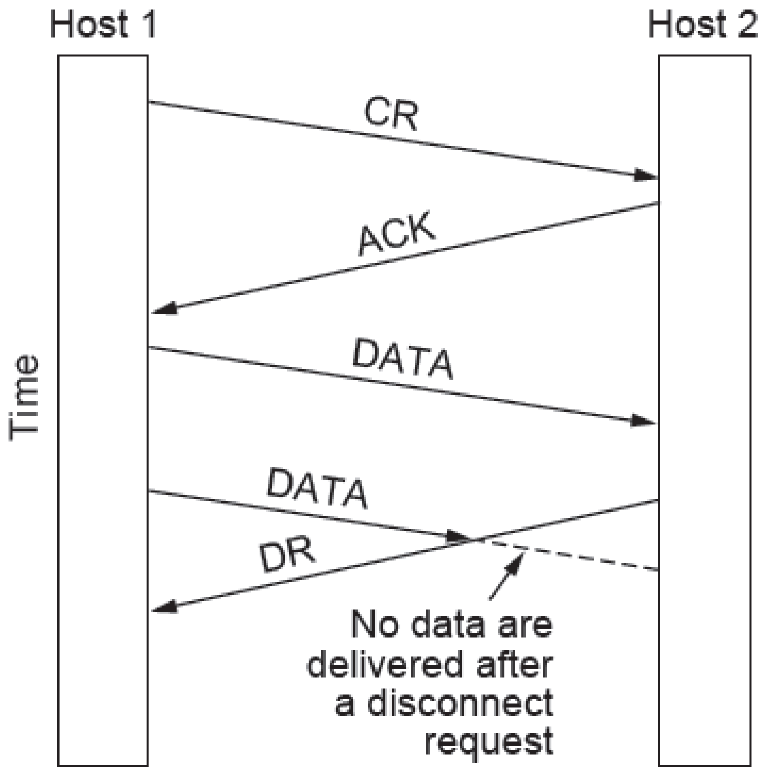
Three protocol scenarios for establishing a connection using a three-way handshake. CR denotes CONNECTION REQUEST. Old duplicate CONNECTION REQUEST appearing out of nowhere.

Connection Establishment



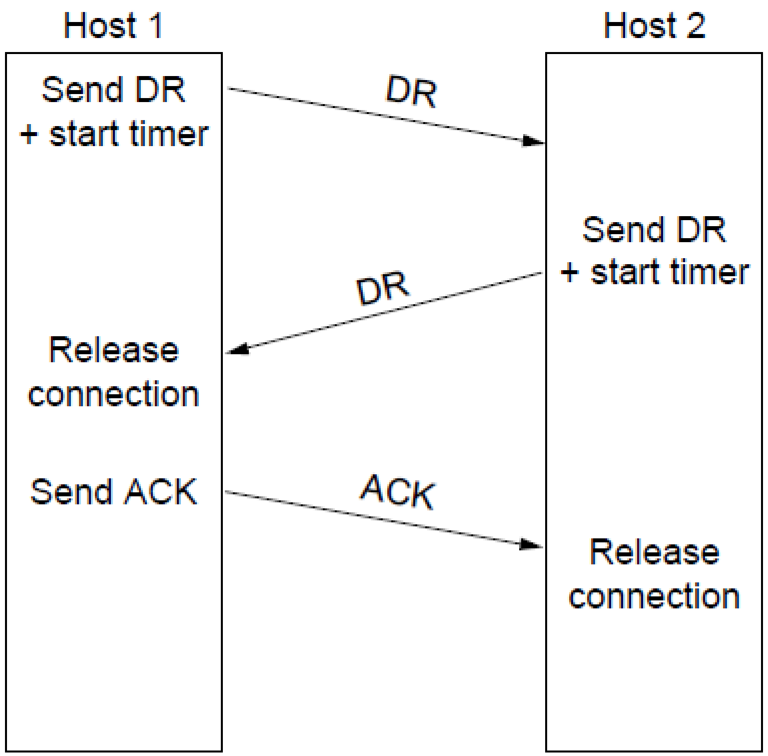
Three protocol scenarios for establishing a connection using a three-way handshake. CR denotes CONNECTION REQUEST. Duplicate CONNECTION REQUEST and duplicate ACK

Connection Release



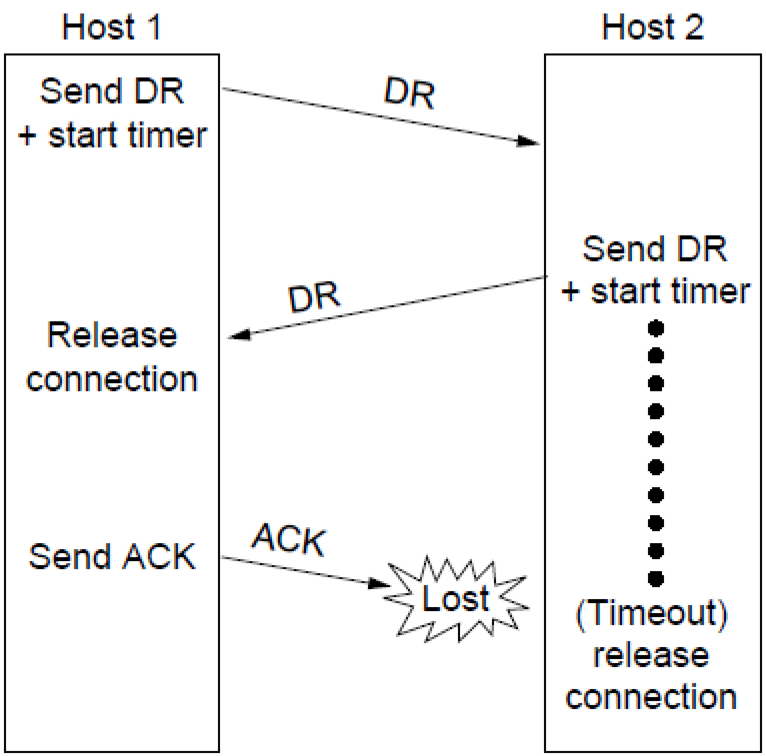
Abrupt disconnection with loss of data

Connection Release



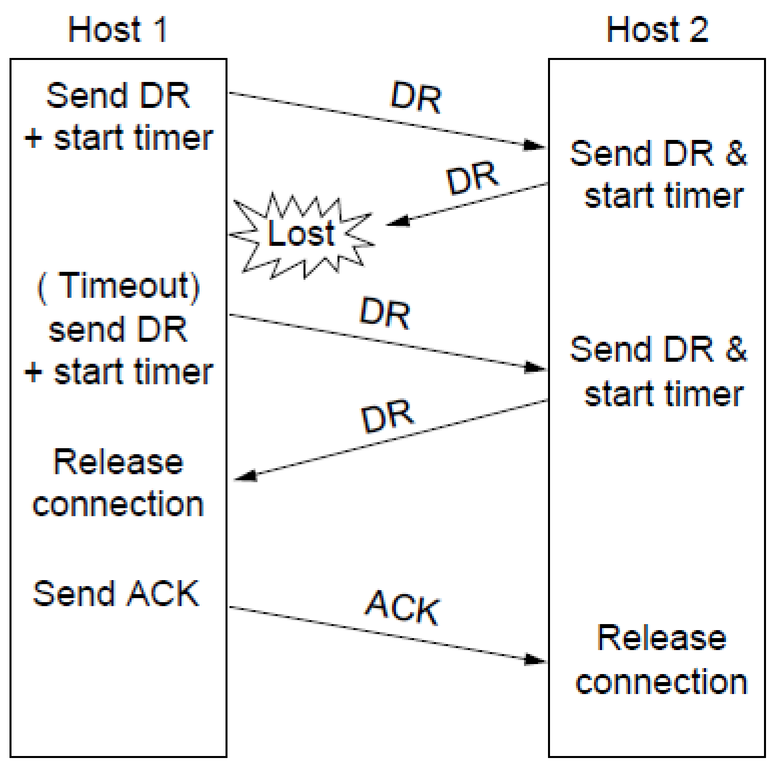
Four protocol scenarios for releasing a connection.
(a) Normal case of three-way handshake

Connection Release



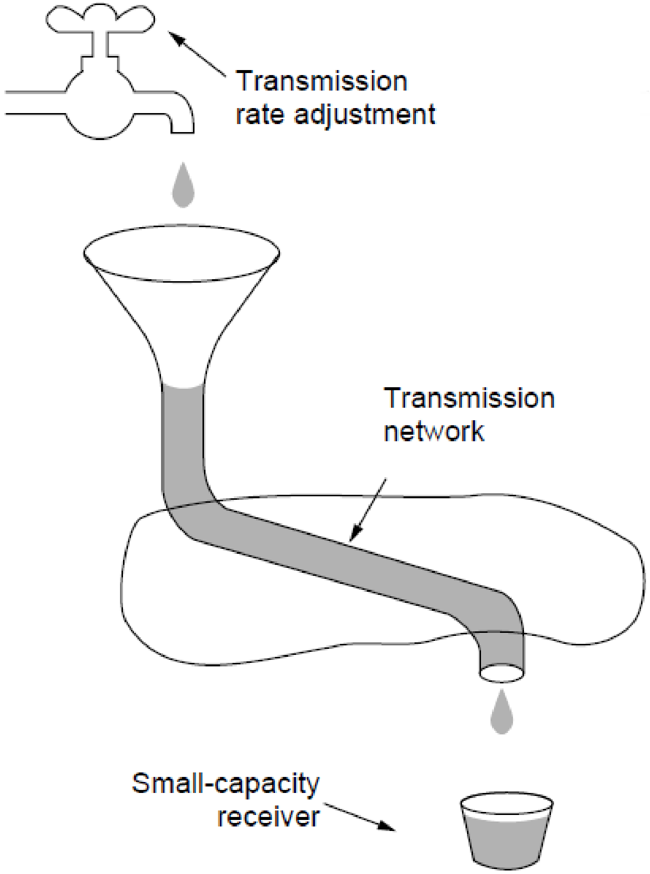
Four protocol scenarios for releasing a connection.
(b) Final ACK lost.

Connection Release



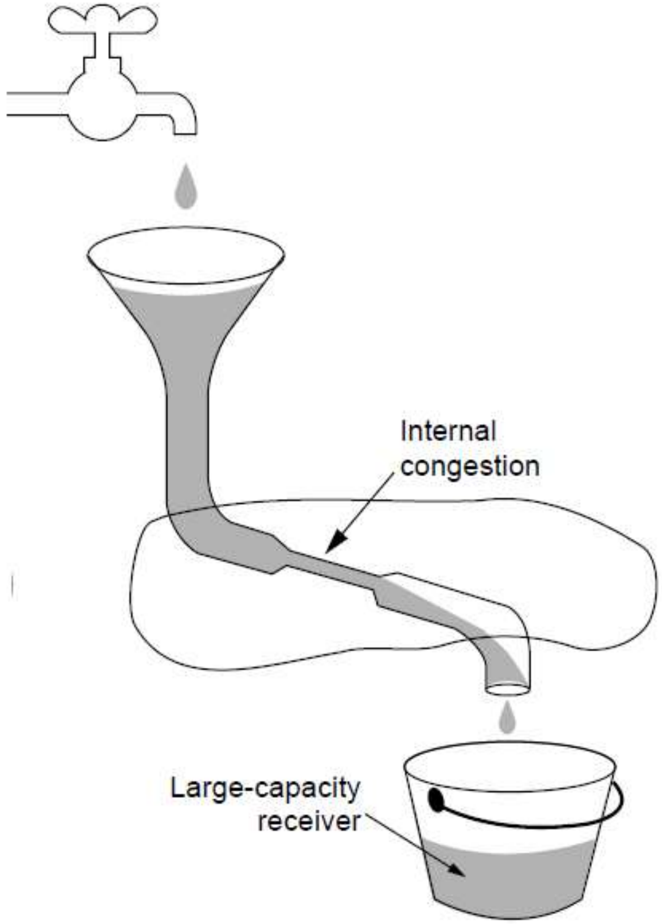
Four protocol scenarios for releasing a connection.
(c) Response lost

Regulating the Sending Rate



A fast network feeding a low-capacity receiver

Regulating the Sending Rate

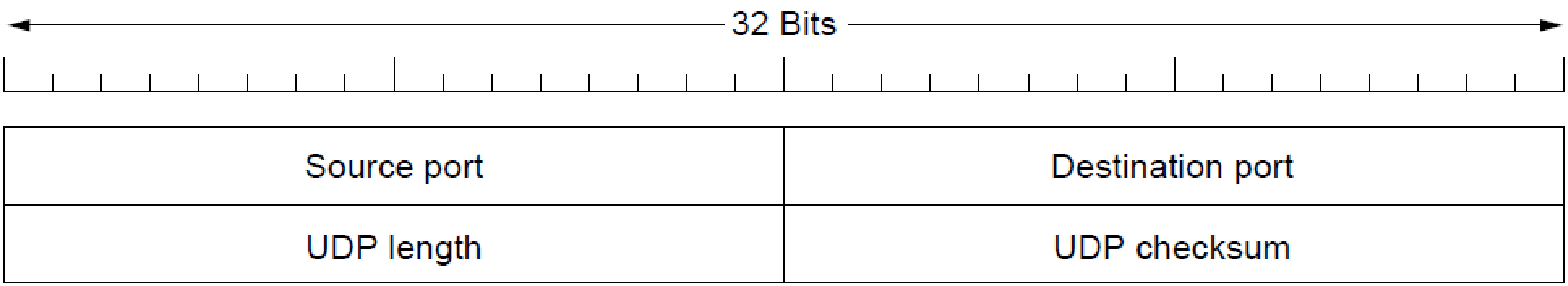


A slow network feeding a high-capacity receiver

Introduction to UDP

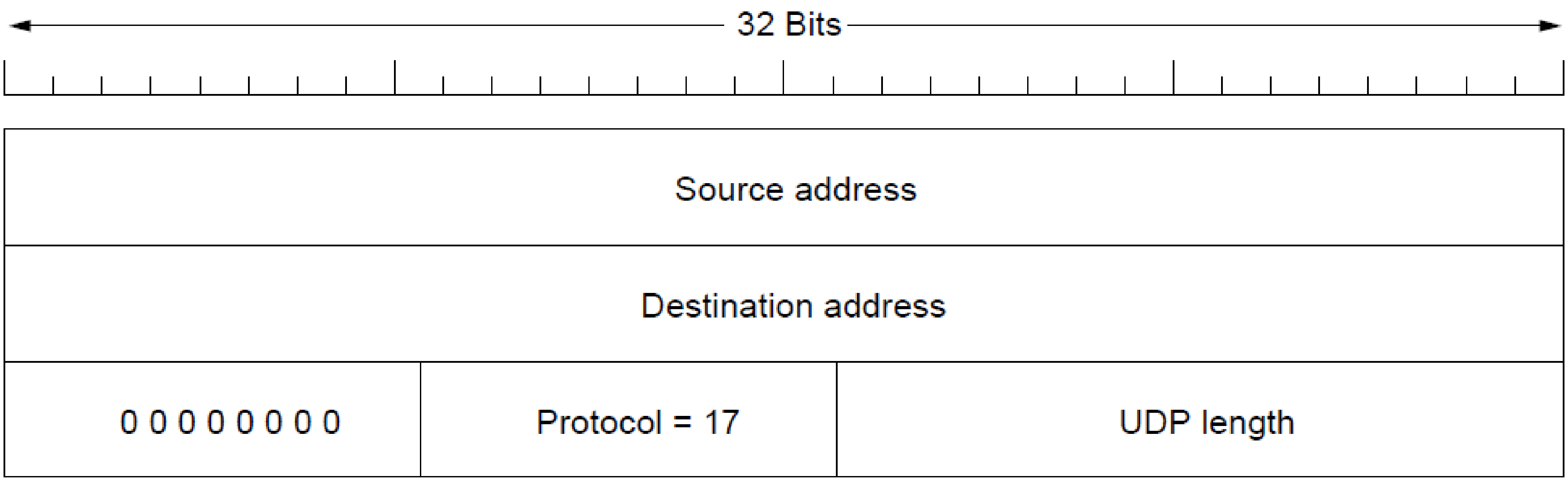
- The Internet protocol suite supports a connectionless transport protocol, UDP (User Datagram Protocol).
- UDP provides a way for applications to send IP datagrams without having to establish a connection.
- UDP transmits segments consisting of an 8-byte header followed by the payload.
- The UDP length field includes the 8-byte header and the data. The UDP checksum is optional and stored as 0 if not computed.

Introduction to UDP



The UDP header.

Introduction to UDP



The IPv4 pseudoheader included in the UDP checksum.

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

Introduction to TCP

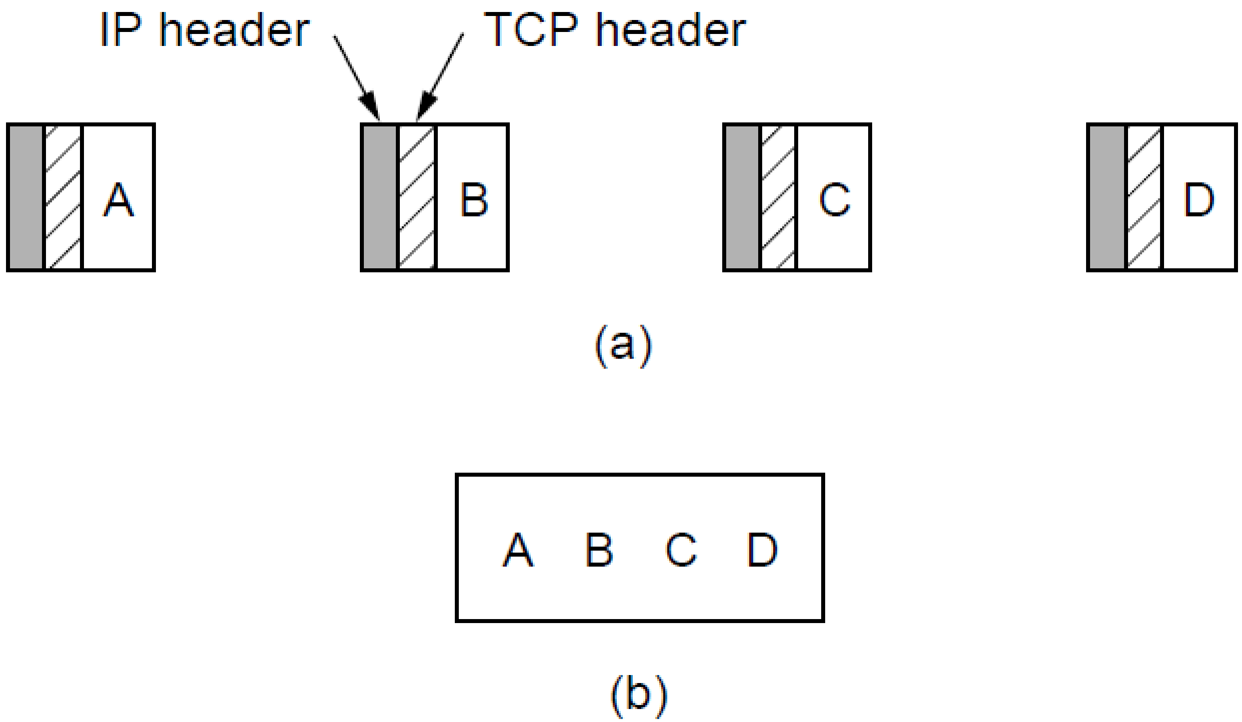
- TCP (Transmission Control Protocol) was specifically designed to provide a reliable end-to-end byte stream over an unreliable internetwork.
- An internetwork differs in terms of different topologies, bandwidths, delays, packet sizes, and other parameters.
- Each machine supporting TCP has a TCP transport entity, either a library procedure, a user process, or part of the kernel.
- TCP service is obtained by both the sender and receiver creating end points, called sockets. Each socket has a socket number (address) consisting of the IP address of the host and a 16-bit number local to that host, called a port.
- Port numbers below 1024 are called well-known ports and are reserved for standard services.

The TCP Service Model

Port	Protocol	Use
20, 21	FTP	File transfer
22	SSH	Remote login, replacement for Telnet
25	SMTP	Email
80	HTTP	World Wide Web
110	POP-3	Remote email access
143	IMAP	Remote email access
443	HTTPS	Secure Web (HTTP over SSL/TLS)
543	RTSP	Media player control
631	IPP	Printer sharing

Some assigned ports

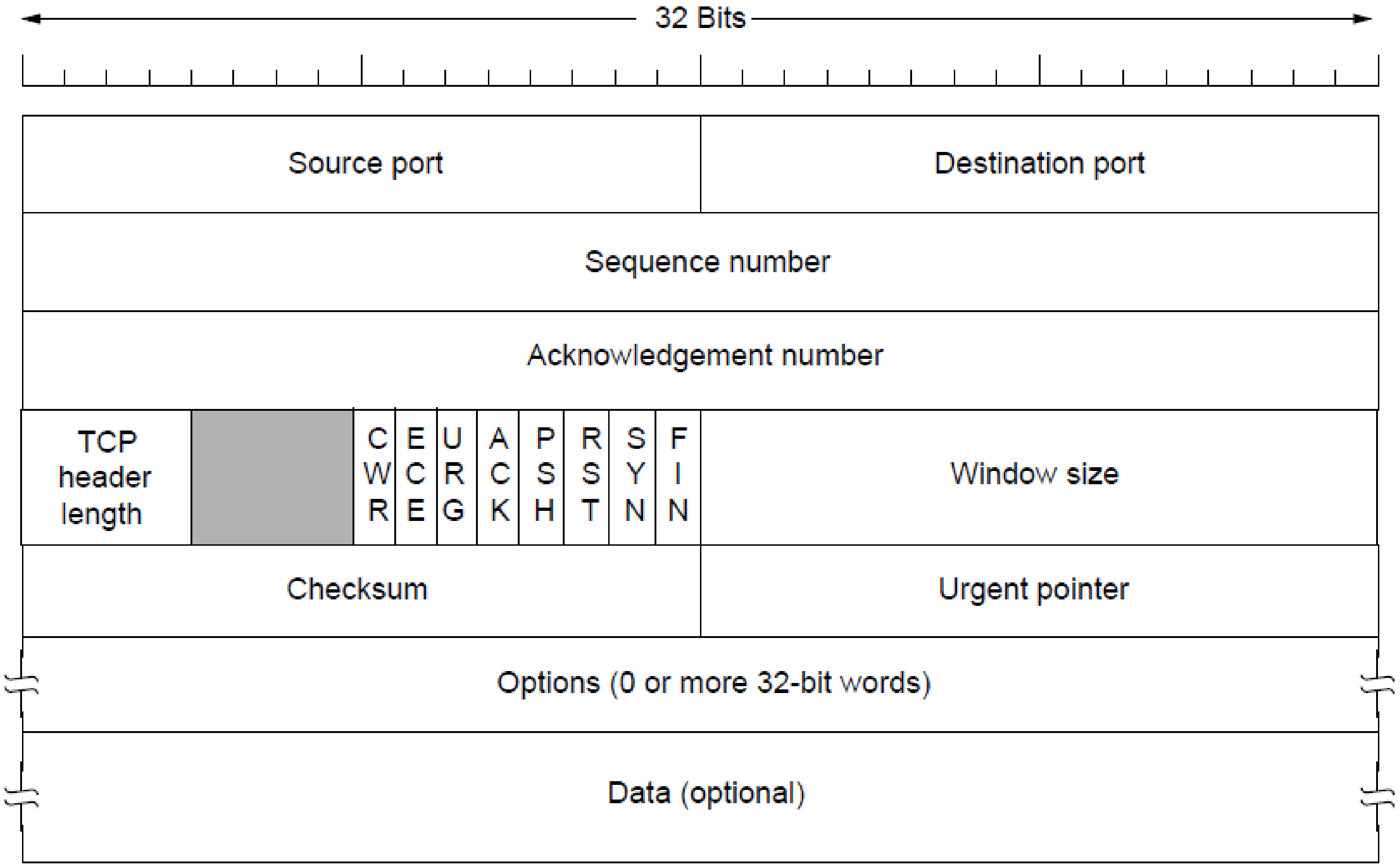
The TCP Service Model



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

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

The TCP Segment Header



The TCP header.

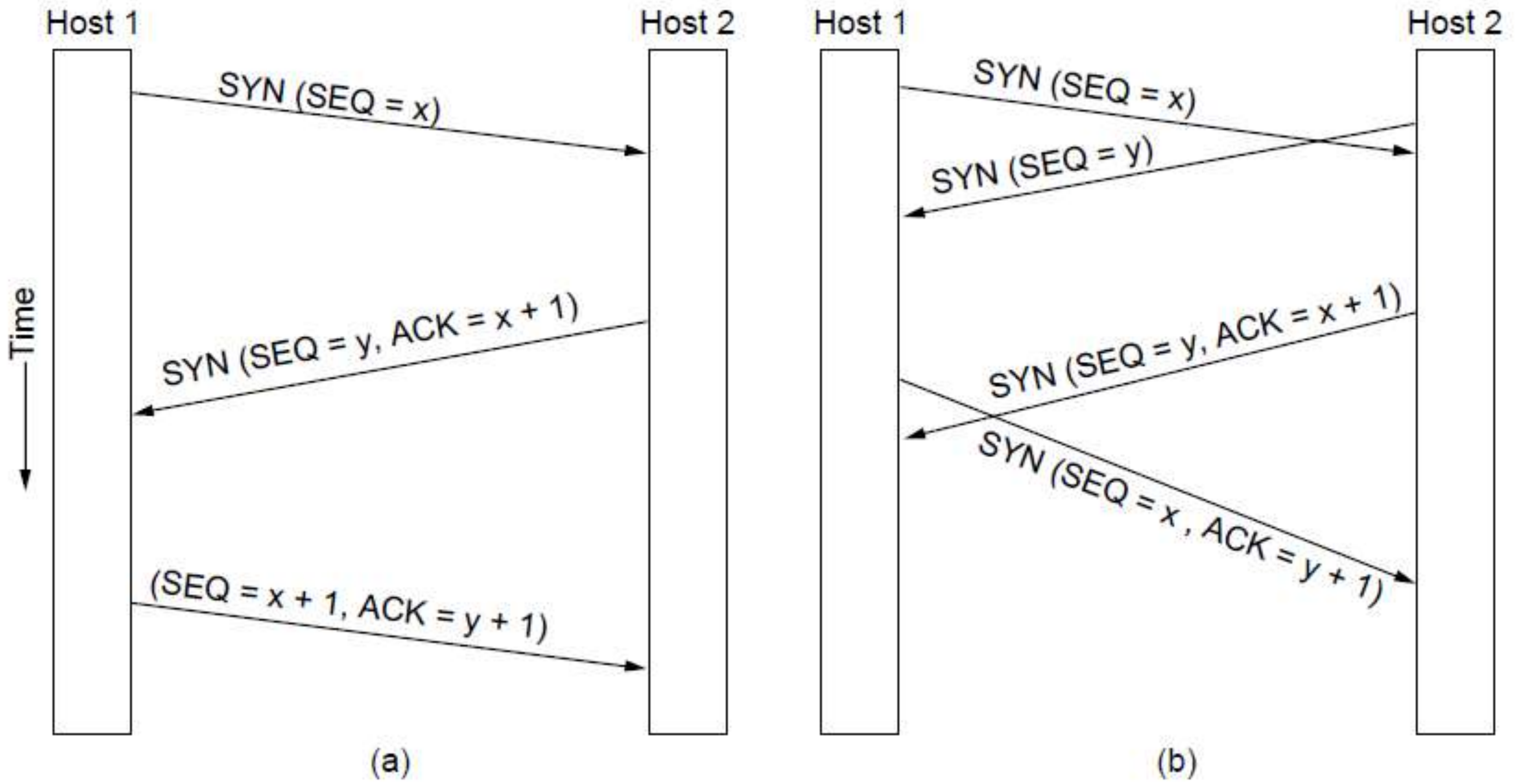
The TCP Segment Header

- TCP connection has its own 32-bit sequence number.
- Separate 32-bit sequence numbers are used for acknowledgements.
- The sending and receiving TCP entities exchange data in the form of segments.
- A TCP segment consists of a fixed 20-byte header (plus an optional part) followed by zero or more data bytes.
- The TCP software decides how big segments should be.
- Two limits restrict the segment size.
- First, each segment, including the TCP header, must fit in the 65,515-byte IP payload.
- Second, each network has a maximum transfer unit, or MTU. In practice, the MTU is generally 1500 bytes and thus defines the upper bound on segment size.

TCP Working

- The basic protocol used by TCP entities is the sliding window protocol.
- When a sender transmits a segment, it also starts a timer.
- When the segment arrives at the destination, the receiving TCP entity sends back a segment (with data if any exist, otherwise without data) bearing an acknowledgement number equal to the next sequence number it expects to receive.
- If the sender's timer goes off before the acknowledgement is received, the sender transmits the segment again.

TCP Connection Establishment



(a)TCP connection establishment in the normal case.
(b) Simultaneous connection establishment on both sides.

TCP Connection Release

- To release a connection, either party can send a TCP segment with the FIN bit set, which means that it has no more data to transmit.
- When the FIN is acknowledged, that direction is shut down for new data.
- Data may continue to flow indefinitely in the other direction, however. When both directions have been shut down, the connection is released.

End

Chapter 6