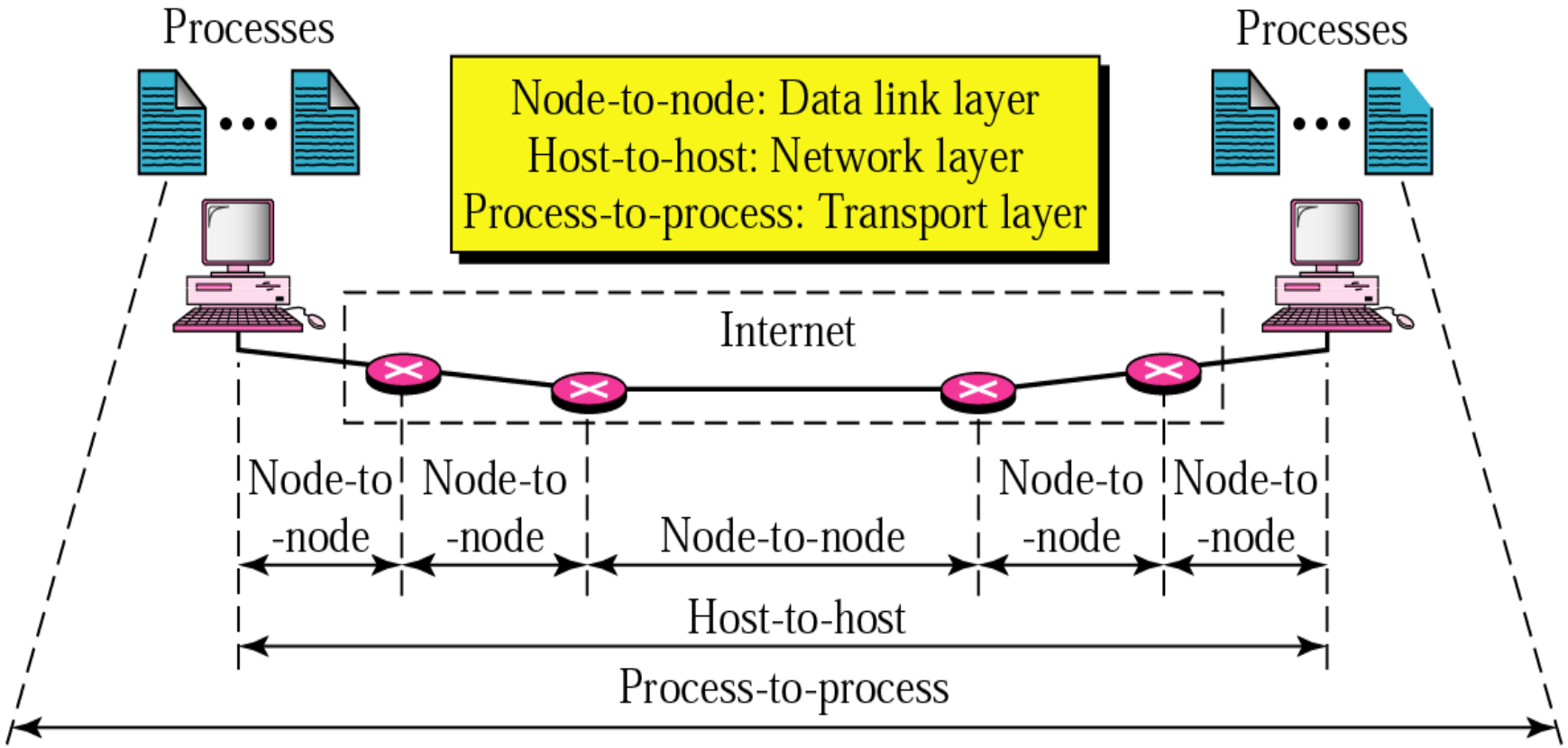
Transport Layer

End to End Delivery

- The transport layer is responsible for process-to-process delivery—the delivery of a packet, part of a message, from one process to another.
- Two processes communicate in a client/server relationship.

Types of data deliveries



End-to-End Protocols

- Common properties that a Transport Layer Protocol can be expected to provide
  - Guarantees message delivery
  - Delivers messages in the same order they were sent
  - Delivers at most one copy of each message
  - Supports arbitrarily large messages
  - Supports synchronization between the sender and the receiver
  - Allows the receiver to apply flow control to the sender
  - Supports multiple application processes on each host

End-to-End Protocols

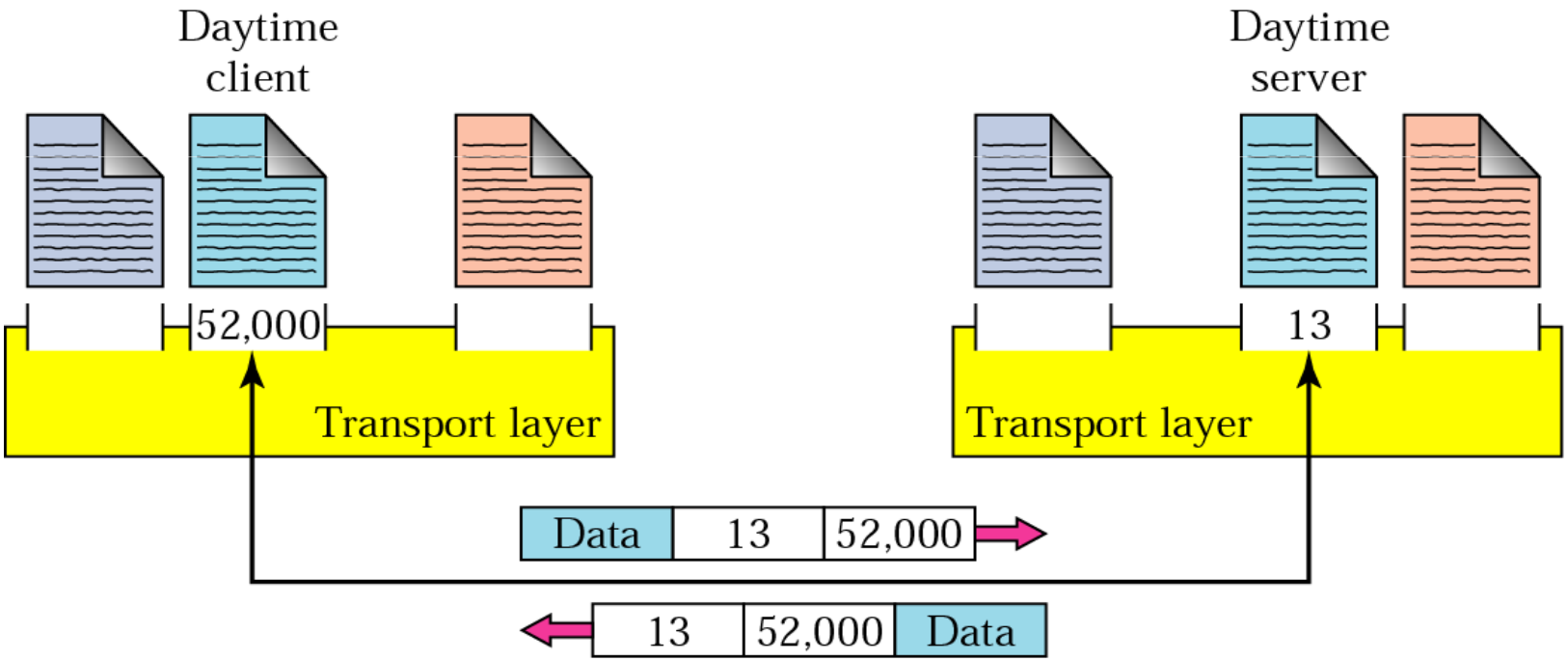
- Typical limitations of the network on which Transport Layer Protocol will operate
  - Drop messages
  - Reorder messages
  - Deliver duplicate copies of a given message
  - Limit messages to some finite size
  - Deliver messages after an arbitrarily long delay

End-to-End Protocols

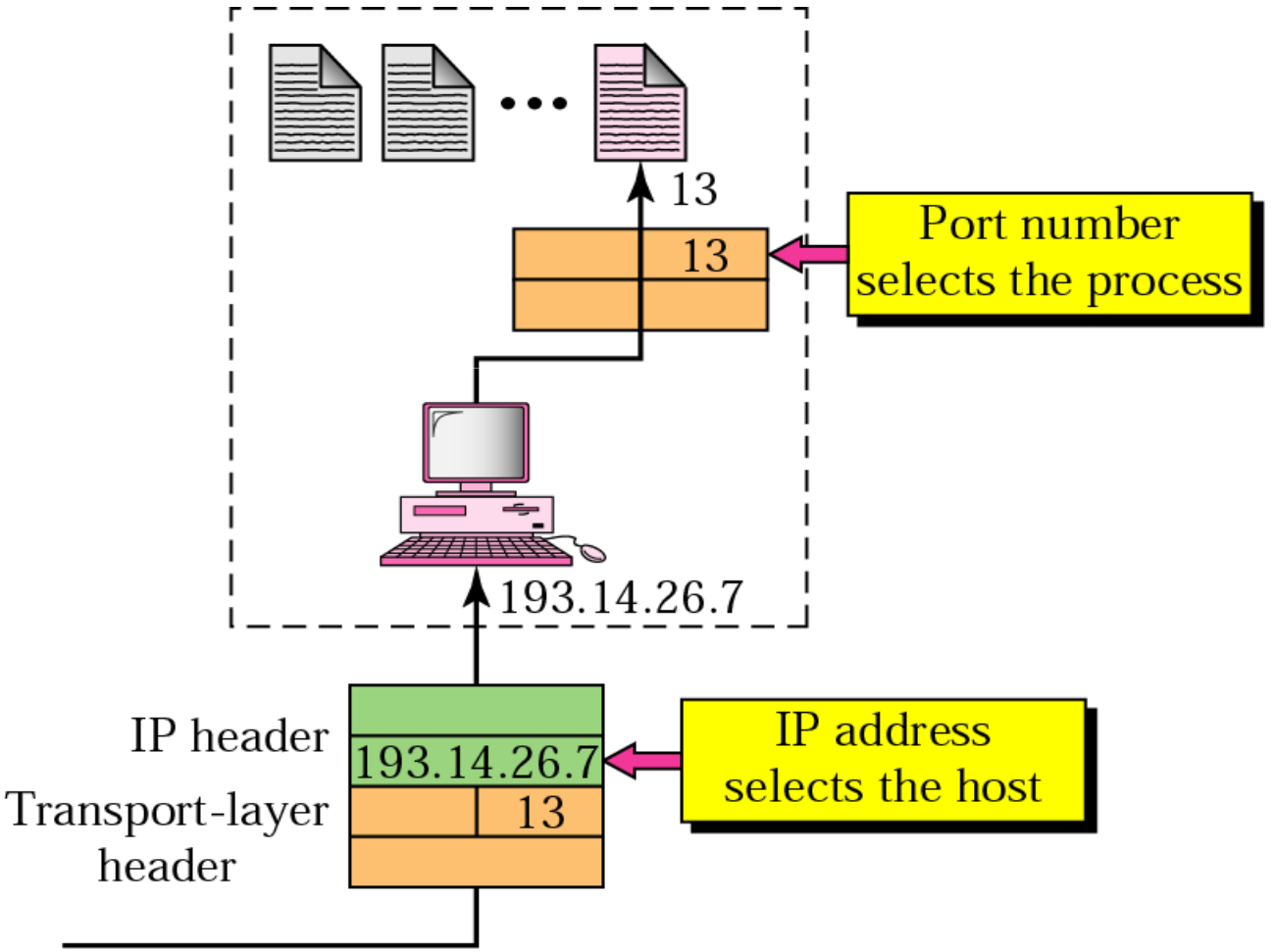
- Challenges for Transport Layer Protocols
  - Develop algorithms that turn the less-than-desirable properties of the underlying network into the high level of service required by application programs

Addressing

Port Numbers

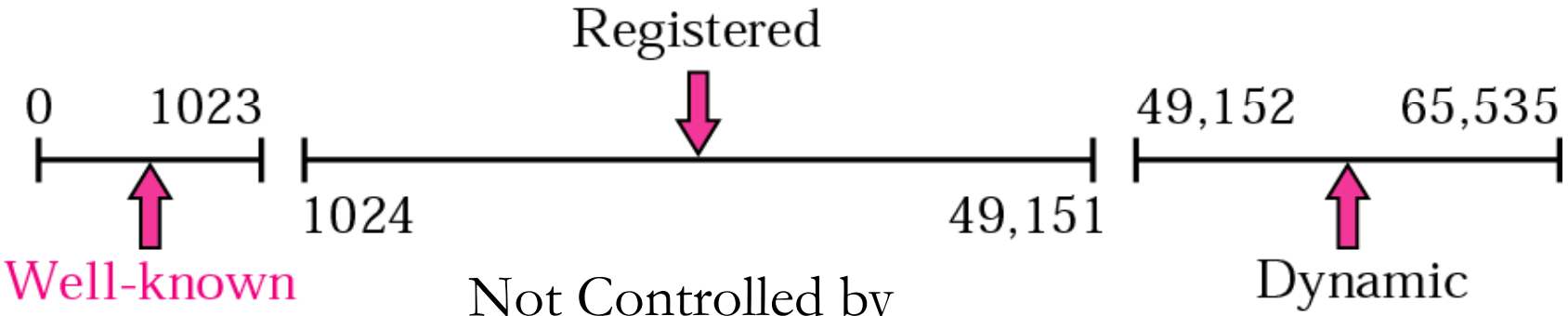


IP Addresses vs Port Numbers



IANA Ranges

- IANA – Internet Assigned Number Authority

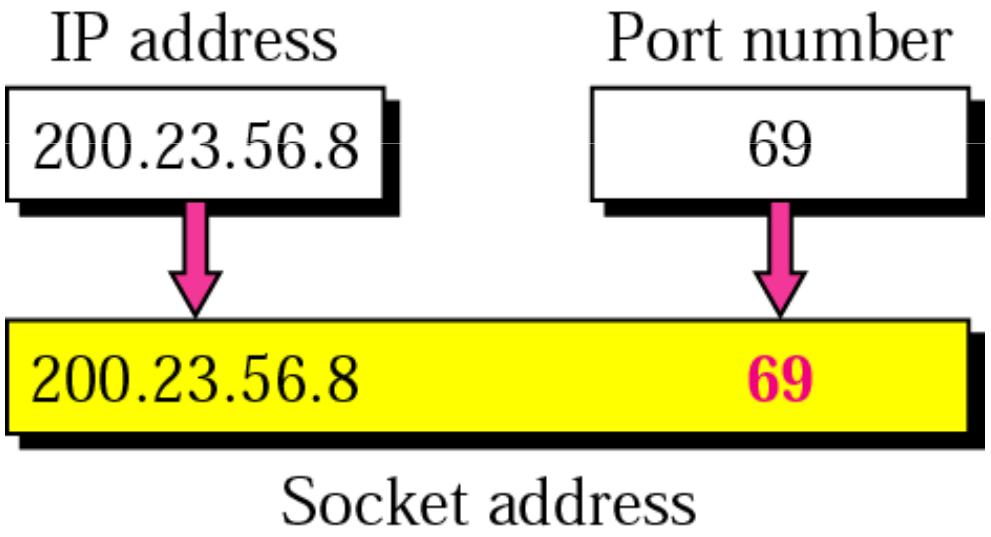


Controlled by IANA

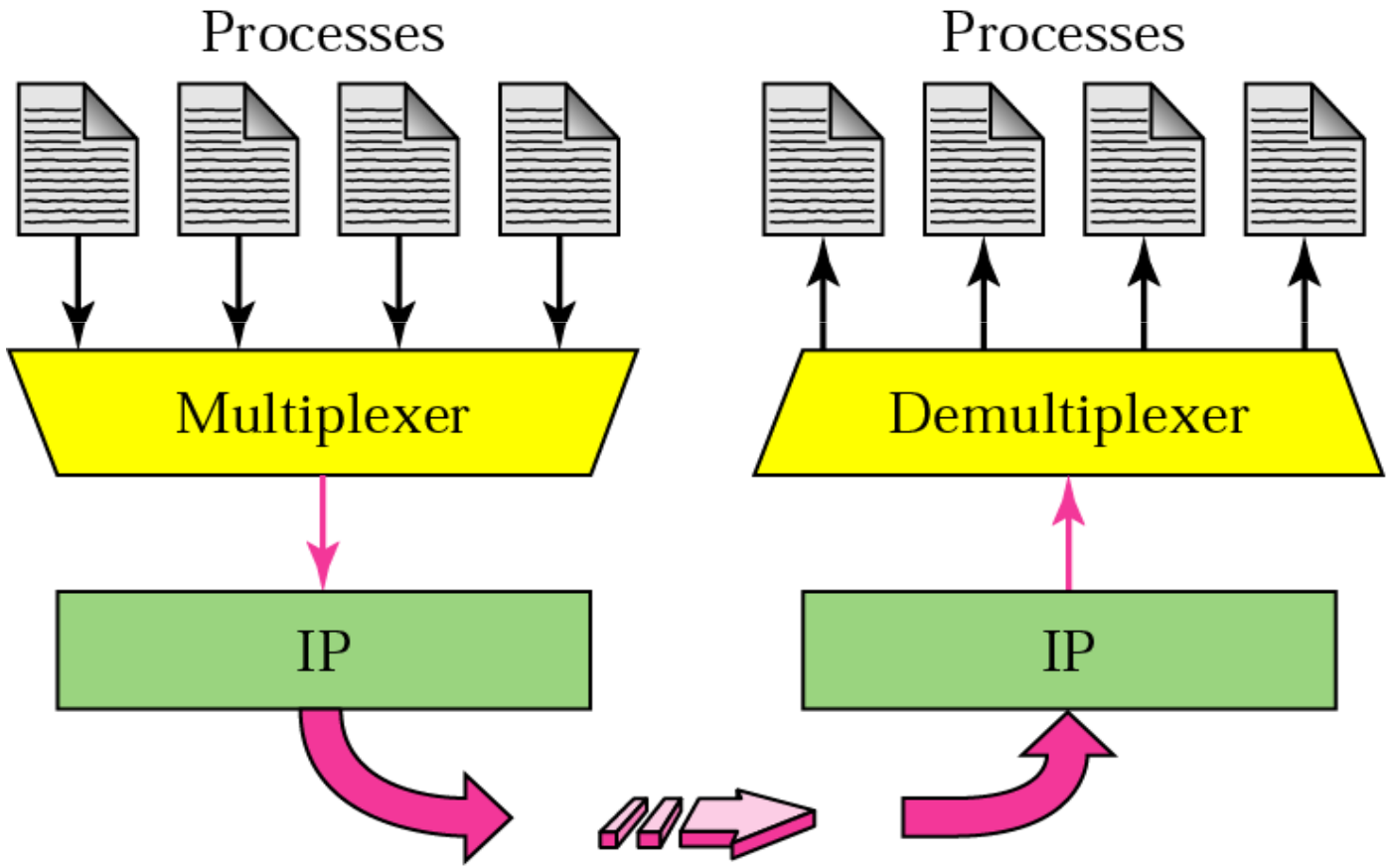
Not Controlled by IANA only registered with IANA

Neither Controlled nor registered with IANA

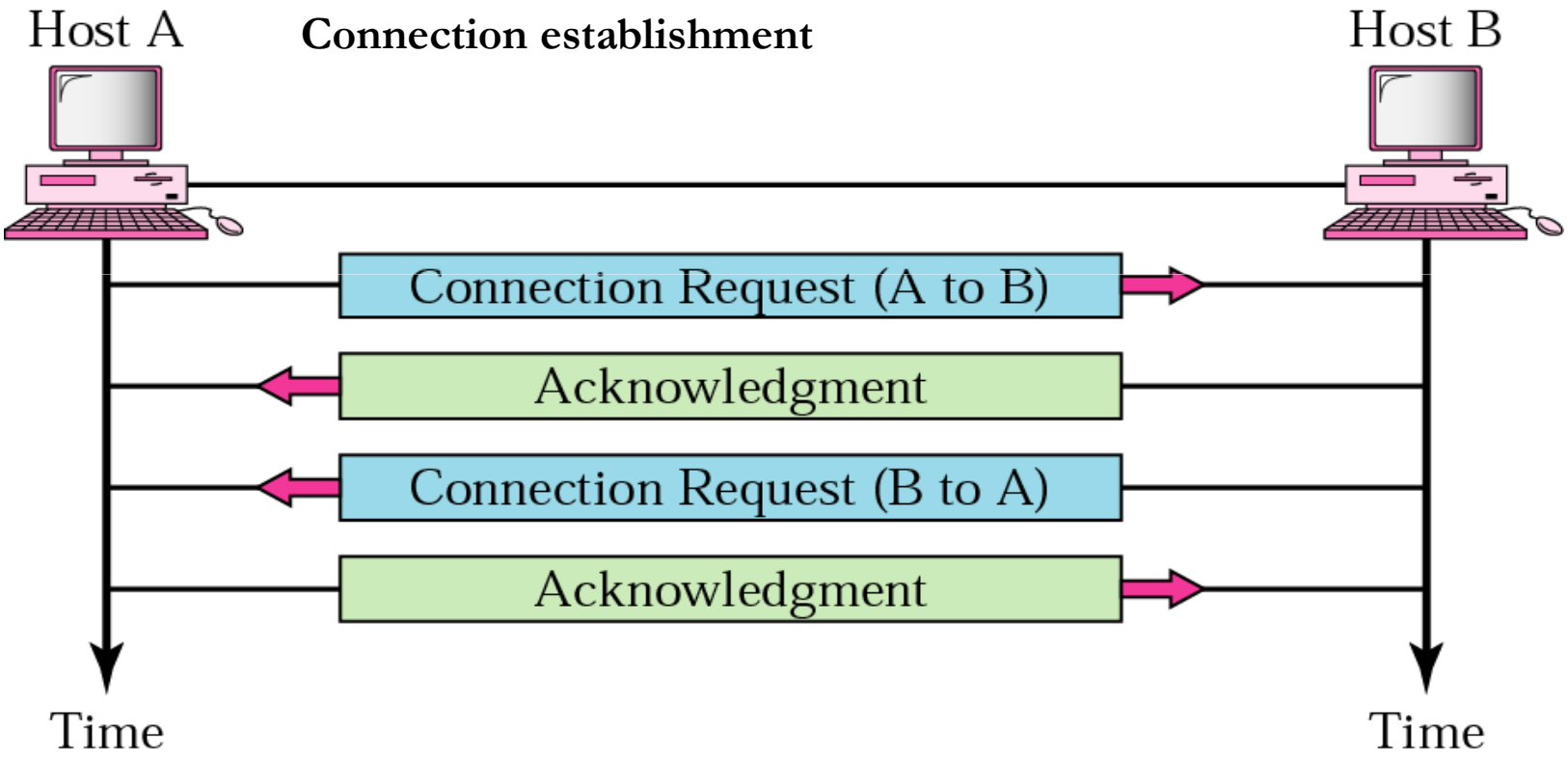
Socket Address



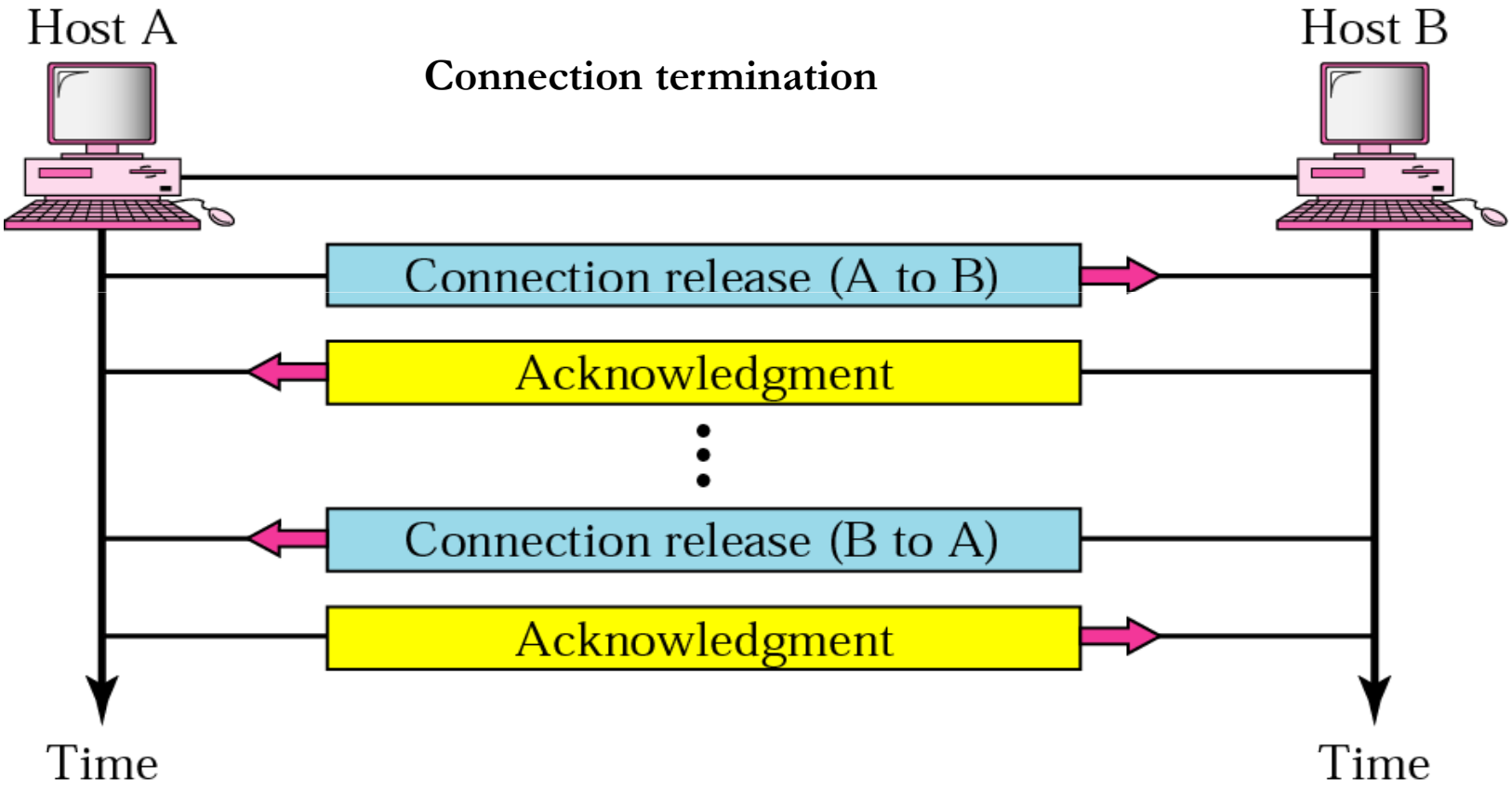
Multiplexing and Demultiplexing



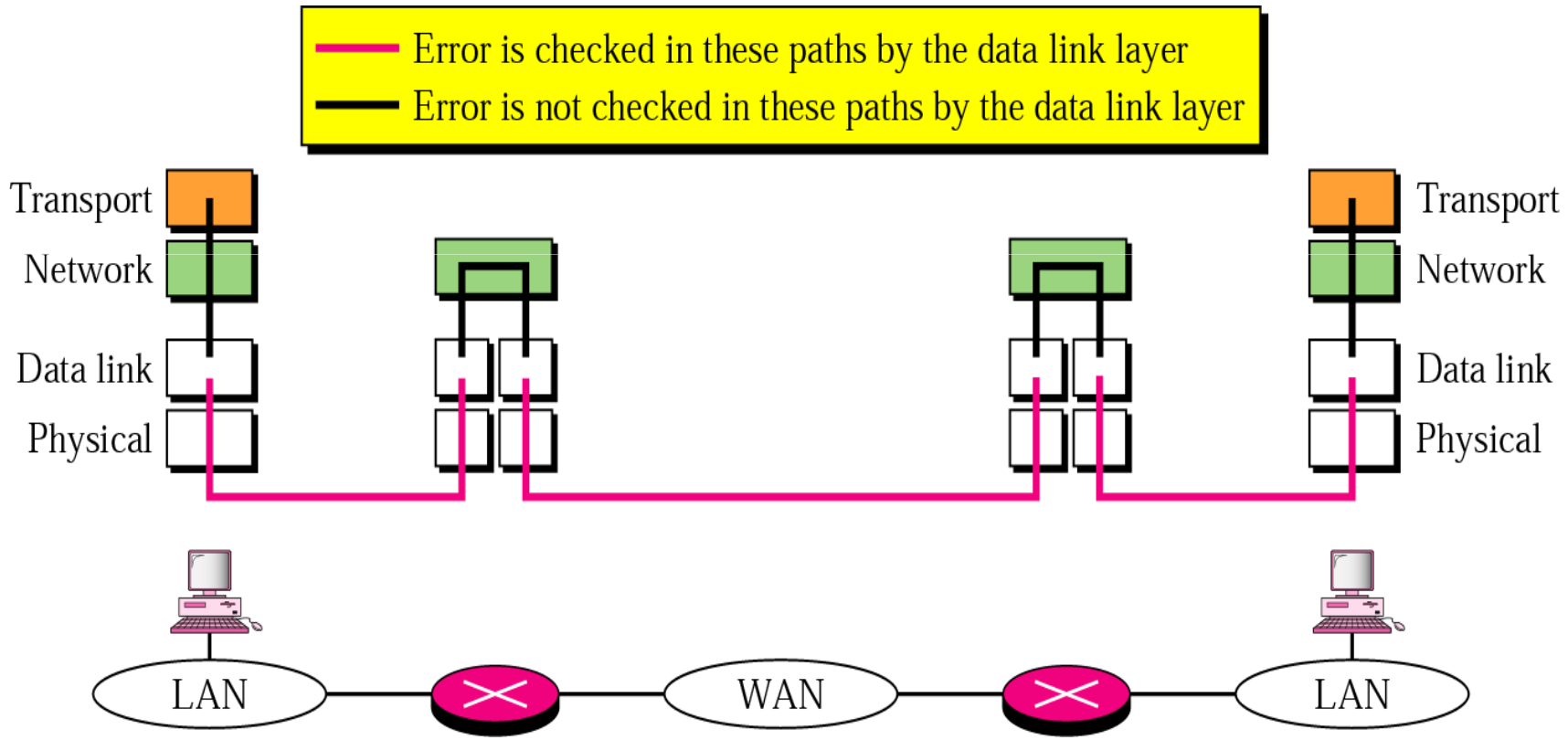
Connection Oriented Vs Connection Less Service



Connection Oriented Vs Connection less Service



Reliable Vs UnReliable



UDP - User Datagram Protocol

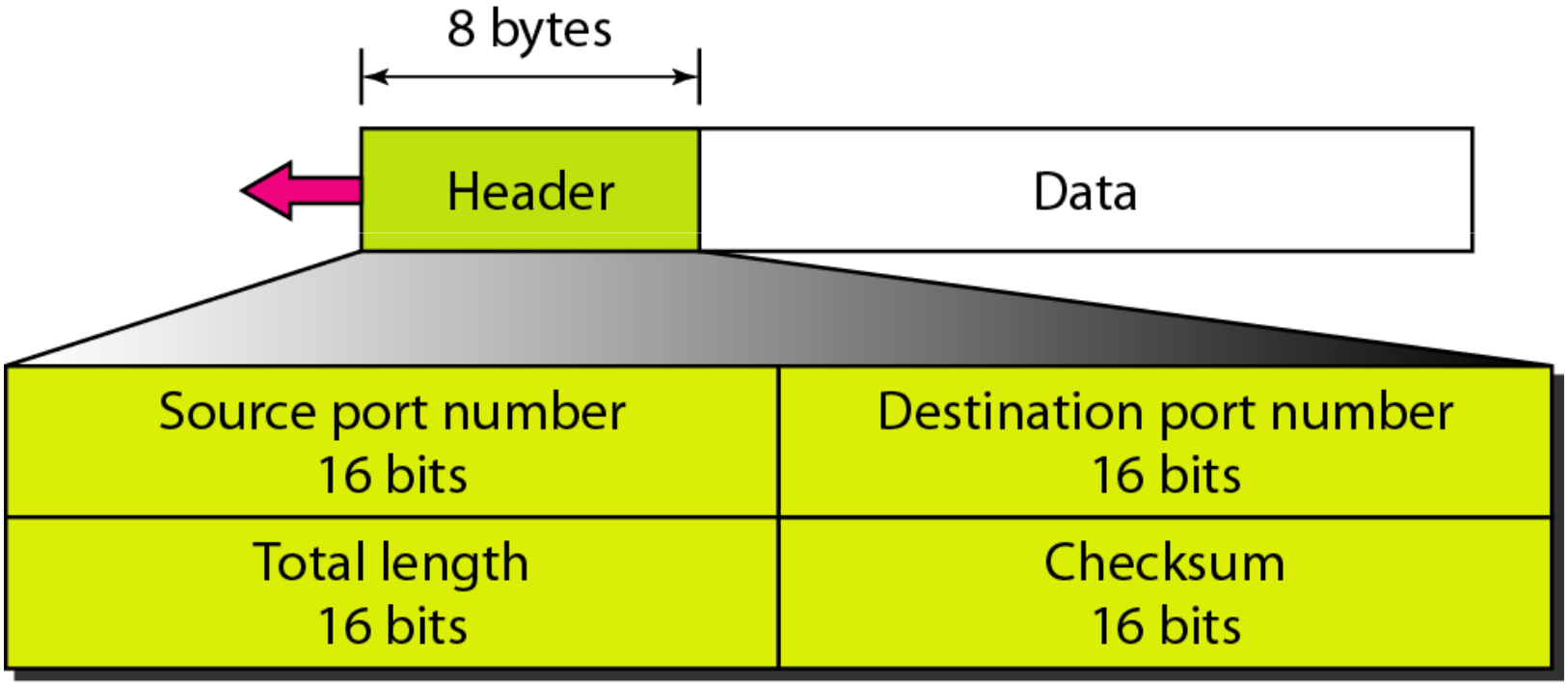
UDP

- UDP is a connectionless, unreliable protocol that has no flow and error control.
- It uses port numbers to multiplex data from the application layer.
- UDP is a convenient Transport Layer Protocol for applications that provide flow and error control.
- It is also used by Multimedia Applications

Well-known ports used with UDP

Port	*Protocol*	*Description*
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
53	Nameserver	Domain Name Service
67	BOOTPs	Server port to download bootstrap information
68	BOOTPc	Client port to download bootstrap information
69	TFTP	Trivial File Transfer Protocol
111	RPC	Remote Procedure Call
123	NTP	Network Time Protocol
161	SNMP	Simple Network Management Protocol
162	SNMP	Simple Network Management Protocol (trap)

UDP Format


8 bytes
Header
Data
Source port number
16 bits
Destination port number
16 bits
Total length
16 bits
Checksum
16 bits

Length

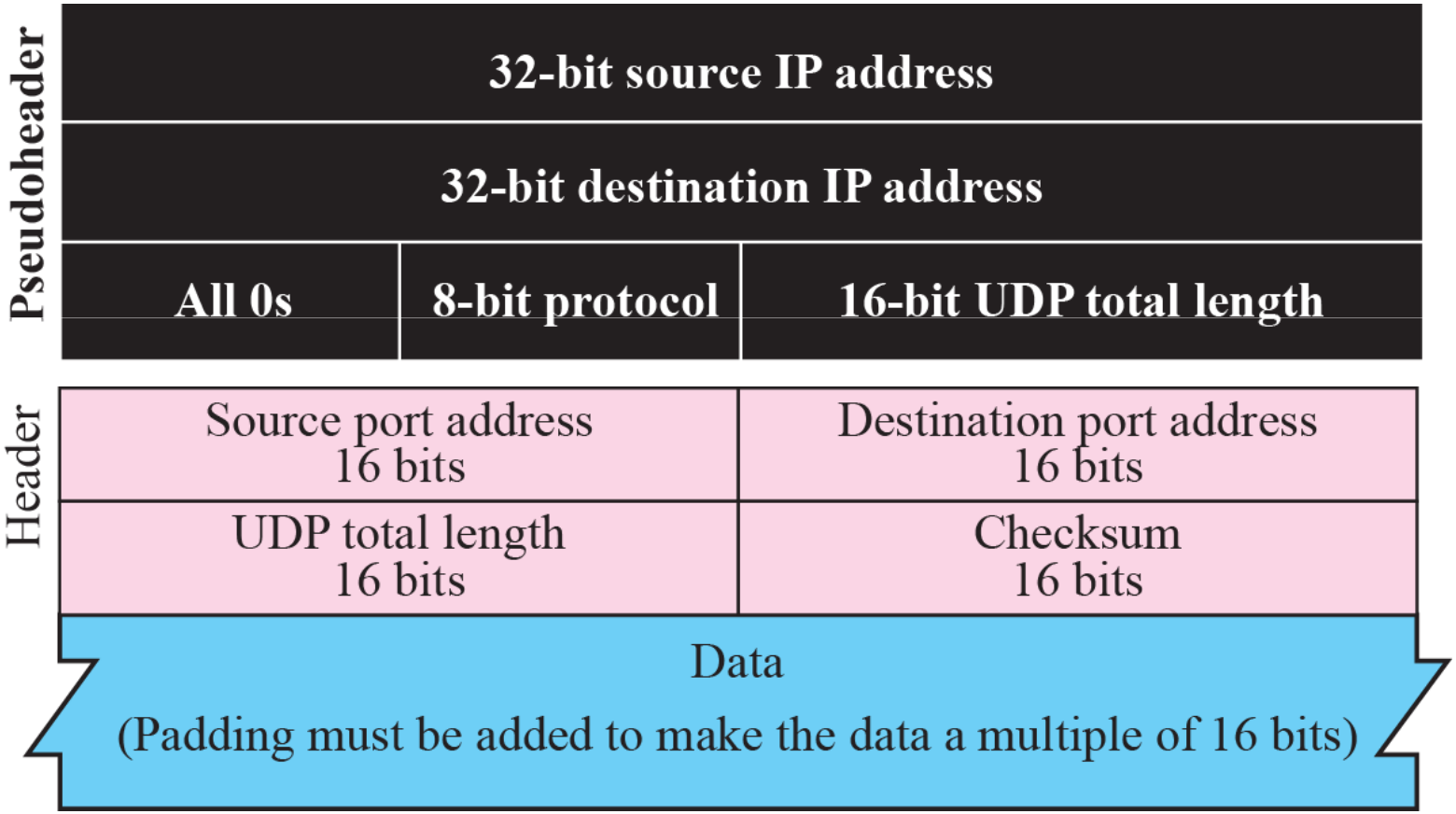
- Total length ie. Header + data
- Maximum Length of IP datagram =65,535 bytes
- UDP data should be stored in IP datagram
- Therefore

UDP length
= IP length – IP header's length

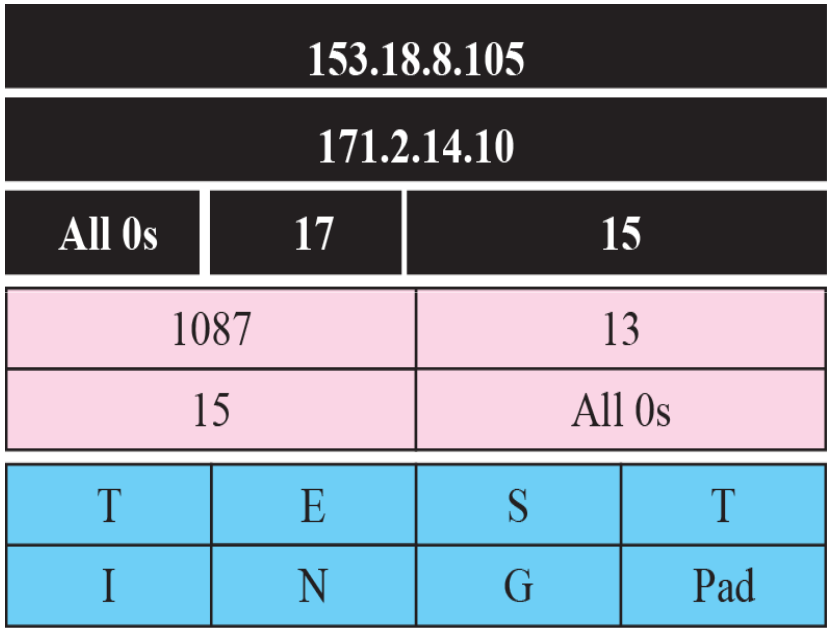
Checksum

- Different from IP and ICMP header checksum
- Adds a pseudo header instead of IP header and find checksum
- 3 sections in Checksum
  - Pseudoheader
  - The UDP header
  - The data from the application layer
- Pseudo header is part of the header of the IP packet

Checksum


Pseudoheader
32-bit source IP address
32-bit destination IP address
All 0s
8-bit protocol
16-bit UDP total length
Header
Source port address
16 bits
Destination port address
16 bits
UDP total length
16 bits
Checksum
16 bits
Data
(Padding must be added to make the data a multiple of 16 bits)

Checksum



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

Example

- The following is a a UDP header in HD format.

0x CB84	0x 000D
0x 001C	0x 001C

1. What is the source port number?
2. What is the destination port number?
3. What is the total length of the user datagram?
4. What is the length of the data?
5. Is the packet directed from a client to a server or vice versa?
6. What is the client process?

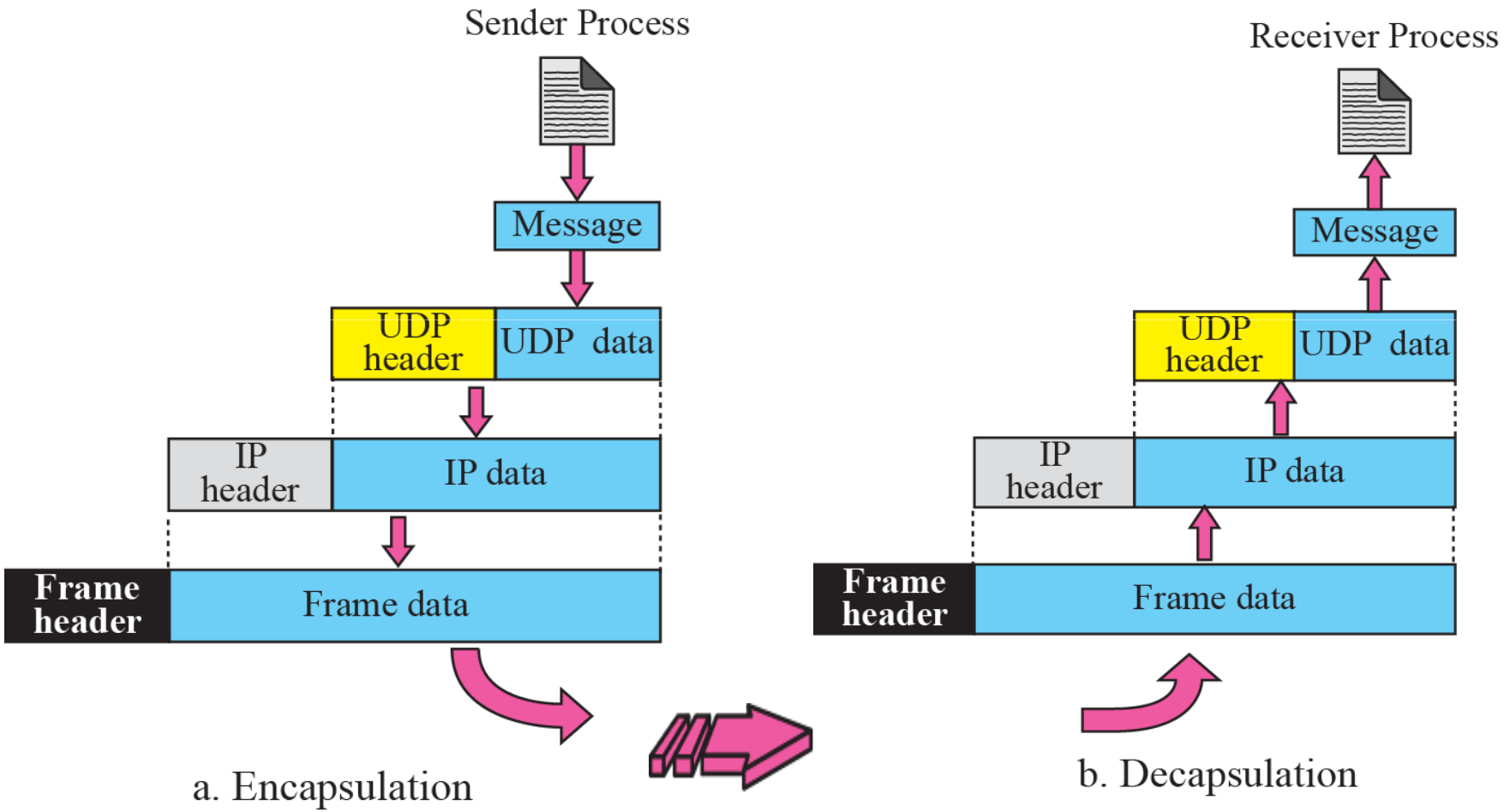
Example

1. The source port number is the first four hexadecimal digits $(CB84)_{16}$ or 52100.
2. The destination port number is the second four hexadecimal digits $(000D)_{16}$ or 13.
3. The third four hexadecimal digits $(001C)_{16}$ define the length of the whole UDP packet as 28 bytes.
4. The length of the data is the length of the whole packet minus the length of the header, or 28 – 8 = 20 bytes.
5. Since the destination port number is 13 (well-known port), the packet is from the client to the server.
6. The client process is the Daytime

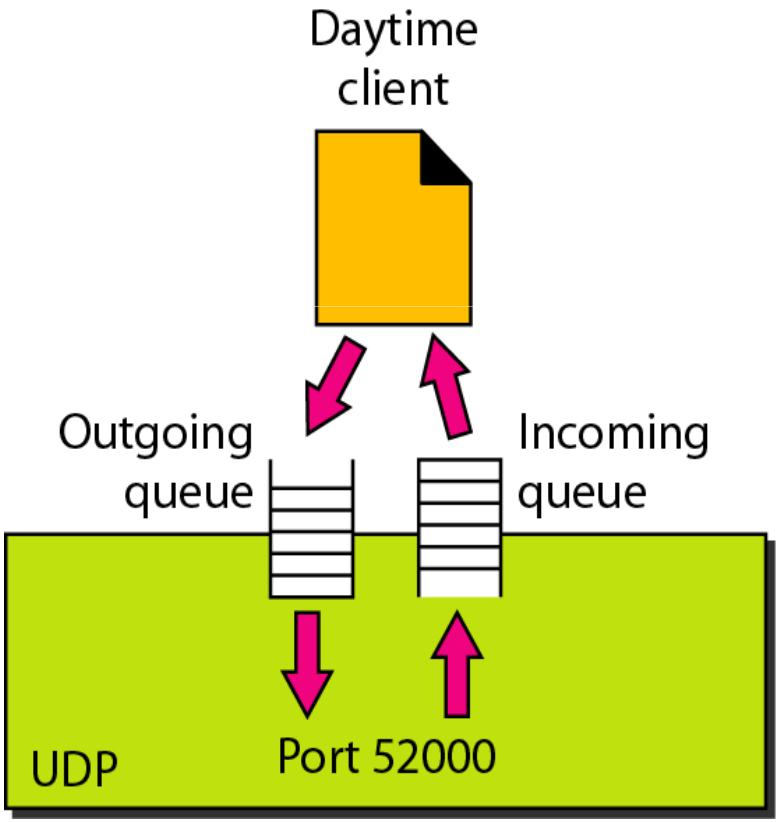
UDP Operation

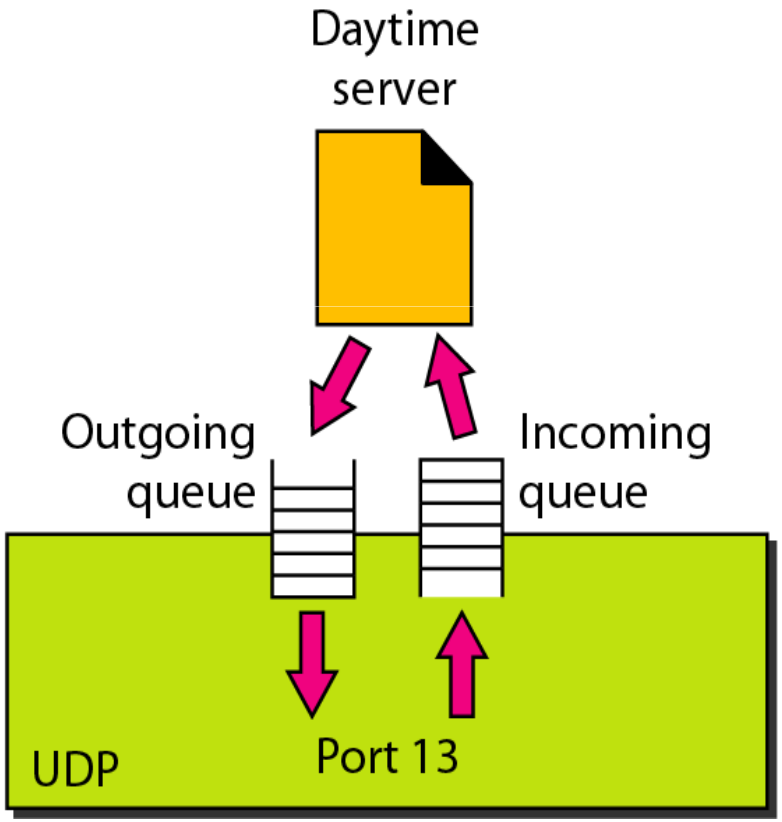
- Connection less service
- Flow control and error control
  - No such thing.
- Encapsulation and Decapsulation
- Queuing

Encapsulation & Decapsulation



Queuing


Daytime
client
Outgoing
queue
Incoming
queue
UDP
Port 52000


Daytime
server
Outgoing
queue
Incoming
queue
UDP
Port 13

Uses of UDP

- Simple request response communication
- Multicasting
- Management process such as SNMP
- Routing protocols such as RIP,OSPF