Intro to LAN/WAN

Transport Layer

What’s Left in this Class?

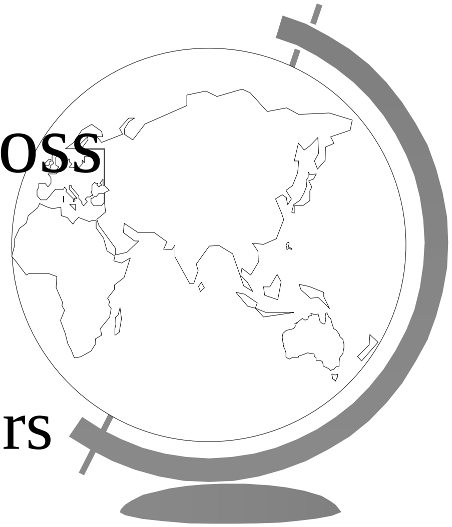
- ☞ Transport Layer ←
- ☞ Application Layer + Exam review
  - – DNS, Email, etc
- ☞ Final exam

Transport Layer Topics

- ☞ Introduction (6.1) ←
- ☞ Elements of Transport Protocols (6.2)
- ☞ Internet Transport Protocols: UDP (6.4)
- ☞ Internet Transport Protocols: TCP (6.5)

Introduction

- Efficient, reliable and cost-effective service to users (application layer)
  - despite limitations of network layer
  - *Example:* routers can drop packets (congestion) but transport must guarantee reliable delivery
  - *Analogy*: physical layer could corrupt bits but data link layer guaranteed reliable delivery
- Features (a lot like the Network layer?)
  - *Connection oriented* vs. *Connectionless*
  - *Addressing*
- If similar, then why replicate functions across layers?
  - Transport layer completely runs on on host
  - Network layer distributed over hosts and routers

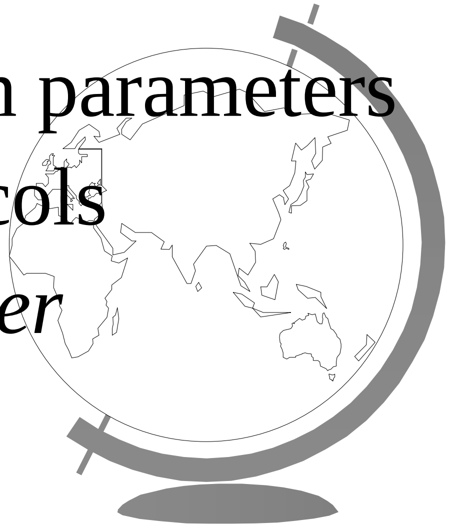


Introduction

- ☞ Similar to data link layer
  - – Mechanisms: (ACKs), sliding window, seq. numbers, etc
  - – Services: error control, sequencing, flow control…
- ☞ Difference?
  - – Data Link is point-to-point,
  - – Transport is end-to-end

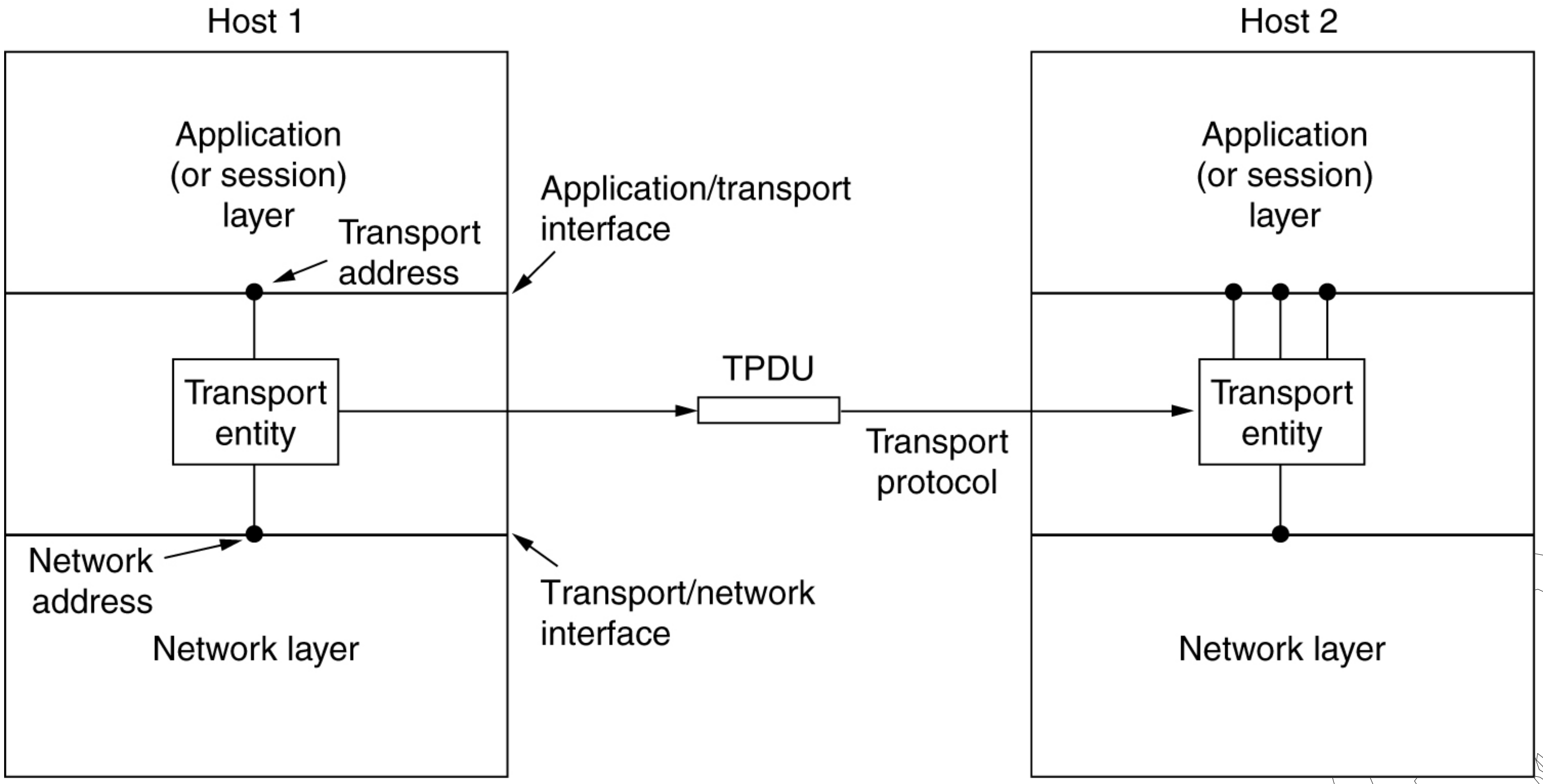
Introduction

- ☞ But Transport layer can make lower subnet *reliable (QoS)*, and gives *standard interface*
- ☞ Major boundary between *user* and *network*!
  - – Few users write code for network layer
  - – Many write code for transport layer (e.g. sockets)
- ☞ Sockets
  - – Provided clean API
  - – Programmer can just make sys calls with parameters
  - – Not worry about nuts and bolts of protocols
  - – Host address: IP address is *Network Layer*
  - – Port number: is *Transport Layer*

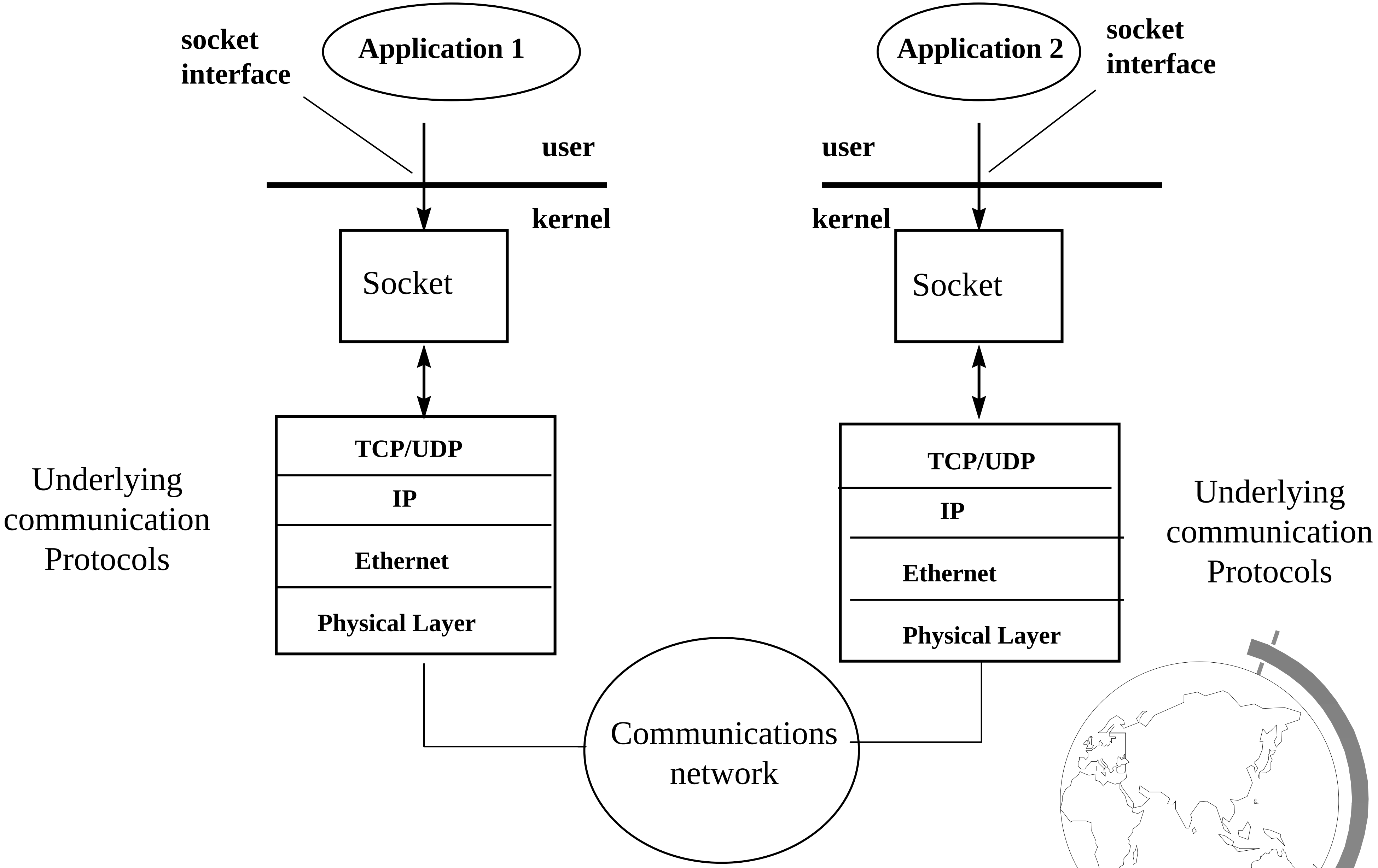


Transport Entity

The network, transport, and application layers.



- ☞ Logical location of transport entity
- ☞ Physical: OS, separate process, network card



Application 1

socket interface

user

kernel

Socket

TCP/UDP

IP

Ethernet

Physical Layer

Underlying communication Protocols

Application 2

socket interface

user

kernel

Socket

TCP/UDP

IP

Ethernet

Physical Layer

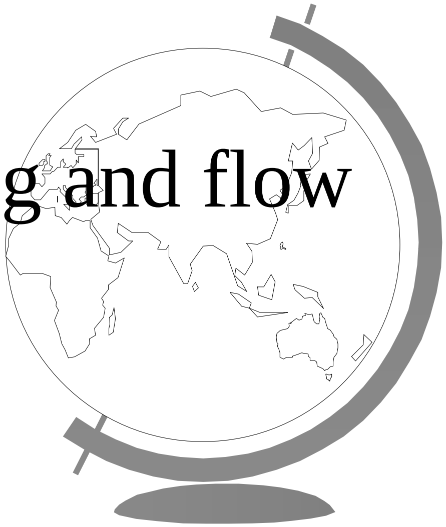
Underlying communication Protocols

Communications network

Figure 2.16

Transport Protocol

- Like Data Link layer:
  - error control, sequencing, flow control…
- But different:
  - must specify router (data link layer always same)
  - destination may be down
  - network may store packets
  - many lines and variance make buffering and flow control different



Generic Transport Layer Primitives

- Already saw these in sockets earlier
- LISTEN:
  - block until a process connects
- CONNECT:
  - attempt to establish connection
- SEND:
  - send information
- RECEIVE:
  - receive information
- DISCONNECT:
  - this side wants to release connection

Transport Layer Topics

- ☞ Introduction (6.1)
- ☞ Elements of Transport Protocols (6.2) ←
- ☞ Internet Transport Protocols: UDP (6.4)
- ☞ Internet Transport Protocols: TCP (6.5)

Elements of Transport Protocols

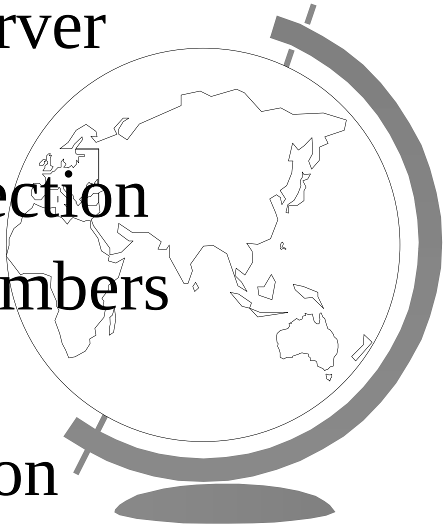
- ☞ Addressing
- ☞ Connection establishment
- ☞ Connection release
- ☞ Flow control and buffering
- ☞ Multiplexing
- ☞ Crash recovery

Finding a Server

- ☞ "Connect to a Server" is a Transport level service
- ☞ How do you find it?
  - – *service mapper* - names to transport layer address
  - – *name server*
- ☞ Analogy
  - – how do you find phone number?
- ☞ Standard servers wait at well-known port
  - – **`ftp 21/tcp`**
  - – **`telnet 23/tcp`**
  - – **`finger 79/tcp`**
  - – **`snmp 161/udp`**

Establishing a Connection

- Imagine severely congested subnet
  - Network can delay, lose, duplicate packets
  - Example: connection to bank to transfer big money!!
- Consequences
  - Connection can happen twice!
- Solution(s)
  - One time only transport address (problem: server mapper cannot work)
  - Use *unique sequence numbers* for each connection
  - When establish connection, exchange seq. numbers
  - *three-way handshake*
  - prevents establishment of unwanted connection

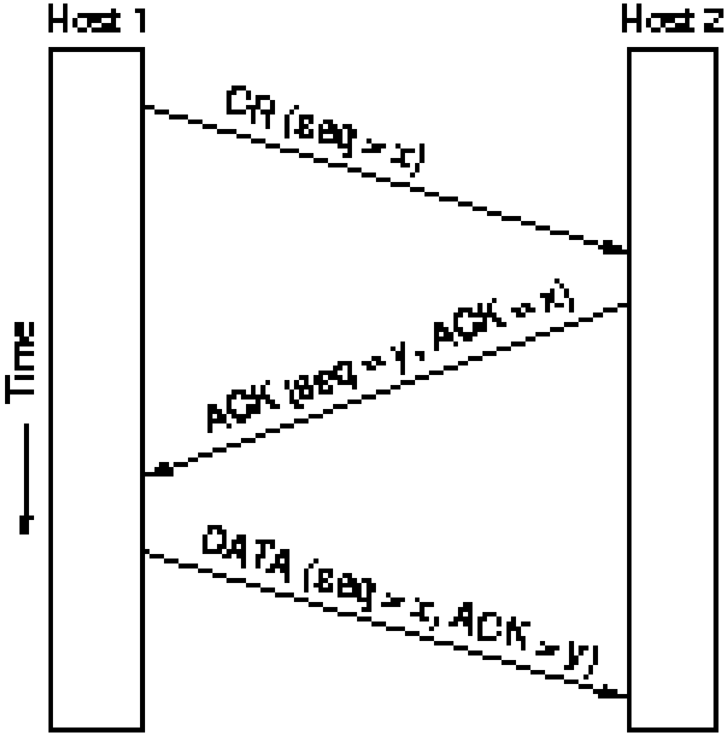


Three-Way Handshake

Tomlinson (1975)

CR = Connection
Request

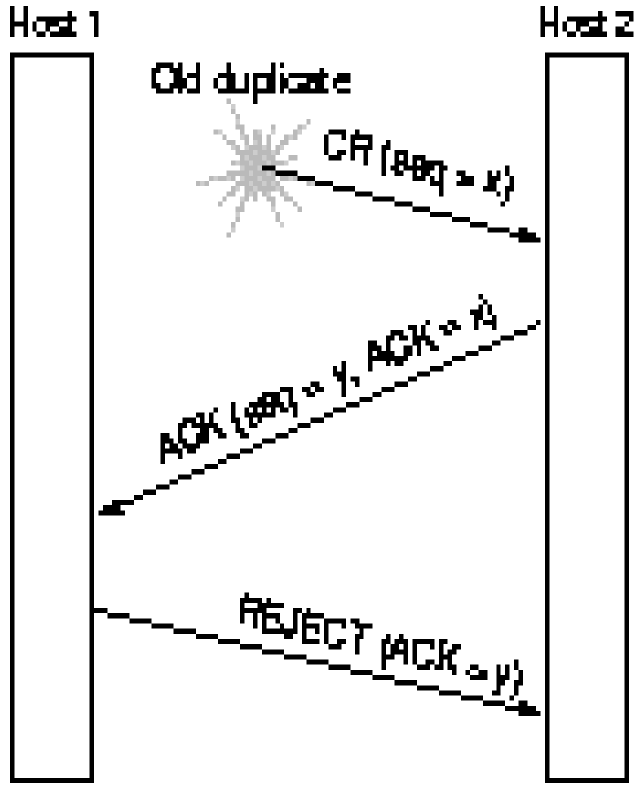
ACK = Connection
Accepted



Three-Way Handshake Handles Problems

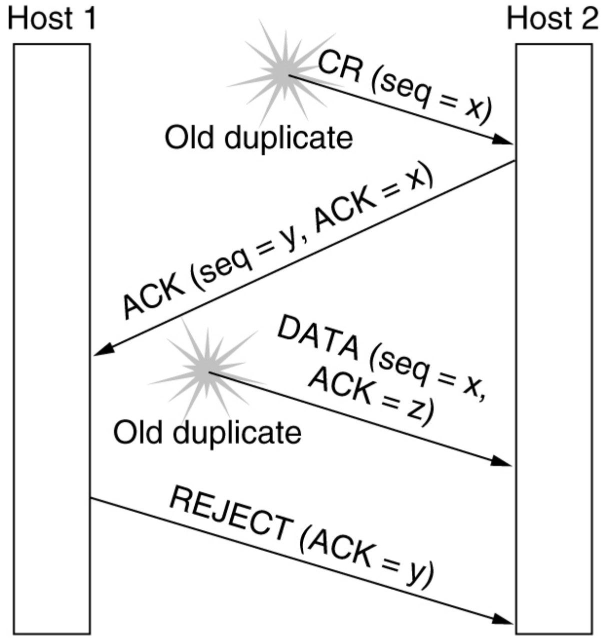
Case 1: delayed duplicate

Key: host 1 knows it has seen ACK before



Case 2: delayed connection request, ACK

Key: Wrong data seq. No.

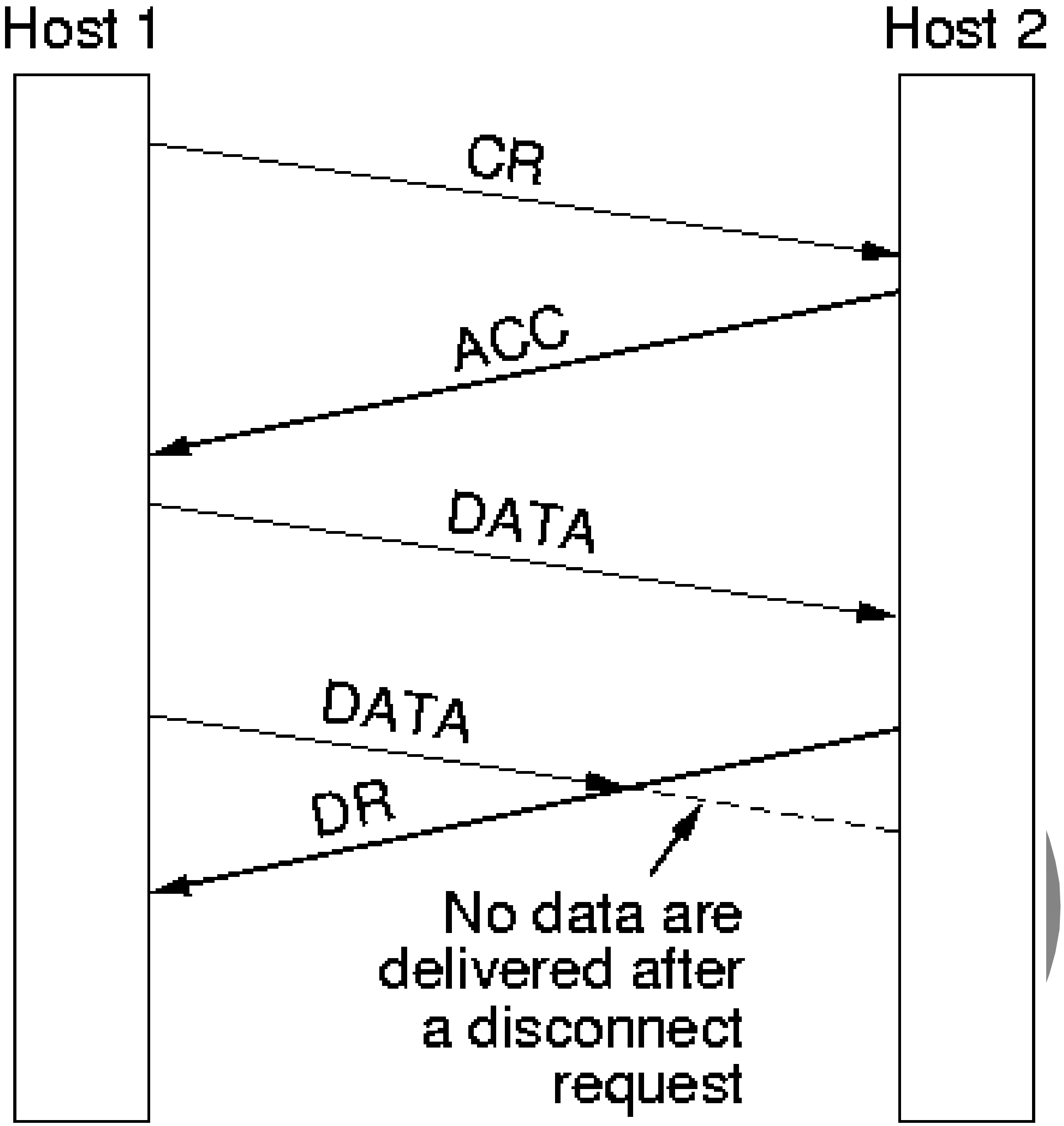


(c)

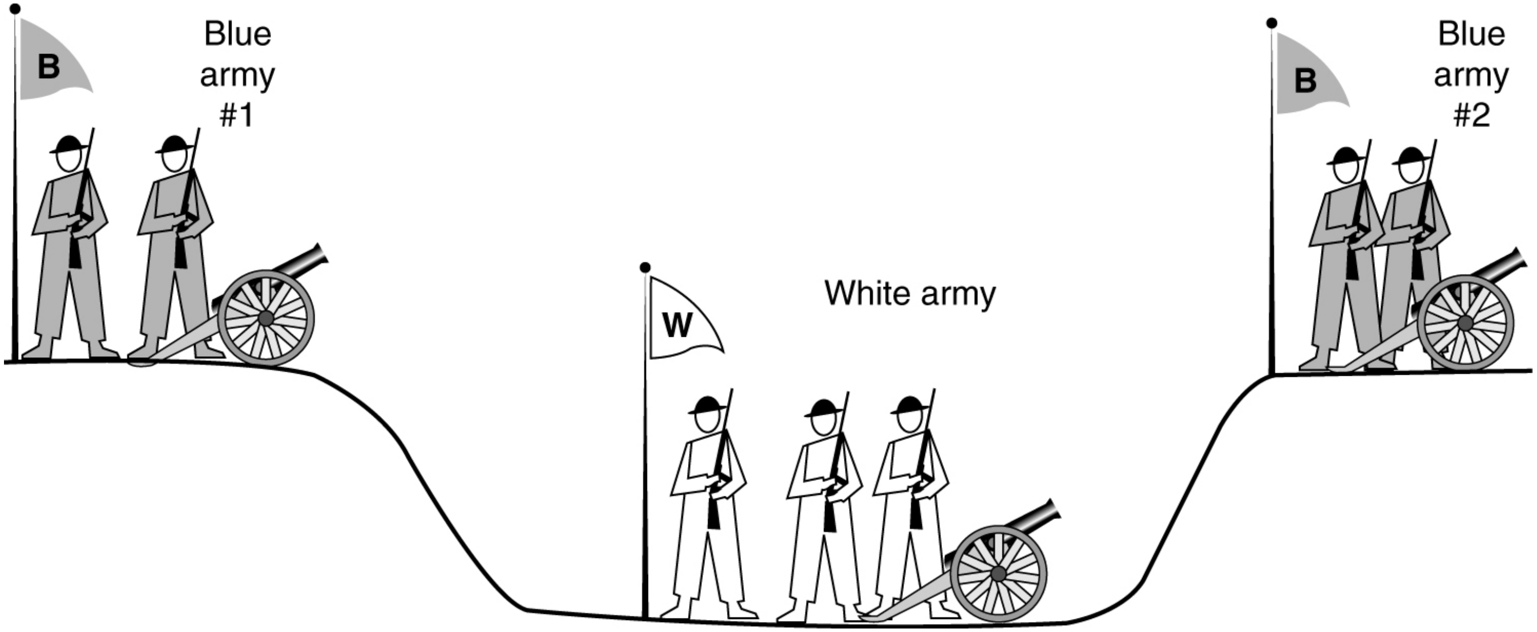
Releasing a Connection

- ☞ Asymmetric release can result in data loss

- ☞ Symmetric release easy?
  - – "I'm done"
  - – "Me, too"



Two-Army Problem



- No safe solution
- Use 3-way handshake with timers (fig 6-14)

Elements of Transport Protocols

- Flow control and buffering
  - Sliding window can still be used (flow control, etc)
  - Host may have several connections, so buffering may be tough
  - Example: protocol 6 with 64 connections and 4-bit sequence numbers require 1024 buffers
- Multiplexing
  - Several applications (email, web, etc) on same host may share one network address
  - Multiplexing: how to combine
- Crash recovery
  - How to recover when host crashes