Computer Networks

Transport Layer

Presented by Hung Ba Ngo

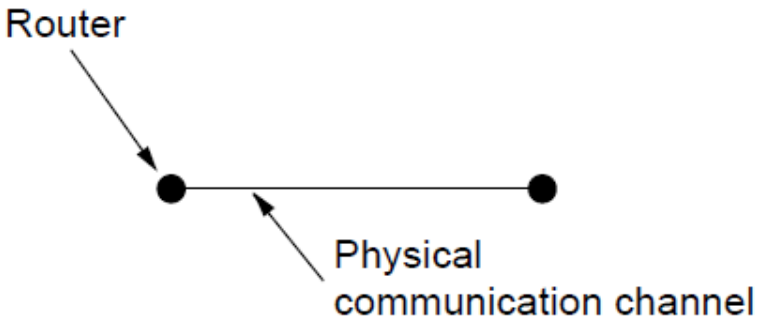
Tháng 08/2015

Objectives

- Roles of the transport layer
- Services provided by the transport layer
- Connection establishment
- Connection release
- TCP and UPD protocols

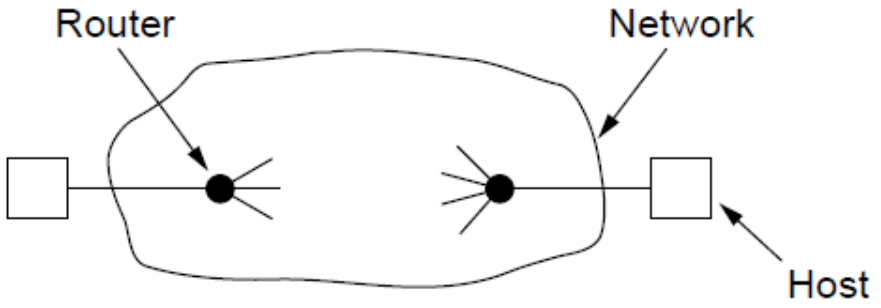
Roles of Transport layer

- While Network layer provides a host-to-host communication, Transport layer provides a End point-to-End point communication
- End points are running applications
- Provides effective, reliable and cost savings packet transmission service for users



(a)

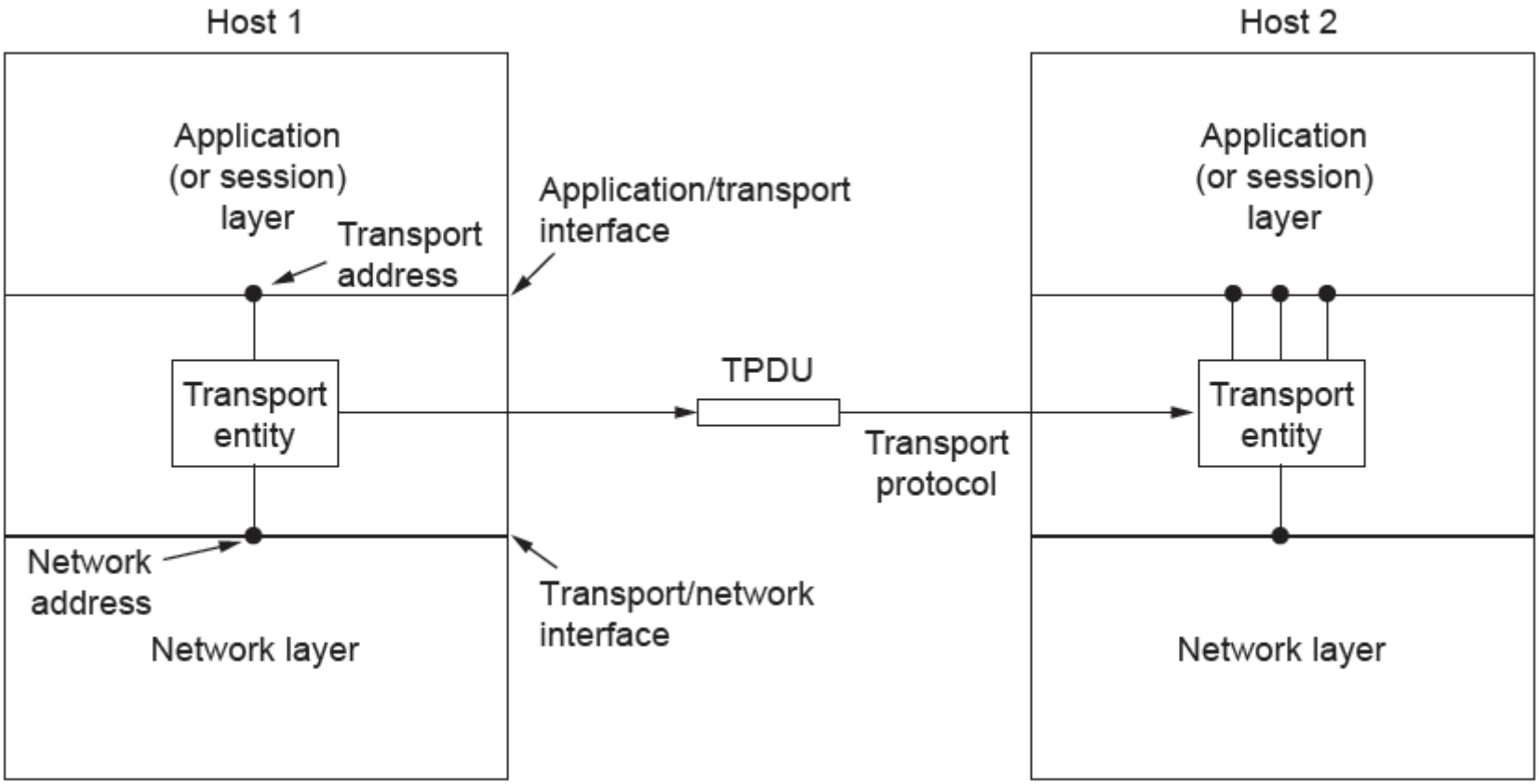
Environment of the data link layer.



(b)

Environment of the transport layer

Services Provided to the Upper Layers

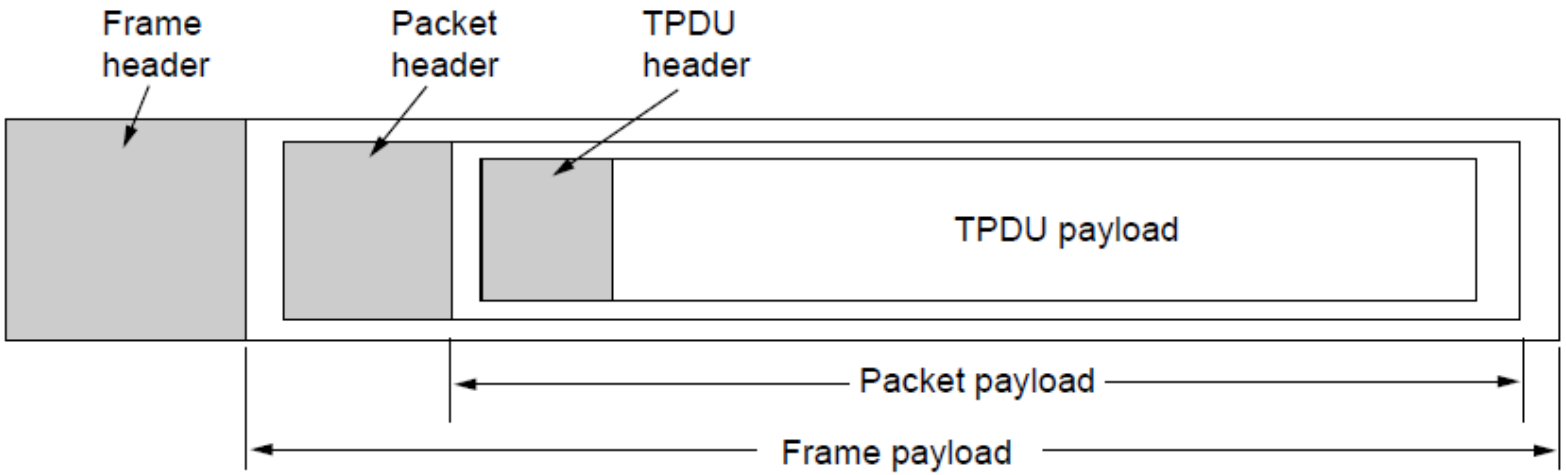


The network, transport, and application layers

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



Nesting of TPDUs, packets, and frames.

Berkeley Sockets

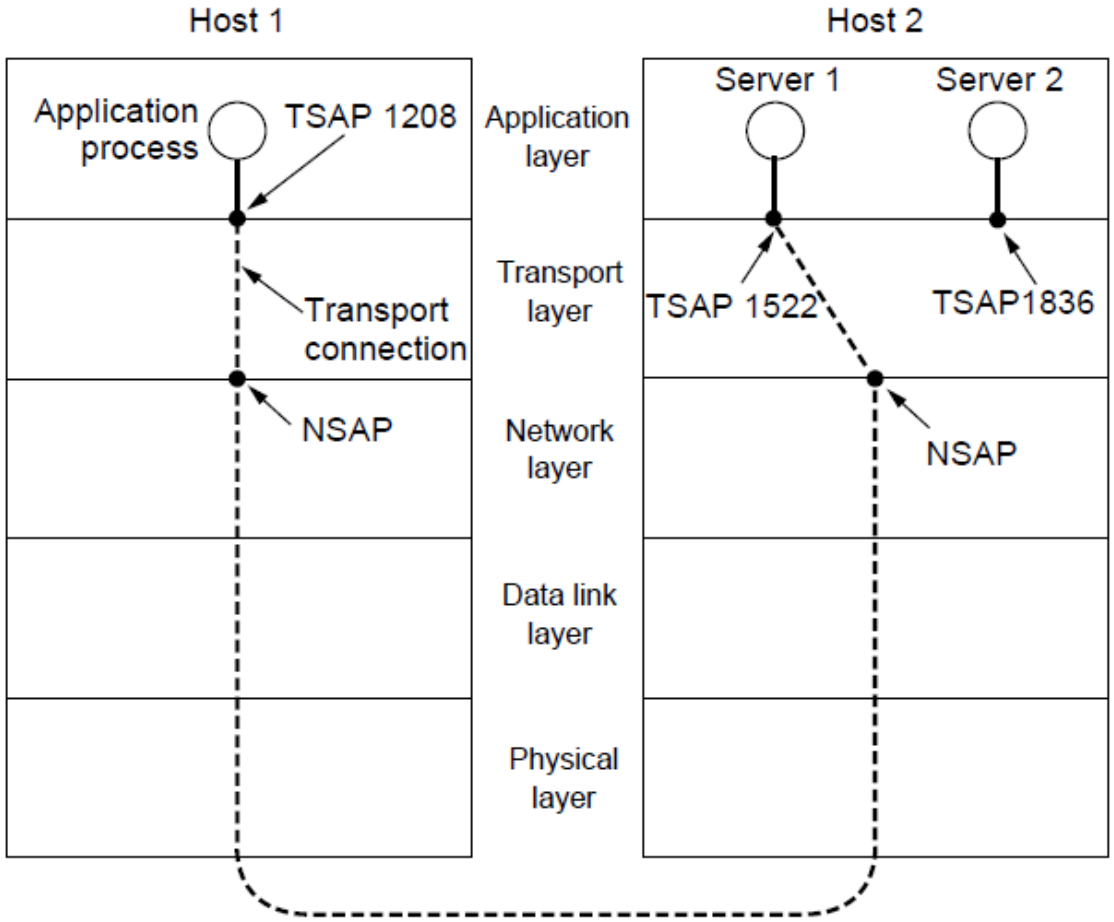
Primitive	Meaning
SOCKET	Create a new communication end point
BIND	Associate a local address with a socket
LISTEN	Announce willingness to accept connections; give queue size
ACCEPT	Passively establish an incoming connection
CONNECT	Actively attempt to establish a connection
SEND	Send some data over the connection
RECEIVE	Receive some data from the connection
CLOSE	Release the connection

The socket primitives for TCP

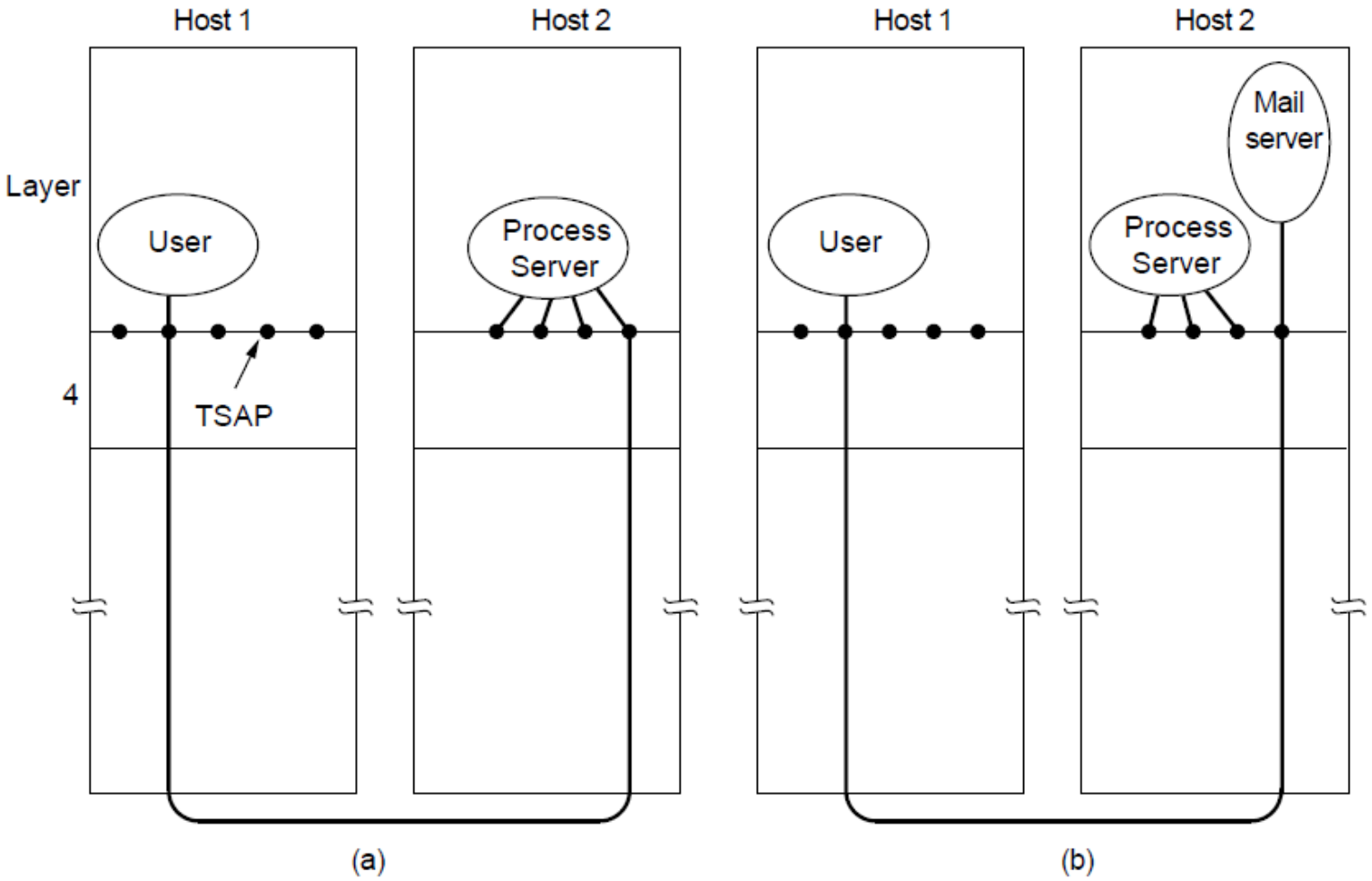
Elements of Transport Protocols

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

Addressing

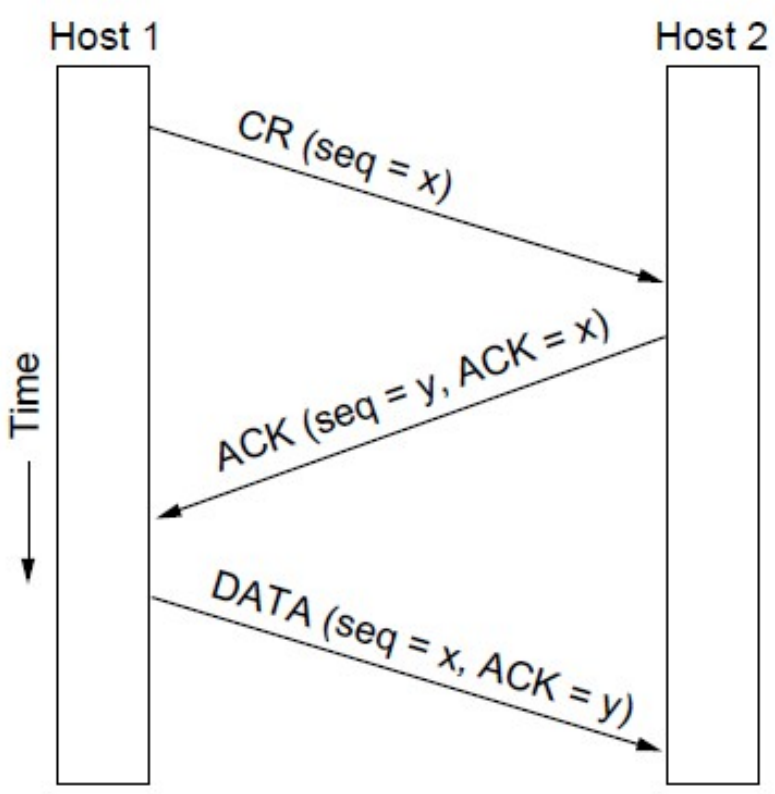


TSAPs, NSAPs, and transport connections



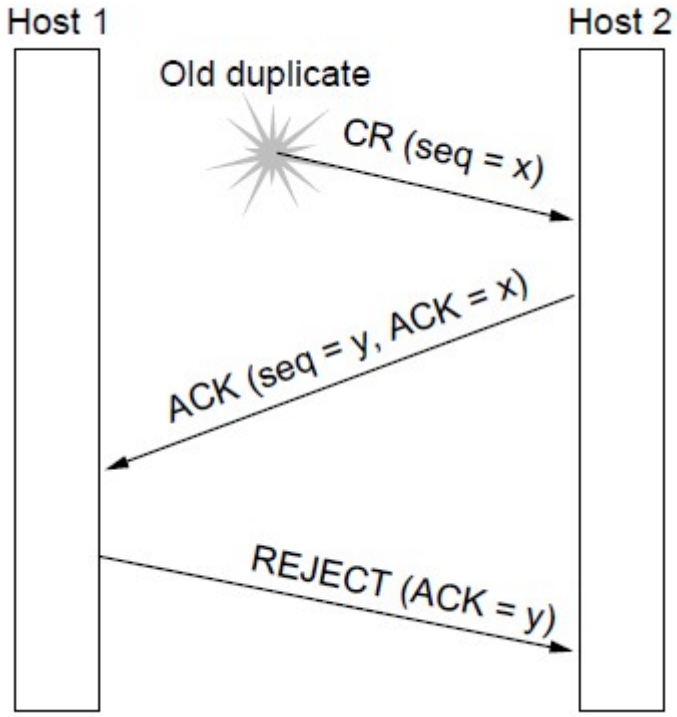
How a user process in host 1 establishes a connection with a mail server in host 2 via a process server.

Connection Establishment



Establishing a connection using a three-way handshake.
CR denotes CONNECTION REQUEST.
Normal operation.

Connection Establishment

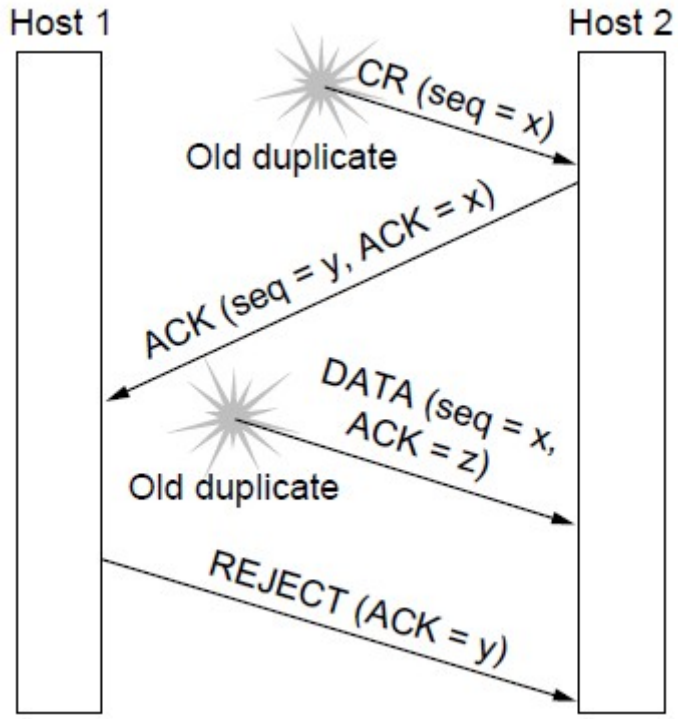


Establishing a connection using a three-way handshake.

CR denotes CONNECTION REQUEST.

Old duplicate CONNECTION REQUEST appearing out of nowhere.

Connection Establishment

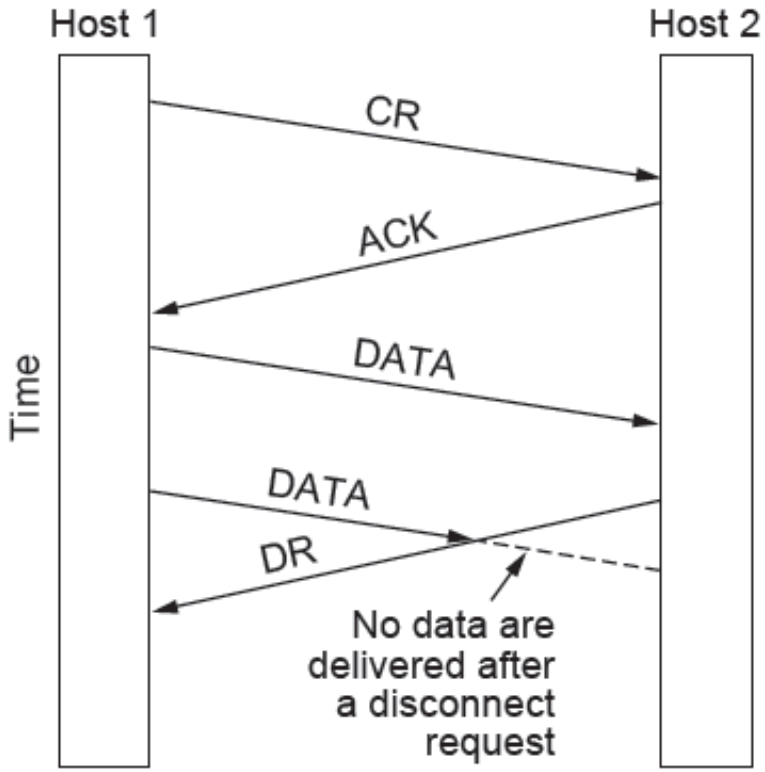


Establishing a connection using a three-way handshake.

CR denotes CONNECTION REQUEST.

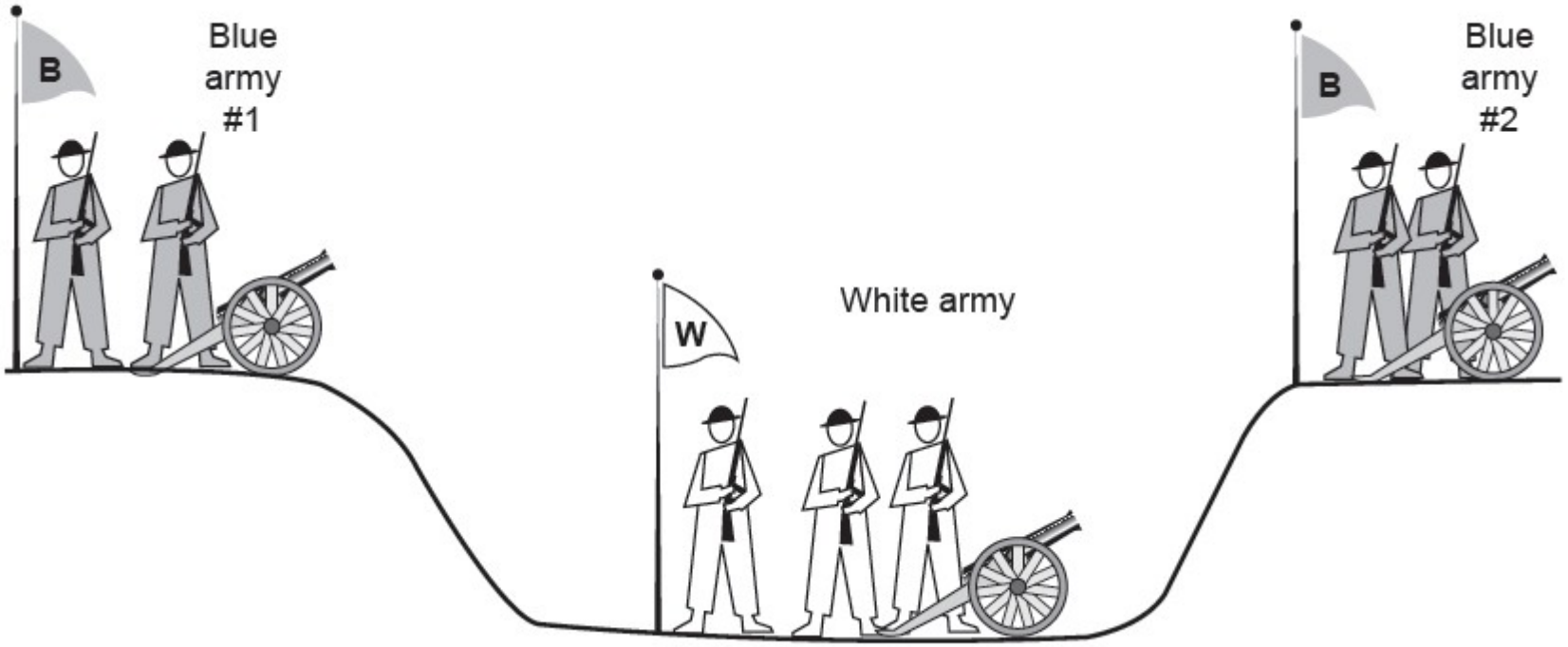
Duplicate CONNECTION REQUEST and duplicate ACK

Connection Release



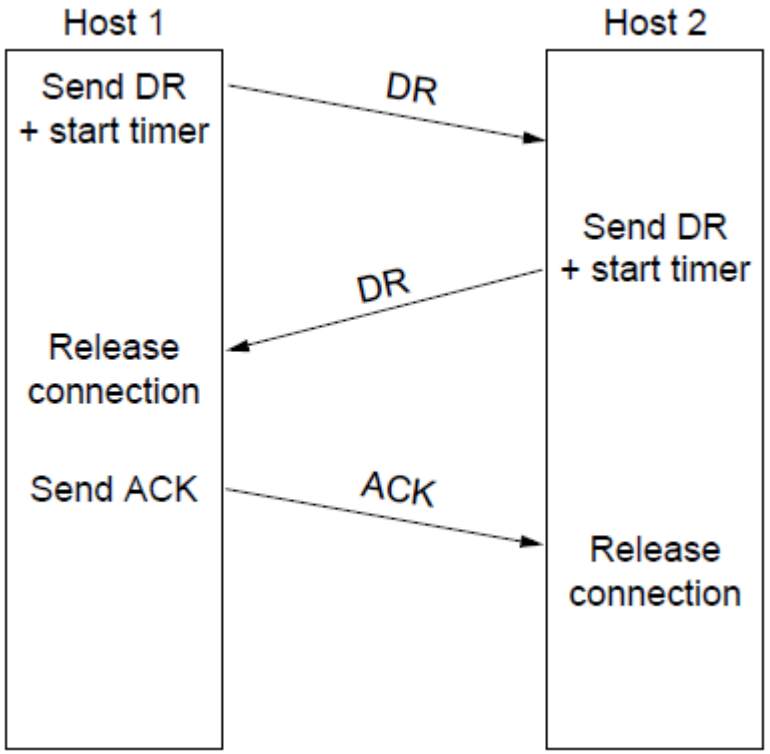
Abrupt disconnection with loss of data

Connection Release



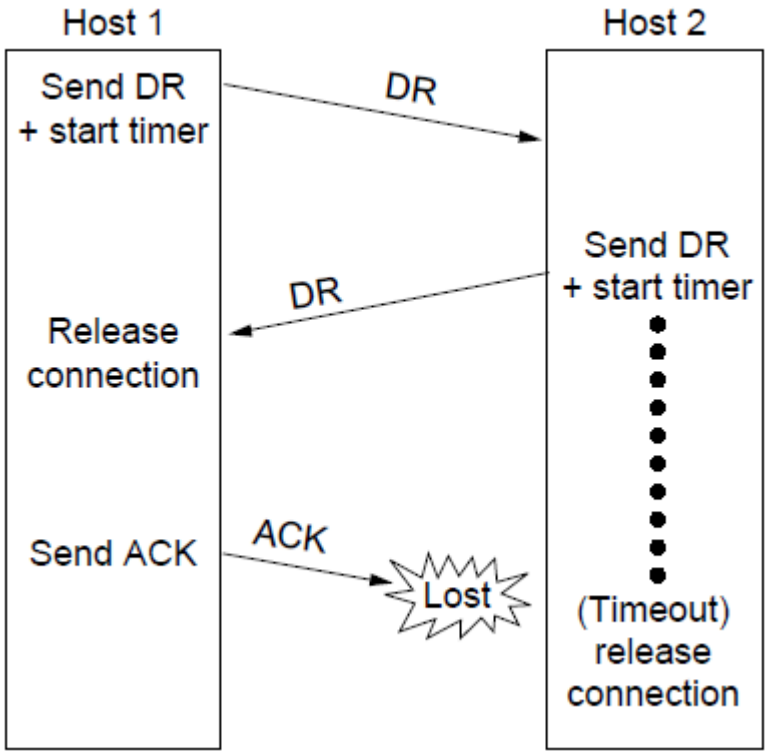
The two-army problem

Connection Release



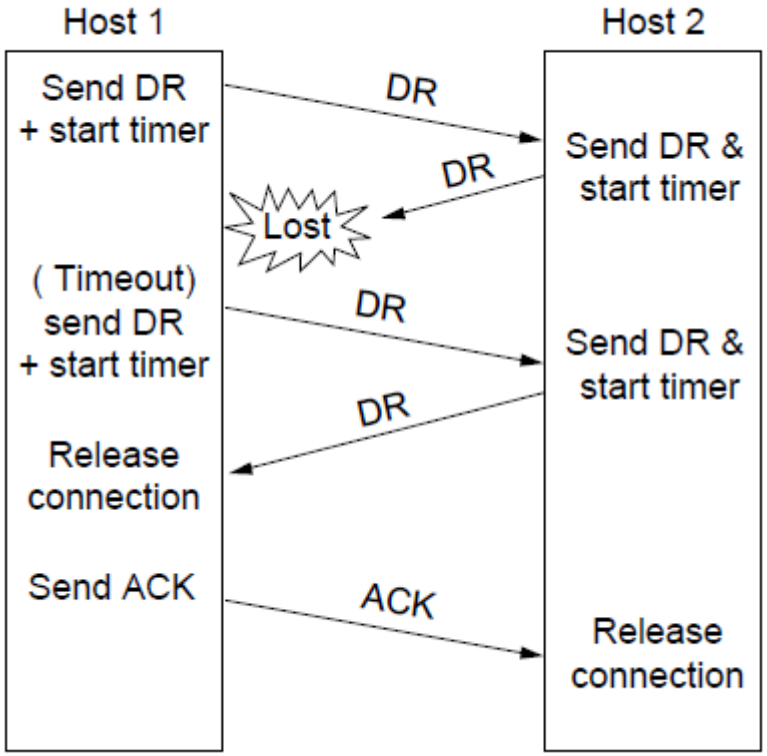
Four protocol scenarios for releasing a connection.
(a) Normal case of three-way handshake

Connection Release



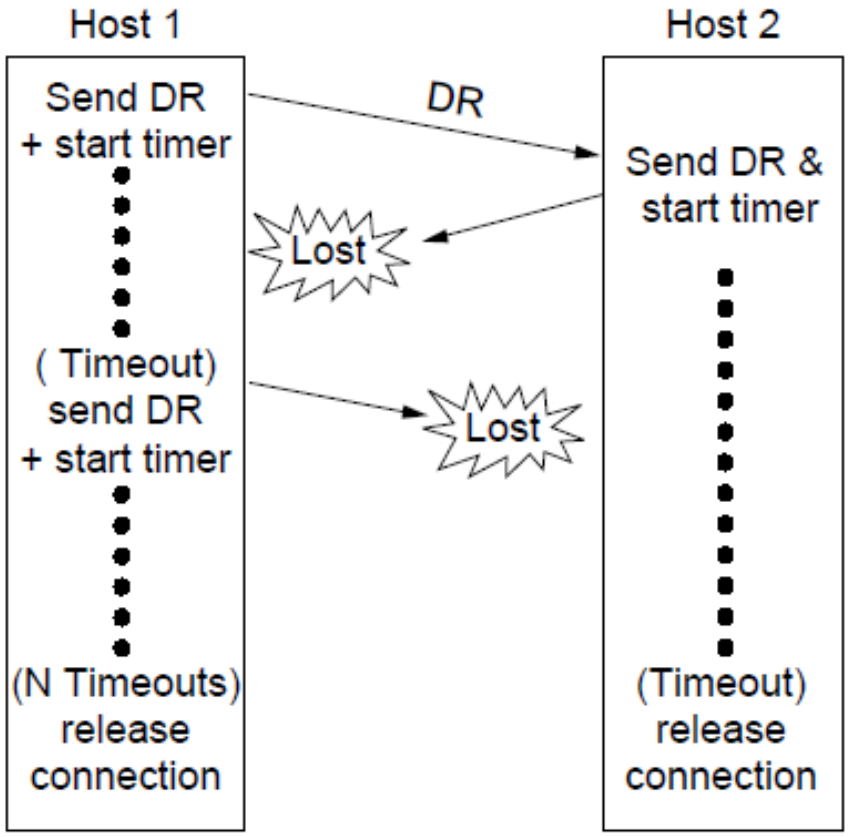
Four protocol scenarios for releasing a connection.
(b) Final ACK lost.

Connection Release



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

Connection Release



Four protocol scenarios for releasing a connection.
(d) Response lost and subsequent DRs lost.

Flow Control

- Using Sliding window protocol where the size of the sending window and the receiving window are different
- Need a scheme for allocating buffer dynamically

Flow Control

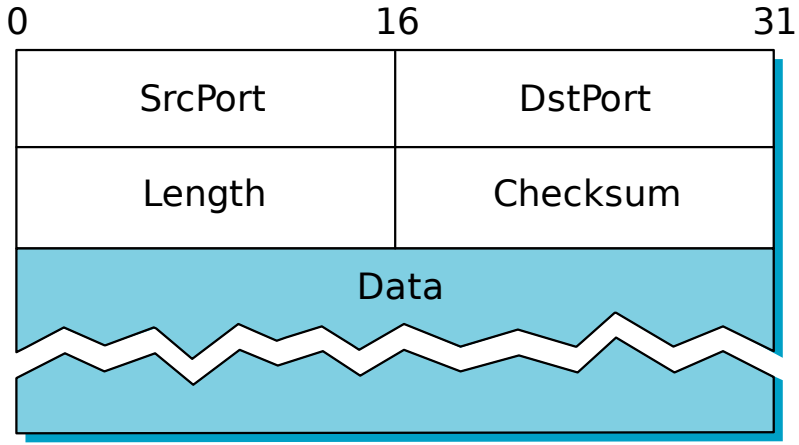
	A	Message	B	Comments
1	→	< request 8 buffers>	→	A wants 8 buffers
2	←	<ack = 15, buf = 4>	←	B grants messages 0-3 only
3	→	<seq = 0, data = m0>	→	A has 3 buffers left now
4	→	<seq = 1, data = m1>	→	A has 2 buffers left now
5	→	<seq = 2, data = m2>	•••	Message lost but A thinks it has 1 left
6	←	<ack = 1, buf = 3>	←	B acknowledges 0 and 1, permits 2-4
7	→	<seq = 3, data = m3>	→	A has 1 buffer left
8	→	<seq = 4, data = m4>	→	A has 0 buffers left, and must stop
9	→	<seq = 2, data = m2>	→	A times out and retransmits
10	←	<ack = 4, buf = 0>	←	Everything acknowledged, but A still blocked
11	←	<ack = 4, buf = 1>	←	A may now send 5
12	←	<ack = 4, buf = 2>	←	B found a new buffer somewhere
13	→	<seq = 5, data = m5>	→	A has 1 buffer left
14	→	<seq = 6, data = m6>	→	A is now blocked again
15	←	<ack = 6, buf = 0>	←	A is still blocked
16	•••	<ack = 6, buf = 4>	←	Potential deadlock

Dynamic buffer allocation. The arrows show the direction of transmission. An ellipsis (...) indicates a lost TPDU

Internet Protocol Suite
UDP & TCP

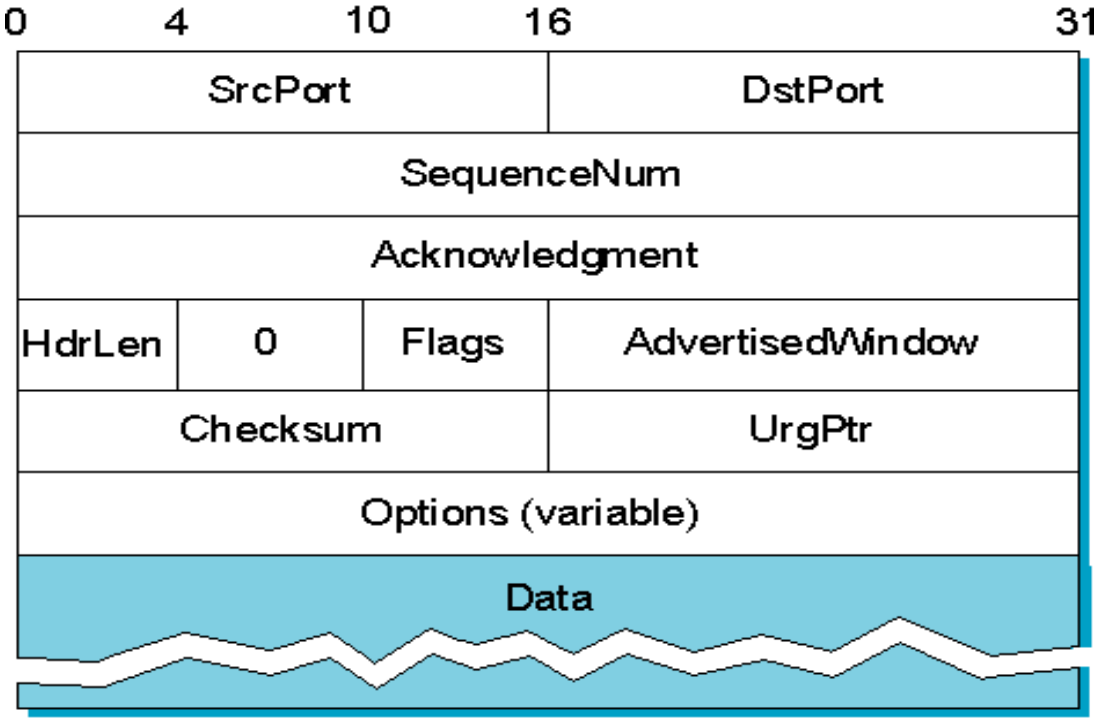
UDP - User Datagram Protocol

- A connectionless communication protocol
- Not required to establish a connection between two end points
- A UDP segment could appear at a destination at anytime
- A UDP segment contains enough information that can be used to forward the segment to the destination



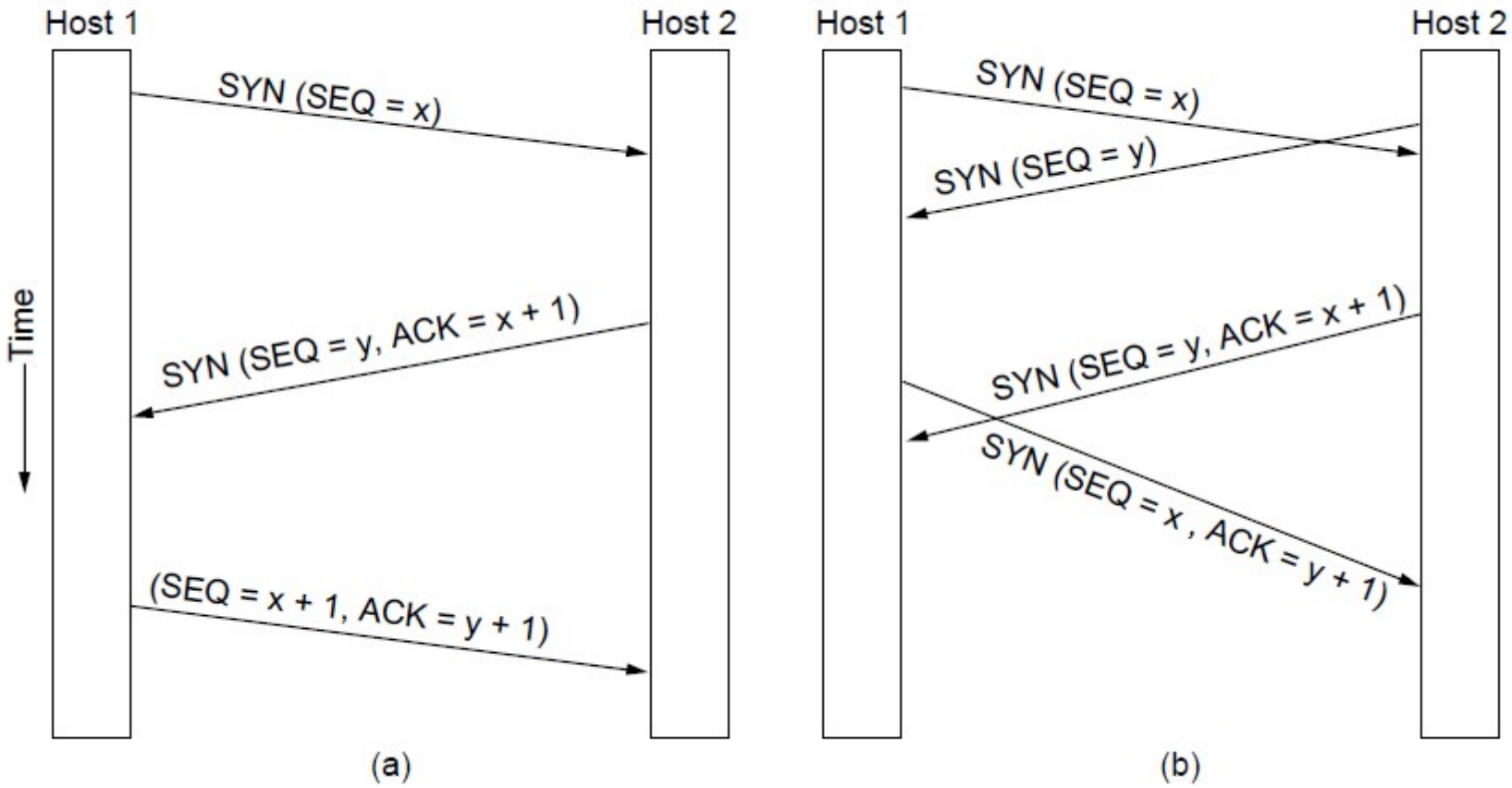
TCP - Transmission Control Protocol

- A connection-oriented communication protocol
- Byte-oriented protocol



Flags = [SYN, FIN, RESET, PUSH, URG, ACK]

TCP – Connection establishment



a) TCP connection establishment in the normal case

b) Simultaneous connection establishment on both sides

Flow control with sliding window protocol