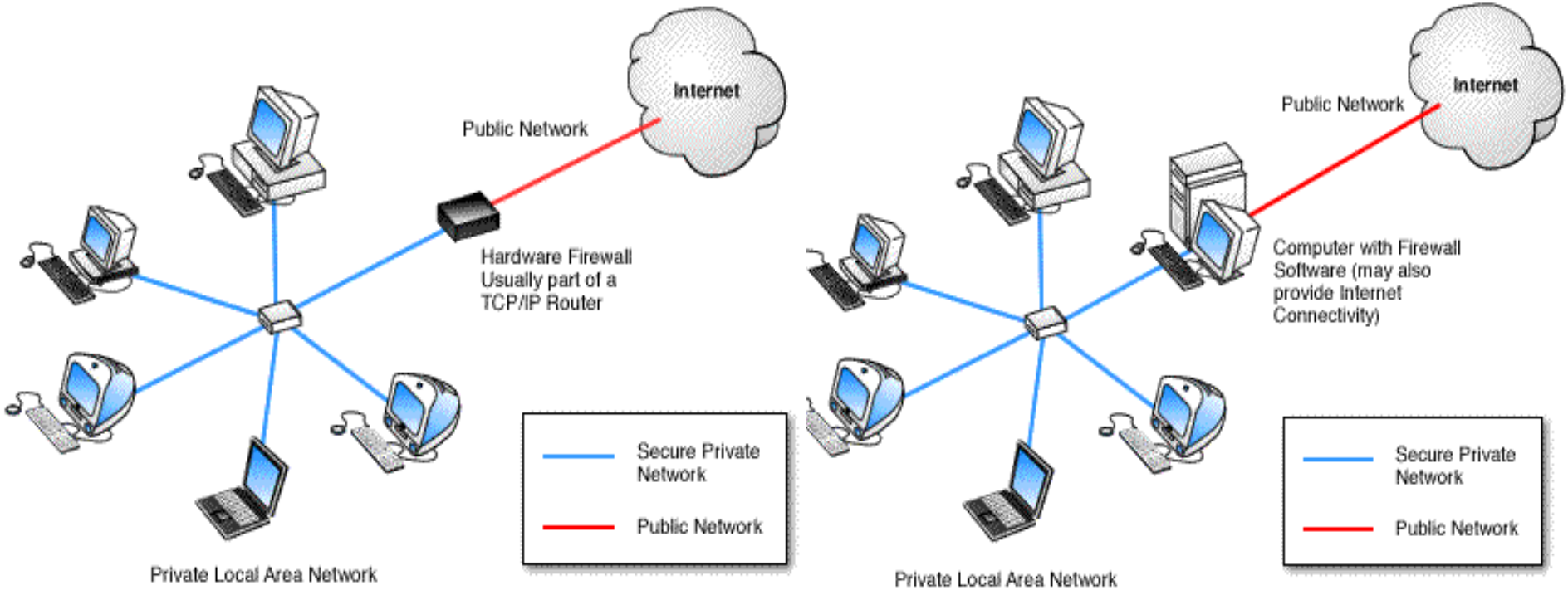
Access Control: Firewalls (1)

World is divided in **good** and **bad** guys --->

access control (security checks) at a single point of entry/exit:

- in medieval castles: drawbridge
- in corporate buildings: security/reception desk
- in computer networks: **firewalls**

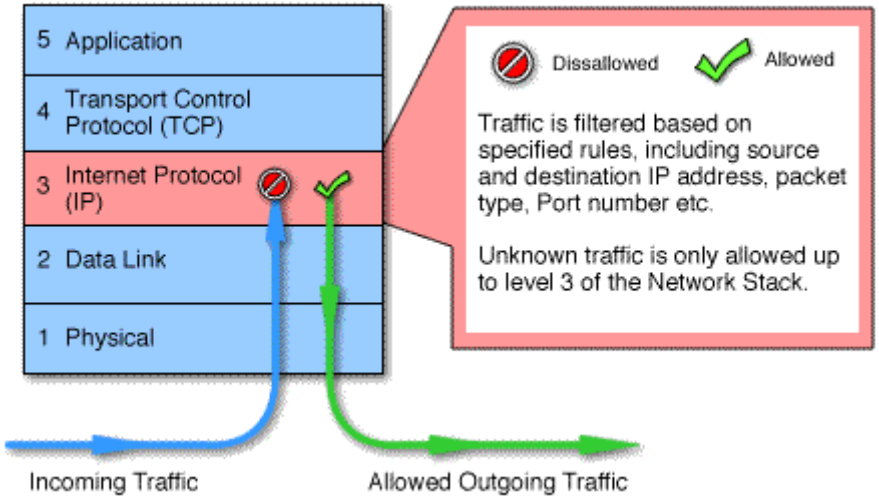


Access Control: Firewalls (2)

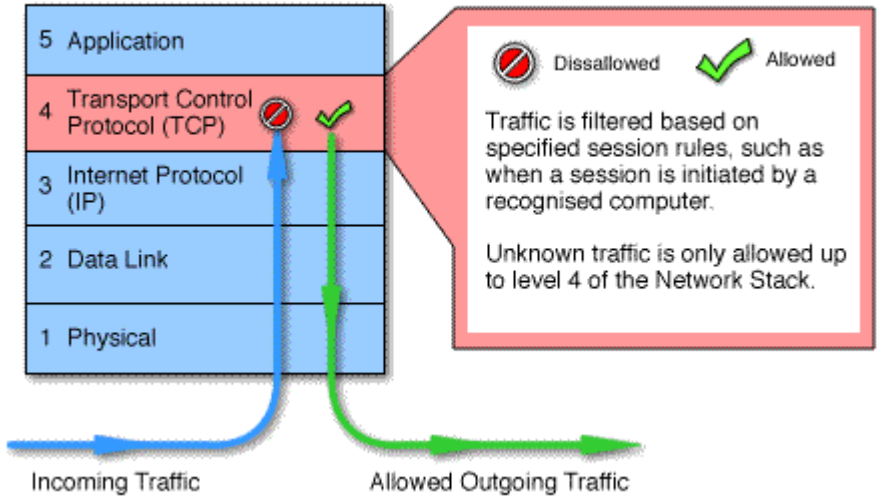
Firewalls fall into four broad categories dependent on which layer they are implemented:

- packet filters
- circuit-level gateways
- application-level gateways
- multilayer inspection firewall

Packet Filtering Firewall



Circuit-Level Gateway



Access Control: Firewalls (3)

Example of packet filtering rules

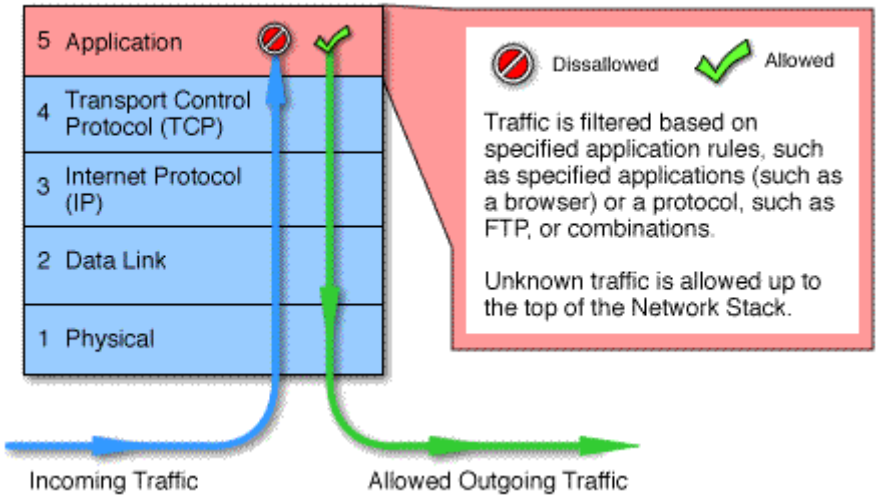
Rule	Source Address	Destination Address	Action	Comments
R1	111.11/16	222.22.22/24	permit	Let datagrams from Bob's university network into a restricted subnet.
R2	111.11.11/24	222.22/16	deny	Don't let traffic from Trudy's subnet into anywhere within Alice's network.
R3	0.0.0.0/0	0.0.0.0/0	deny	Don't let traffic into Alice's network.

Results of packet filtering

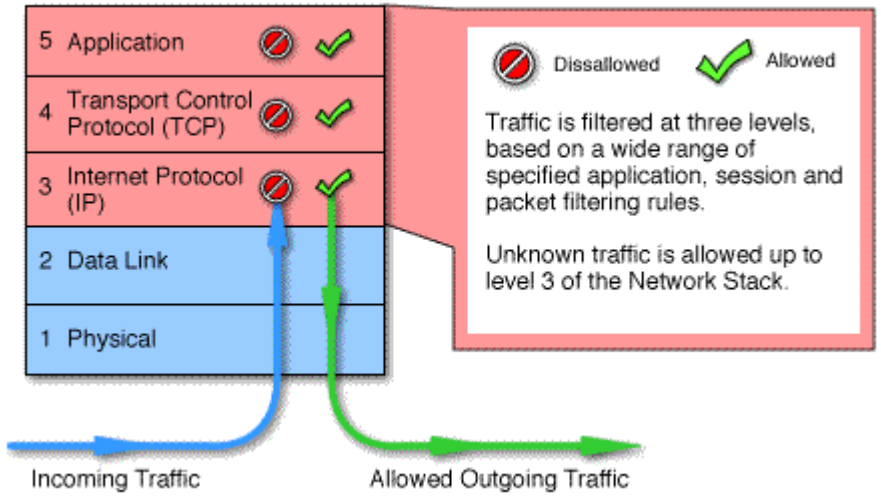
Datagram Number	Source IP Address	Destination IP Address	Desired Action	Action Under R2, R1, R3	Action Under R1, R2, R3
P1	111.11.11.1 (hacker subnet)	222.22.6.6 (corp.net)	deny	deny (R2)	deny (R2)
P2	111.11.11.1 (hacker subnet)	222.22.22.2 (special subnet)	deny	deny (R2)	permit (R1)
P3	111.11.6.6 (univ. net, not the hacker subnet)	222.22.22.2 (special subnet)	permit	permit (R1)	permit (R1)
P4	111.11.6.6 (univ. net, not the hacker subnet)	222.22.6.6 (corp. net)	deny	deny (R3)	deny (R3)

Access Control: Firewalls (4)

Application-Level Gateway



Multilayer Inspection Firewall



Multilayering

- offer a high level of security and good performance but are expensive
- due to their complexity they are potentially less secure than simpler types of firewalls ---> require administration by highly competent personnel

Firewalls introduce a trade-off between the degree of communication with the outside world and the level of security

- filters **cannot** prevent spoofing of IP addresses and port numbers ---> often use a all-or-nothing policy
- gateways can have software bugs ---> may allow attackers to penetrate them
- using wireless communication or dial-up modems ---> possibility of bypassing firewalls

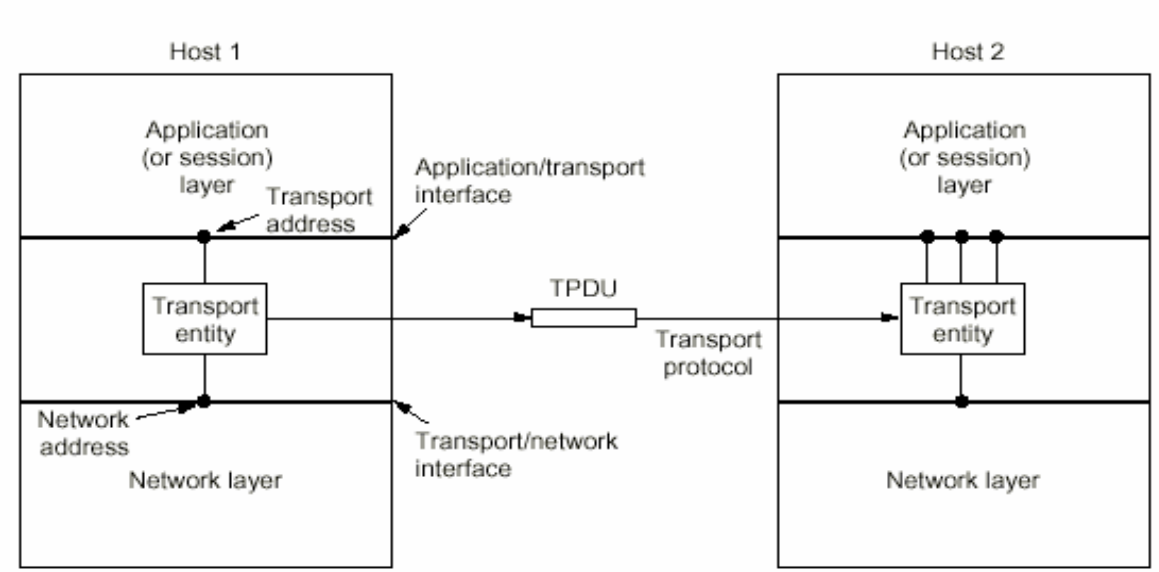
Transport Layer(1)

Ultimate goal:

To provide efficient, **reliable**, and cost-effective service to its users, i.e. processes in the application layer.

- To achieve this goal, the transport layer makes use of the services provided by the network layer.
- This work is done by the so-called **transport entity.**
- It may be implemented
  - in the OS kernel
  - in a separate user process
  - in a library package bound into network applications
  - on a separate network interface card

Logical relationship between the involved layers



Transport Layer(2)

Why using two distinct layers (network and transport) for similar services:

The network layer is part of the communication subnet and is run by the carrier (at least for WAN´s)

---> The transport layer forms the major boundary between the provider and the user of a (reliable) data transmission service

---> Thanks to the transport layer, it is possible for application programs to run, using a standard set of primitives, on different networks with possibly different subnet interfaces and reliability features

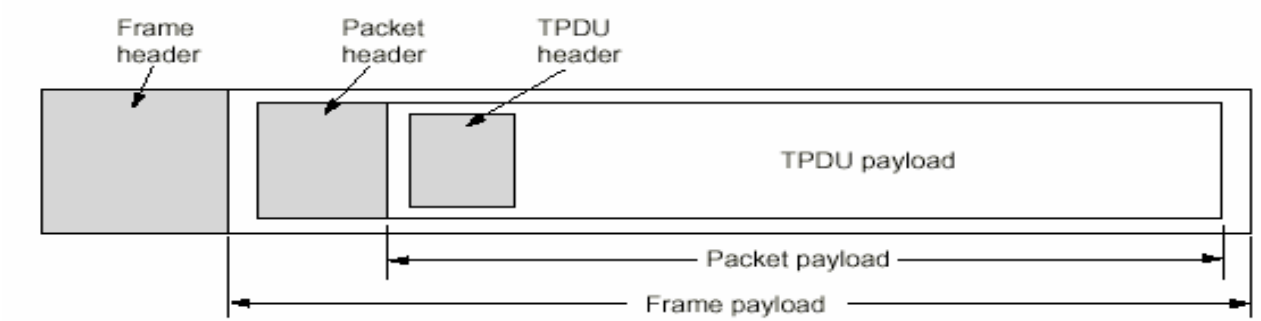
---> It fulfills the key function of isolating the upper layers from the technology, design, and imperfections of the subnet

---> In essence, it makes it possible for the transport service to be more reliable than the underlying network service

---> This is the main difference between the two services:

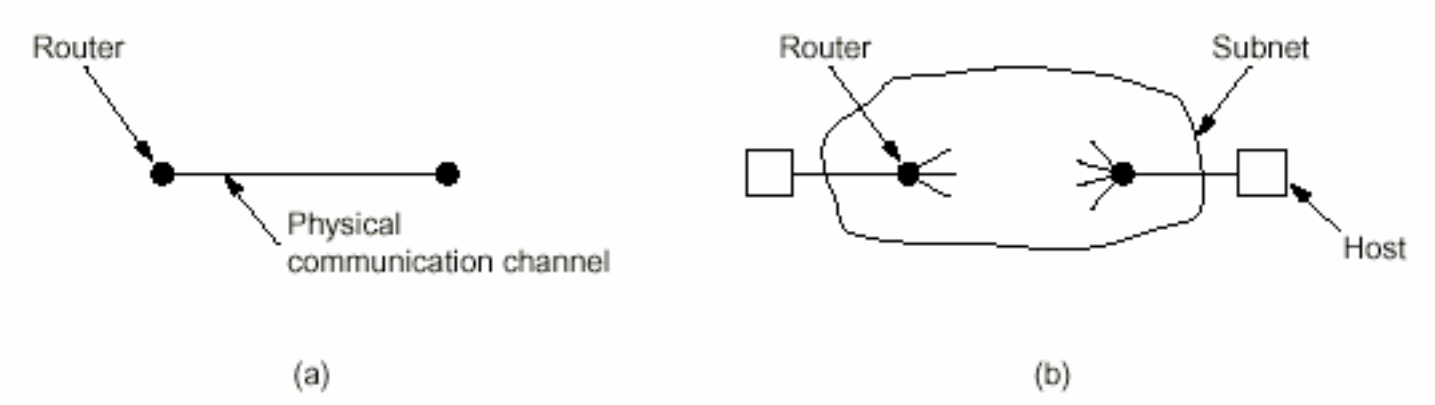
- real networks can lose packets, are not error-free because of its connectionless (datagram) type
- the (connection-oriented) transport service provides a reliable service on top

Nesting of TPDU´s, packets, and frames



Transport Layer(4)

Different environments for data link and transport protocols, resp.

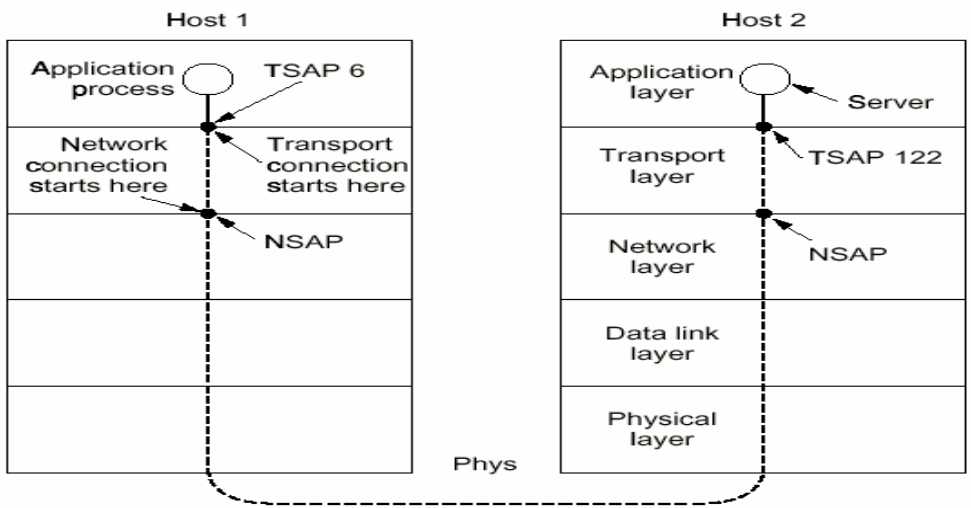


Implications for transport protocols:

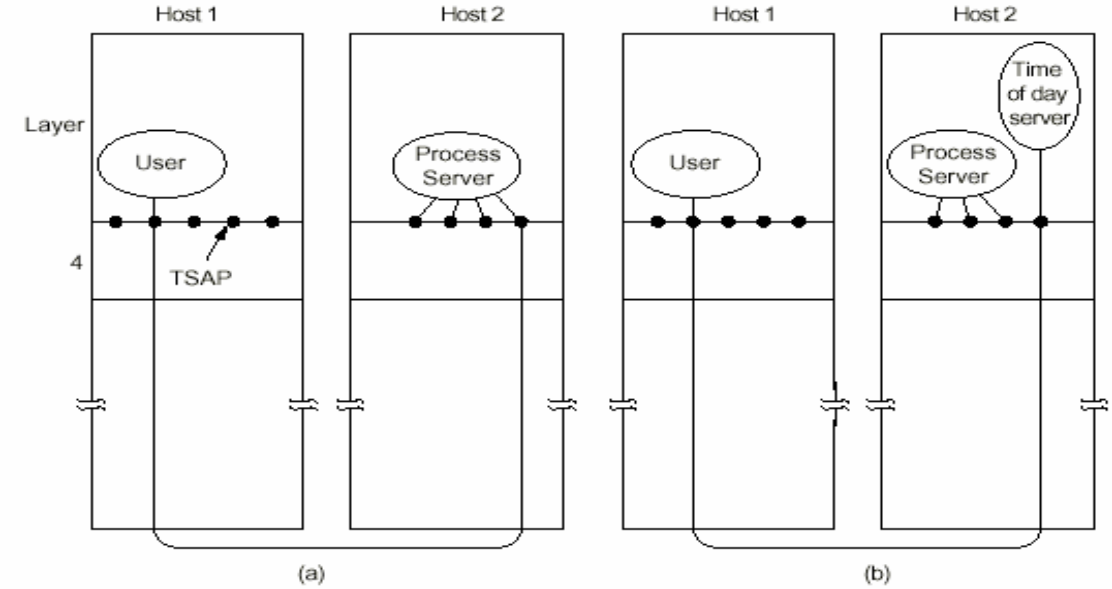
- explicit addressing of destinations is required
- initial connection establishment is more complicated
- potential existence of storage capacity in the subnet
- larger and dynamically varying number of connections

Transport Layer(5)

Addressing (first approach)



Addressing (second approach)



Transport Layer(6)

How to deal with delayed duplicates?

We must devise mechanisms to kill aged packets that are still wandering about --->

Packet lifetime can be restricted to a known maximum using one of the following techniques:

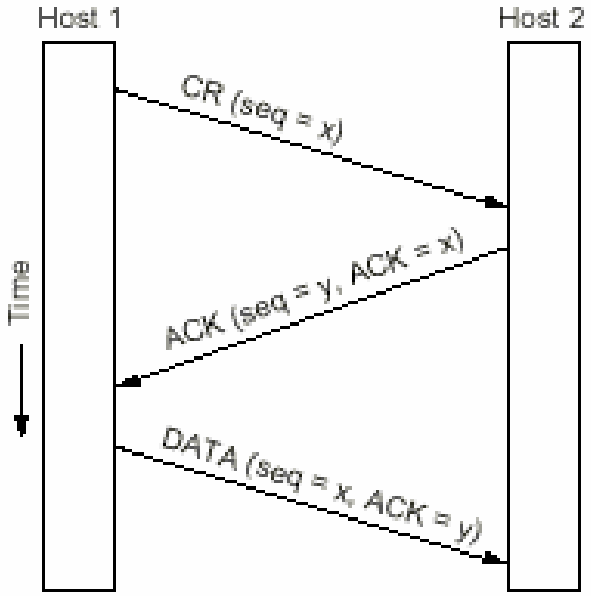
- restricted subnet design
- putting a hop counter in each packet
- timestamping each packet

Establishing a Connection

Problem: Getting both sides (sender and receiver) to agree on the initial sequence number

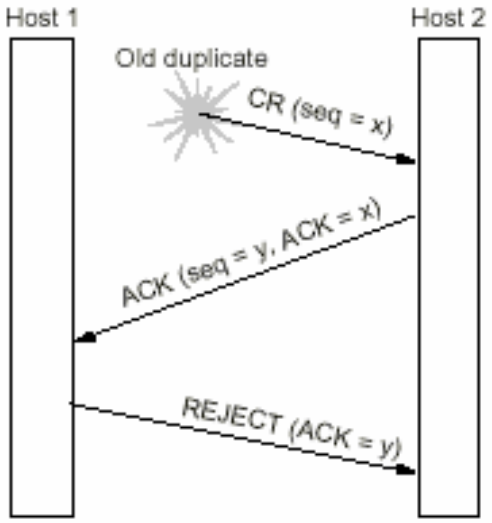
Solution: The three-way handshake (also adopted in TCP)

Normal operation

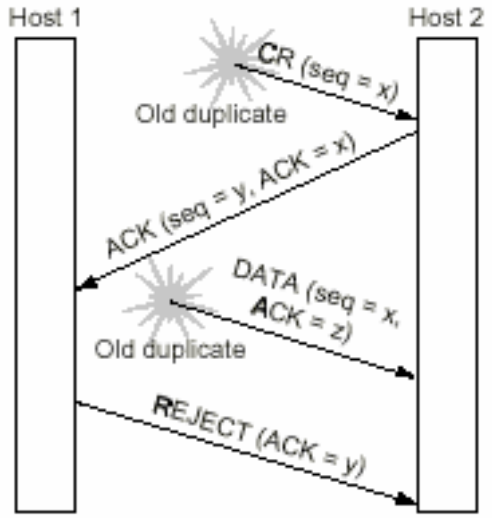


Transport Layer(8)

Duplicate CR (Connection request)



Duplicate CR and ACK

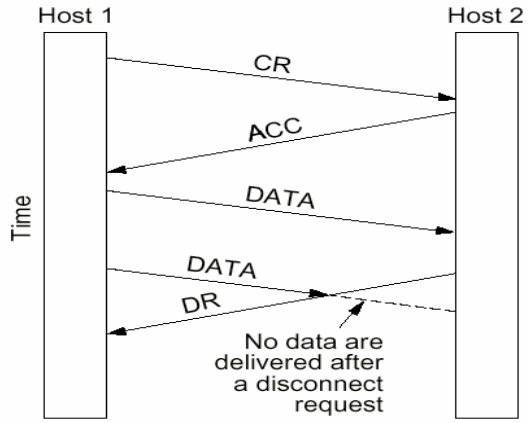


Transport Layer(9)

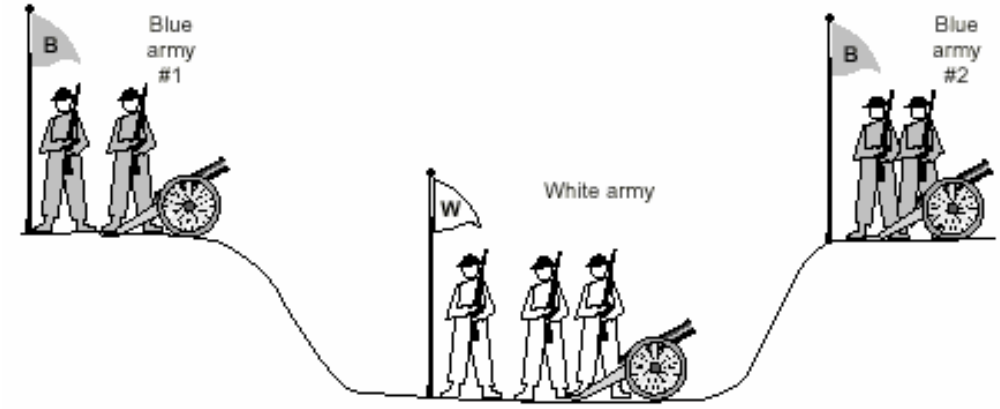
Releasing a Connection

Easier than establishing one, but still with pitfalls

Abrupt disconnection with loss of data



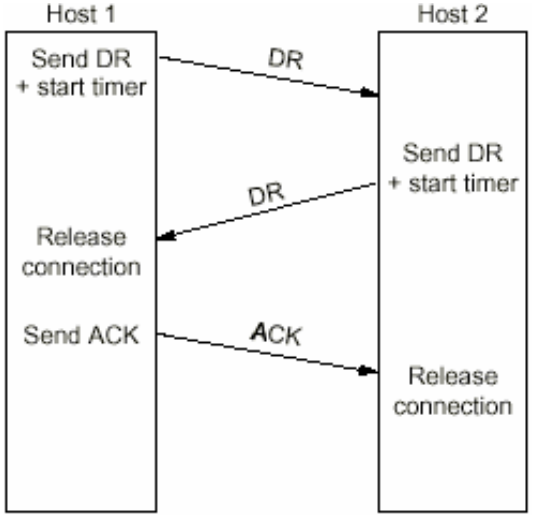
The two-army problem



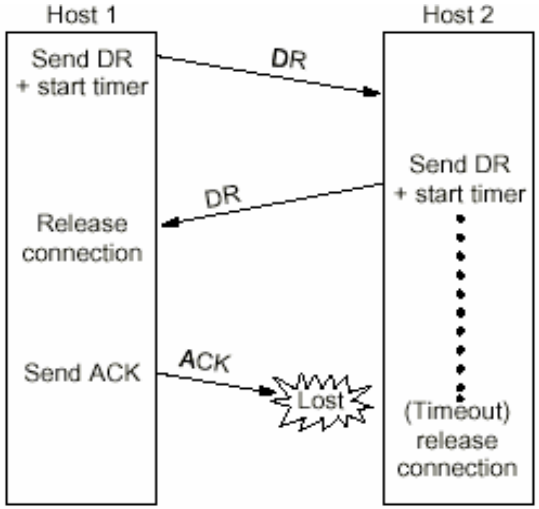
Transport Layer(10)

Four scenarios of releasing using a three-way handshake

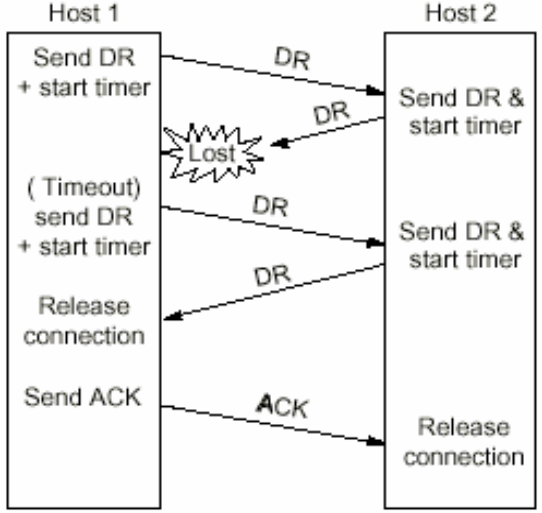
a) normal use



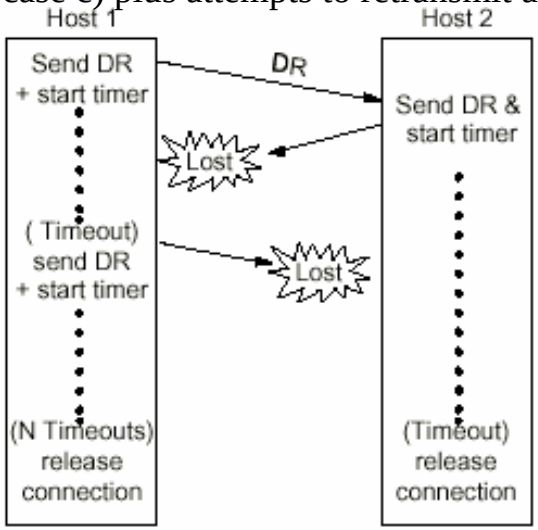
b) final *ACK* TPDU is lost



c) second DR lost



d) case c) plus attempts to retransmit also fail



Transport Layer(11)

Flow control and buffering

There are similarities and differences w.r.t. to the same topic in the data link layer.

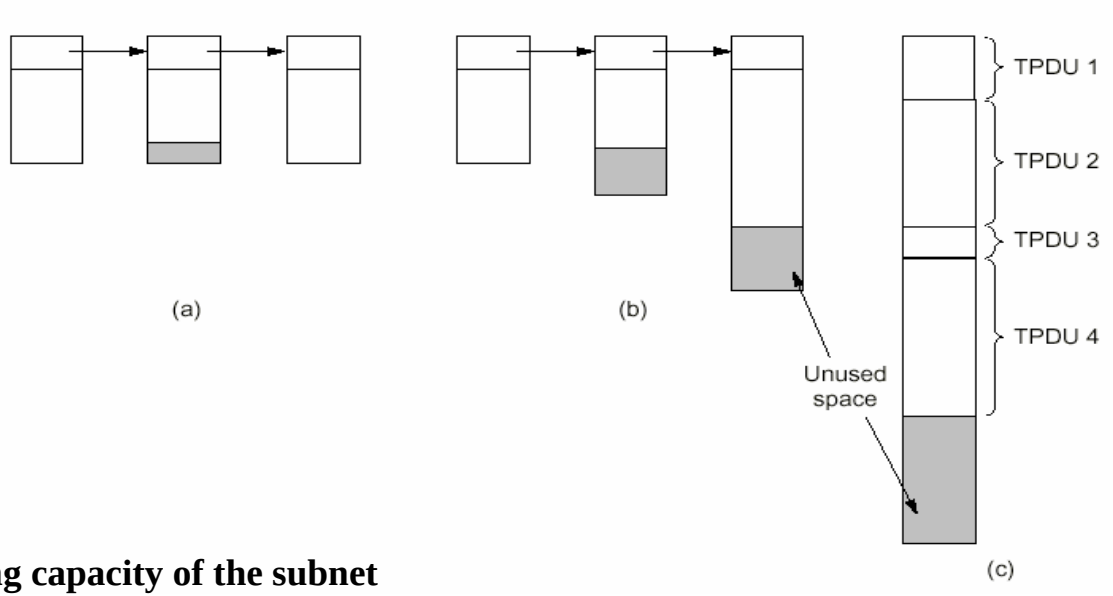
Basic Similarity:

sliding window or other scheme is needed on each connection to keep a fast transmitter from overrunning a slow receiver

Main difference:

a router usually has relatively few lines whereas a host may have numerous connections

Question of buffer size



Another bottleneck: the carrying capacity of the subnet

---> congestion problem

Solution approach (also adopted in TCP):

A sliding window flow control scheme in which the sender adjusts dynamically the window size to match the network's carrying capacity

Transport Layer(14)

The Internet TCP (Transmission Control Protocol)

Goal:

Provide a reliable (connection-oriented) end-to-end byte stream over an unreliable internetwork

Service to provide:

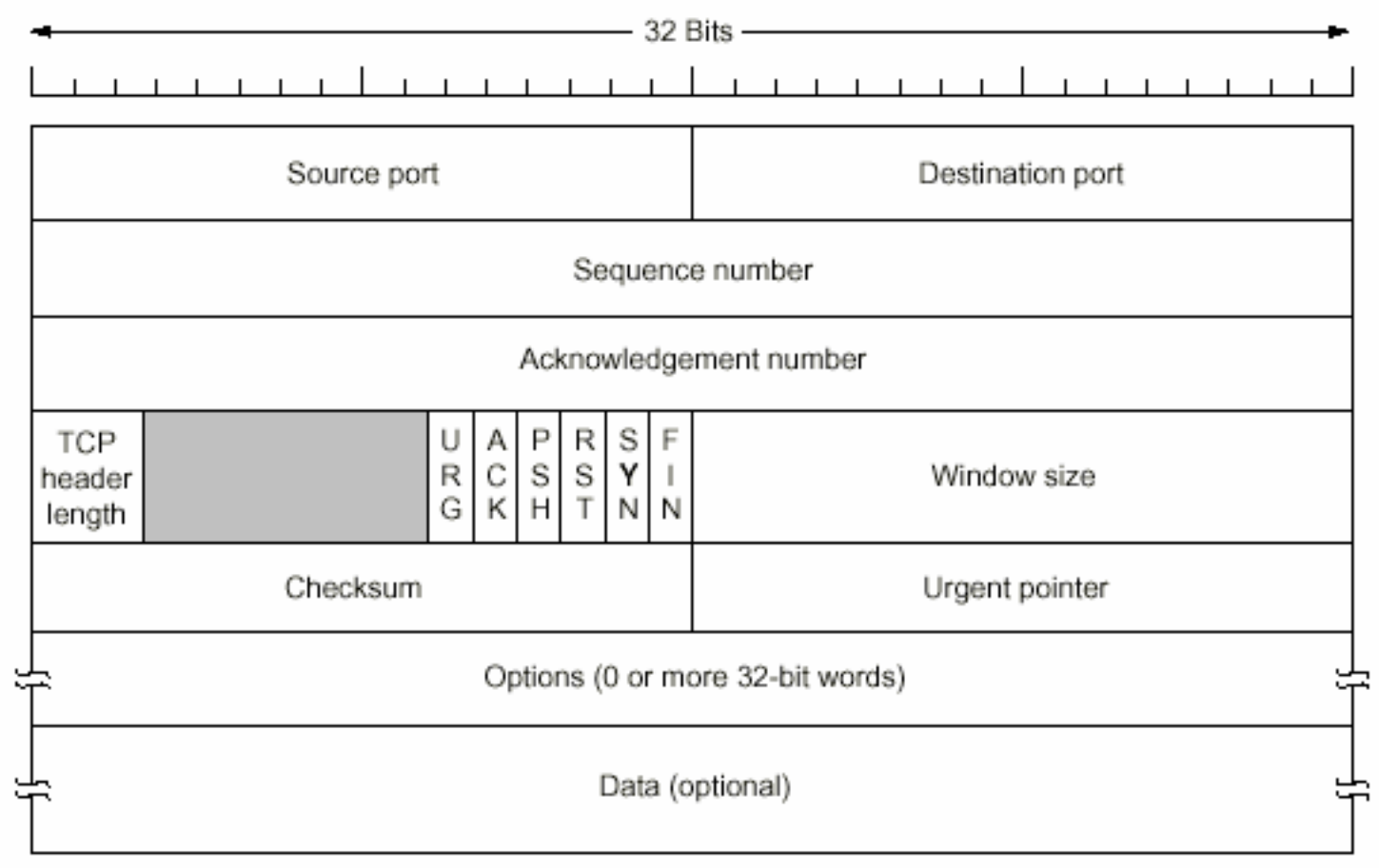
- Accepted user data streams from local processes are broken into pieces not exceeding 64K bytes
- Reconstruction of the original byte stream by reassembling the pieces in the proper sequence
- Time out and retransmission in order to guarantee proper delivery

The TCP service model

- both the sender and receiver create end points called sockets
- Each socket has an identifier (address):= (IP address, 16-bit number local to that host called port)
- To obtain TCP service, a connection must be established between sockets of the sender and receiver
- A socket may be used for multiple connections at the same time
- Connections are identified by the socket identifiers at both ends: (socket1, socket2)

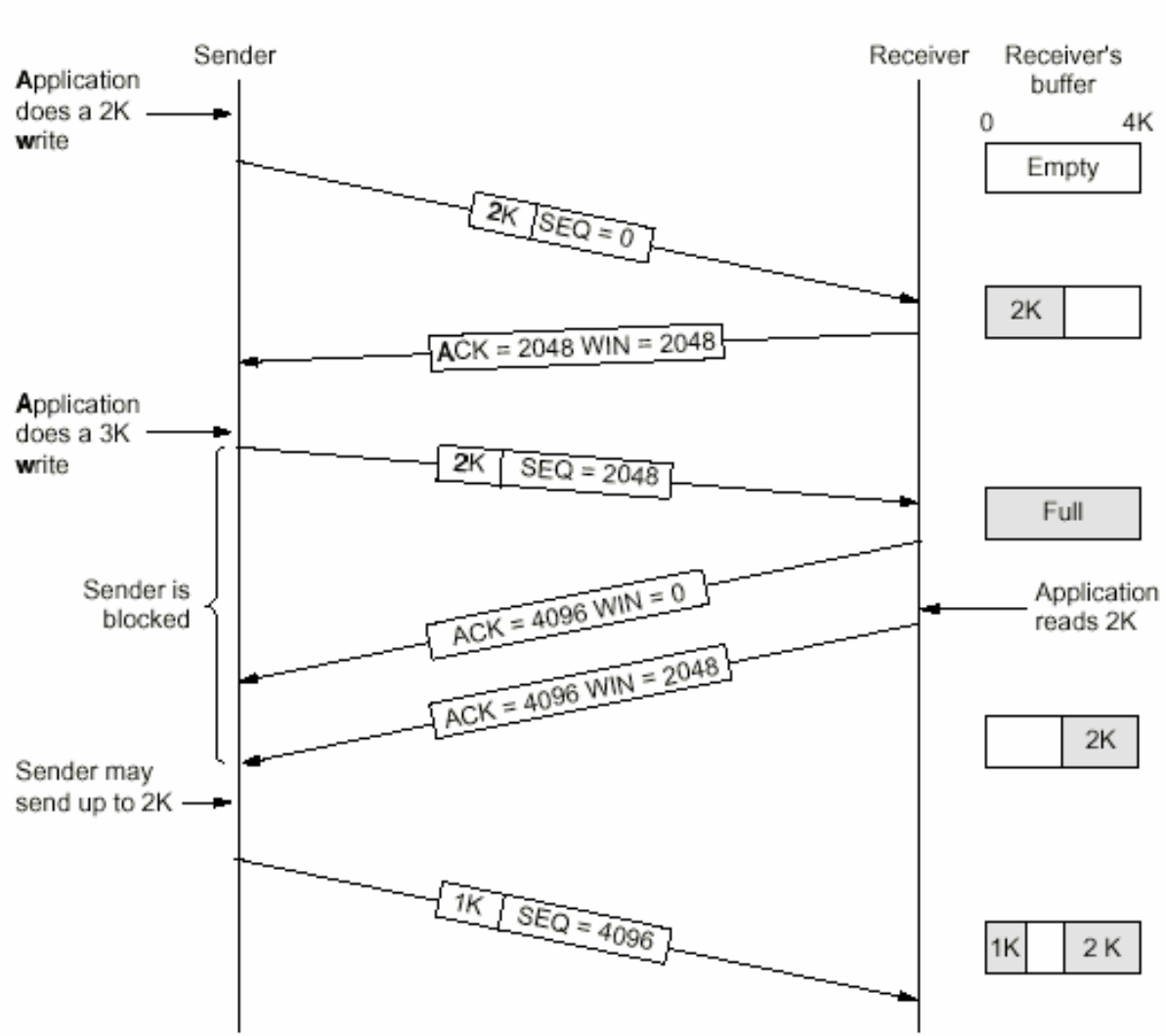
Transport Layer(16)

The TCP Segment header



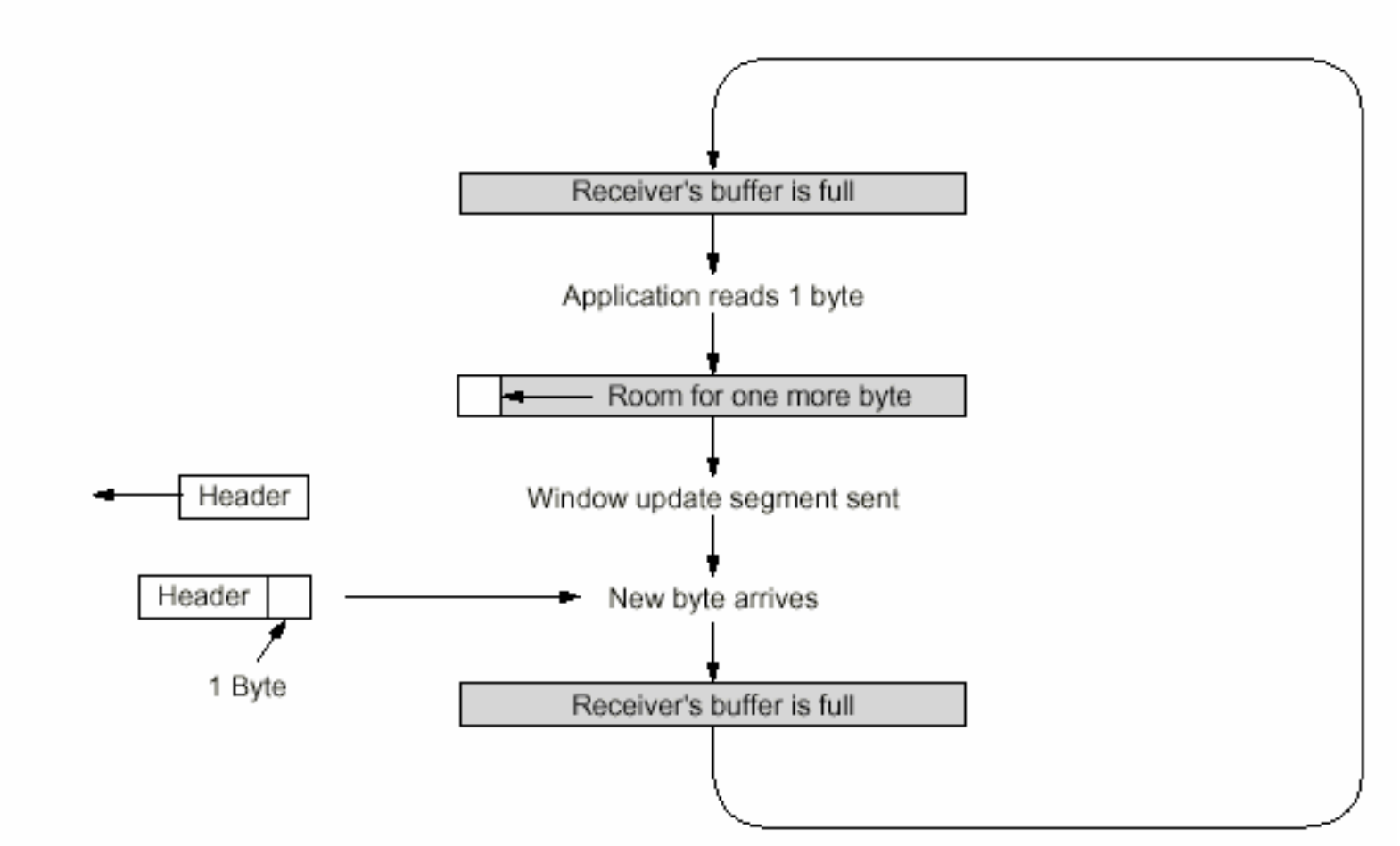
Transport Layer(17)

The window management (transmission policy) in TCP



Transport Layer(18)

The silly window syndrome



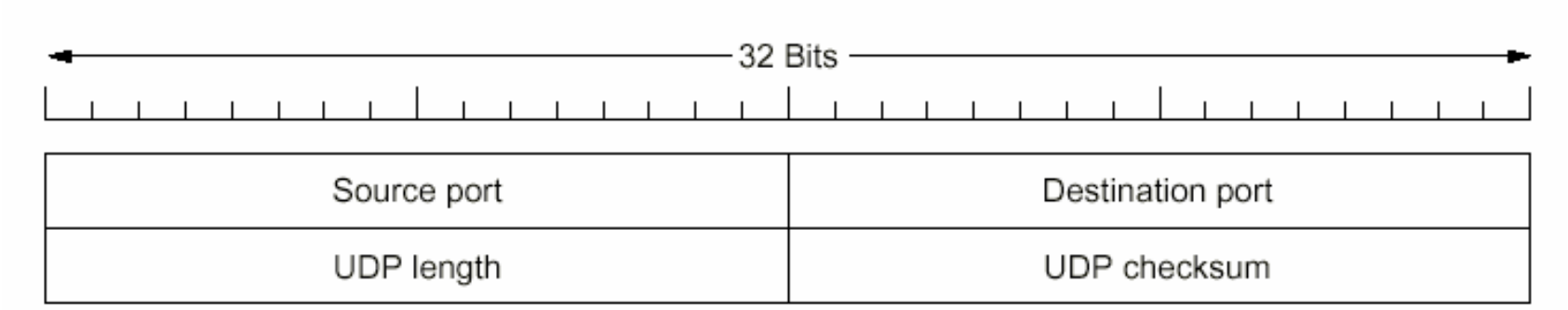
Transport Layer(19)

UDP (User Data Protocol)

Goal:

Provide a connectionless (unreliable) way for applications to send encapsulated raw IP datagrams --->
UDP is nothing else than IP + transport layer header

The UDP header



Typical use:

Client-server applications that have one request and one response

Transport Layer(20)

Summary

The transport layer is the key to understanding layered protocols. Its most important service is to provide an end-to-end, reliable, connection-oriented byte stream from sender to receiver. To do so, it must

- establish connections over unreliable networks
  ---> it must cope with delayed duplicate packets
  ---> it is done by means of a three-way-handshake
- release connections which is easier but still has to face the two-army problem
- handle the service primitives that permit establishing, using, and releasing of connections
- manage connections(transmission policy) and timer

The main Internet transport protocol is TCP

- data is exchanged in the form of segments
- it uses a fixed 20-byte header + optional part + zero or more data bytes
- the basic transmission protocol used is the sliding window protocol
- a great deal of work has gone into optimizing TCP performance using various algorithms