BRAC UNIVERSITY

Inspiring Excellence

# Introduction to Transport Layer

Lecture 4 | CSE421 – Computer Networks

Department of Computer Science and Engineering
School of Data & Science

# Our goals: Objectives

- understand principles behind transport layer services
- learn about two transport layer protocols:
  - UDP: User Datagram Protocol
  - TCP: Transmission Control Protocol

# Transport vs. Network layer

- ***transport layer:*** logical communication between **processes**
- ***network layer:*** logical communication between **hosts**
- ***Segments:*** Transport Layer PDU

***household analogy:***

*12 kids in Ann's house sending letters to 12 kids in Bill's house:*

- processes = kids
- app messages = letters in envelopes
- hosts = houses
- transport protocol = Ann and Bill
- network-layer protocol = postal service

# Functions of the Transport Layer

- Primary responsibilities:

1. Segmenting the data and managing each piece.
2. Reassembling the segments into streams of application data.
3. Identifying the different applications.
4. Multiplexing
5. Initiati[Reliability]ssion.
6. Performing flow control between end users.
7. Enabling error recovery.

# Different Applications

## Different Requirements

Web Applications






Online Games

- Some applications need their data to be complete with no errors or gaps.
- But can accept a slight delay to ensure this.

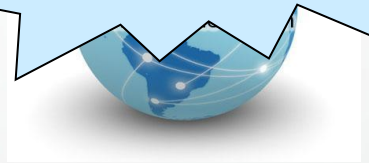
- Some applications can accept occasional errors or gaps in the data.
- But they cannot accept any delay.

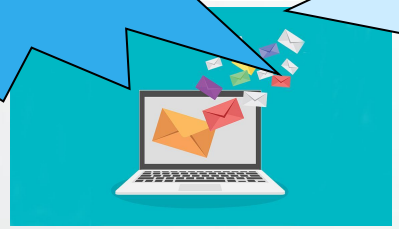
# Solution : Two transport protocols?

Web Applications

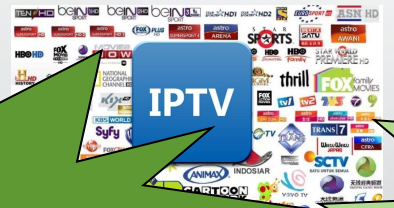
Reliable

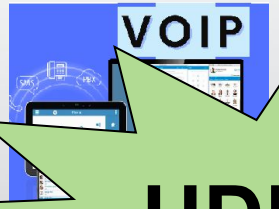
TCP





Emails




Fast

UDP


Online Games

# UDP: User Datagram Protocol [RFC 768]

- UDP : User Datagram Protocol
  - Best Effort Service
  - Used by applications that requires no delay in data delivery
- How does UDP deliver fast?
  - no connection establishment (which can add delay)
  - small header size
  - no error or flow or congestion control: UDP can blast away as fast as desired

# User Datagram Protocol (UDP)

- UDP is used by:
  - streaming multimedia apps (loss tolerant, rate sensitive)
  - SNMP

  But sometimes ……
  - DNS
  - HTTP/3

- reliable transfer over UDP:
  - add reliability at application layer
  - application-specific error recovery!

# Functions of the Transport Layer

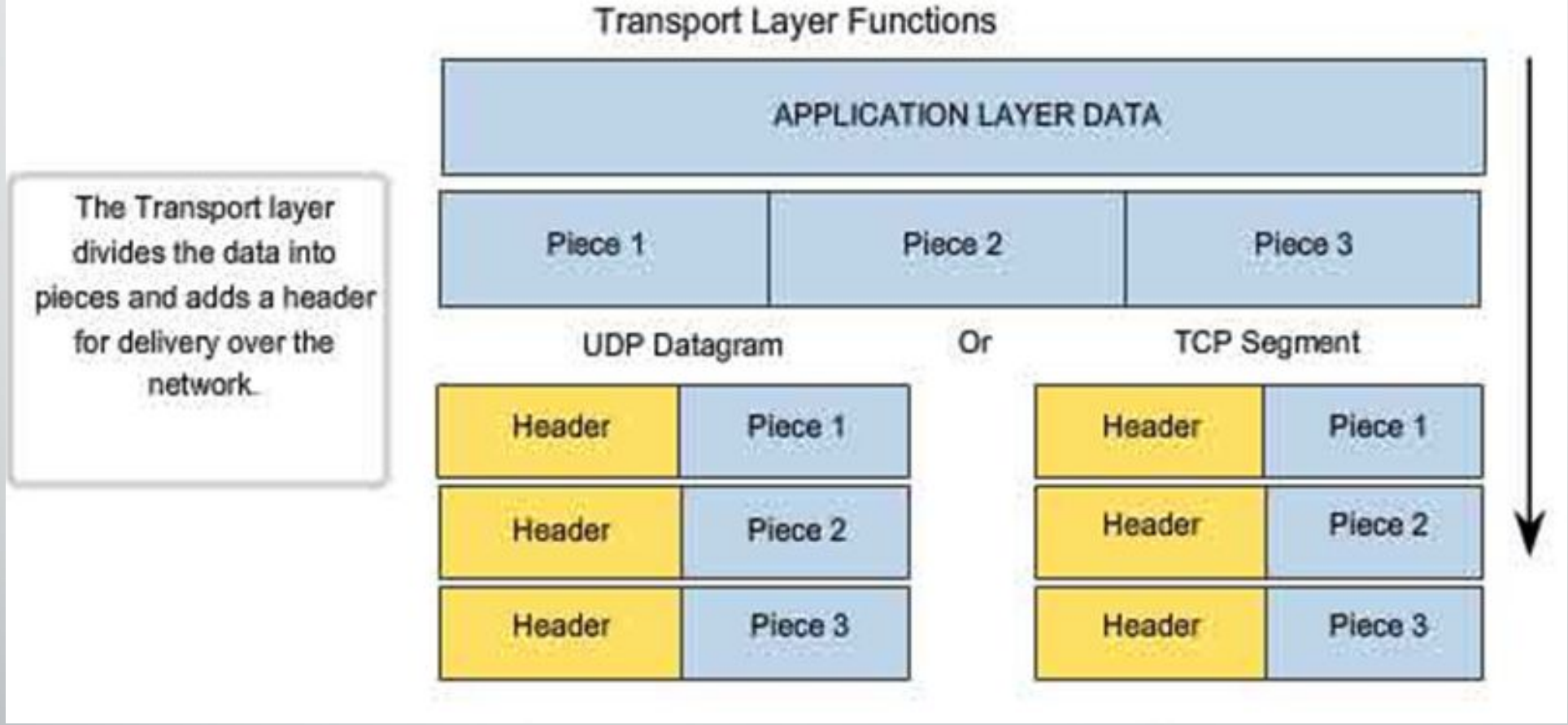
- Primary responsibilities:

1. Segmenting the data and managing each piece.
2. Reassembling the segments into streams of application data.
3. Identifying the different applications.
4. Multiplexing
5. Establishing and terminating a connection
6. Enabling error recovery.
7. Performing flow control between end users.

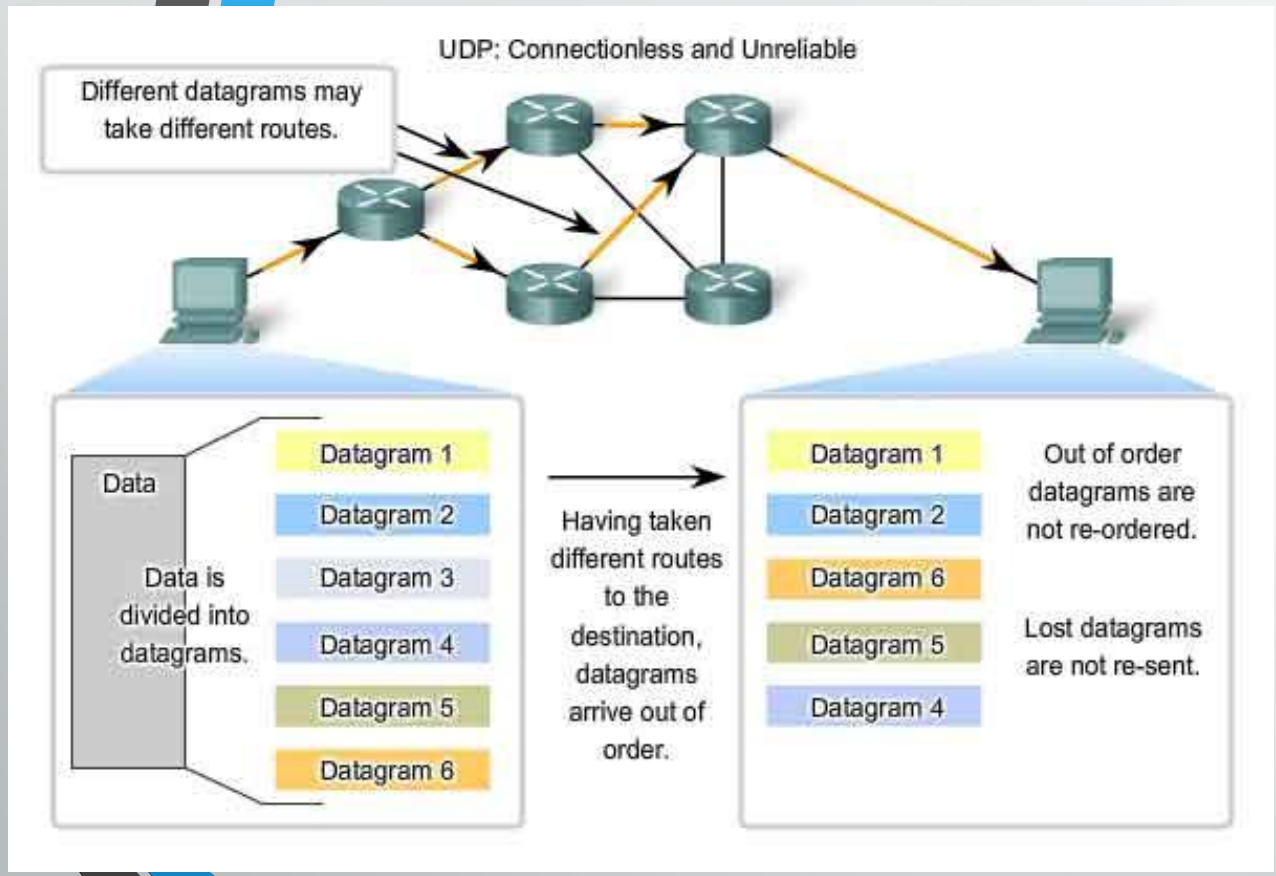
UDP &TCP

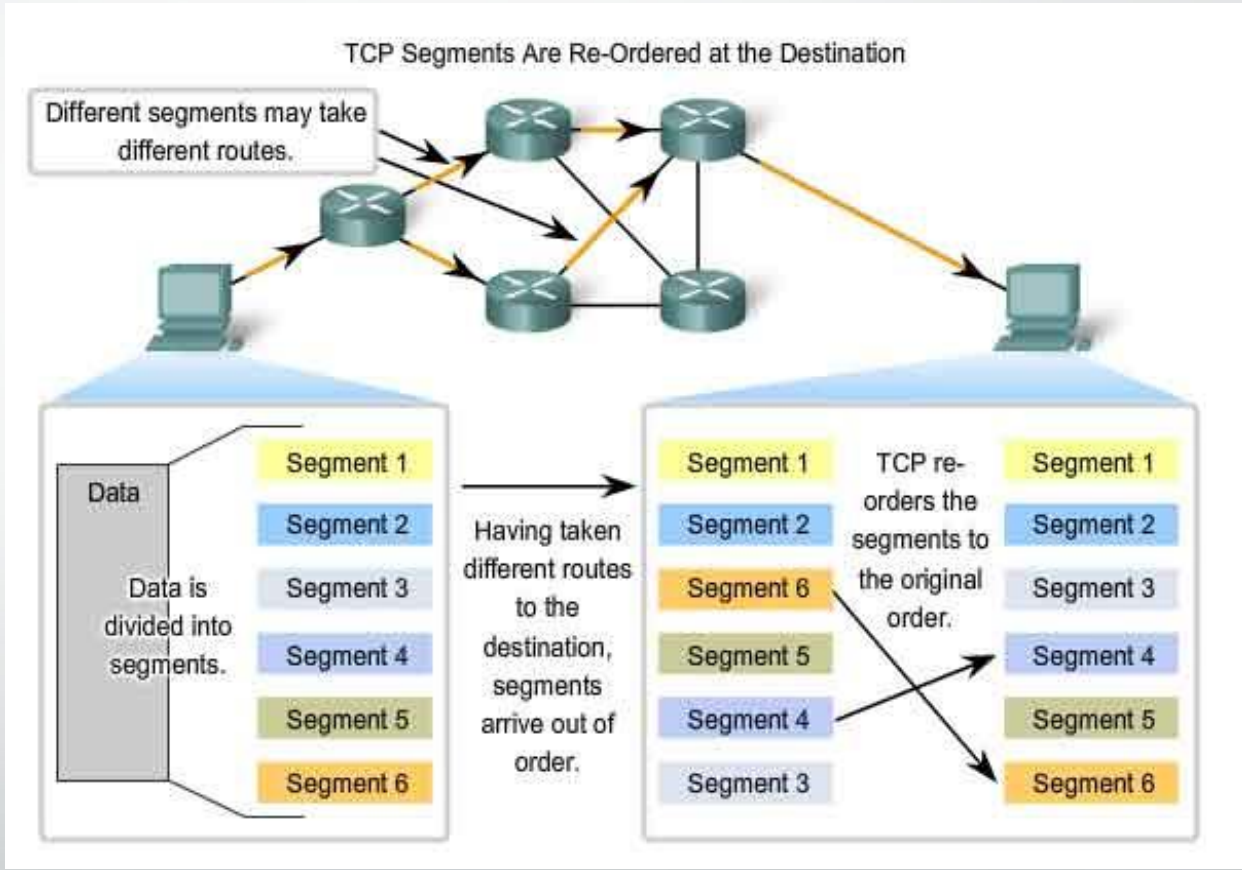
Only TCP

# Function 1&2 – Segmentation and Reassembly



# Function 2 – Reassembly




**UDP**

**TCP**

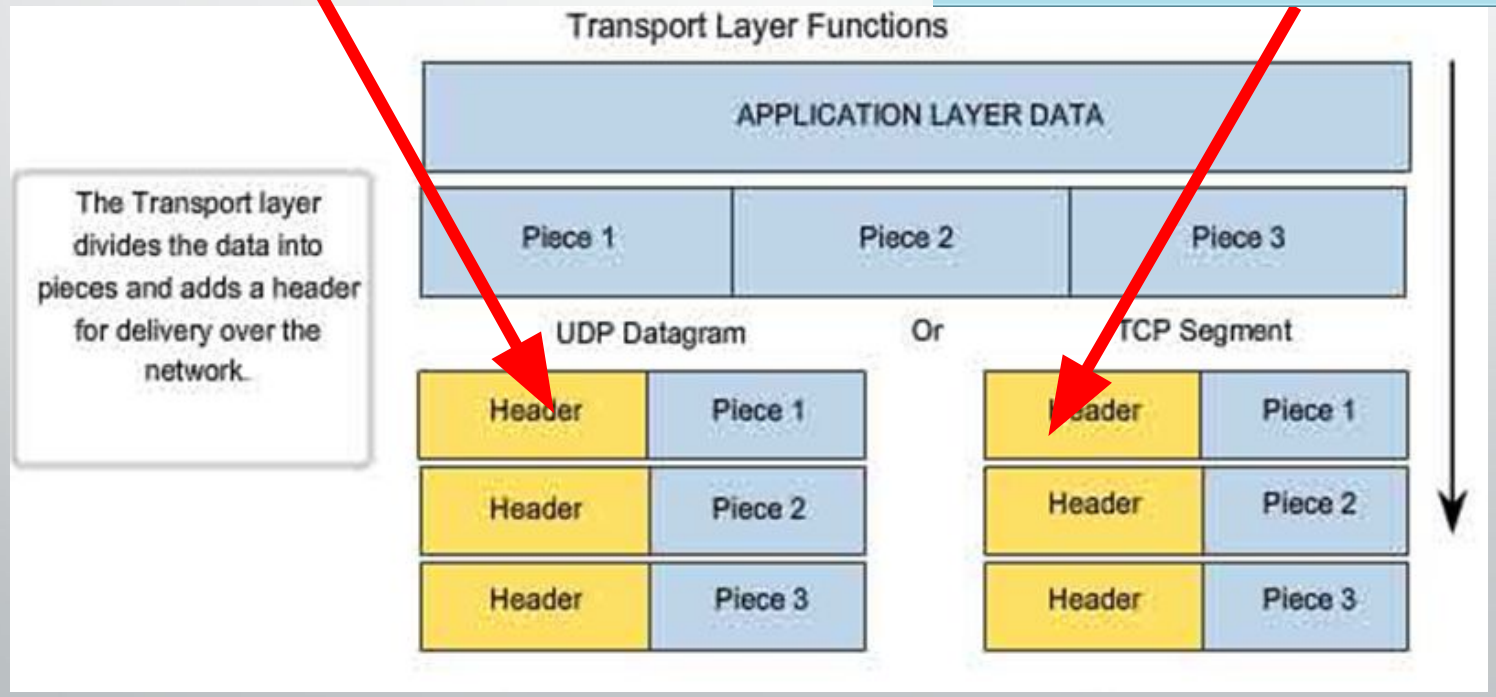
# TCP and UDP Headers

**UDP HEADER**

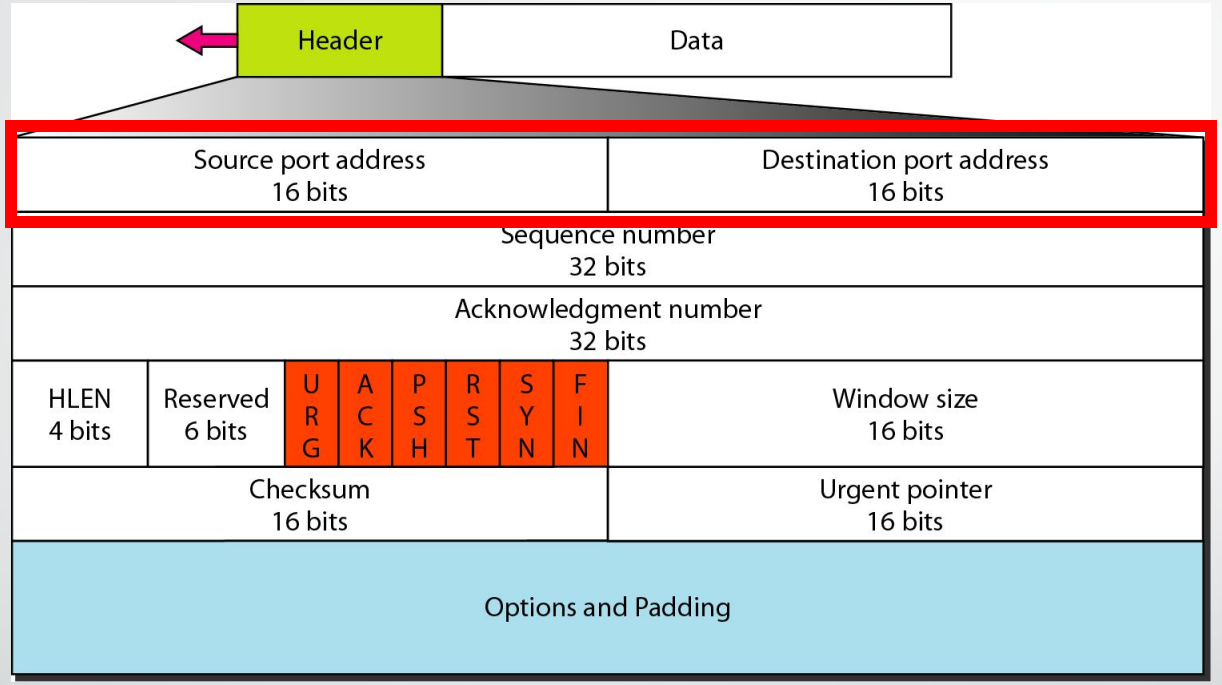
| Source port number<br>16 bits | Destination port number<br>16 bits |
|---|---|
| Total length<br>16 bits | Checksum<br>16 bits |

**TCP HEADER**

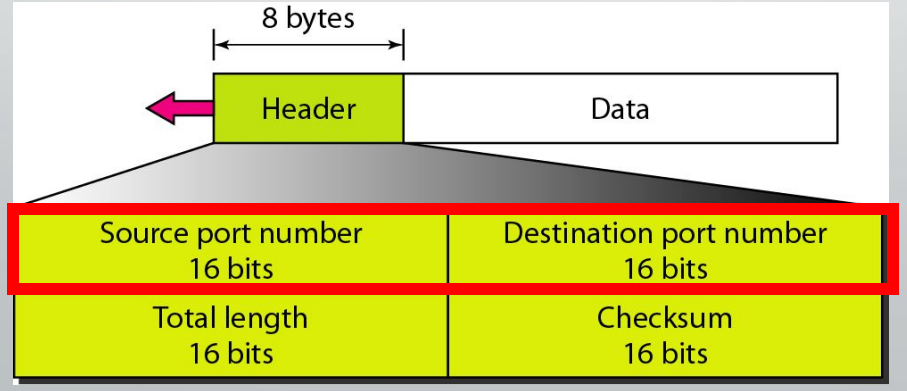
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Source port address<br>16 bits | | | | | | | | Destination port address<br>16 bits |
| Sequence number<br>32 bits | | | | | | | | |
| Acknowledgment number<br>32 bits | | | | | | | | |
| HLEN<br>4 bits | Reserved<br>6 bits | URG | ACK | PSH | RST | SYN | FIN | Window size<br>16 bits |
| Checksum<br>16 bits | | | | | | | | Urgent pointer<br>16 bits |
| Options and Padding | | | | | | | | |



# TCP and UDP Headers



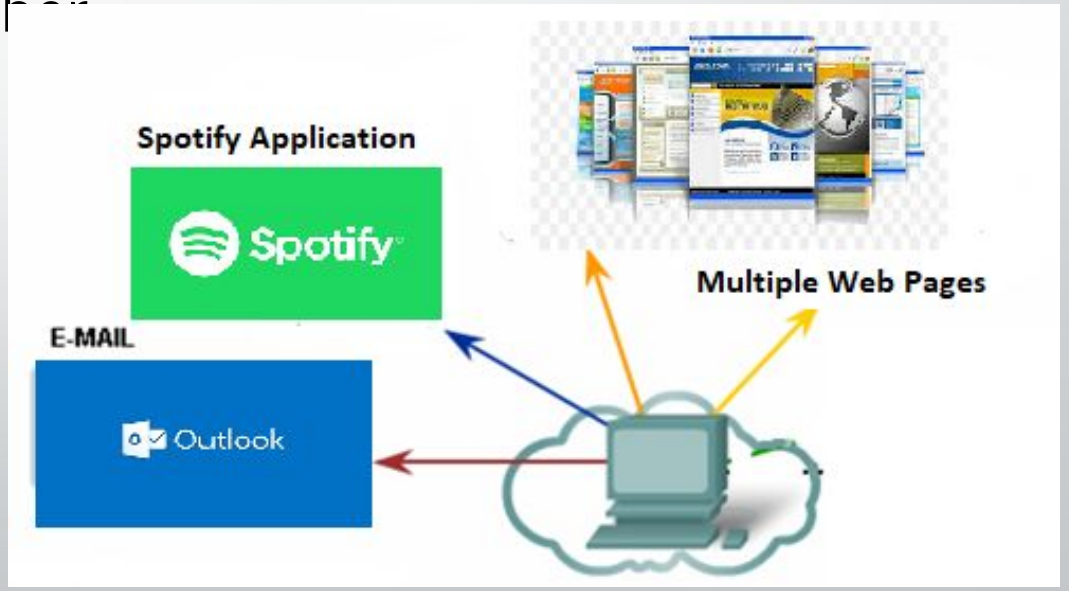

- TCP Header




- UDP Header

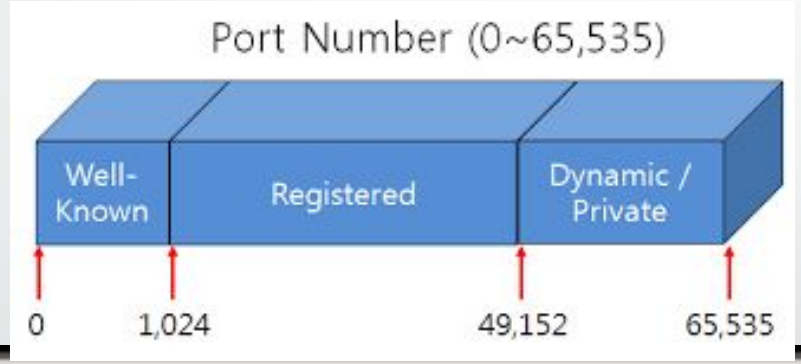
# Function 3 – Identifying Different Applications

- Port Numbers/Addresses are used to identify different applications/processes running in a computer
- 16-bits in length
  - Represented as one single decimal number
  - Range **0 - 65535**
  - e.g. **80 – Web**
  - **25 – SMTP**
  - **4070 – Spotify**



# Port Numbers

- Internet Corporation for Assigned Names and Numbers (ICANN) assigns port numbers
- **Three** categories:




| Port Number Range | Port Group |
|---|---|
| 0 to 1023 | Well Known (Contact) Ports |
| 1024 to 49151 | Registered Ports |
| 49152 to 65535 | Private and/or Dynamic Ports |

# Port Number Types

- **Well-Known Ports:**
  - Assigned and controlled by IANA for standard services
  - Commonly used by system processes and standardized services and applications.

| Port Number Range | Port Group |
|---|---|
| 0 to 1023 | Well Known (Contact) Ports |
| 1024 to 49151 | Registered Ports |
| 49152 to 65535 | Private and/or Dynamic Ports |

67&68 – DHCP

123 – NTP

25 – SMTP

110 – POP3

143 – IMAP

443 – HTTPS

80 – HTTP

53 - DNS

# Port Number Types

- **Registered Ports:**
  - Assigned by IANA but for specific applications requested by developers or organizations.
  - Can be registered for a lot of not-so-well-known, especially corporate/proprietary protocols.

| Port Number Range | Port Group |
|---|---|
| 0 to 1023 | Well Known (Contact) Ports |
| 1024 to 49151 | Registered Ports |
| 49152 to 65535 | Private and/or Dynamic Ports |

8008 – Alternate HTTP

23399 – Skype

8080 – Alternate HTTP

4070 – Spotify

5060 – SIP (VoIP)

3306 – MySQL

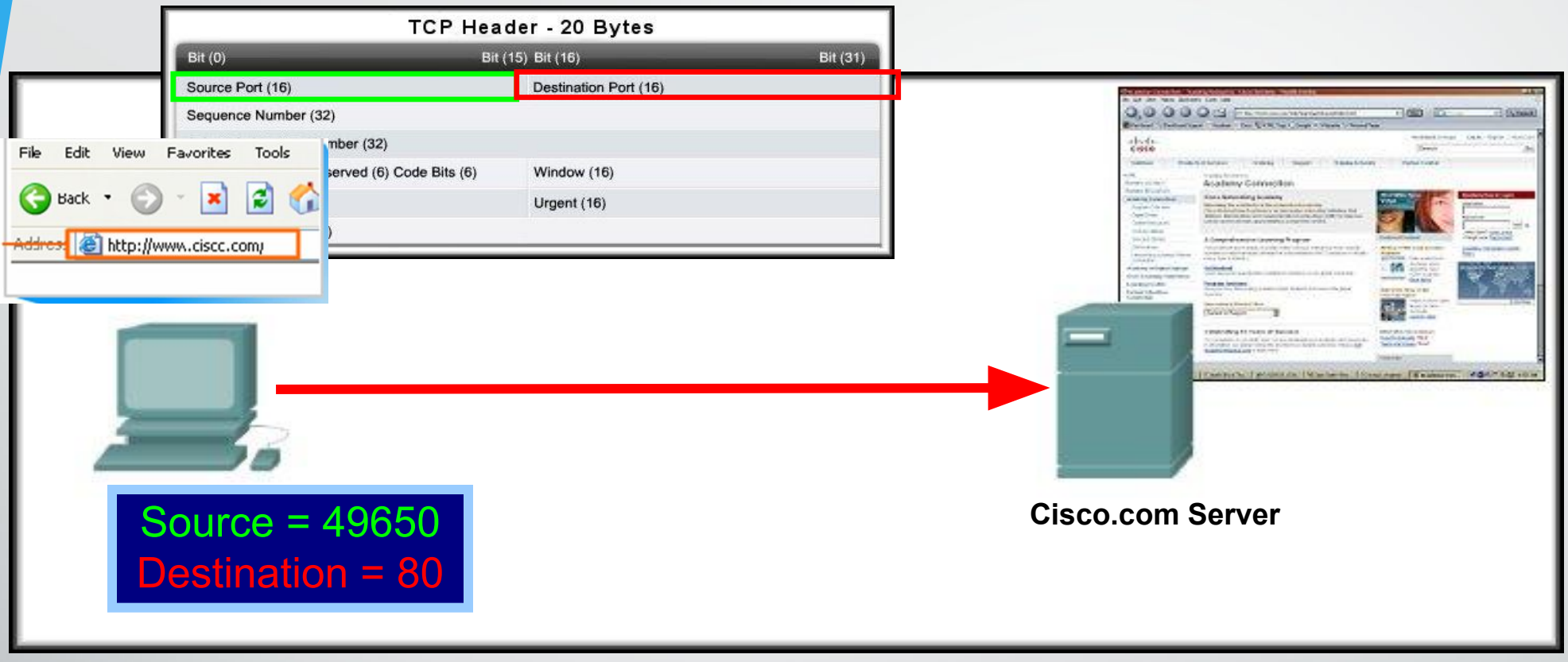
# Port Number Types

- **Dynamic Ports:**
  - Also known as private or ephemeral ports
  - Never assigned or controlled by IANA.

| Port Number Range | Port Group |
|---|---|
| 0 to 1023 | Well Known (Contact) Ports |
| 1024 to 49151 | Registered Ports |
| 49152 to 65535 | Private and/or Dynamic Ports |

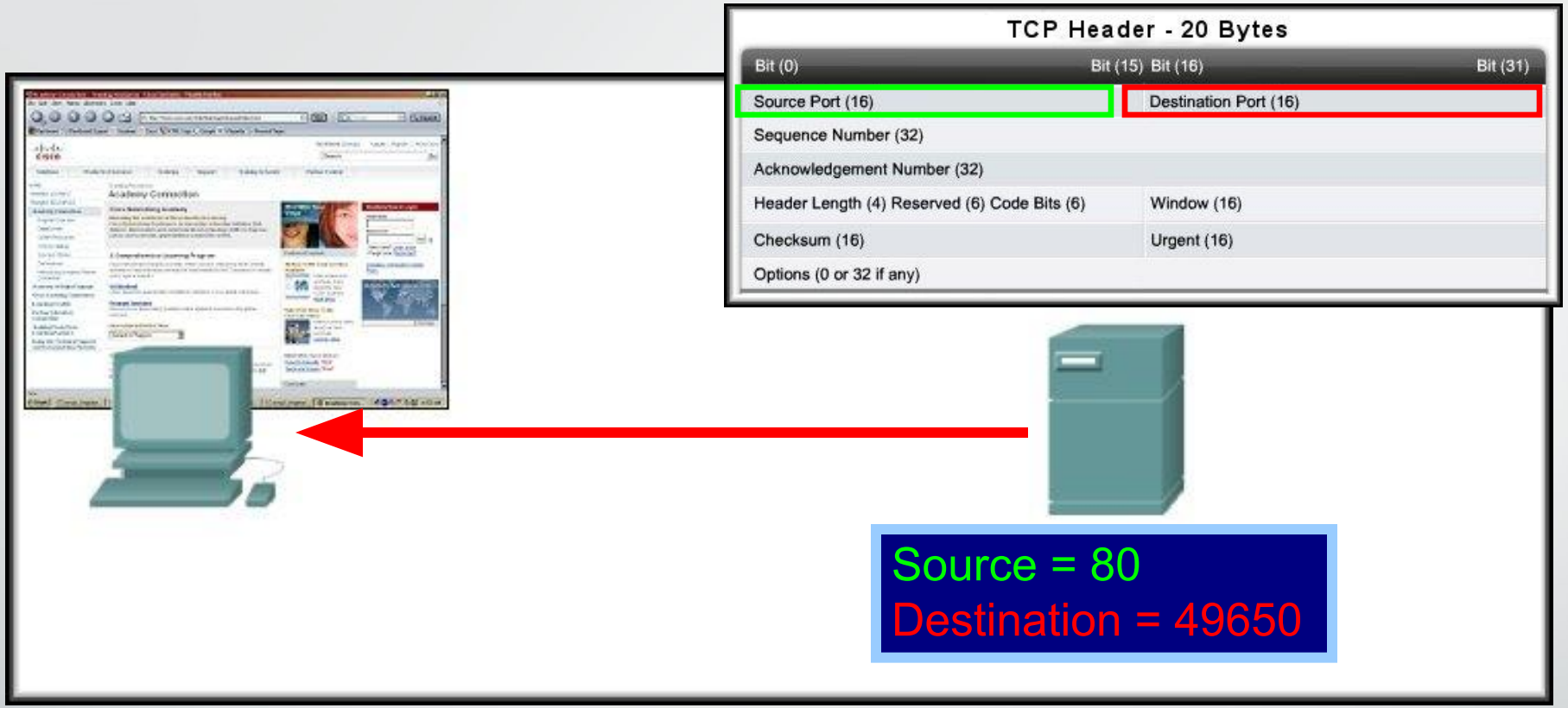
Dynamic port usage will become clearer as we move through the material.

# More on Port Numbers



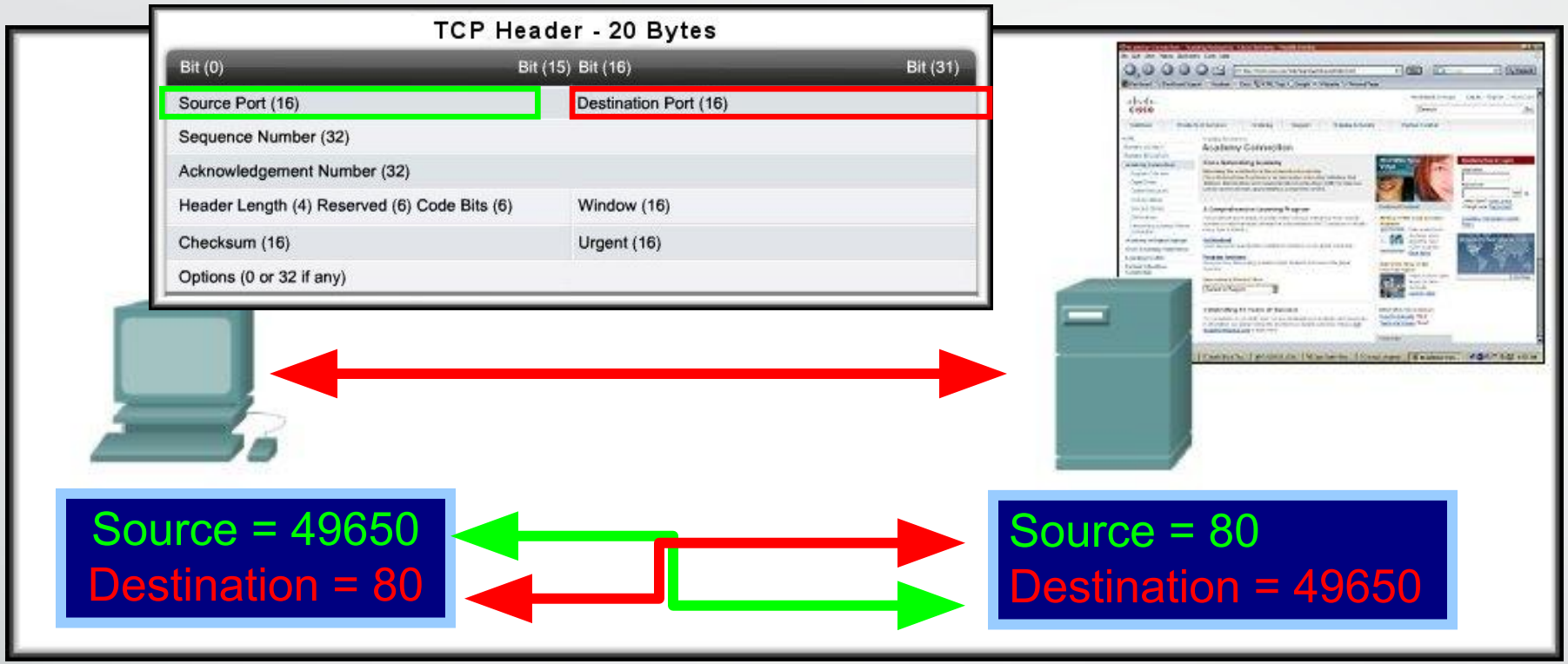

- Server is listening on Port 80 for HTTP connections.
- The client sets the destination port to 80 and uses a dynamic port as its source.

# Port Numbers in Action



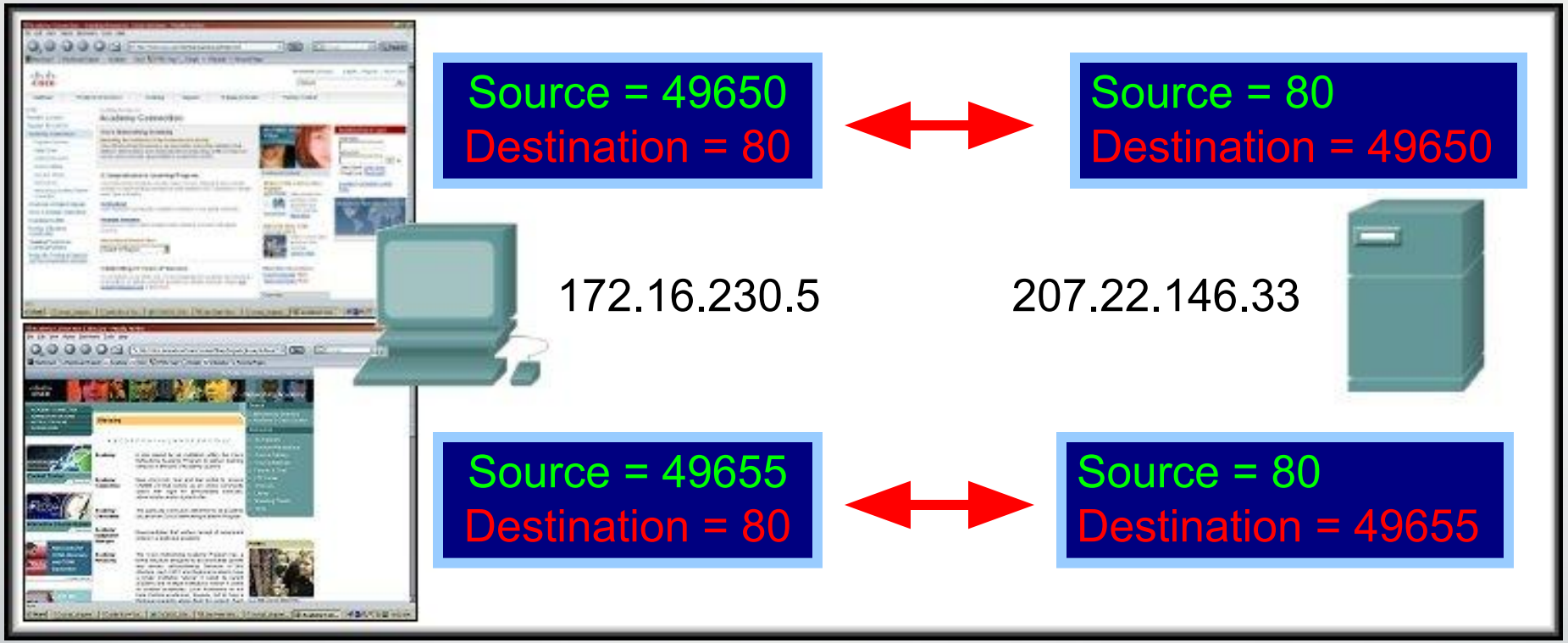

- Server replies with the web page.
  - Sets the source port to 80 and uses the client's source port as the destination.

# Port Numbers in Action



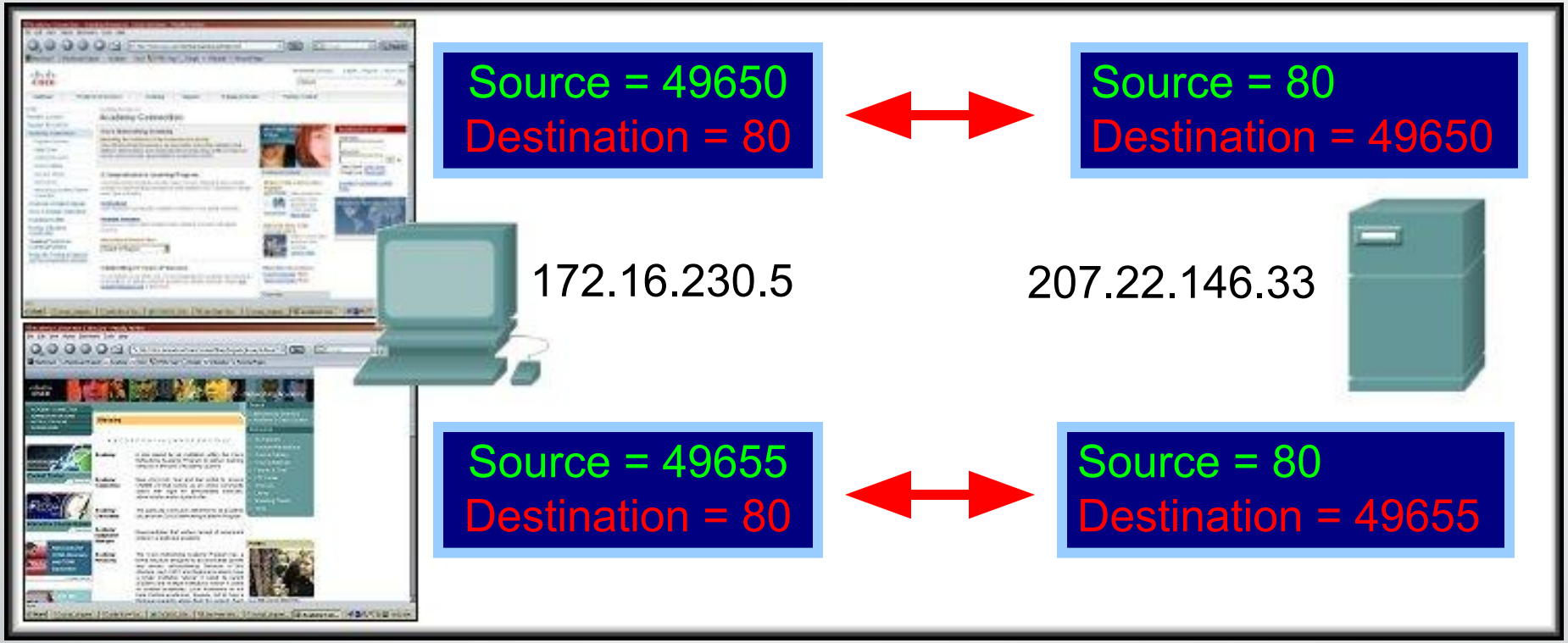

- Clients can use any random port number, Servers can't.
  - Because clients won't be able to identify server process otherwise
  - Servers thus must use **well-known port numbers**!

# Port Numbers in Action



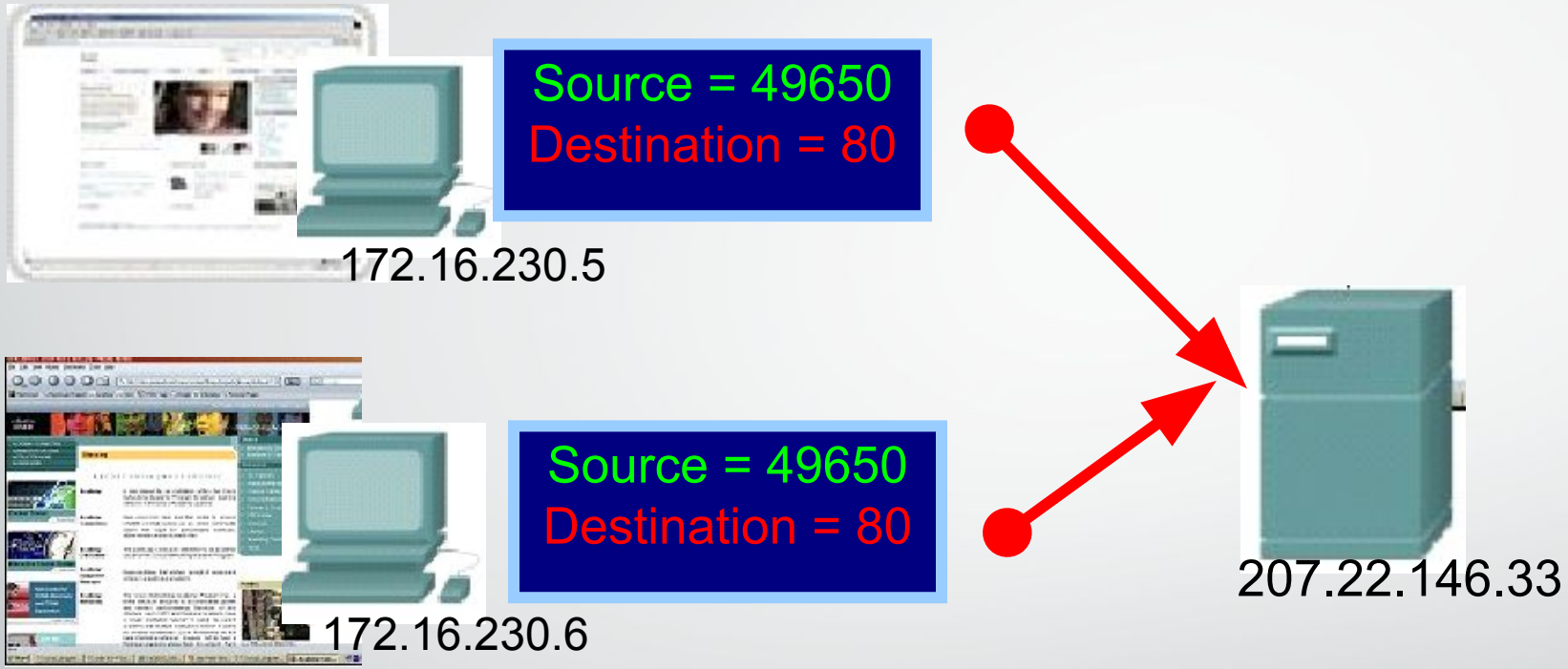

- What if there are two sessions to the same server?
  - The client uses another dynamic port as its source and the destination is still port 80.
  - Different source ports keep the sessions unique on the server.

# Port Numbers in Action




- There are two tabs in the same PC, then?
  - The client uses another dynamic port as its source and the destination is still port 80.
  - Different source ports keep the sessions unique.

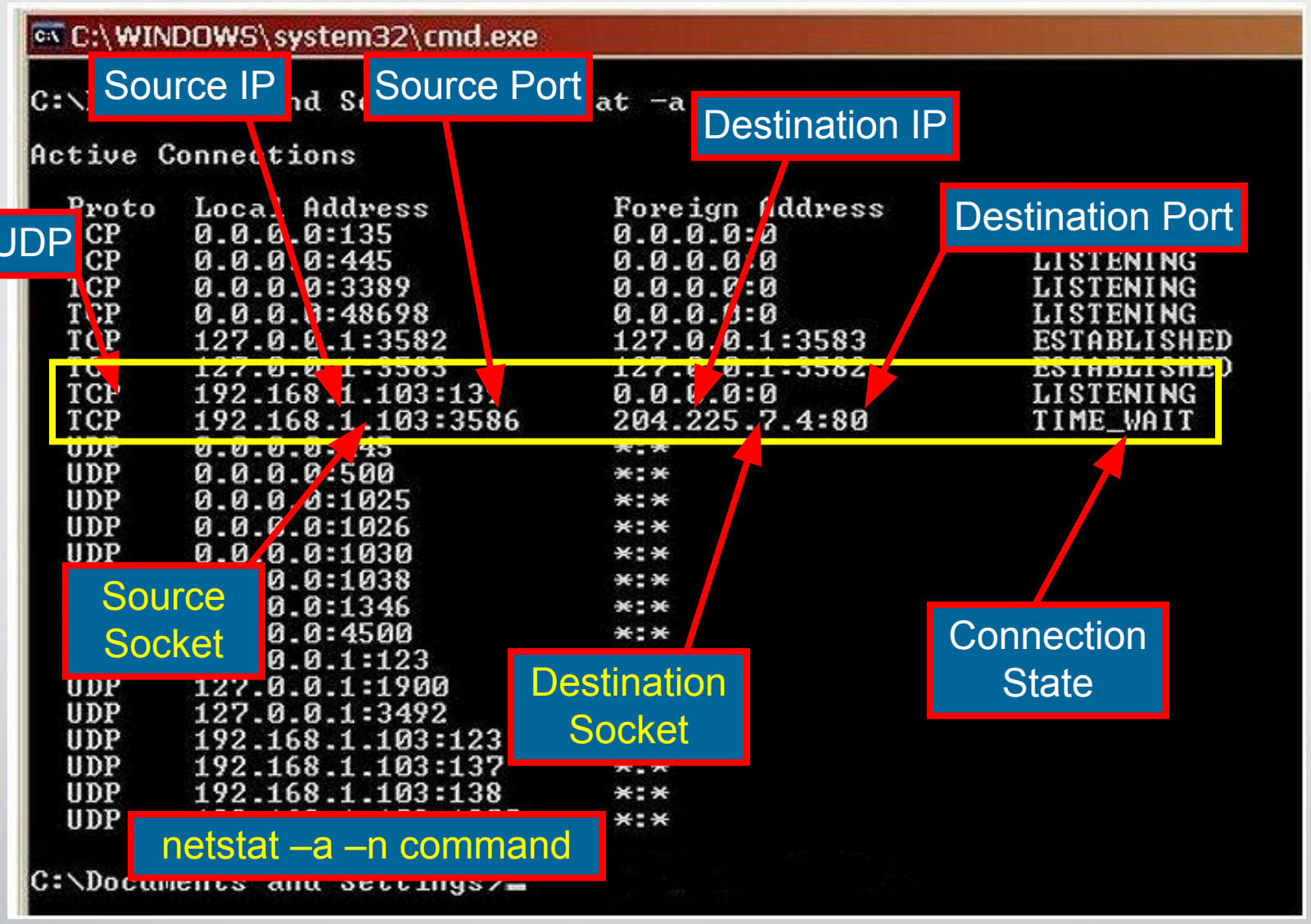
# More on Port Numbers in Action




- How does the Server's Transport Layer keep them separate?
  - The socket (IP Address:Port)

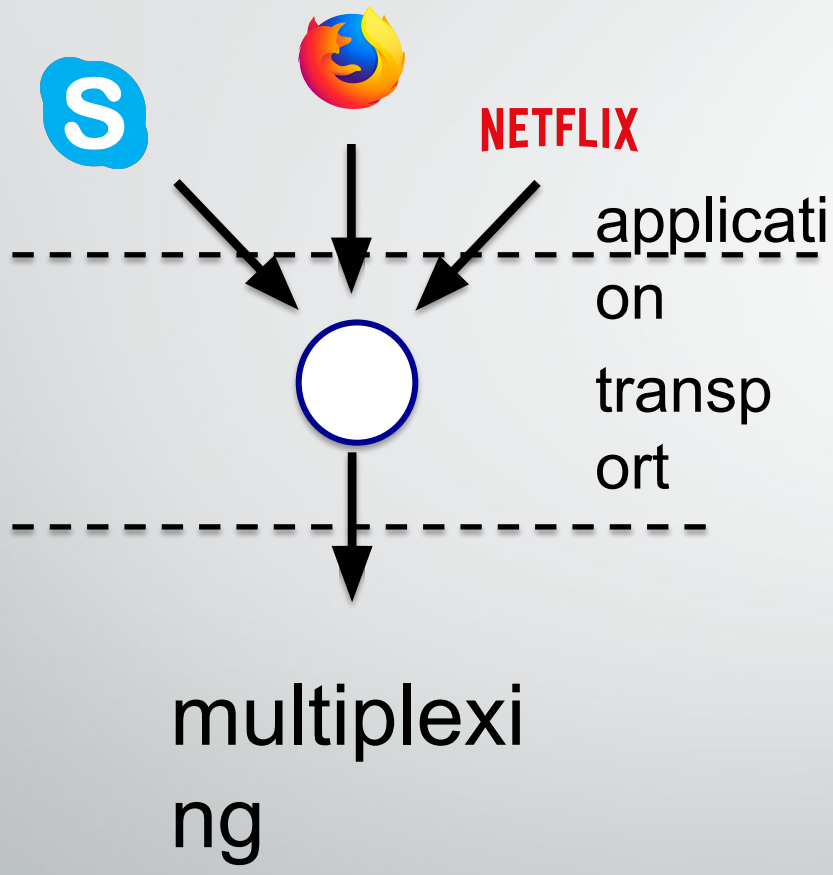
**172.16.230.5:49650** ⟷ **207.22.146.33:80**

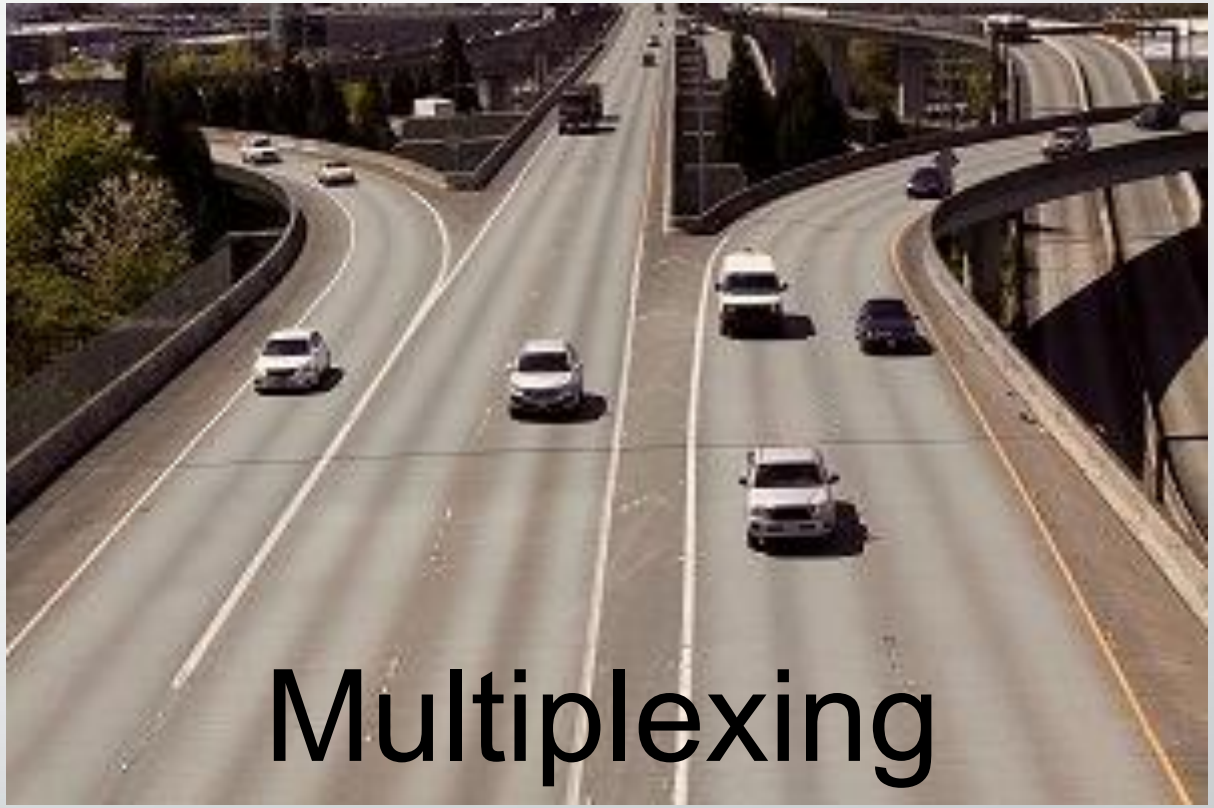
**172.16.230.6:49650** ⟷ **207.22.146.33:80**



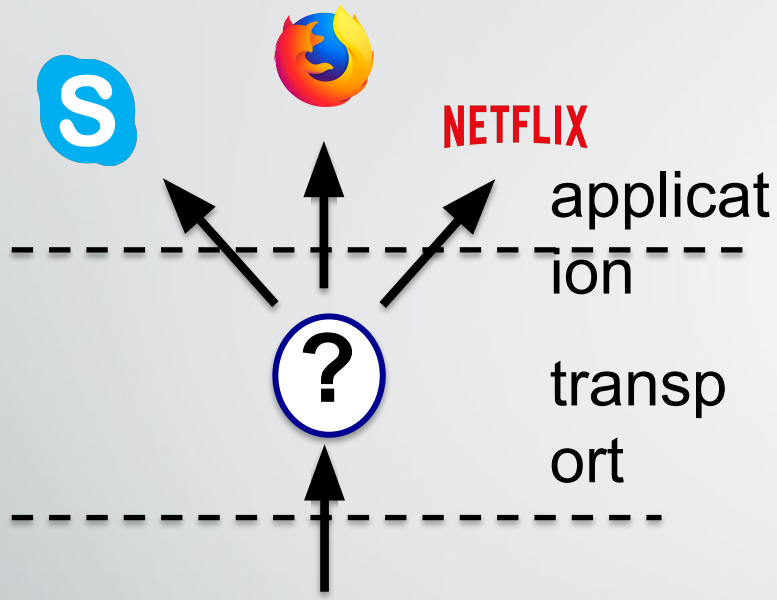

# Netstat - Network Utility Tool

# Function 4 –Multiplexing



# Function 4 –DeMultiplexing




Demultiplexing


Demultiplexing

# DeMultiplexing/ Multiplexing

- Multiplexing, demultiplexing: based on segment, datagram header field values
- **UDP:** demultiplexing using **destination port number** (only)
- **TCP:** demultiplexing using 4-tuple: **source and destination IP addresses, and port numbers**

# And Now more on TCP!