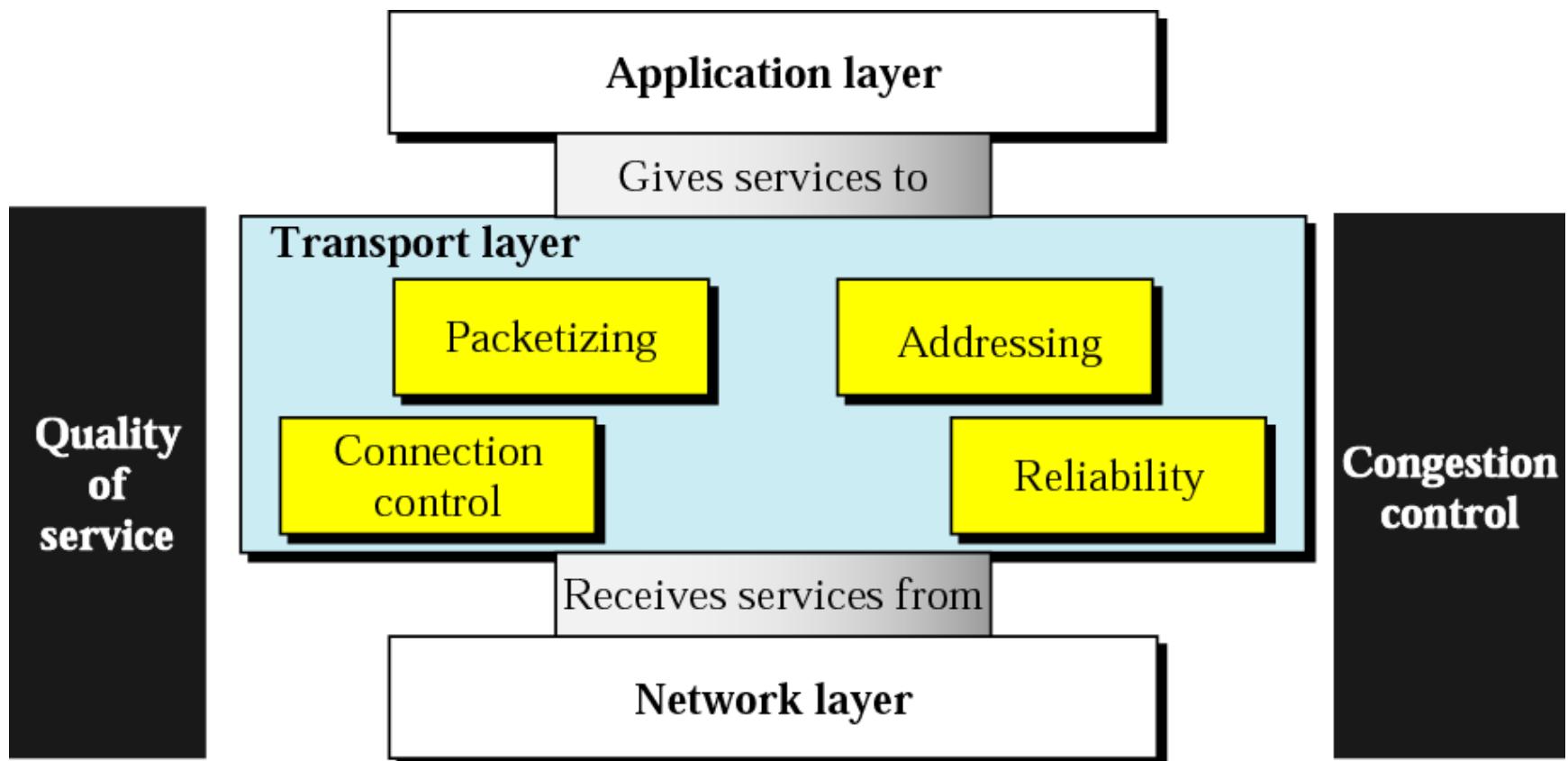


*Transport Layer*

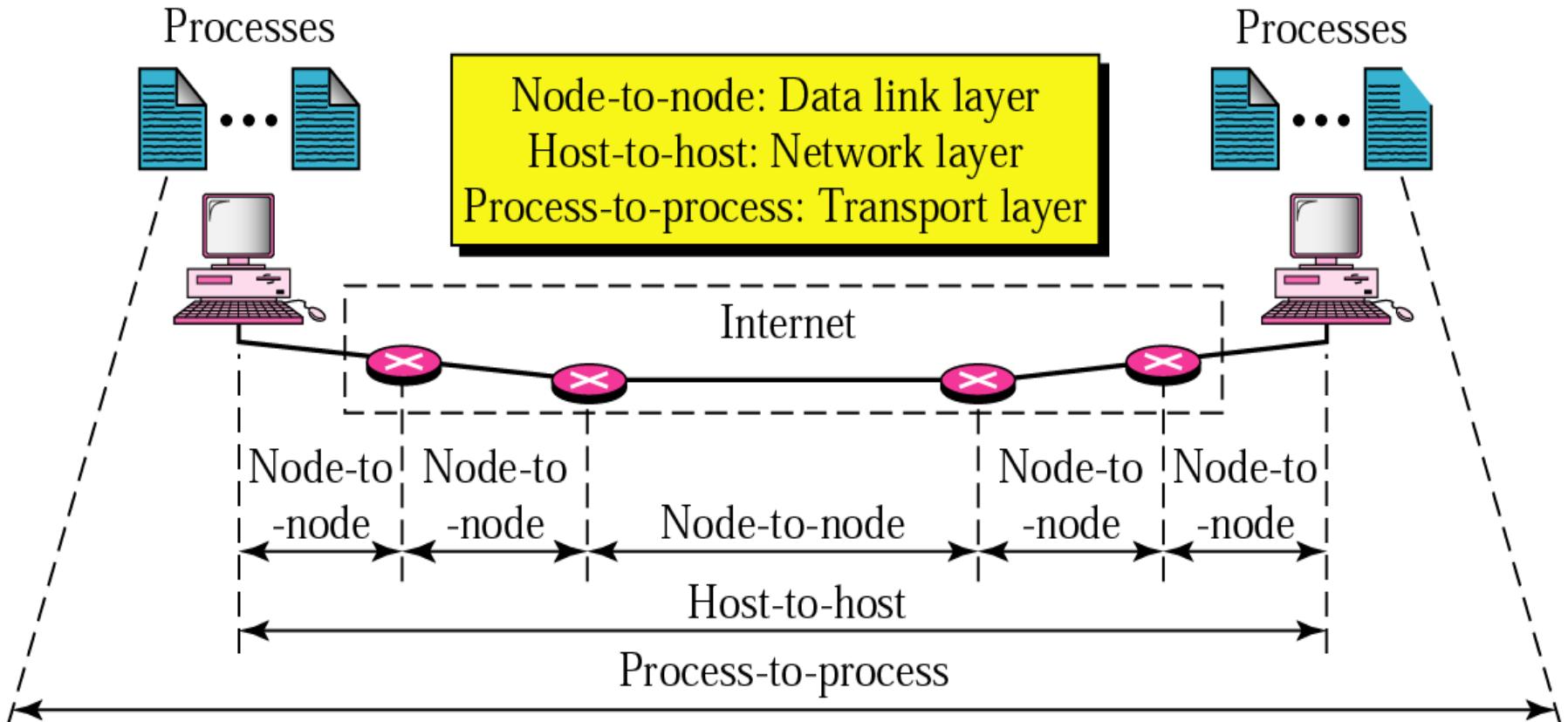
# Position of transport layer



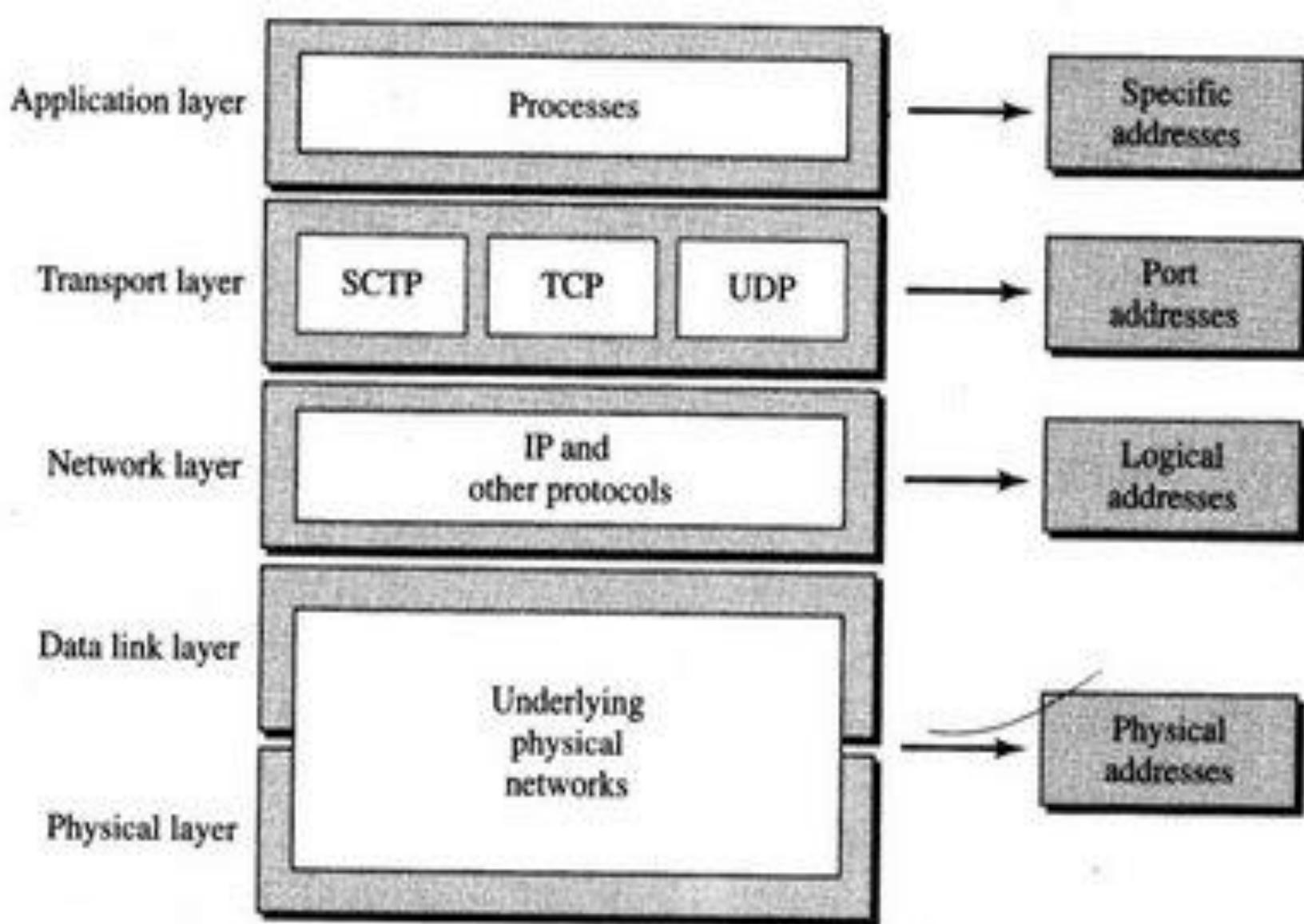
# Introduction:

The transport layer is responsible for process-to-process delivery.

# Types of data deliveries



# Position of Transport layer protocol in TCP/IP



1] UDP:

- Connectionless Protocol.
- Unreliable Protocol.

2] TCP:

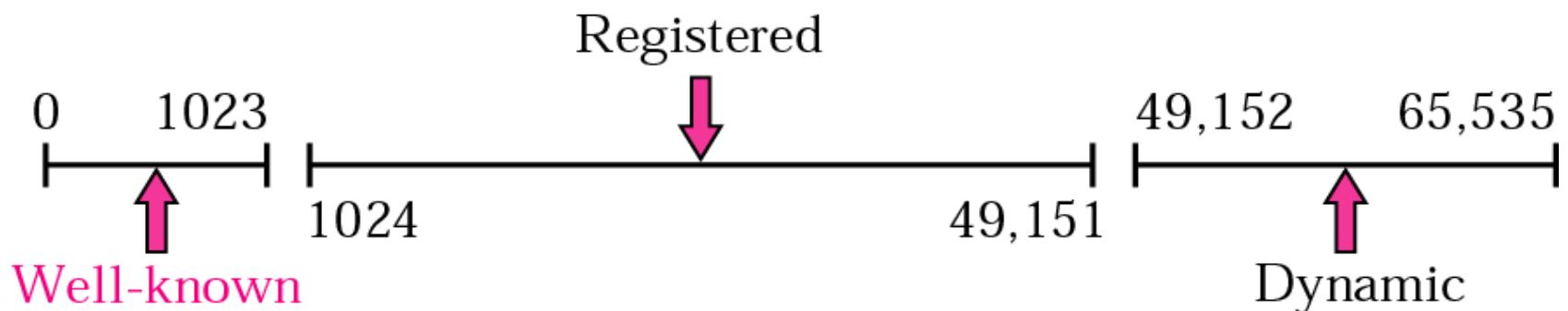
- Connection Oriented Protocol.
- Reliable Protocol.

3] SCTP: Stream Control Transmission Protocol

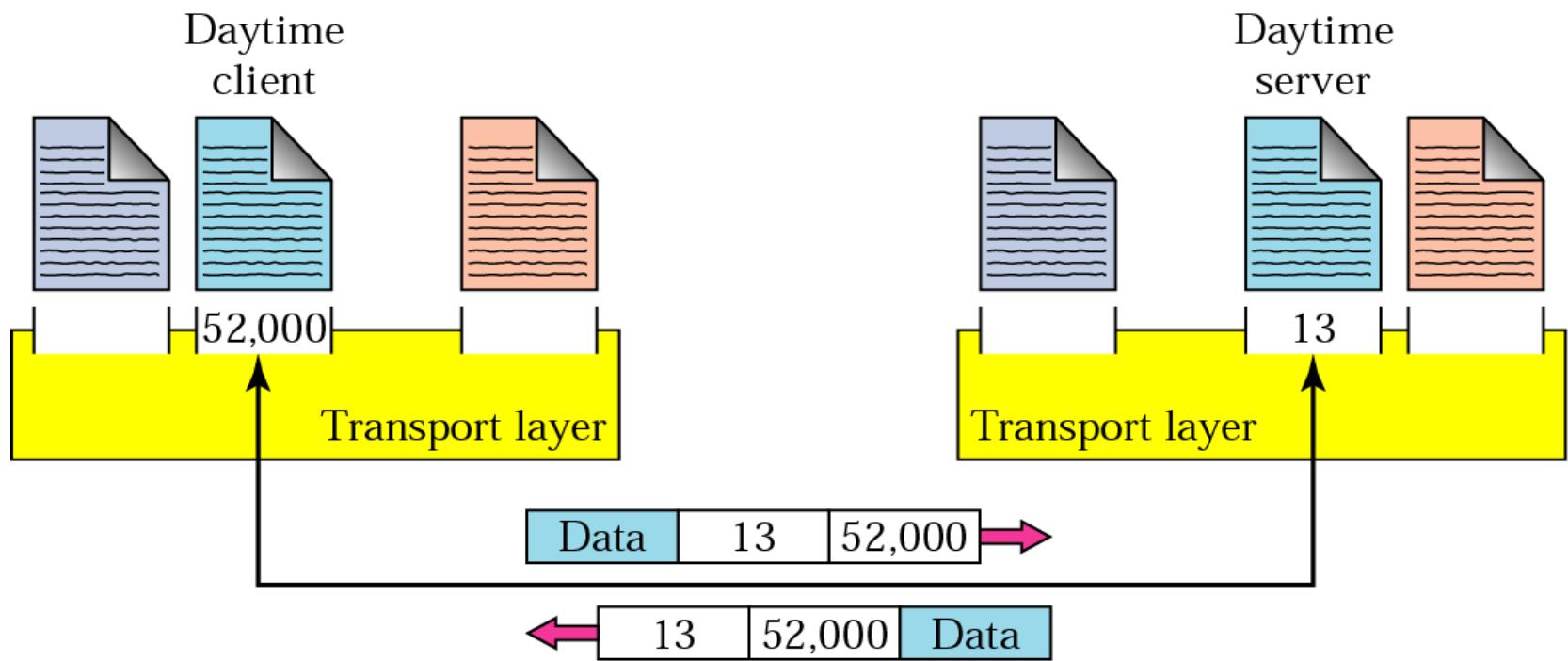
- Designed for multimedia, multihomed and multistream application

# Port numbers

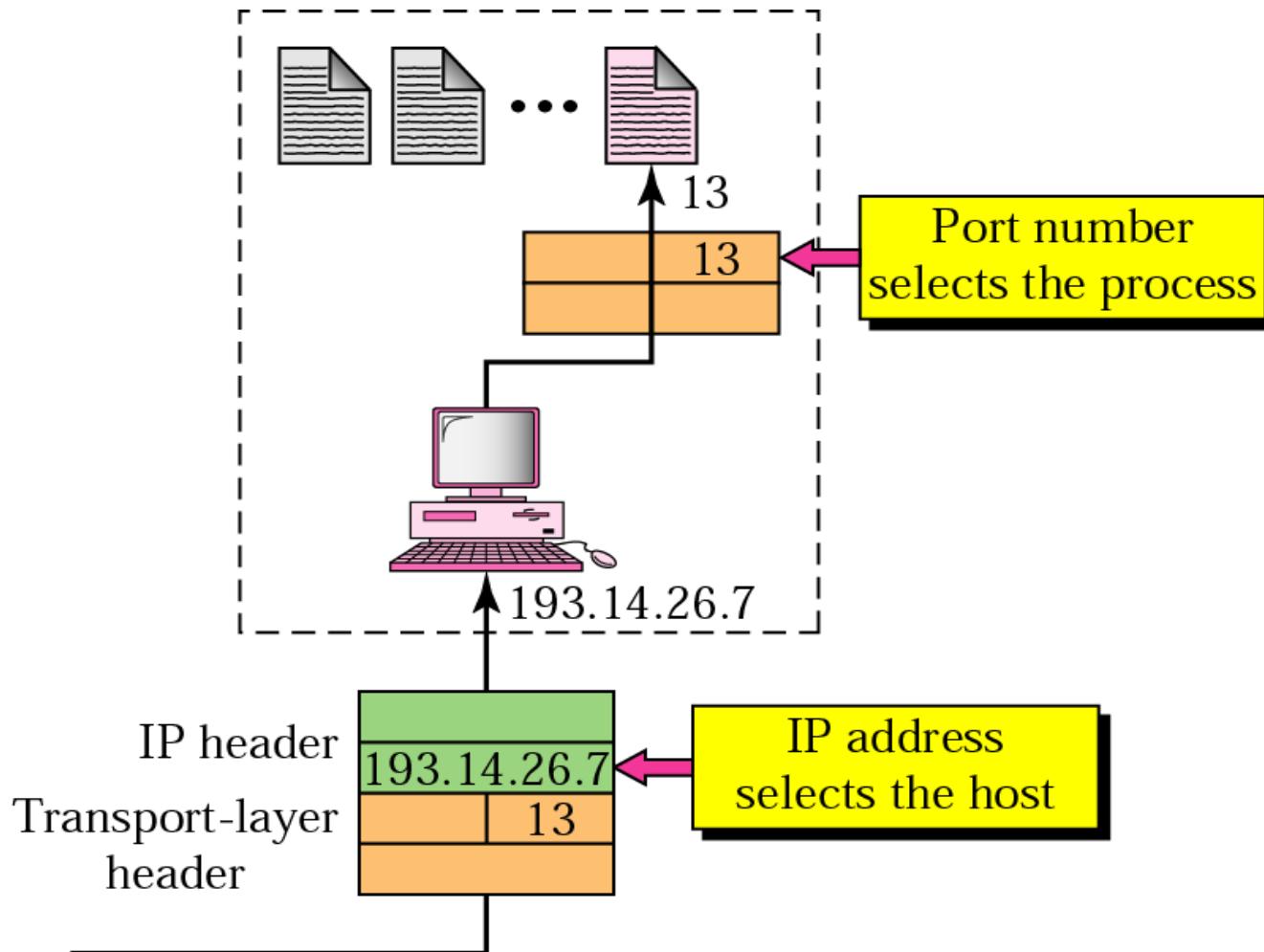
IANA (Internet Assigned Number Authority) ranges:



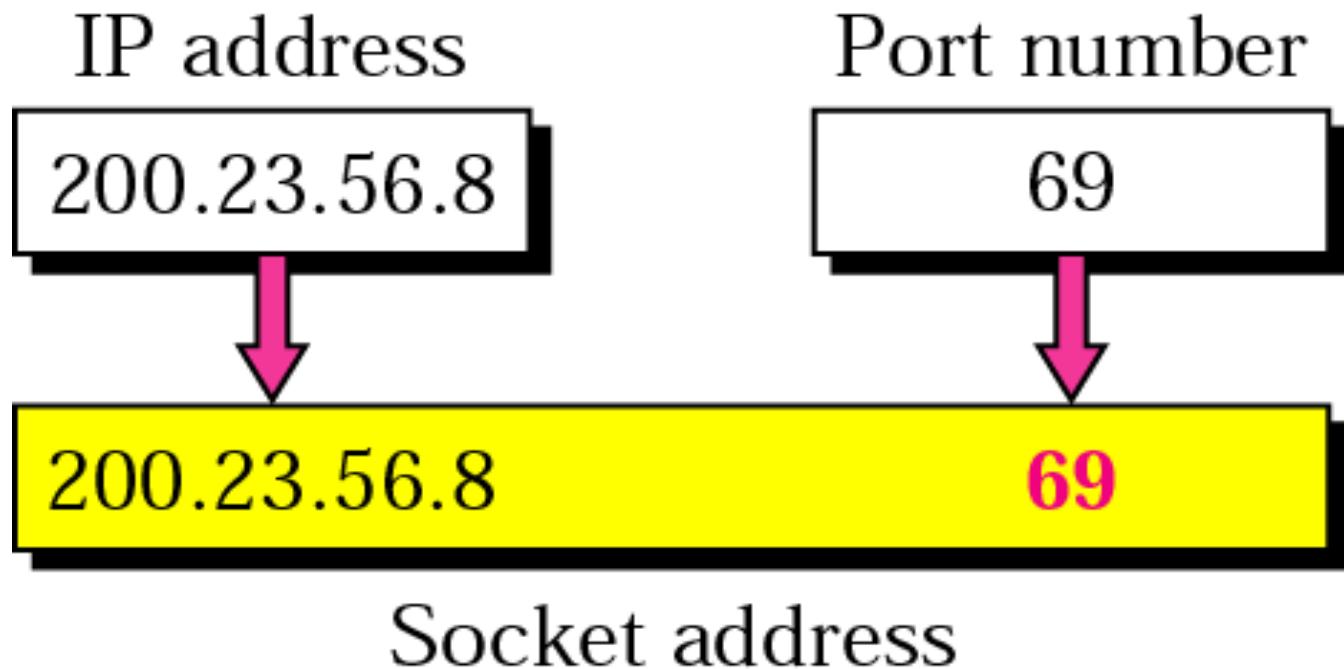
# Port numbers



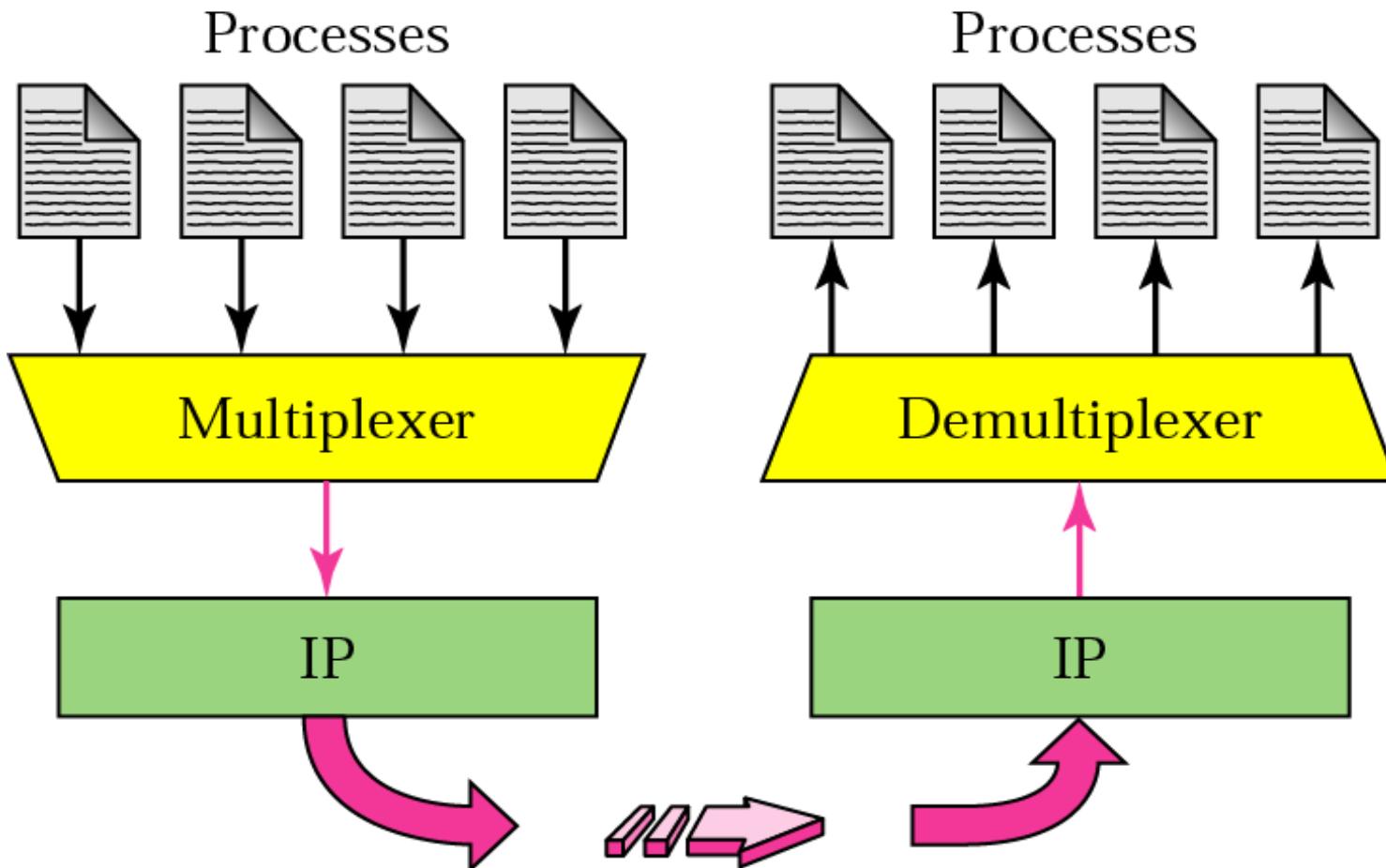
# IP addresses versus port numbers



# Socket address



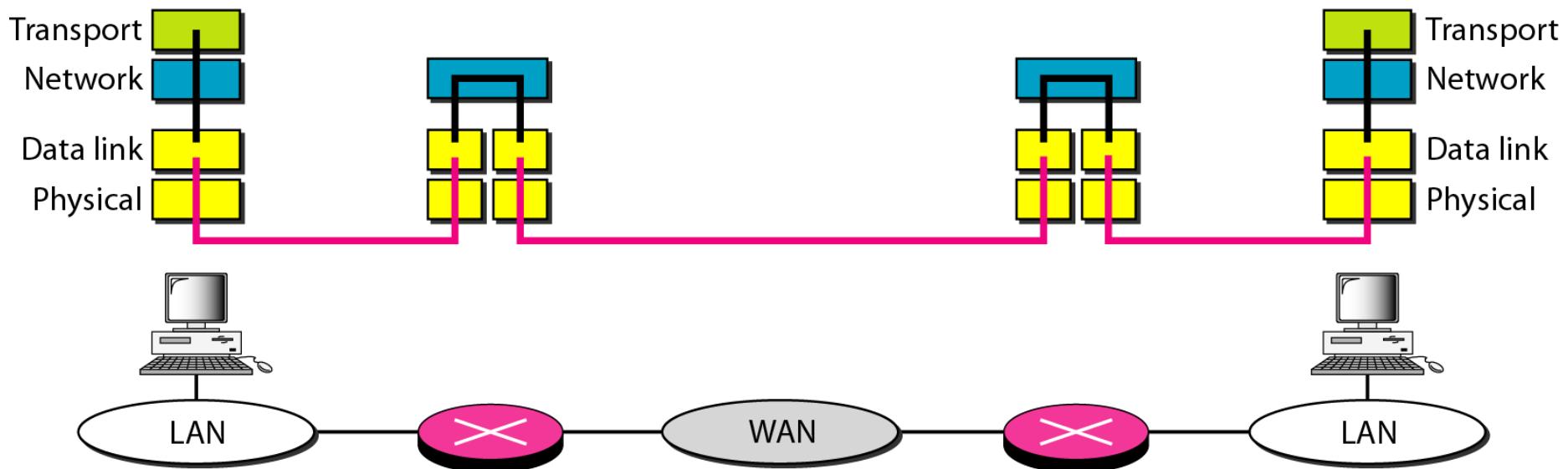
# Multiplexing and Demultiplexing



- ❑ Connectionless Protocol Vs Connection Oriented Protocol.
- ❑ Reliable Versus Unreliable.

# Error control

- Error is checked in these paths by the data link layer
- Error is not checked in these paths by the data link layer

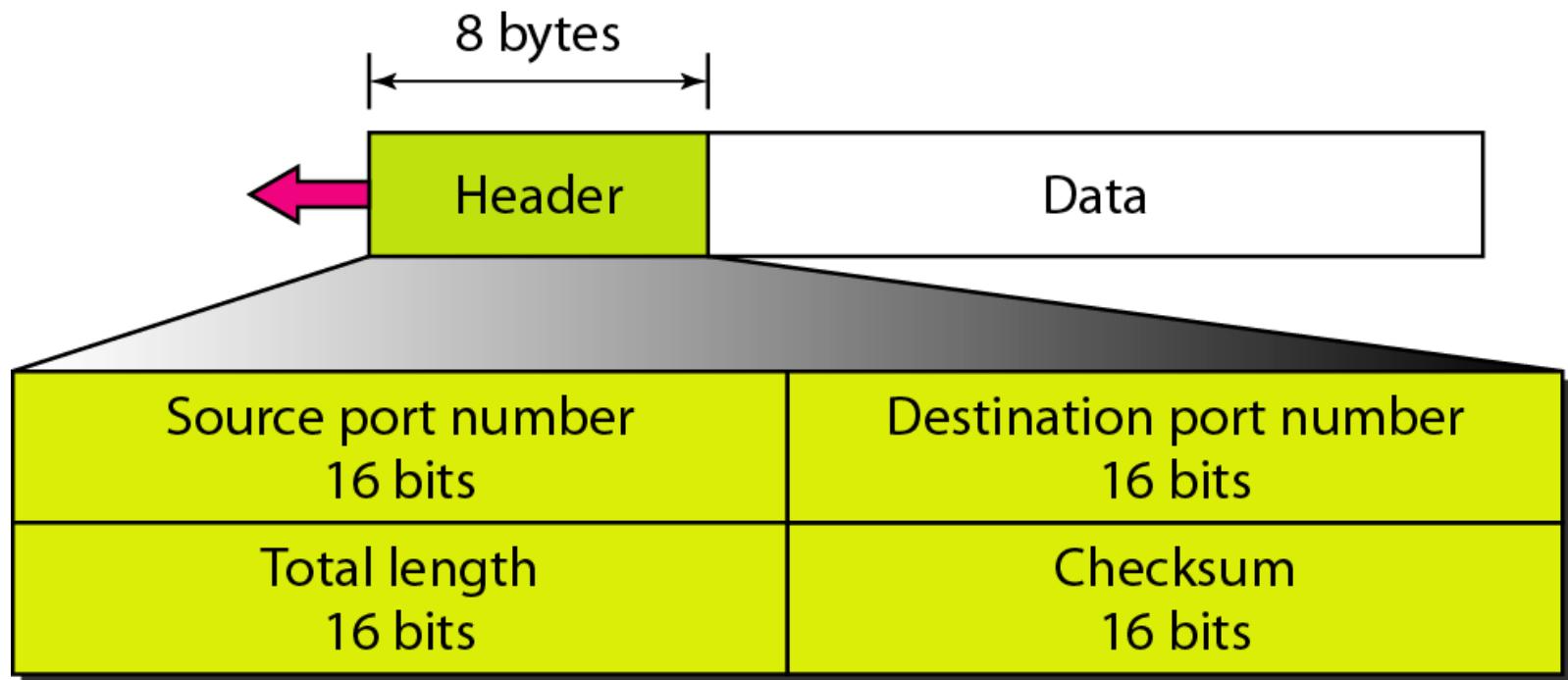


# USER DATAGRAM PROTOCOL (UDP)

The User Datagram Protocol (UDP) is called a connectionless, unreliable transport protocol. It does not add anything to the services of IP except to provide process-to-process communication instead of host-to-host communication.

Also, it performs very limited error checking. UDP is so powerless.

# User datagram format



# Checksum calculation of a simple UDP user datagram

153.18.8.105			
171.2.14.10			
All 0s	17	15	
1087		13	
15		All 0s	
T	E	S	T
I	N	G	All 0s

10011001 00010010	→	153.18
00001000 01101001	→	8.105
10101011 00000010	→	171.2
00001110 00001010	→	14.10
00000000 00010001	→	0 and 17
00000000 00001111	→	15
00000100 00111111	→	1087
00000000 00001101	→	13
00000000 00001111	→	15
00000000 00000000	→	0 (checksum)
01010100 01000101	→	T and E
01010011 01010100	→	S and T
01001001 01001110	→	I and N
01000111 00000000	→	G and 0 (padding)
10010110 11101011	→	Sum
01101001 00010100	→	Checksum

# TCP

TCP is a connection-oriented protocol; it creates a virtual connection between two TCPs to send data. In addition, TCP uses flow and error control mechanisms at the transport level.

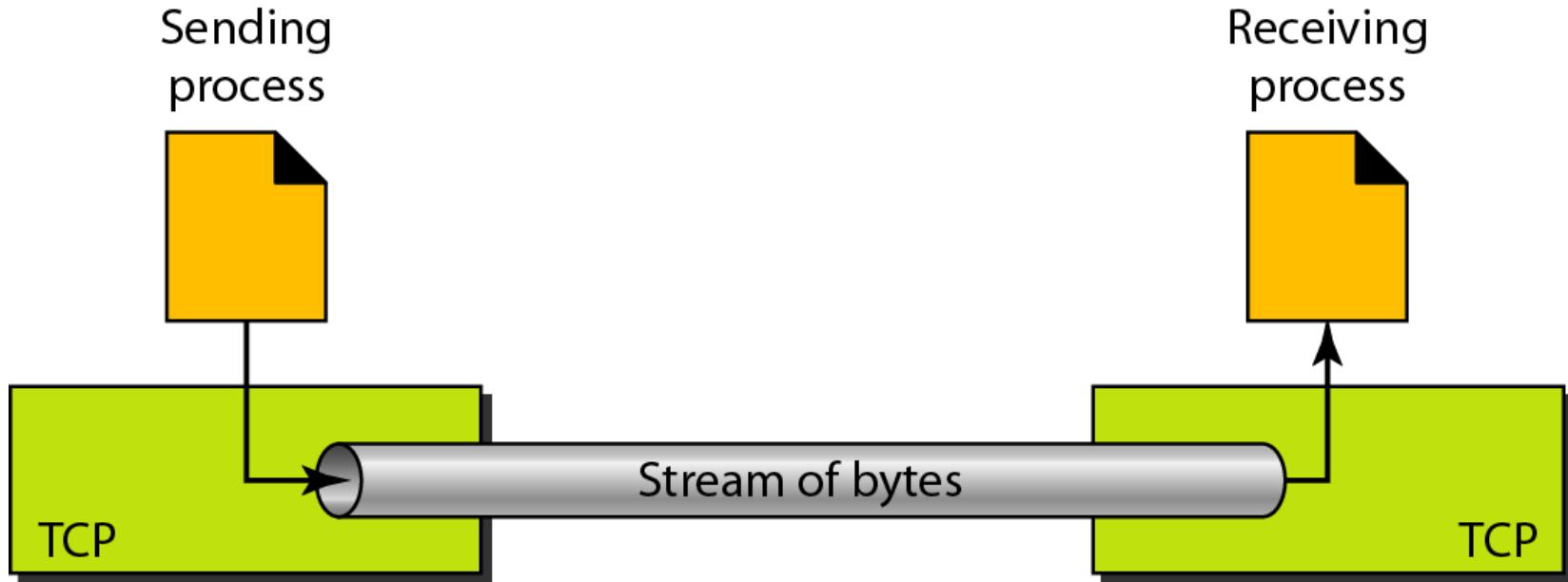
# TCP Services

## *1] Process-to-Process Communication*

# Well-known ports used by TCP

<i>Port</i>	<i>Protocol</i>	<i>Description</i>
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
20	FTP, Data	File Transfer Protocol (data connection)
21	FTP, Control	File Transfer Protocol (control connection)
23	TELNET	Terminal Network
25	SMTP	Simple Mail Transfer Protocol
53	DNS	Domain Name Server
67	BOOTP	Bootstrap Protocol
79	Finger	Finger
80	HTTP	Hypertext Transfer Protocol
111	RPC	Remote Procedure Call

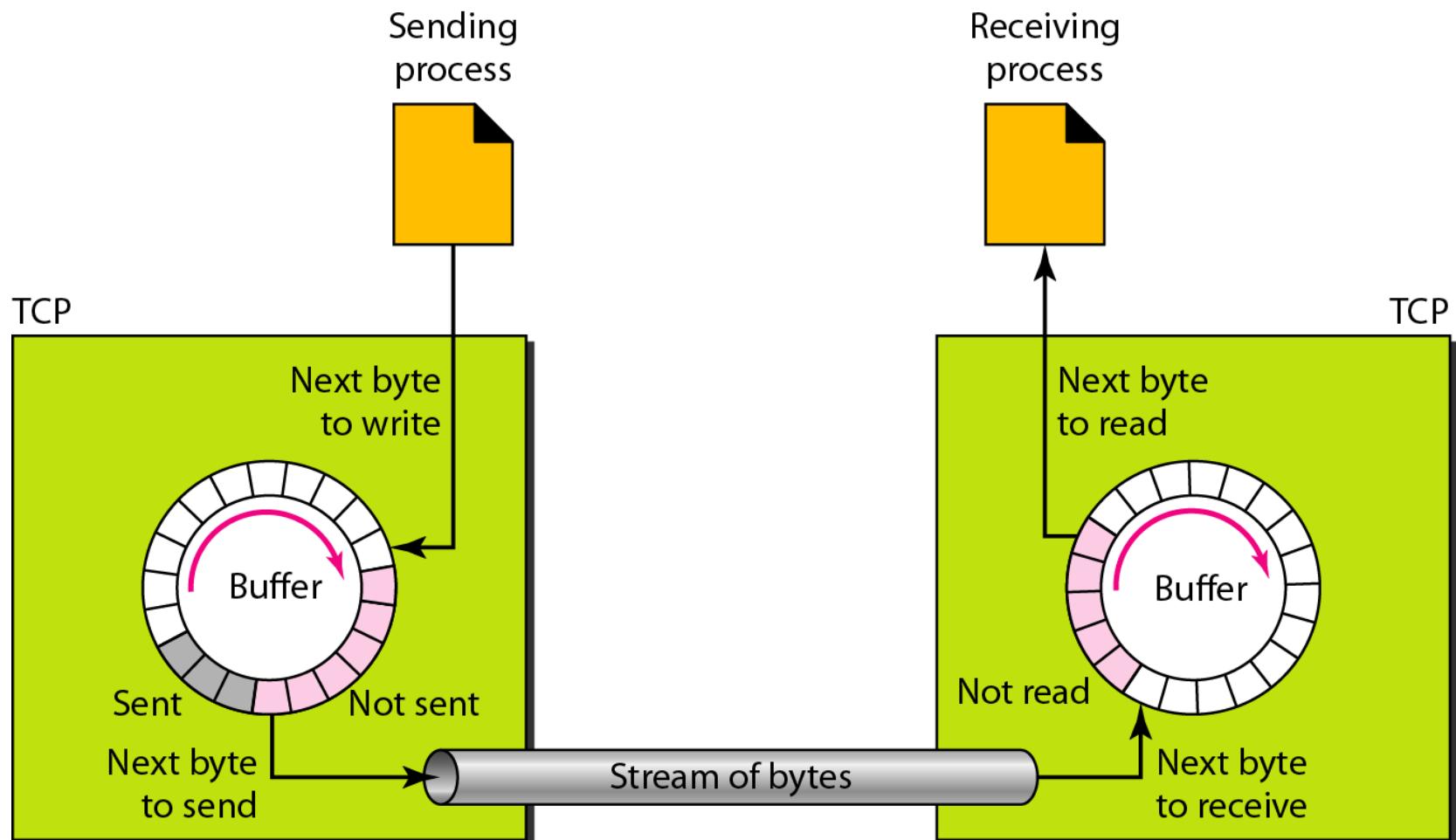
## 2] Stream delivery Service



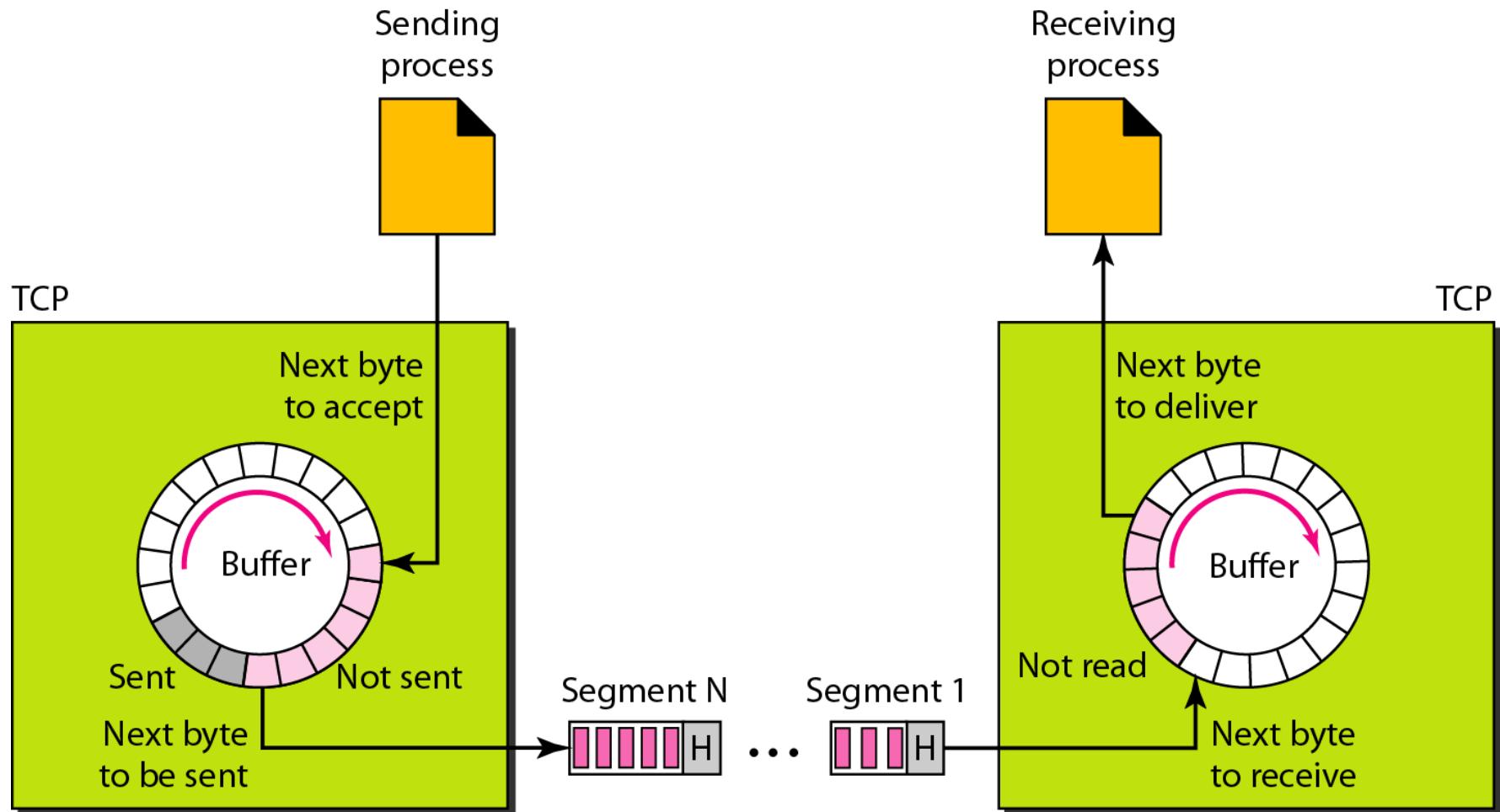
Two processes seem to be connected by an imaginary "tube" that carries their data across the Internet.

The sending process produces (writes to) the stream of bytes, and the receiving process consumes (reads from) them.

# 3] Sending and receiving buffers



# 4] TCP segments



5] Full-Duplex Communication

6] Connection-Oriented Service

7] Reliable Service.

# TCP Features

1] Numbering System

2] Byte Number

3] Sequence Number

4] Acknowledgment Number

5] Flow Control

6] Error Control

7] Congestion Control

## 2] Byte Number:

The numbering does not necessarily start from 0. Instead, TCP generates a random number between 0 and  $2^{32} - 1$  for the number of the first byte.

Byte numbering is used for flow and error control.

### 3] Sequence Number:

After the bytes have been numbered, TCP Assigns a sequence number to each segment that is being sent. The sequence number for each segment is the number of the first byte carried in that segment.

## 4] Acknowledgment Number:

The acknowledgment number defines the number of the next byte that the party expects to receive.

The acknowledgment number is cumulative, which means that the party takes the number of the last byte that it has received, safe and sound, adds 1 to it, and announces this sum as the **acknowledgment number**.