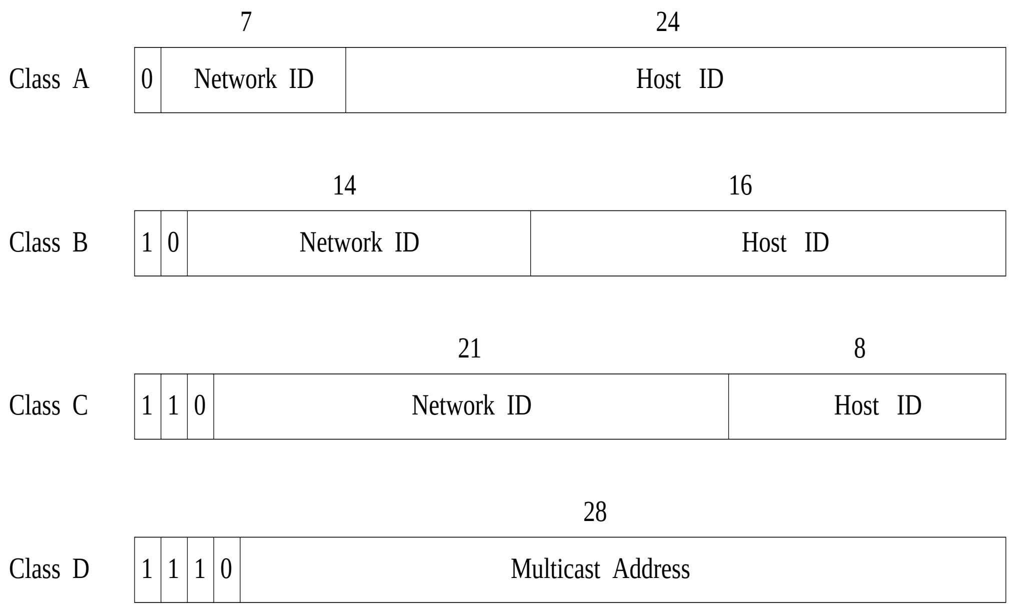
IP address format:



Dotted decimal notation: 10000000 00001011 00000011 00011111 $\leftrightarrow$ 128.11.3.31

Symbolic name to IP address translation: domain name server (DNS).

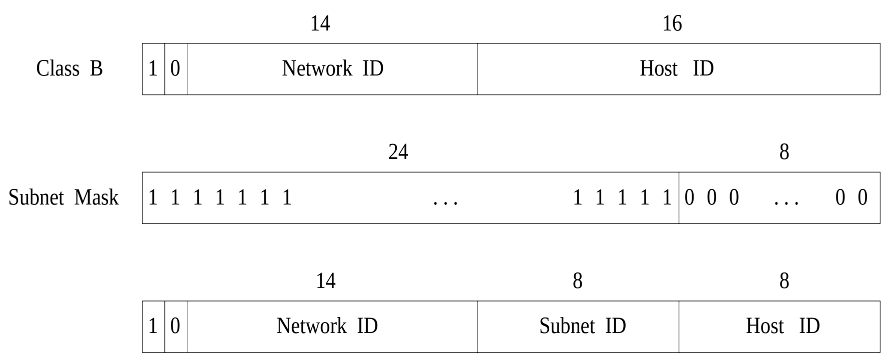
Notice hierarchical organization ("2-level").

Each interface (NIU) has an IP address; single host can have multiple IP addresses.

Running out of addresses. . .

Potential problem: Waste of address space. Giving each network own network ID is inefficient.

Solution: *Subnetting*; group several physical networks into one.



To determine subnet ID:

- Perform ANDing of IP address and subnet mask
- 3-level hierarchy

Forwarding and address resolution:

$\longrightarrow$ mechanics of routing

$\longrightarrow$ routing table is given

Subnet ID	Subnet Mask	Next Hop
128.10.2.0	255.255.255.0	Interface 0
128.10.3.0	255.255.255.0	Interface 1
128.10.4.0	255.255.255.0	128.10.4.250

Either destination host is directly connected on the same LAN or not.

Table look-up I:

- For each entry, compute *DestSubnetID* = *DestAddr* AND *SubnetMask*.
- Compare *DestSubnetID* with *SubnetID* and take forwarding action.

One more task left: Translate destination (or next hop IP address into LAN address.

$\longrightarrow$ required to hand-off packet to LAN layer

$\longrightarrow$ address resolution protocol (ARP)

Table look-up II:

- If ARP table contains entry, using LAN address send to destination.
- If ARP table does not contain entry, broadcast ARP Request packet with destination IP address.

  $\rightarrow$ e.g., Ethernet broadcast address: all 1's
- Encapsulate ARP packet into LAN frame.
- Update ARP table upon receipt of reply.

Dynamically maintain ARP table: use timer for each entry (15 min) to invalidate entries.

$\longrightarrow$ aging

Other approaches to solve address depletion problem:

- IPv6

  $\rightarrow$ 128 bits

- classless IP addressing

  $\rightarrow$ $a.b.c.d/x$

  $\rightarrow$ variable length subnetting

  $\rightarrow$ interdomain routing

  $\rightarrow$ CIDR (classless interdomain routing)

- dynamically assigned IP addresses

  $\rightarrow$ reusable

  $\rightarrow$ e.g., DHCP (dynamic host configuration protocol)

IETF standards: RFC (Request for Comments)

- RFC 791 (IP)
- RFC 826 (ARP)
- RFC 903 (RARP)
- RFC 894 (Ethernet)
- RFC 793 (TCP)
- RFC 768 (UDP)
- etc.

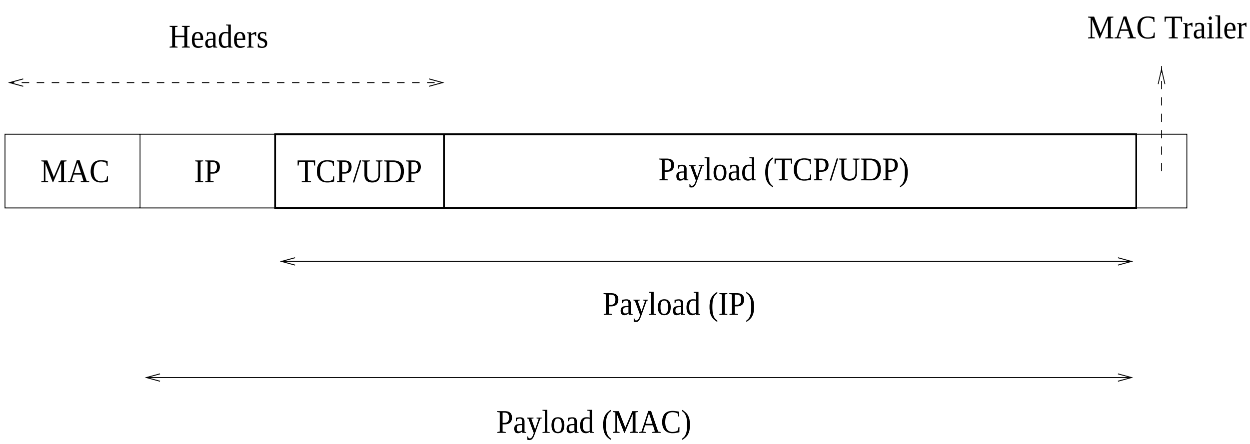
Transport Protocols: TCP/UDP Structure

⟶ end-to-end mechanism

⟶ runs on top of link-based mechanism

⟶ treat network layer as black box

Three-level encapsulation:



Network layer assumptions:

- unreliable
- out-of-order delivery (not frequent)
- absence of QoS guarantees (delay, throughput, etc.)
- insecure (IPv4)

  $\rightarrow$ IPsec

Additional (informal) performance properties:

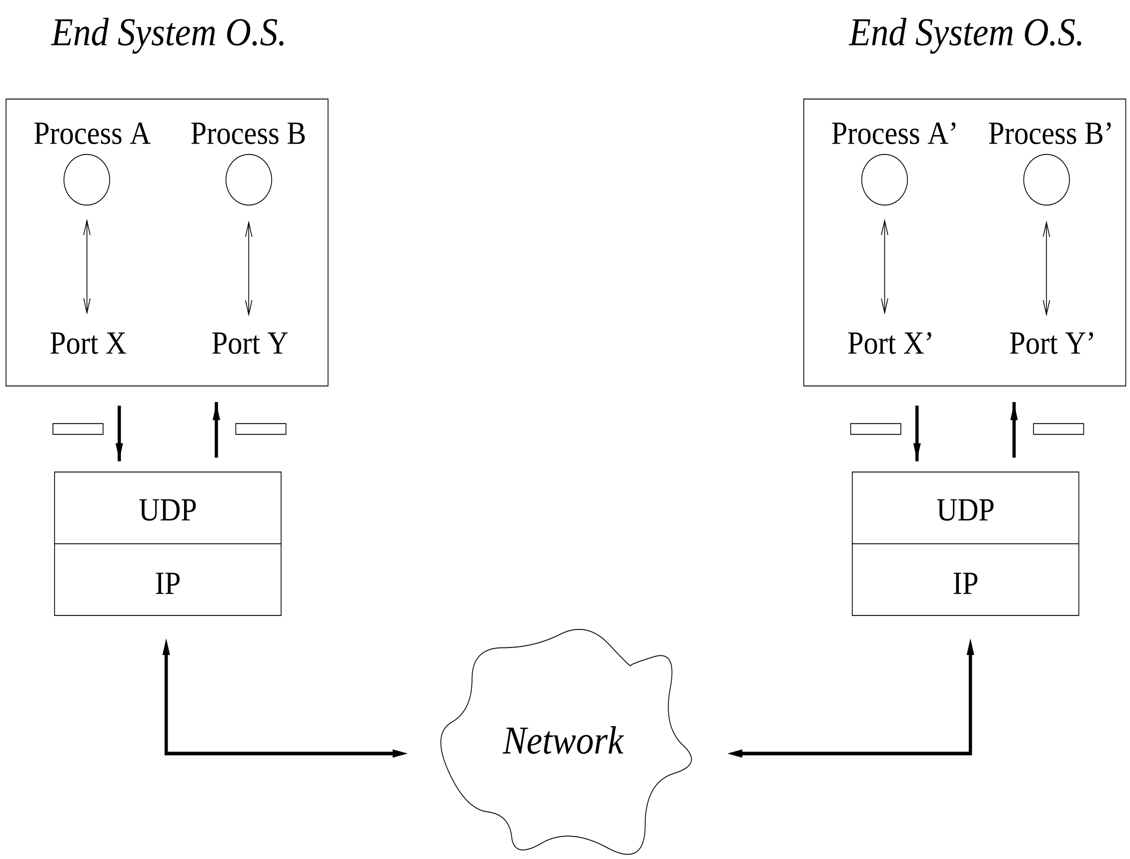
- Works "fine" under low load conditions
- Can break down under high load conditions
- Behavioral range wide: to some extent predictable

Goal of UDP (user datagram protocol):

$\longrightarrow$ process identification

$\longrightarrow$ port number as demux key

$\longrightarrow$ minimal support beyond IP



UDP packet format:

2	2
Source Port	Destination Port
Length	Checksum
Payload	

Checksum calculation (pseudo header):

4		
Source Address		
Destination Address		
$00 \cdots 0$	Protocol	UDP Length

$\longrightarrow$ pseudo header, UDP header and payload

UDP usage:

- Multimedia streamining

  → at minimum requires process identification

  → congestion control carried out above UDP

- Stateless client/server applications

  → persistent state a hinderance

  → lightweight

Goals of TCP (transmission control protocol):

- process identification
- reliable communication: ARQ
- speedy communication: congestion control
- segmentation

$\longrightarrow$ connection-oriented, i.e., stateful

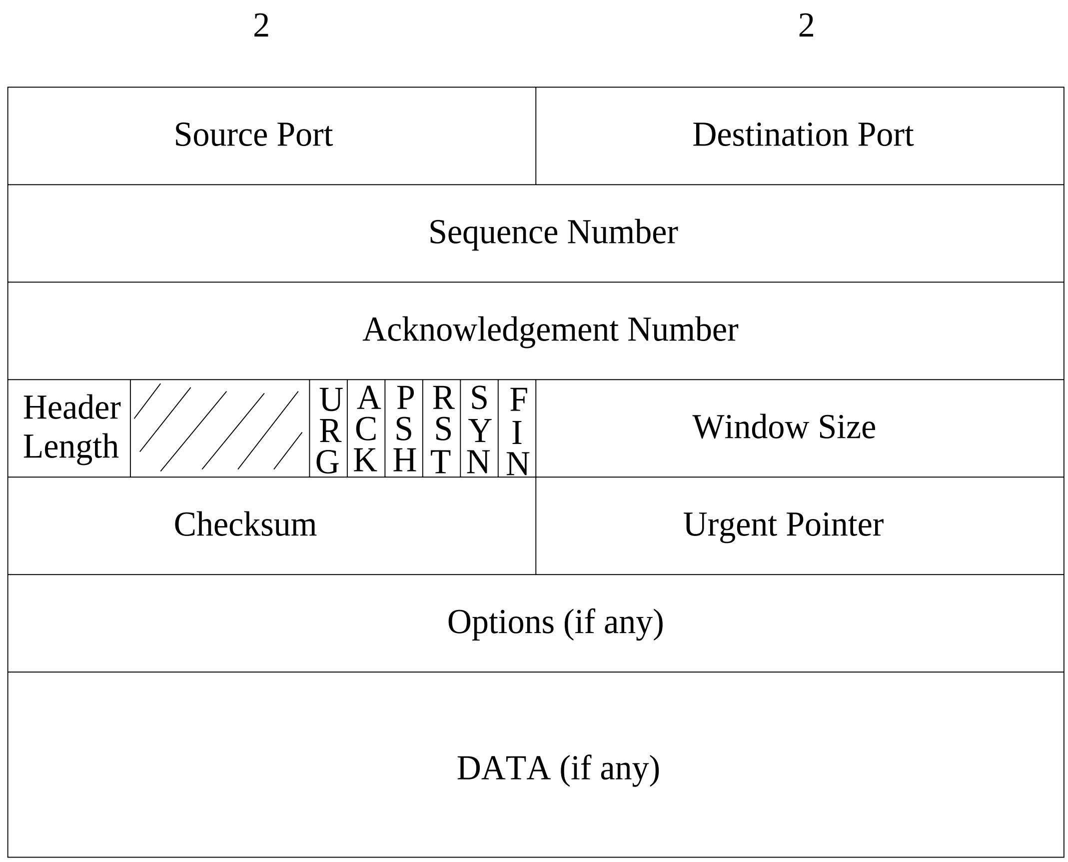
$\longrightarrow$ complex mixture of functionalities

Segmentation task: Provide "stream" interface to higher level protocols

$\longrightarrow$ view: contiguous stream of bytes

- segment stream of bytes into blocks or *segments* of fixed size
- segment size determined by TCP MTU (Maximum Transmission Unit)
- use also for reliability mechanism

TCP packet format:



- Sequence Number: position of first byte of payload
- Acknowledgement: next byte of data expected (receiver)
- Header Length (4 bits): 4 B units
- URG: urgent pointer flag
- ACK: ACK packet flag
- PSH: override TCP buffering
- RST: reset connection
- SYN: establish connection
- FIN: close connection
- Window Size: receiver's advertised window size
- Checksum: prepend pseudo-header
- Urgent Pointer: byte offset in current payload where urgent data begins
- Options: MTU; take min of sender & receiver (default 556 B)

Checksum calculation (pseudo header):

4

Source Address		
Destination Address		
00 ··· 0	Protocol	TCP Segment Length

$\longrightarrow$ pseudo header, TCP header and payload