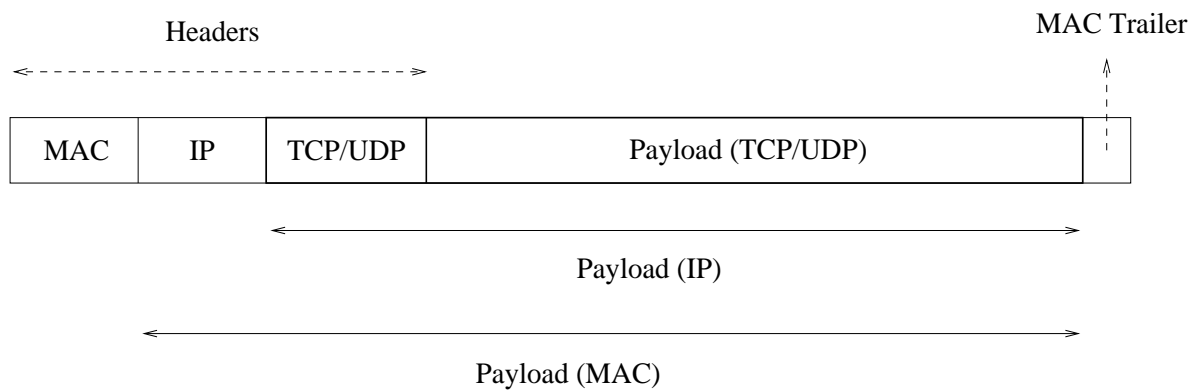# Transport Layer Protocols

→ end-to-end

→ runs on top of network layer protocols

→ treat network layer & below as black box

Three-level encapsulation: e.g., TCP and UDP

Meaning of protocol "stack"

→ push/pop headers and trailers

Network layer (IP) assumptions:

- unreliable

- out-of-order delivery

- absence of QoS guarantees (delay, throughput, etc.)

- insecure (IPv4)

    $\rightarrow$ IPsec

Additional performance properties:

- works "ok"

- can break down under high load conditions

    $\rightarrow$ e.g., flash crowds, DDoS and worm attacks

- wide behavioral range

    $\rightarrow$ from good to bad

Goal of UDP (User Datagram Protocol):

$\longrightarrow$ process identification

$\longrightarrow$ port number as demux key

$\longrightarrow$ minimal support beyond IP
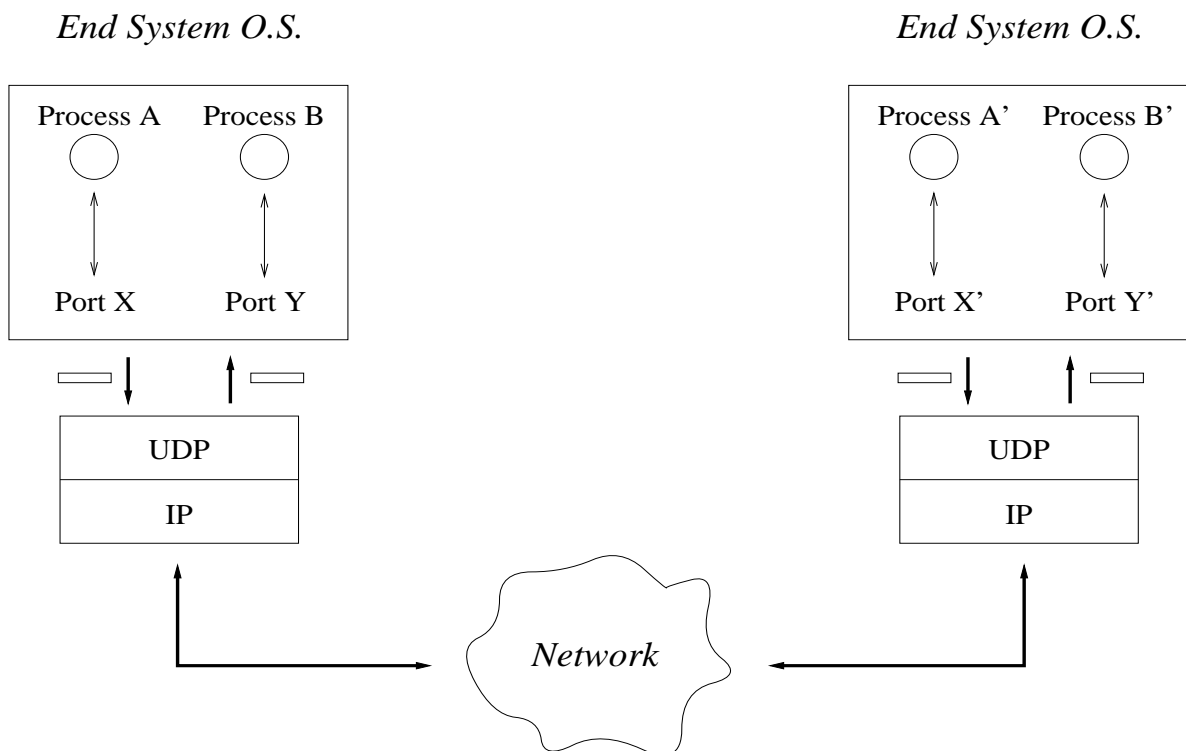
UDP packet format:

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

  $\rightarrow$ lean and nimble

  $\rightarrow$ at minimum requires process identification

  $\rightarrow$ reliability addressed above UDP if needed: FEC or ARQ

  $\rightarrow$ congestion control addressed above UDP

- lightweight client/server applications

  $\rightarrow$ persistent state leads to overhead

Goals of TCP (Transmission Control Protocol):

- process identification

- reliable communication: ARQ

- speedy communication: congestion control

- segmentation and MTU

- connection establishment and tear-down

$\rightarrow$ complex mixture of functionalities

Provide stream interface to higher level protocols

$\rightarrow$ exported semantics: contiguous byte stream

$\rightarrow$ e.g., accessed using read(), write() system calls

Segmentation:

- segment stream of bytes into blocks of fixed size

- segment size determined by TCP MTU (Maximum Transmission Unit)

- actual unit of transmission in ARQ

- efficiency and reduced fragmentation

TCP packet format:

2	2

Source Port	Destination Port
Sequence Number	
Acknowledgement Number	
Header Length	//////	U R G	A C K	P S H	R S T	S Y N	F I N	Window Size
Checksum	Urgent Pointer
Options (if any)	
DATA (if any)	

- 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

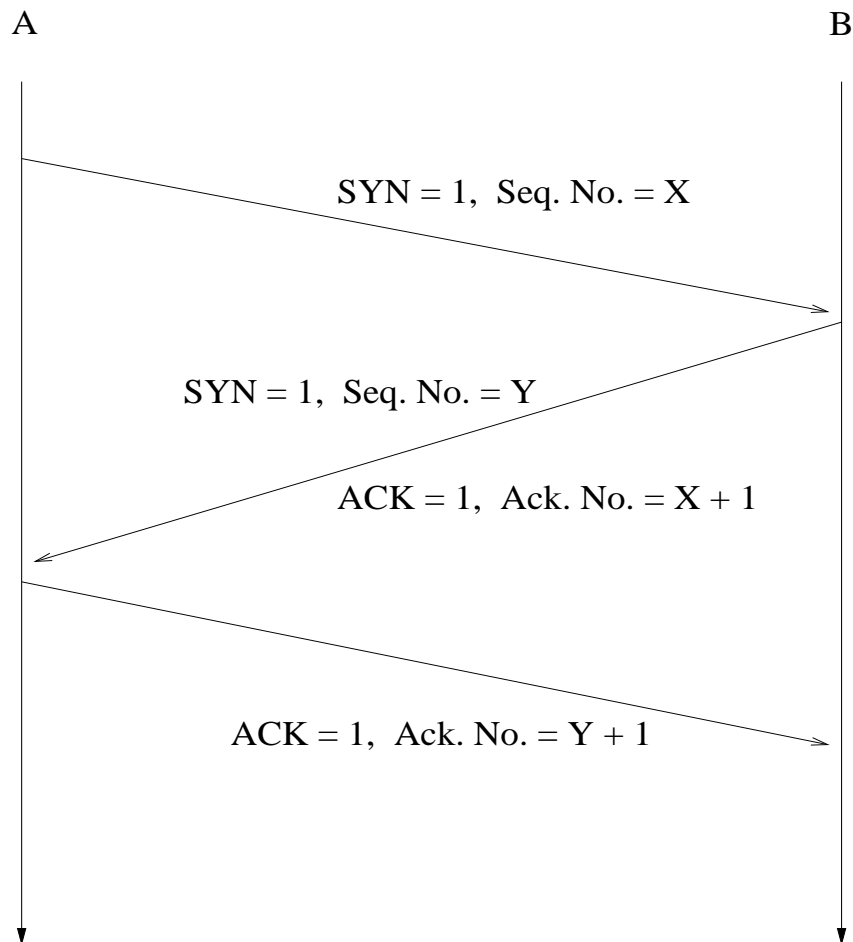- Options: MTU; take min of sender and receiver

Checksum calculation: pseudo header

4

Source Address
Destination Address

00 $\cdots$ 0	Protocol	TCP Segment Length

$\longrightarrow$ pseudo header, TCP header and payload

TCP connection establishment (3-way handshake):

A   B

SYN = 1, Seq. No. = X

SYN = 1, Seq. No. = Y

ACK = 1, Ack. No. = X + 1

ACK = 1, Ack. No. = Y + 1

- X, Y are chosen randomly

  $\rightarrow$ mitigate sequence number prediction

- piggybacking

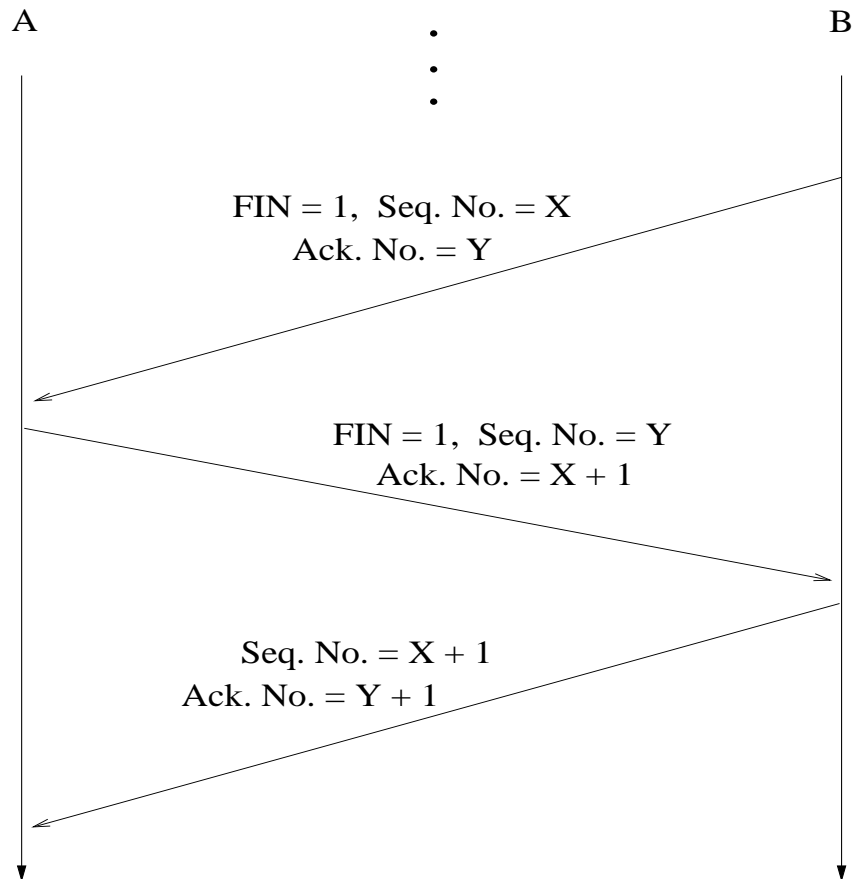2-party consensus problem: are A and B in agreement about the state of affairs after 3-way handshake?

$\rightarrow$ in general: in networks with unbounded delay, there is no solution

$\rightarrow$ acknowledging the ACK . . .

$\rightarrow$ lunch date problem

$\rightarrow$ also TCP session termination

TCP connection termination:

A   •
                                     •
                                     •
                                                                      B

FIN = 1, Seq. No. = X
Ack. No. = Y

FIN = 1, Seq. No. = Y
Ack. No. = X + 1

Seq. No. = X + 1
Ack. No. = Y + 1

- full duplex

- half duplex

Finite state machine representation: