Computer Networks
COL 334/672

Transport Layer

Tarun Mangla

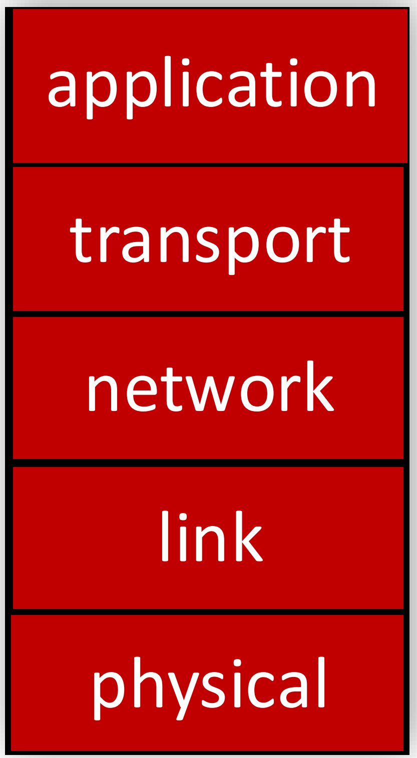
Sem 1, 2024-25

Quiz on Moodle

- Password: sdn

Course Recap

- *application:* supporting network applications
  - HTTP, IMAP, SMTP, DNS
- *transport:* process-process data transfer
  - TCP, UDP
- *network:* routing of datagrams from source to destination
  - IP, routing protocols
- *link:* data transfer between neighboring network elements
  - Ethernet, 802.11 (WiFi), PPP
- *physical:* bits “on the wire”



application

transport

network

link

physical

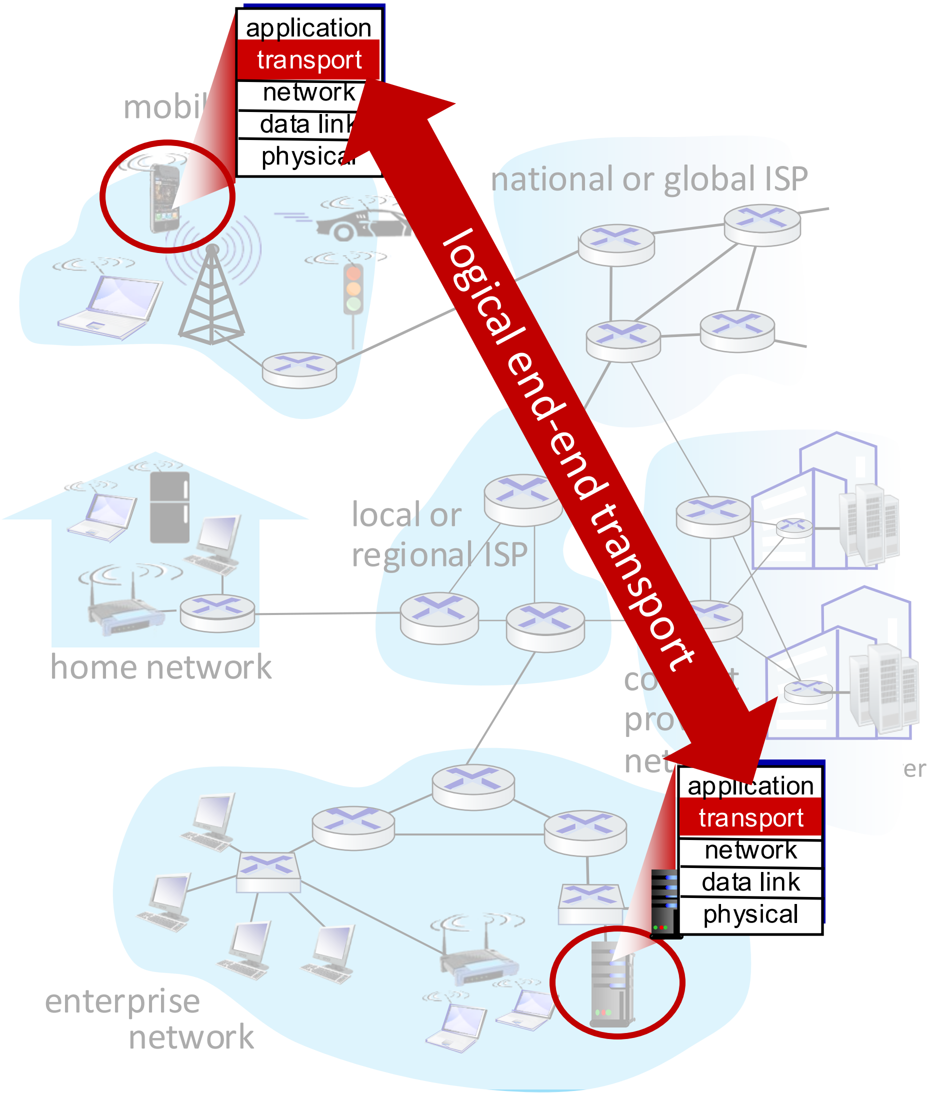
Transport layer: overview

Our goal:

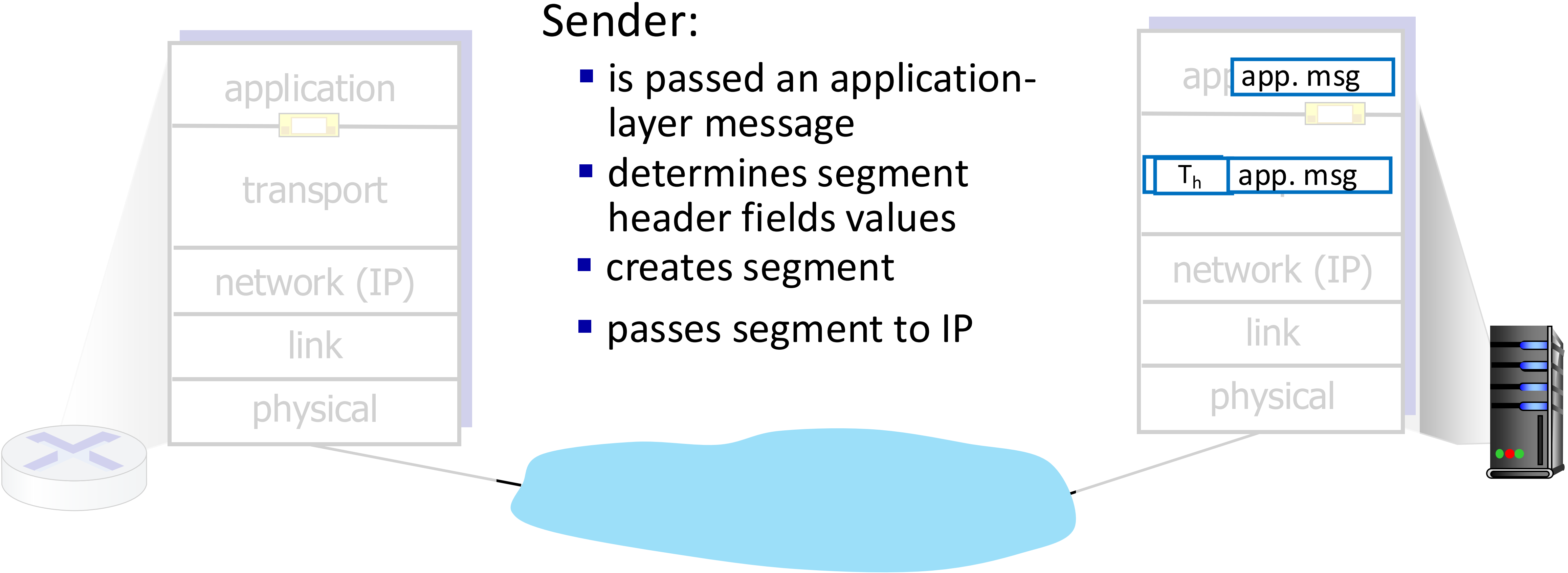
- Understand principles behind transport layer services:
  - multiplexing, demultiplexing
  - reliable data transfer
  - flow control
  - congestion control
- Learn about Internet transport layer protocols:
  - UDP:
    - connectionless transport
  - TCP:
    - connection-oriented reliable transport
    - TCP congestion control

Transport services and protocols

- provide *logical communication* between application processes running on different hosts
- transport protocols actions in end systems:
  - sender: breaks application messages into *segments*, passes to network layer
  - receiver: reassembles segments into messages, passes to application layer
- two transport protocols available to Internet applications
  - TCP, UDP



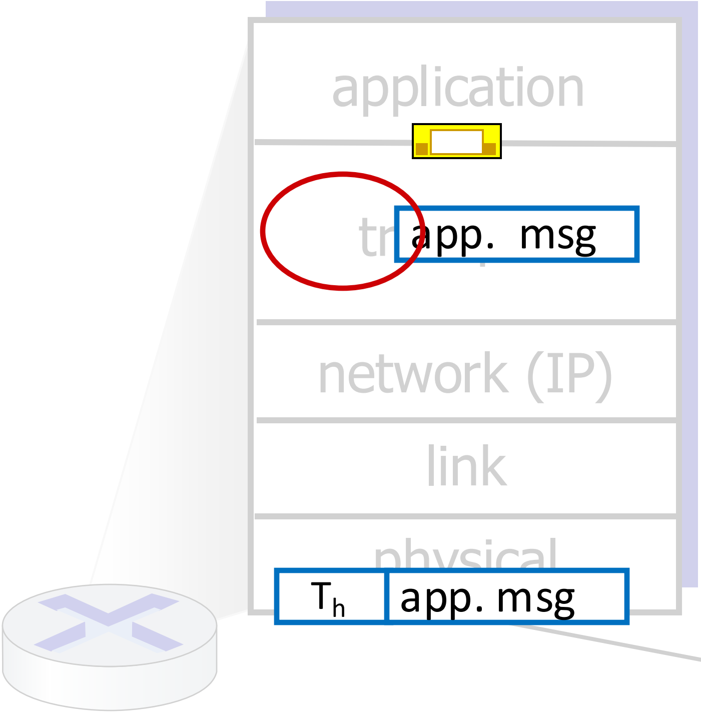
Transport Layer Actions



Sender:

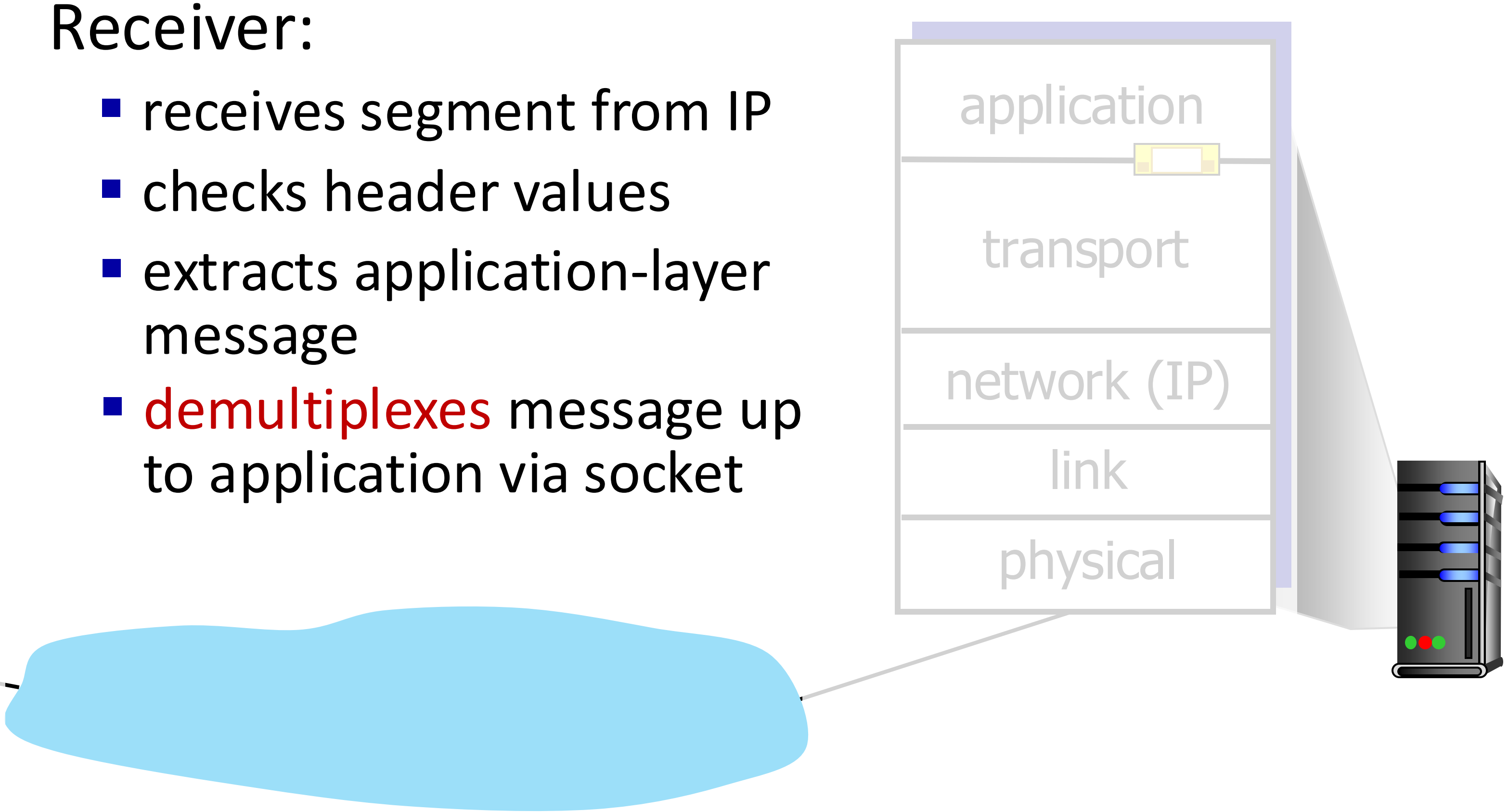
- is passed an application-layer message
- determines segment header fields values
- creates segment
- passes segment to IP

Transport Layer Actions



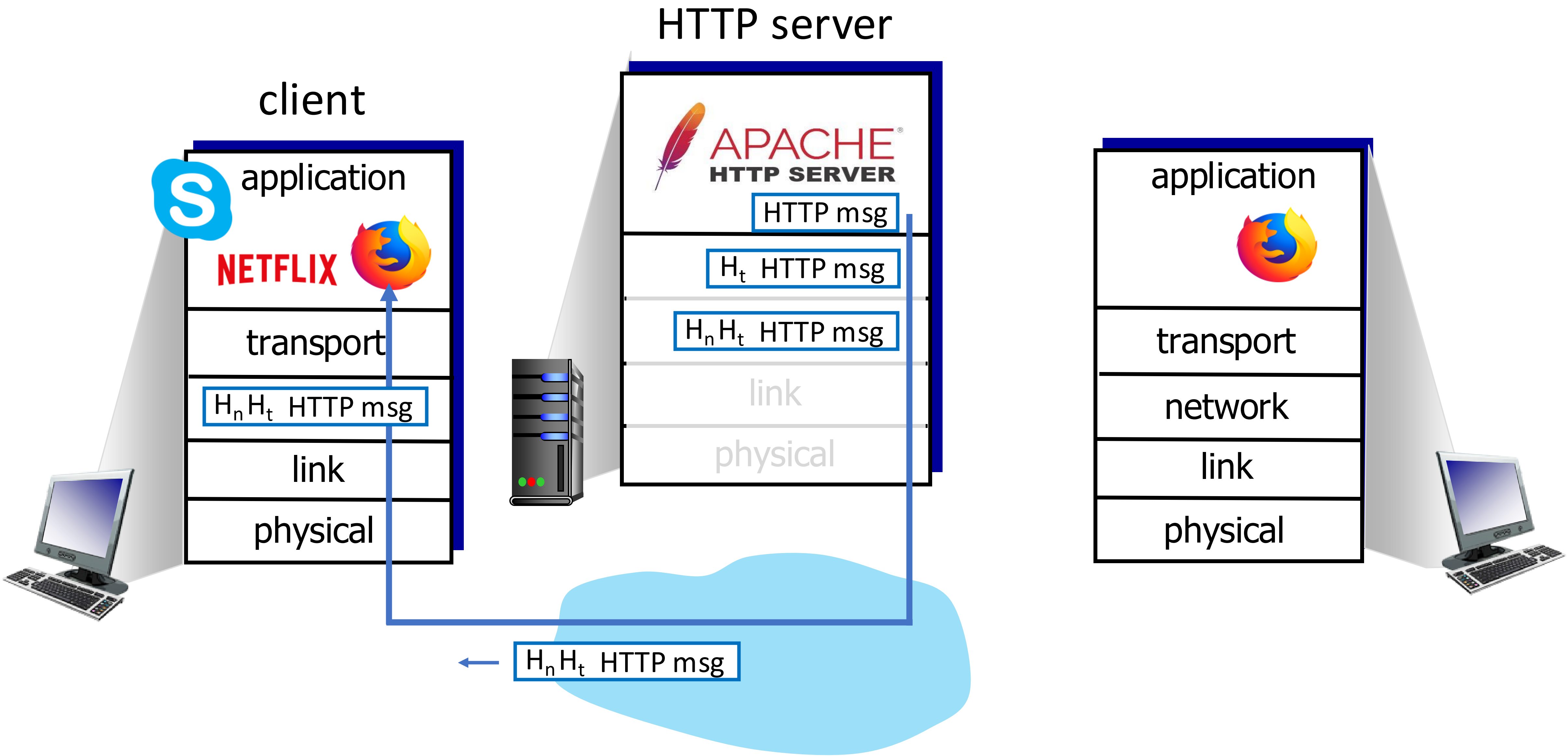
Receiver:

- receives segment from IP
- checks header values
- extracts application-layer message
- demultiplexes message up to application via socket

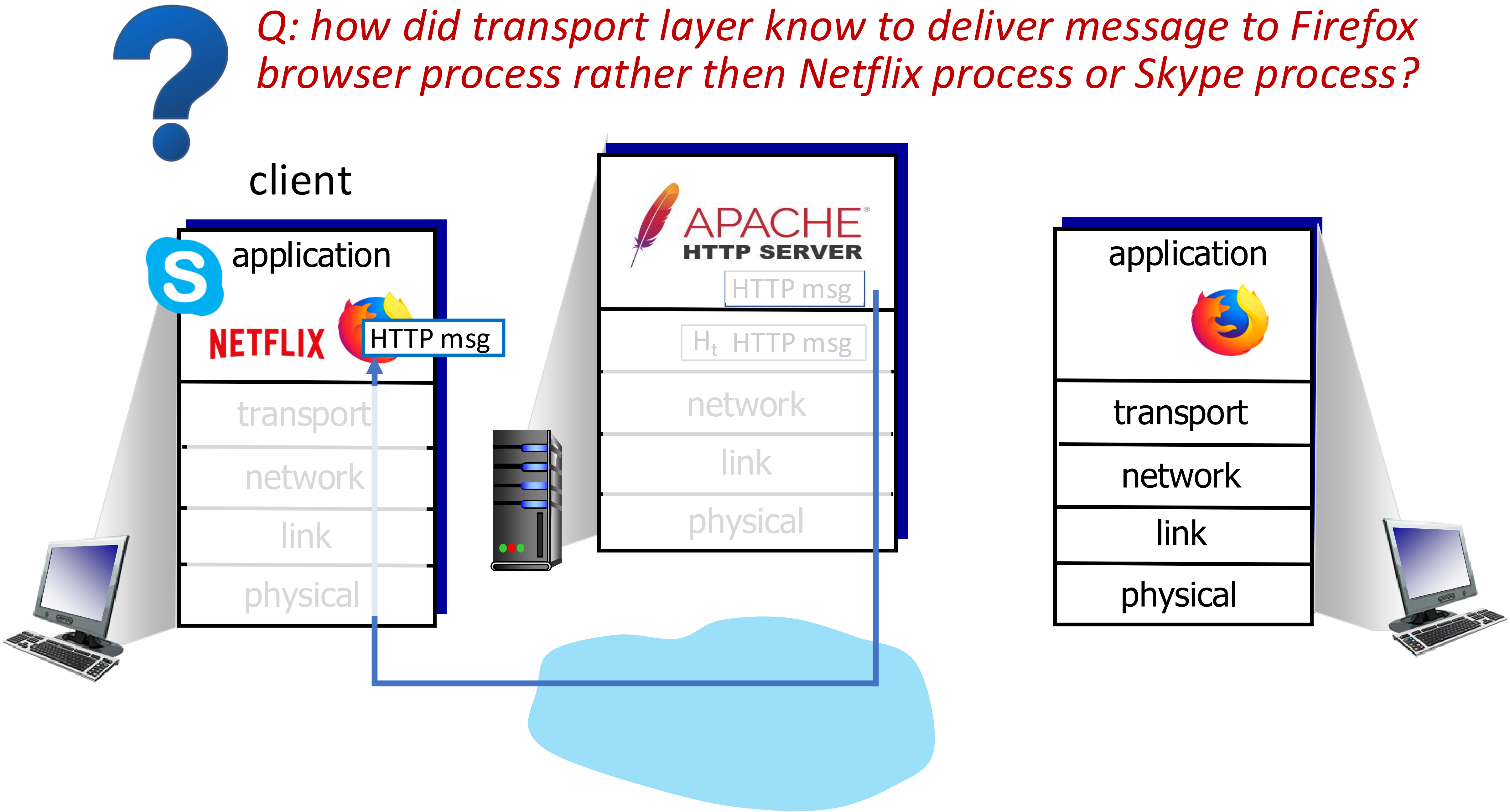


Transport-layer services

- Multiplexing/demultiplexing
- Reliable delivery
- Flow control
- Congestion control

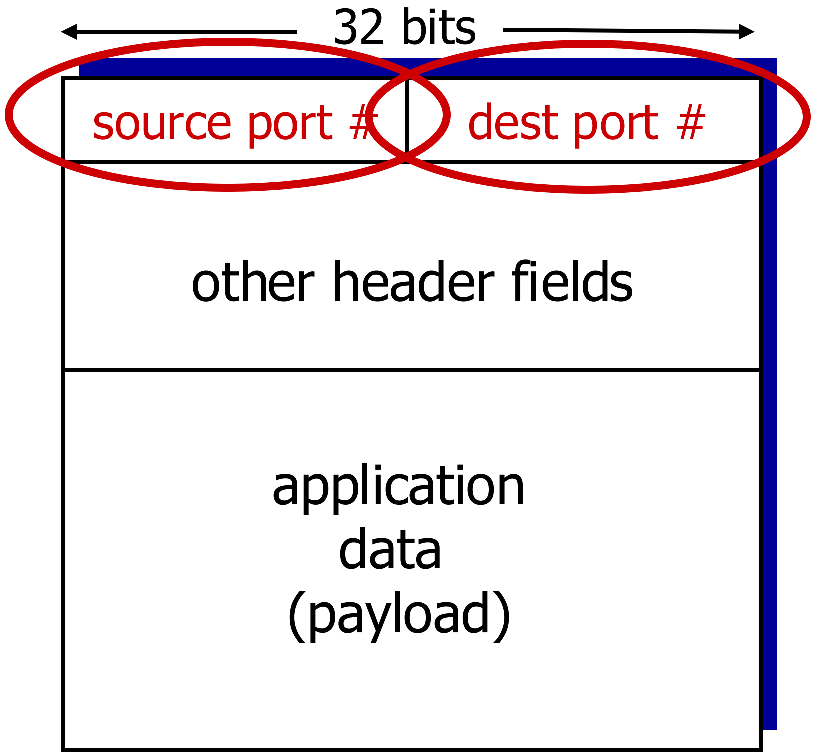

HTTP server
client
application
NETFLIX
transport
$H_n H_t$ HTTP msg
link
physical
APACHE
HTTP SERVER
HTTP msg
H_t HTTP msg
$H_n H_t$ HTTP msg
link
physical
$H_n H_t$ HTTP msg
application
transport
network
link
physical

Q: how did transport layer know to deliver message to Firefox browser process rather then Netflix process or Skype process?



How demultiplexing works

- host receives IP datagrams
  - each datagram has source IP address, destination IP address
  - each datagram carries one transport-layer segment
  - each segment has source, destination port number
- host uses *IP addresses & port numbers* to direct segment to appropriate socket



TCP/UDP segment format

Multiplexing/Demultiplexing

- Implemented using logical ports

- Difference in UDP/TCP multiplexing

- UDP or connection-less: Uses 2-tuple
  - (Dst IP, Dst Port) of the incoming packet aka bind (IP, port)

- TCP or connection-oriented: Uses 4-tuple
  - (Src IP, Src Port, Dst IP, Dst Port)

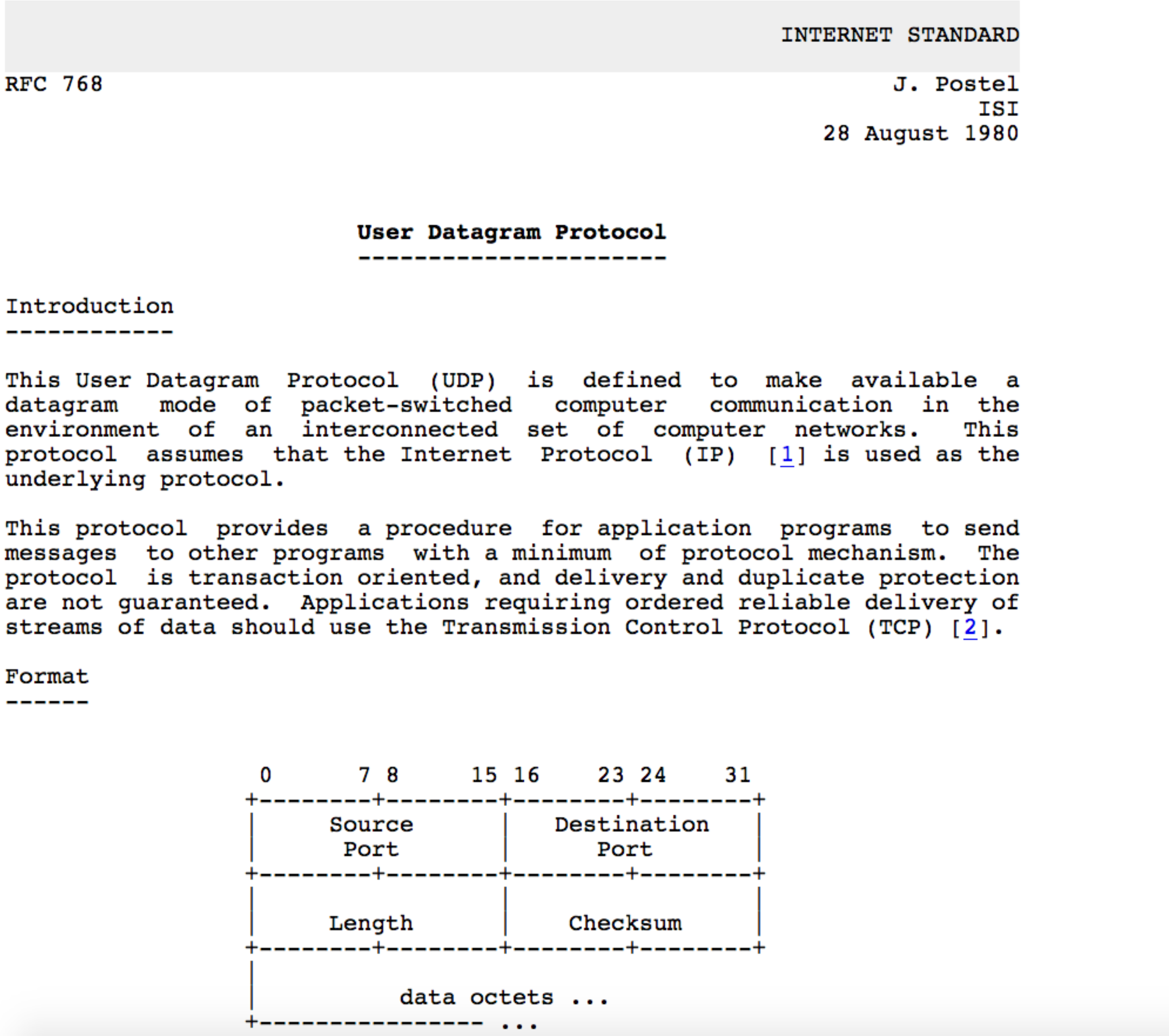
UDP: User Datagram Protocol

- “no frills,” “bare bones” Internet transport protocol
- “best effort” service, UDP segments may be:
  - lost
  - delivered out-of-order to app
- *connectionless:*
  - no handshaking between UDP sender, receiver
  - each UDP segment handled independently of others

Why is there a UDP?

- No connection establishment delay. Makes it particularly suitable for applications involving short transactions:
  - DNS
  - SNMP
  - DHCP
- Provides applications with the flexibility to implement only a subset of features
  - E.g., reliability is not needed for video conferencing applications

UDP: User Datagram Protocol [RFC 768]



INTERNET STANDARD

RFC 768
J. Postel
ISI
28 August 1980

User Datagram Protocol

Introduction

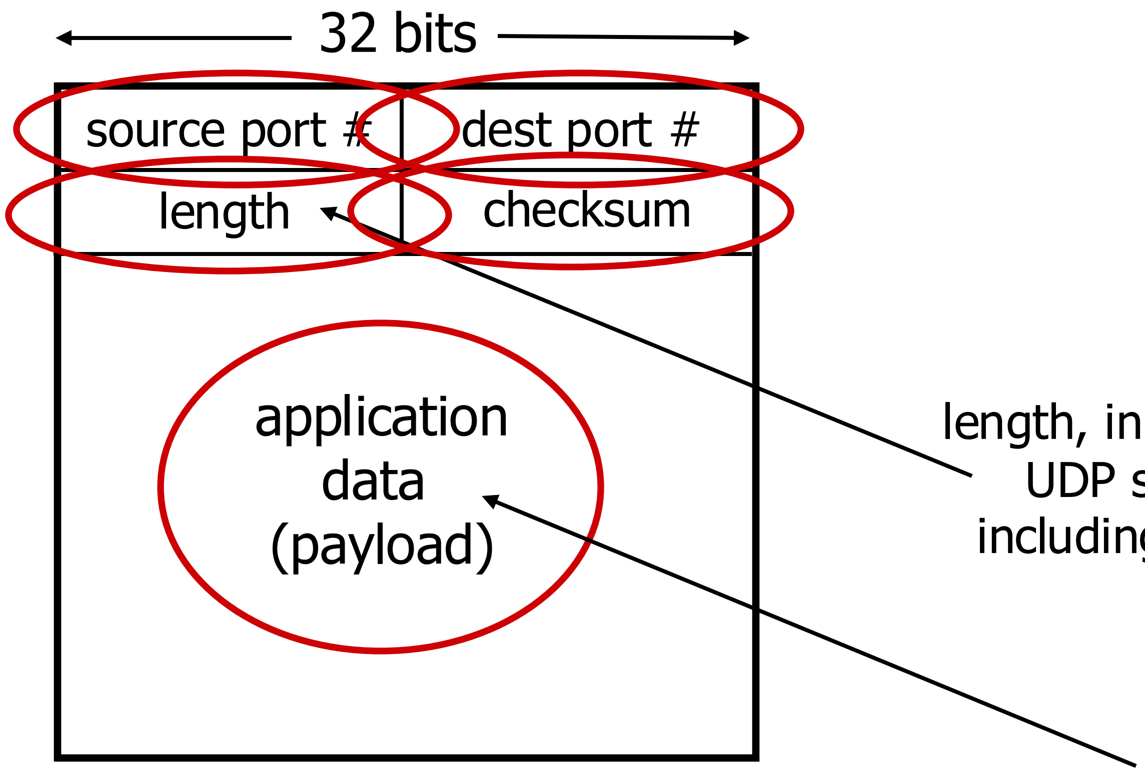
This User Datagram Protocol (UDP) is defined to make available a datagram mode of packet-switched computer communication in the environment of an interconnected set of computer networks. This protocol assumes that the Internet Protocol (IP) [1] is used as the underlying protocol.

This protocol provides a procedure for application programs to send messages to other programs with a minimum of protocol mechanism. The protocol is transaction oriented, and delivery and duplicate protection are not guaranteed. Applications requiring ordered reliable delivery of streams of data should use the Transmission Control Protocol (TCP) [2].

Format

```
 0      7 8     15 16    23 24    31
+--------+--------+--------+--------+
|     Source      |   Destination   |
|      Port       |      Port       |
+--------+--------+--------+--------+
|                 |                 |
|     Length      |    Checksum     |
+--------+--------+--------+--------+
|
|          data octets ...
+---------------- ...
```

UDP segment header



length, in bytes of UDP segment, including header

data to/from application layer

UDP segment format

```
0      7 8     15 16    23 24    31
+--------+--------+--------+--------+
|          source address           |
+--------+--------+--------+--------+
|        destination address        |
+--------+--------+--------+--------+
|  zero  |protocol|   UDP length    |
+--------+--------+--------+--------+
```

Pseudo-header