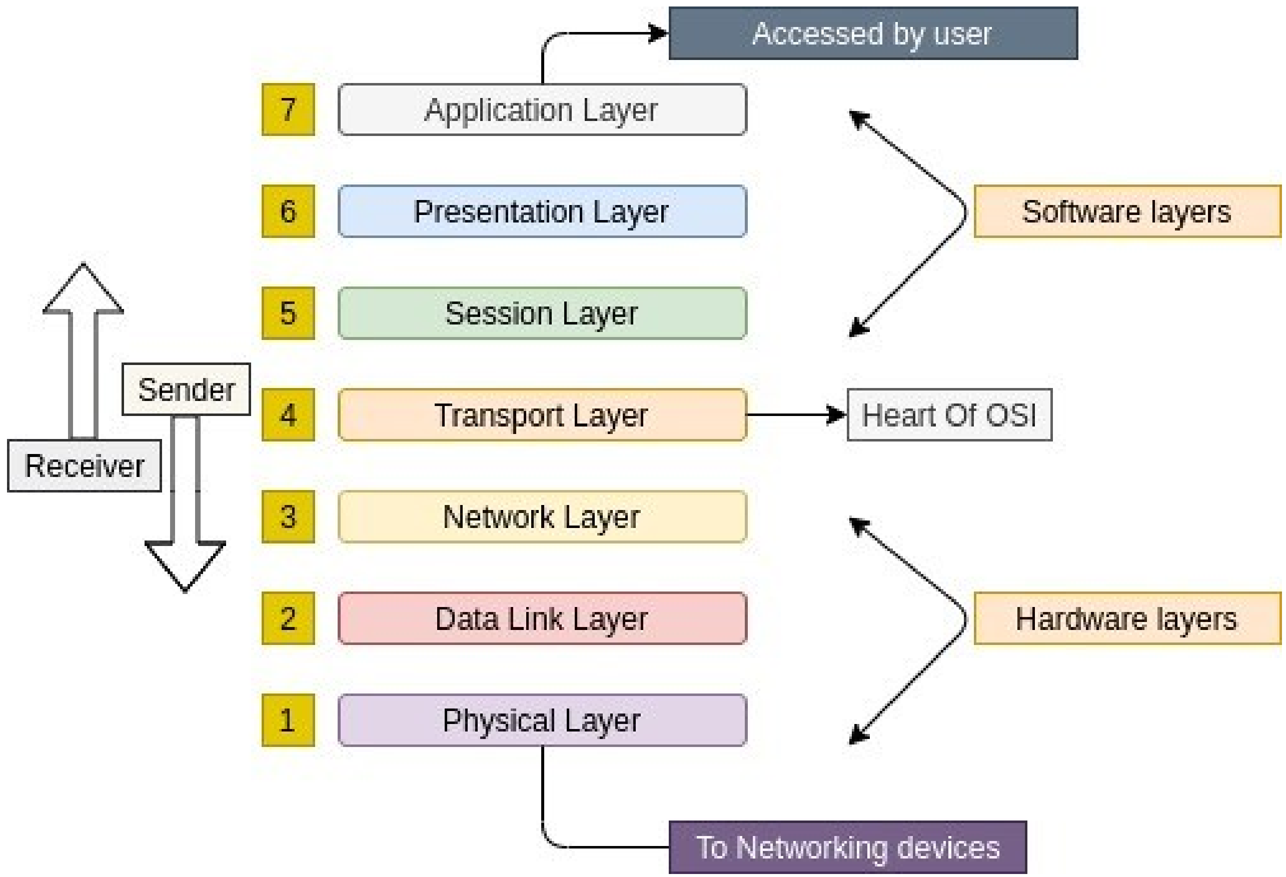
Transport Layer

Chapter 6



In the OSI layer model, the Lower layer provides services to the upper layer

Transport Layer

- Transport layer is a process-to-process communication layer that is responsible for delivering the entire message.

- A process is an application running on the host system. For process-to-process communication, the transport layer uses a port number assigned to each process uniquely.

Data Link vs Transport Layer

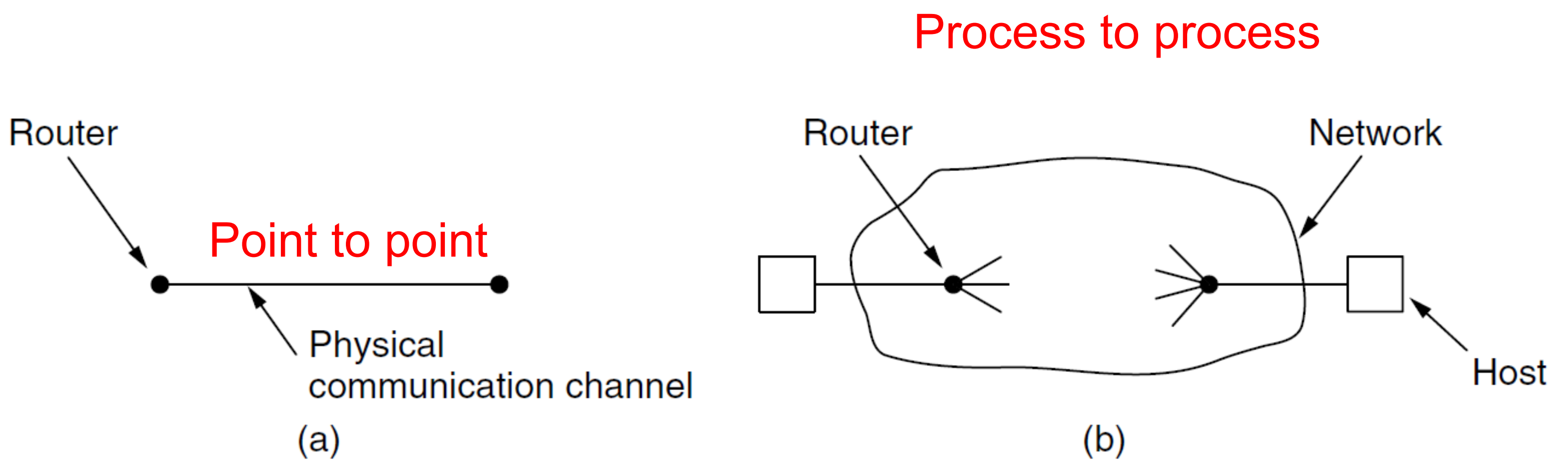
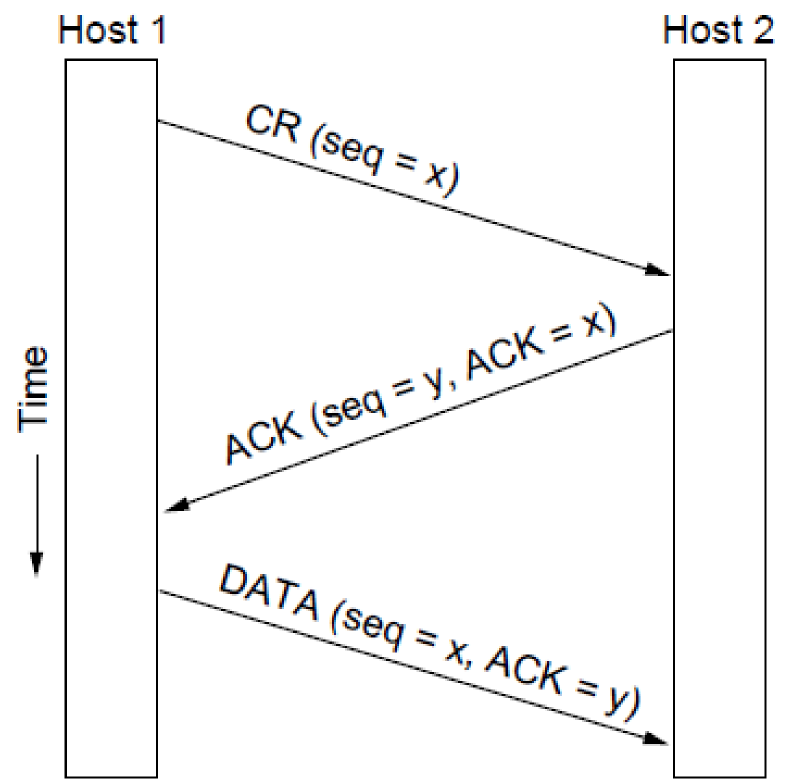


Figure 6-7. (a) Environment of the data link layer. (b) Environment of the transport layer.

Connection Establishment



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

Connection Establishment

Establishing a connection sounds easy, but it is actually surprisingly tricky.

- The problem occurs when the network can lose, delay, corrupt, and duplicate packets.
- Only the **common cases need be implemented efficiently** to obtain good network performance, but the protocol must be able to cope with the uncommon cases without breaking.

It's **not necessary to perform well** for uncommon cases. It's enough, if the protocol is able to cope with it without breaking.

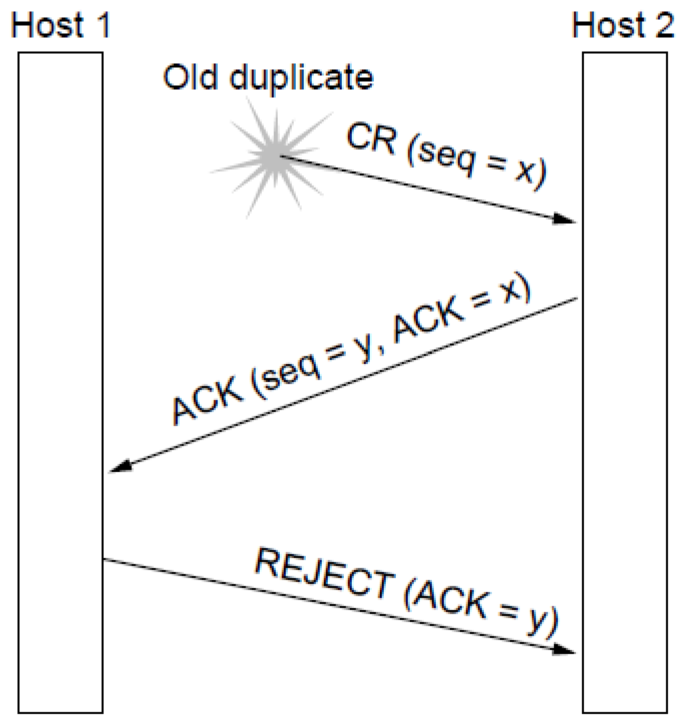
Connection Establishment

The crux of the problem is that the delayed duplicates are thought to be new packets.

We cannot prevent packets from being duplicated and delayed. But if and when this happens, the packets must be rejected as duplicates and not processed as fresh packets.

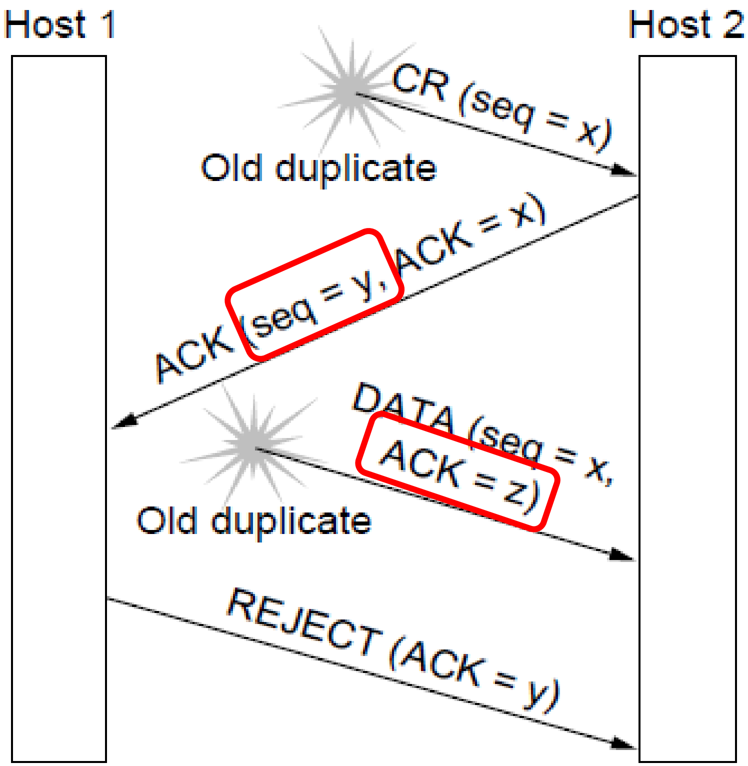
The problem can be addressed in various ways, none of them very satisfactory.
Finally to solve this specific problem, Tomlinson (1975) introduced **the three-way handshake.**

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.

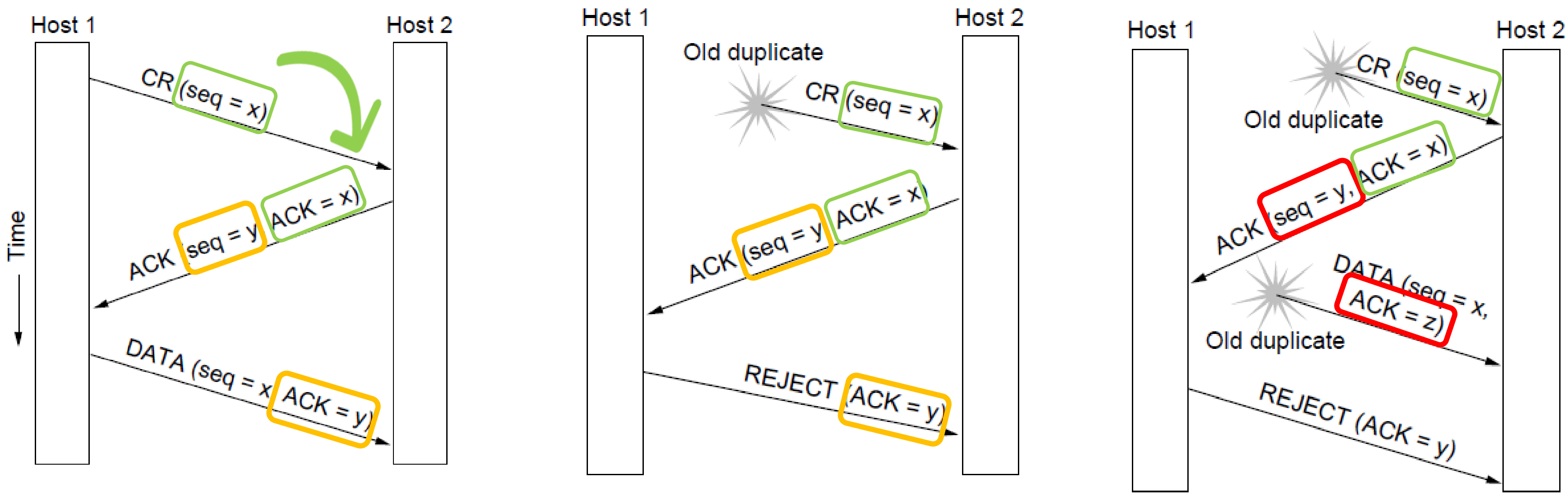
Connection Establishment



Expected ACK=y.
When the second delayed segment arrives at host 2, the fact that *z* **has been acknowledged** rather than *y* tells host 2 that this is an old duplicate.

The important thing to realize here is that there is **no combination of old segments** that can cause the **protocol to fail** and have a connection set up by accident when no one wants it.

Connection Establishment



Connection Release

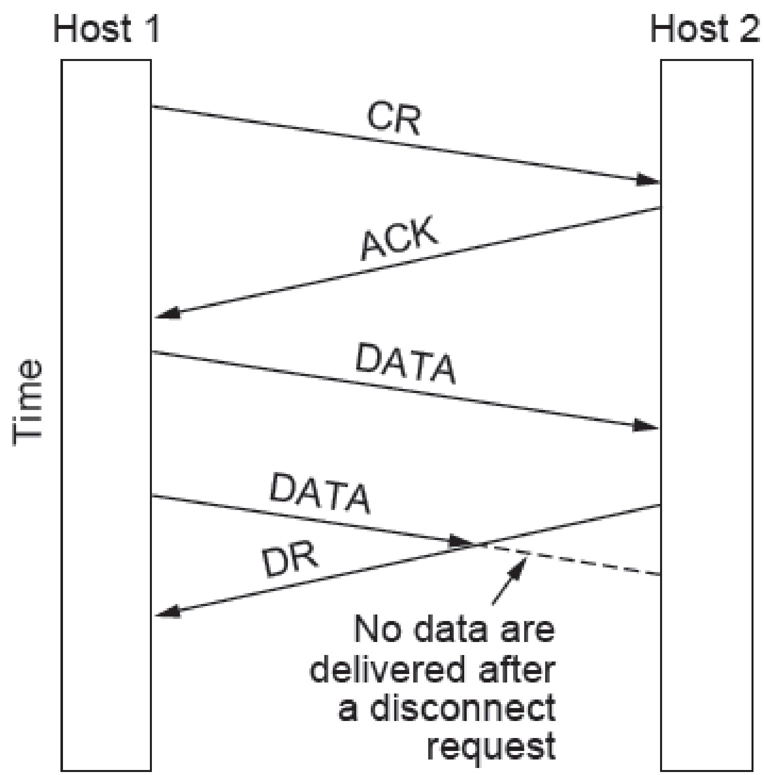
There are two styles of terminating a connection:

- Asymmetric release and
- symmetric release.

- **Asymmetric release** is the way the telephone system works: when one party hangs up, the connection is broken.
  - Asymmetric release is abrupt and may result in data loss.
- **Symmetric release** treats the connection as two separate unidirectional connections and requires each one to be released separately.
  - Each direction is released independently of the other one.
  - Here, a host can continue to receive data even after it has sent a DISCONNECT segment.

Connection Release

Asymmetric release



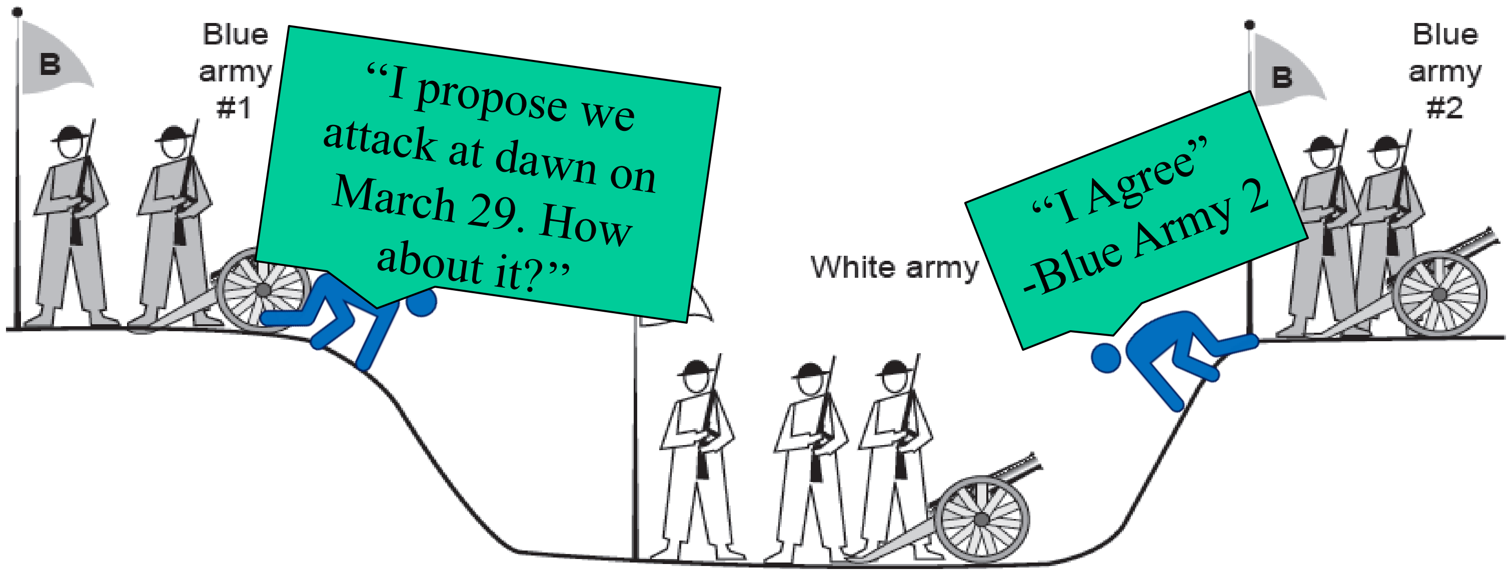
Abrupt disconnection with loss of data

Connection Release

Symmetric release

Two blue armies want to synchronize their attacks to defeat White army.
The question is: does a protocol exist that allows the blue armies to win?

Will the attack happen?



The two-army problem

Connection Release

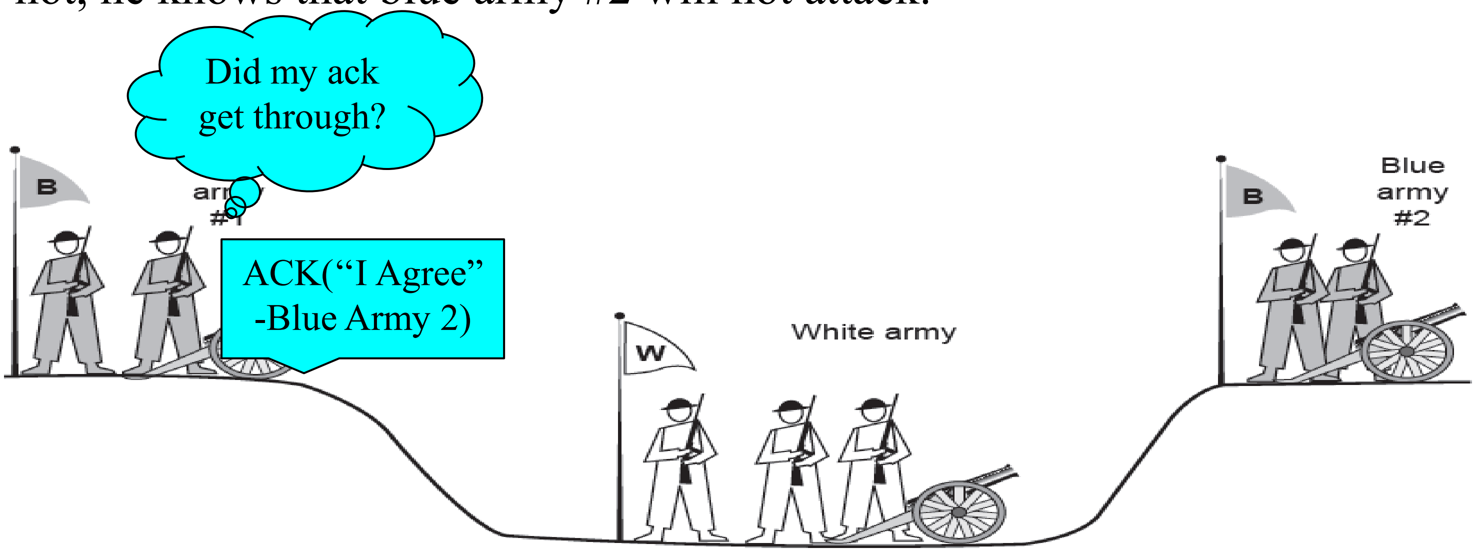
Symmetric release

The question is: does a protocol exist that allows the blue armies to win?

Will the attack happen?

let us improve the protocol by making it a three-way handshake. Assuming no messages are lost**, blue army #2 will get the acknowledgement**.

- ➢ But the commander of blue army #1 will now hesitate.
- ➢ After all, he does not know if his acknowledgement got through, and if it did not, he knows that blue army #2 will not attack.



The two-army problem

Connection Release

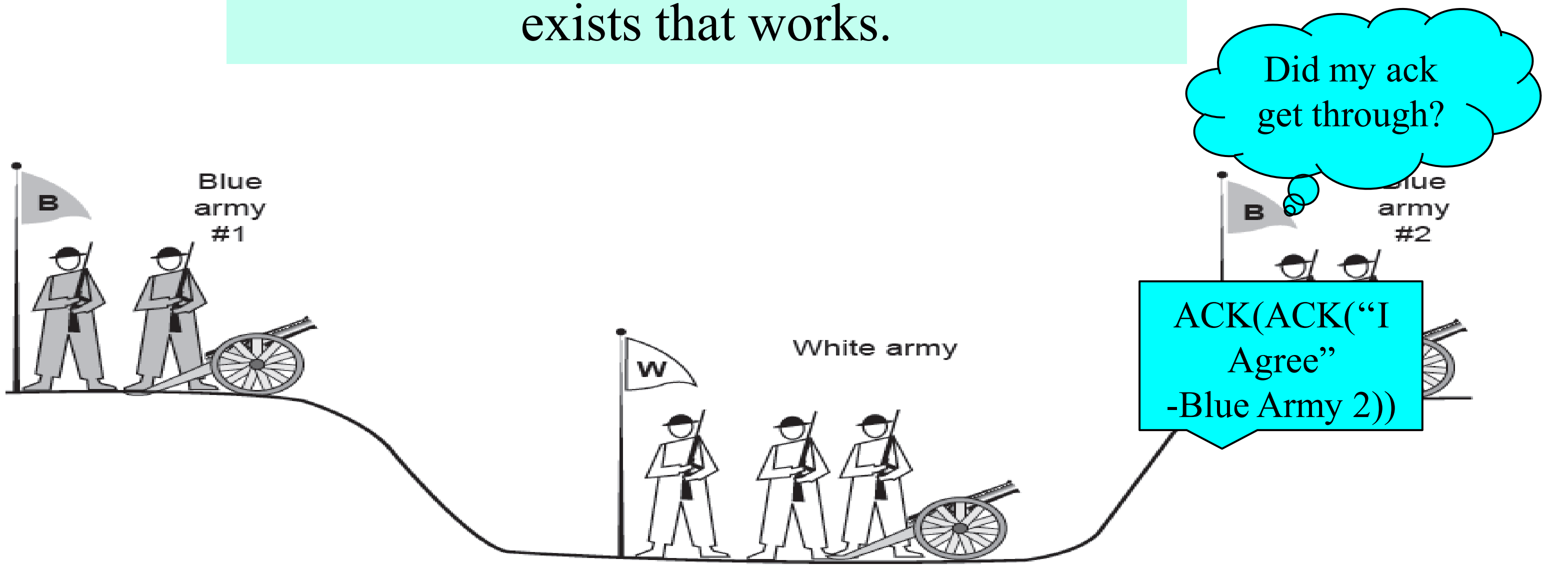
Symmetric release

The question is: does a protocol exist that allows the blue armies to win?

Will the attack happen?

We could now make a four-way handshake protocol, but that does not help either.

In fact, it can be proven that no protocol exists that works.



Since **the sender** of the **final message** can never be sure of its arrival, he will not risk attacking.

Connection Release

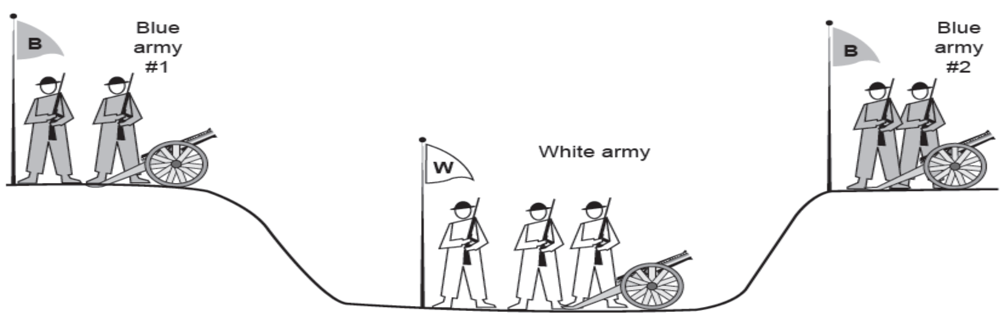
Symmetric release

- To see the relevance of the two-army problem to releasing connections, rather than to military affairs, **just substitute "disconnect" for "attack.**
- If neither side is prepared to disconnect until it is convinced that the other side is prepared to disconnect too**, the disconnection will never happen.**

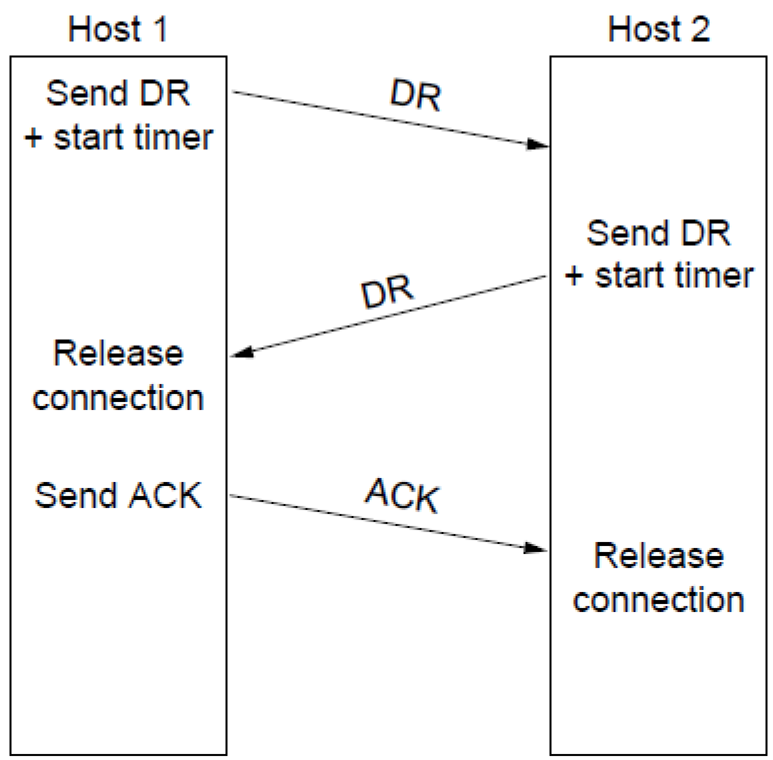
So if there is dependency disconnection will never happen.

We can avoid this quandary by

- Foregoing the need for agreement and
- Pushing the problem up to the transport user, letting each side independently decide when it is done.



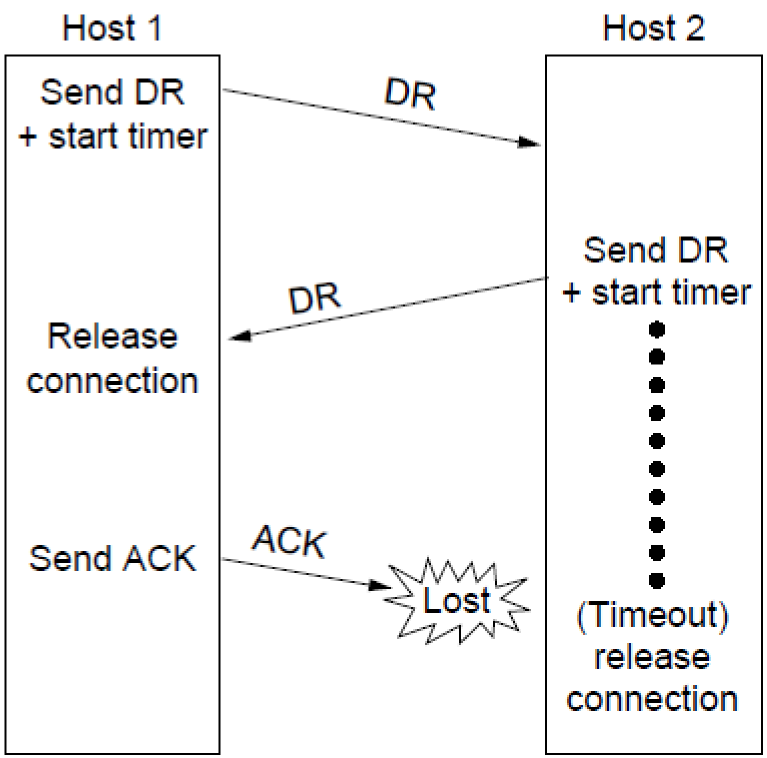
Connection Release



Scenarios for releasing a connection.

(a) Normal case of three-way handshake

Connection Release



Final ACK lost:

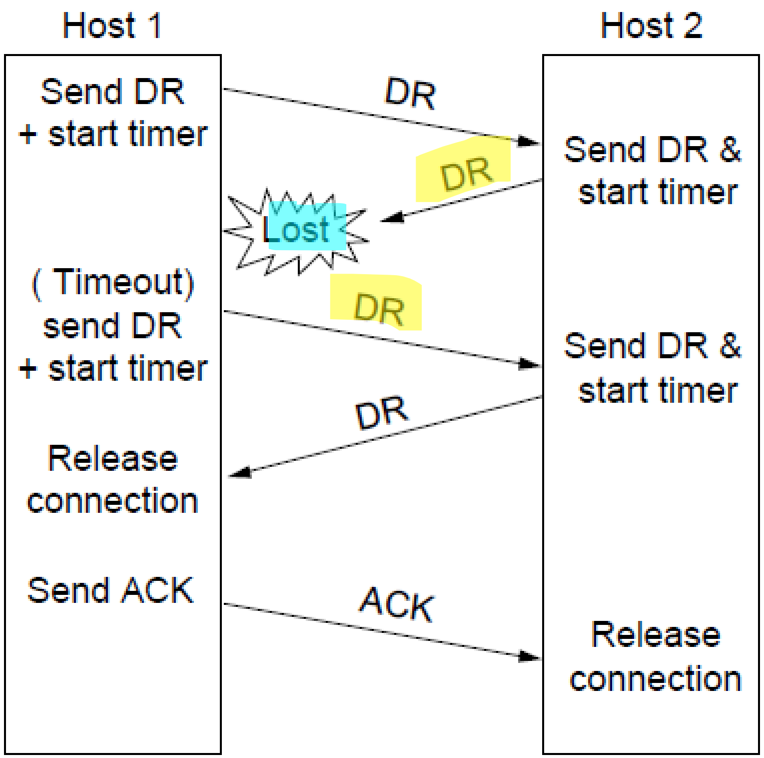
The situation is saved by the timer.

When the timer expires, the connection is released anyway.

Scenarios for releasing a connection.

(b) **Final ACK lost.**

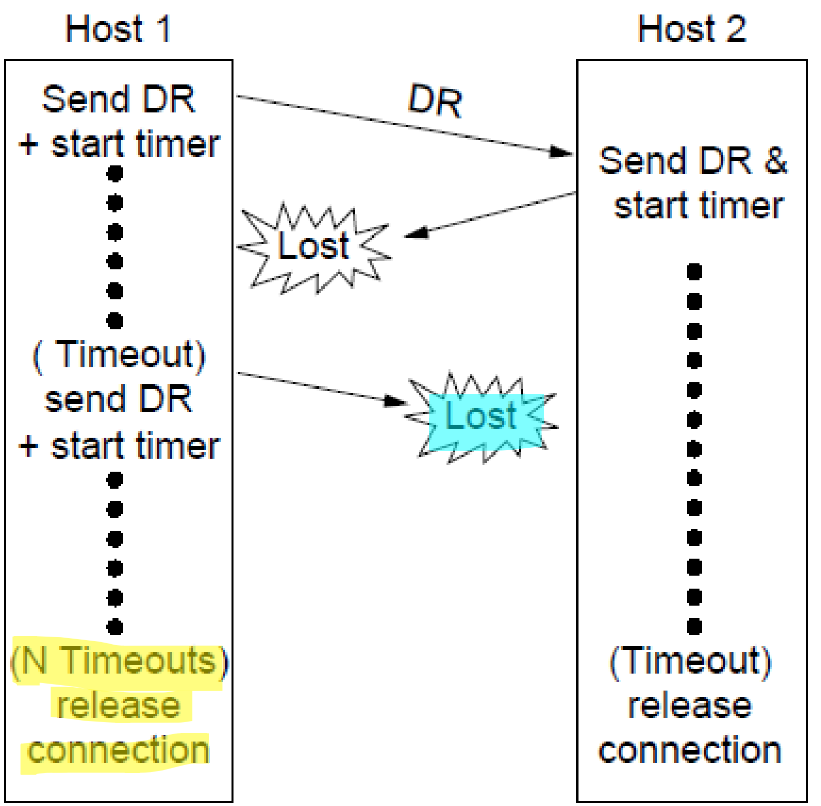
Connection Release



Second DR being lost:
The user initiating the disconnection will not receive the **expected response**, will time out, and will start all over again.

Scenarios for releasing a connection.
(c) Response lost

Connection Release



Response lost and subsequent DRs lost:
This is the same as Fig (c) except that now we assume all the repeated attempts to retransmit the DR also fail due to lost segments.
After *N* retries, the sender just gives up and releases the connection.
Meanwhile, the receiver times out and also exits.

Scenarios for releasing a connection.
(d) Response lost and subsequent DRs lost.

Introduction to UDP (1)

32 Bits

Source port	Destination port
UDP length	UDP checksum

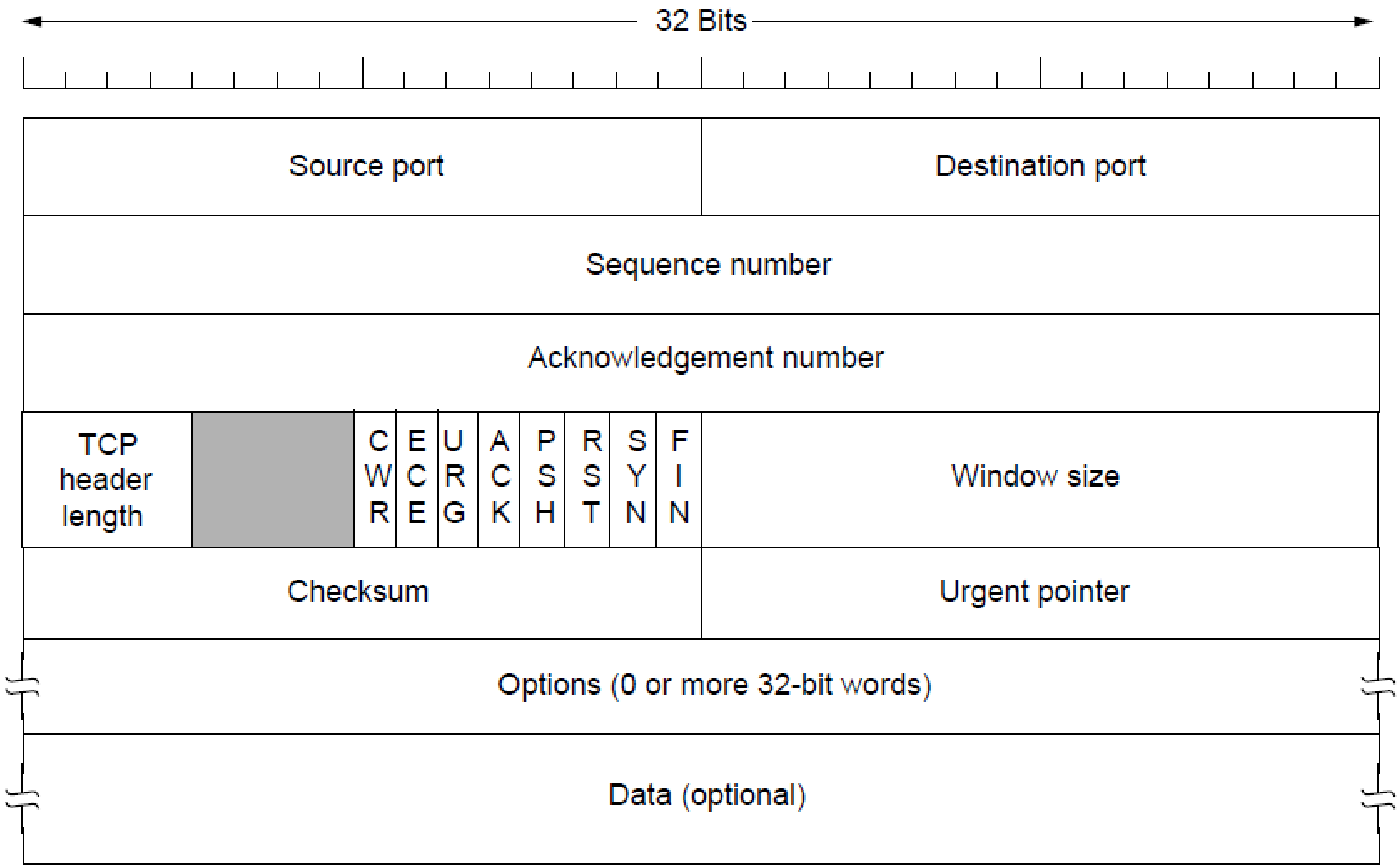
The UDP header.

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

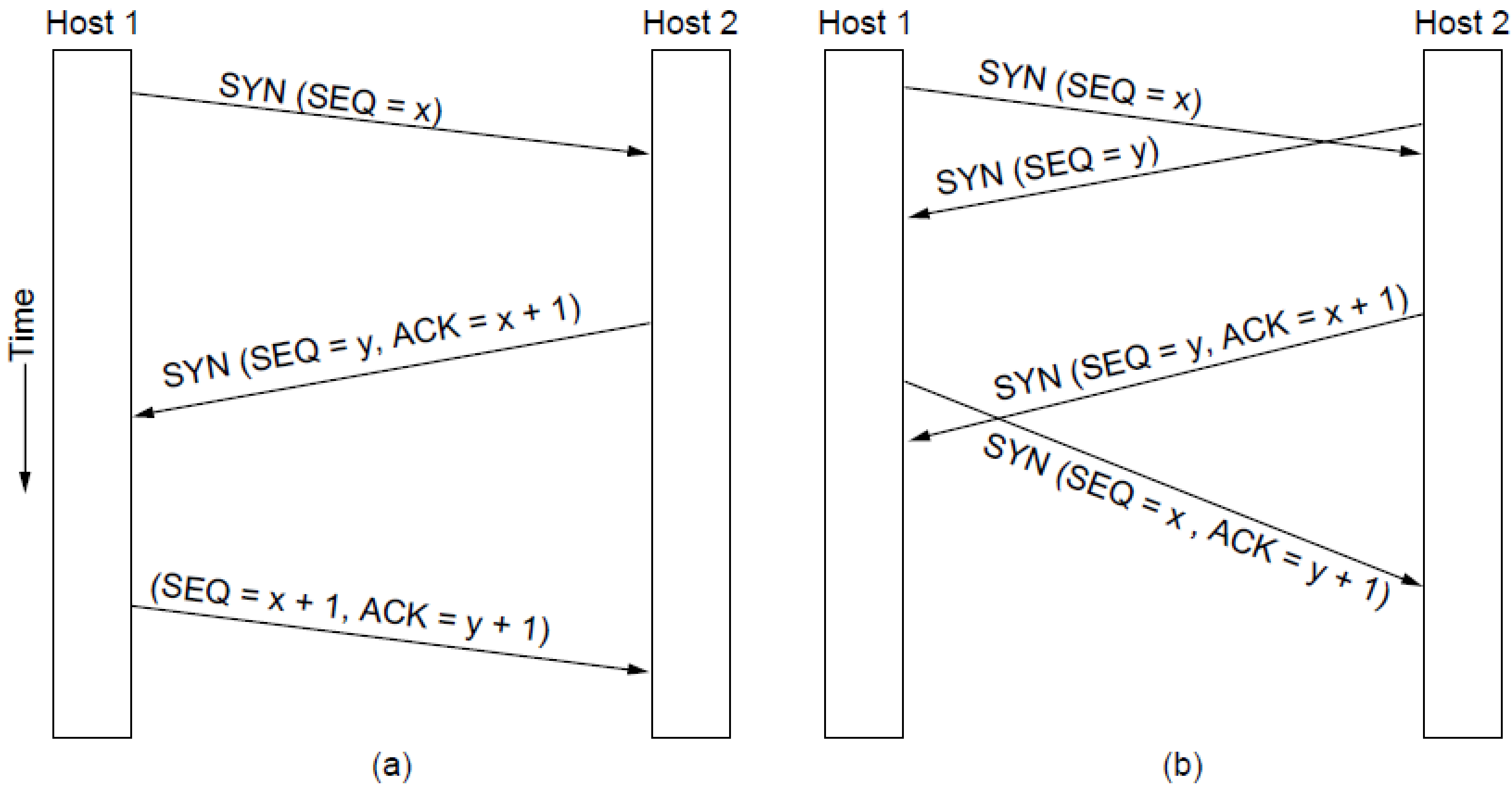
Some assigned ports

The TCP Segment Header



The TCP header.

TCP Connection Establishment



(a) TCP connection establishment in the normal case.

(b) Simultaneous connection establishment on both sides.