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 sion.
6. Performing flow control between end users.
7. Enabling error recovery.

Reliability

Different Applications

Different Requirements

Web Applications

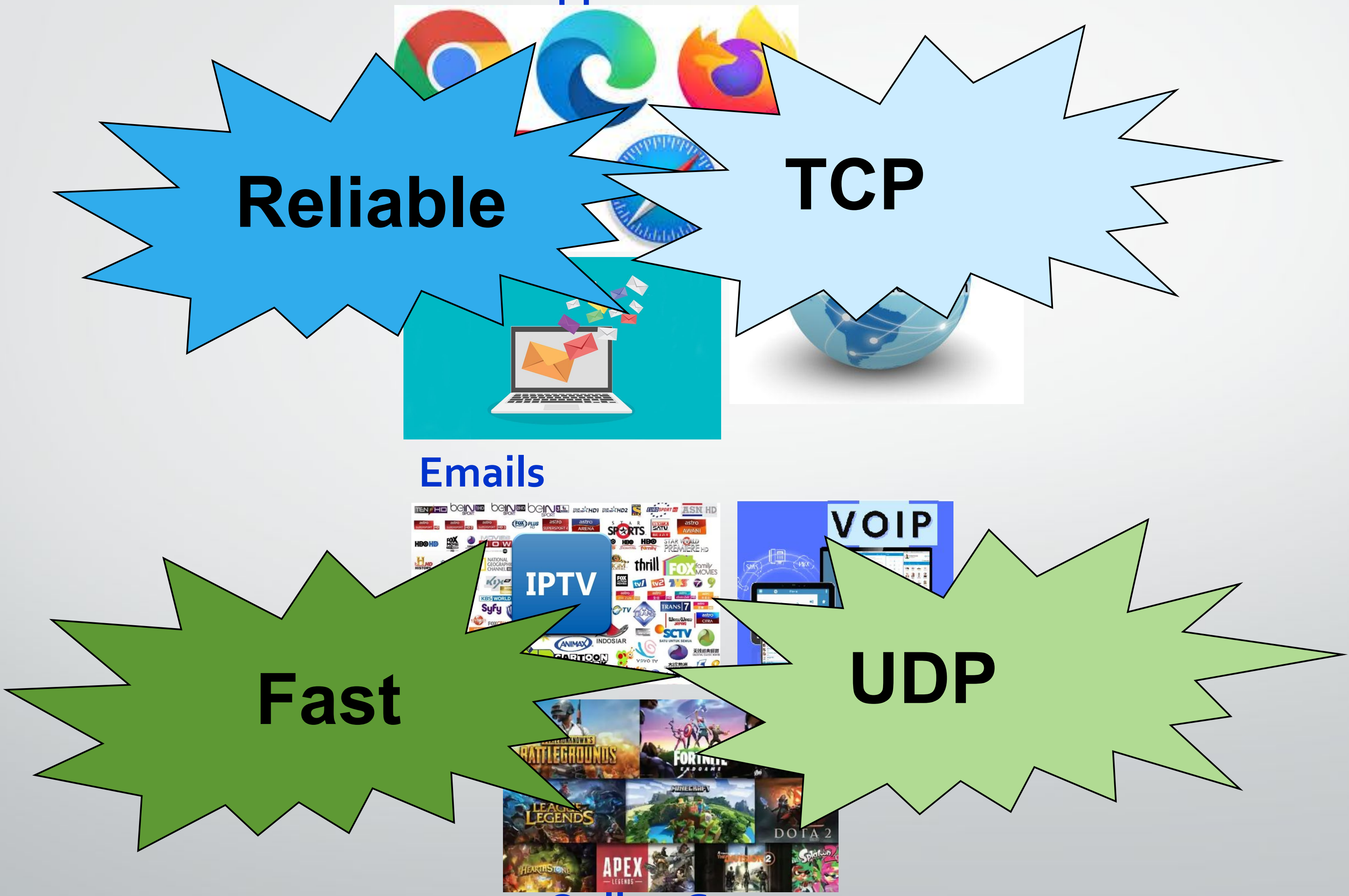
Online Games

- 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.

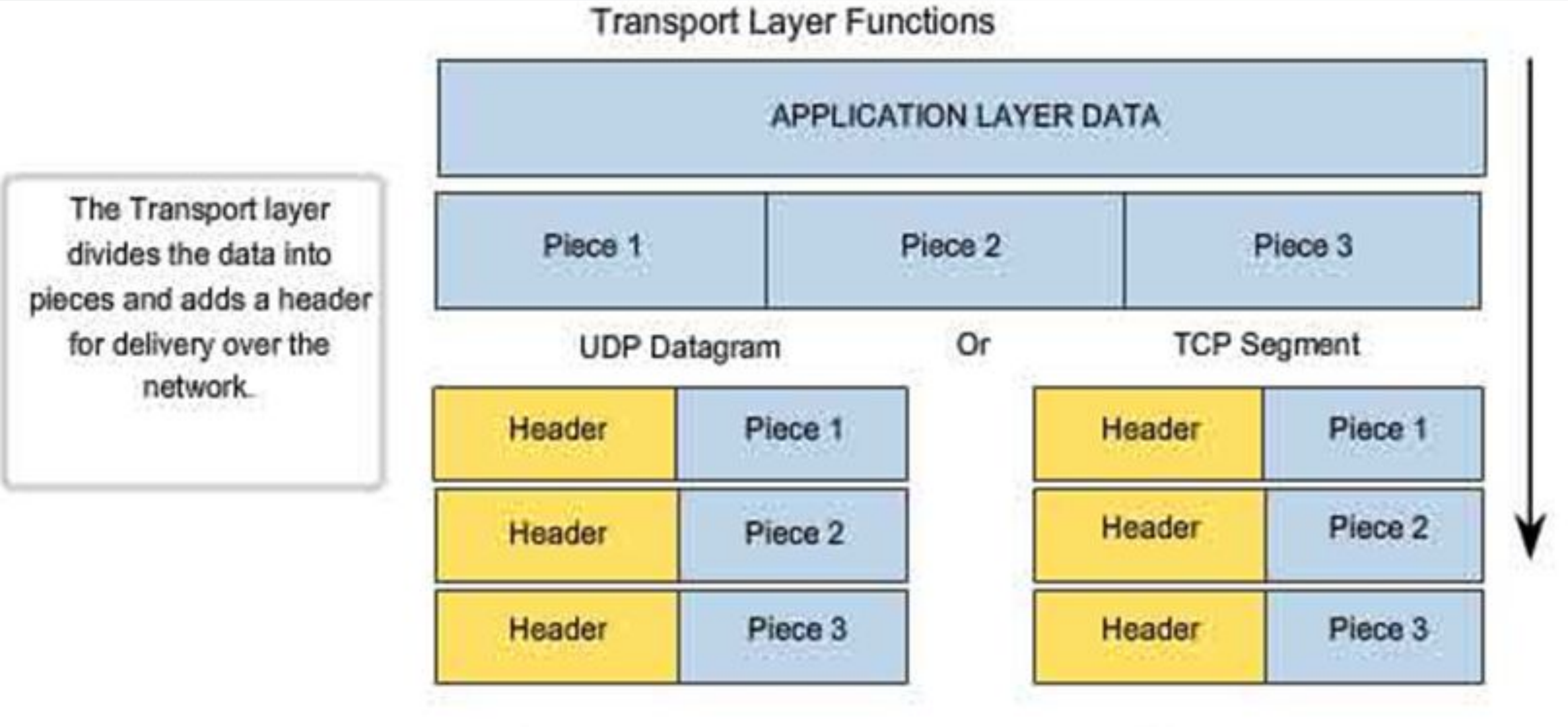
UDP &TCP

2. Reassembling the segments into streams of application data.
3. Identifying the different applications.
4. Multiplexing

Only TCP

5. Establishing and terminating a connection
6. Enabling error recovery.
7. Performing flow control between end users.

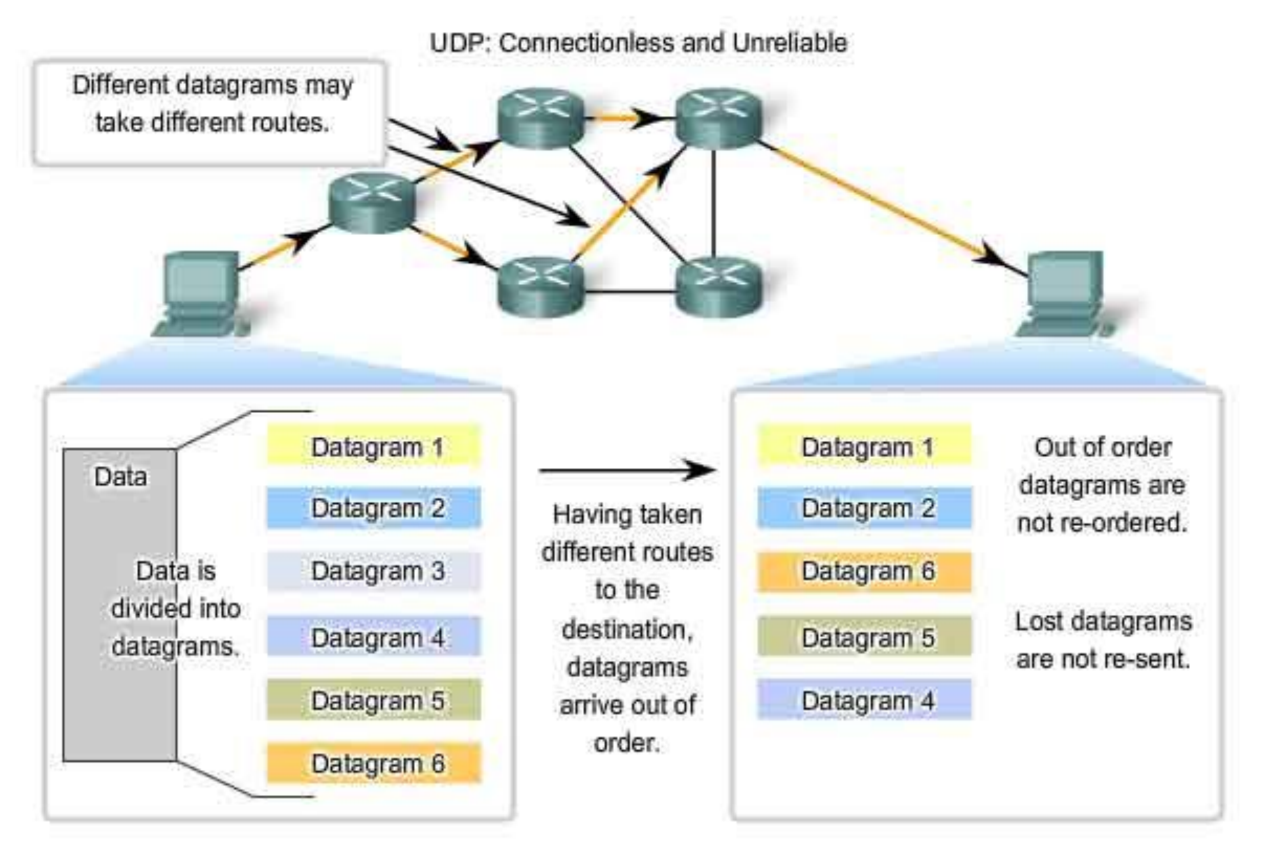
Function 1&2 – Segmentation and Reassembly

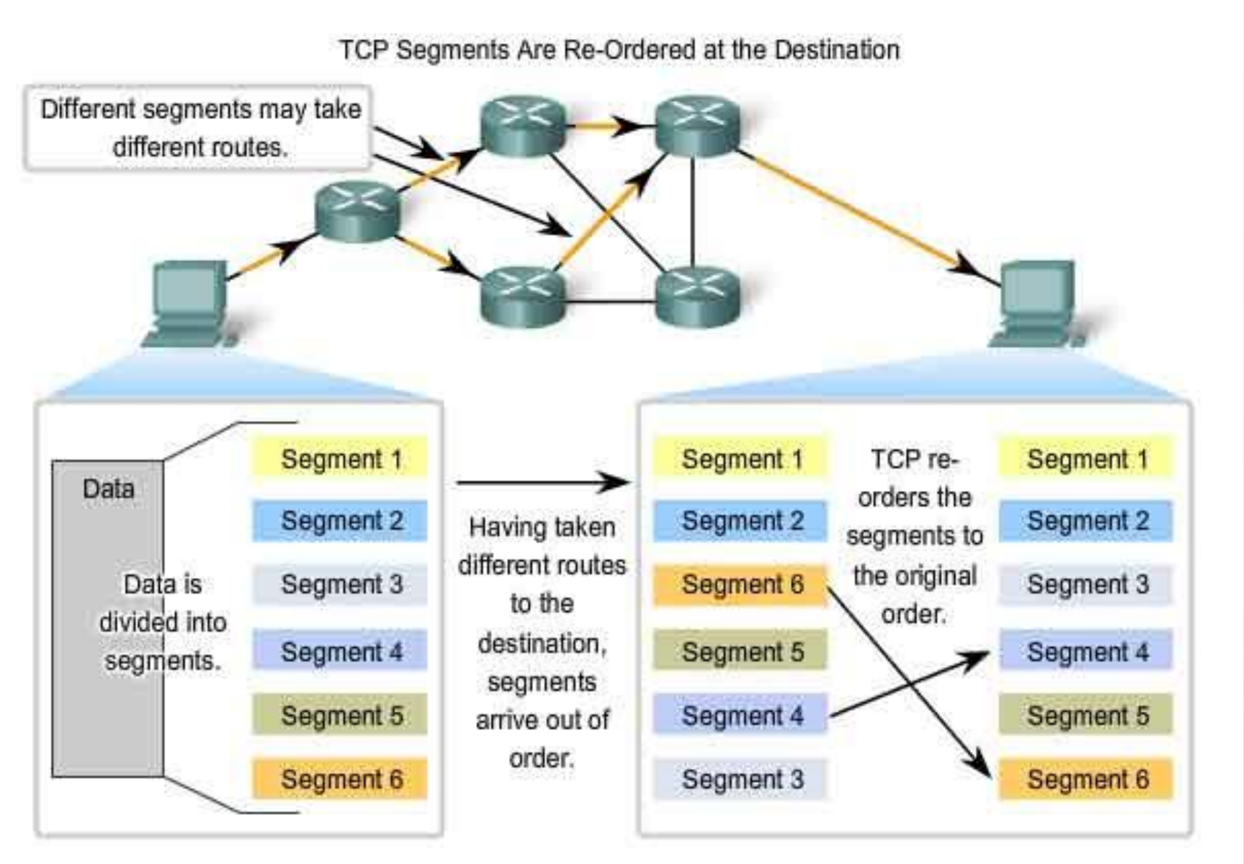


Transport Layer Functions

The Transport layer divides the data into pieces and adds a header for delivery over the network.

Function 2 – Reassembly


UDP: Connectionless and Unreliable
Different datagrams may take different routes.
Data
Data is divided into datagrams.
Datagram 1
Datagram 2
Datagram 3
Datagram 4
Datagram 5
Datagram 6
Having taken different routes to the destination, datagrams arrive out of order.
Datagram 1
Datagram 2
Datagram 6
Datagram 5
Datagram 4
Out of order datagrams are not re-ordered.
Lost datagrams are not re-sent.

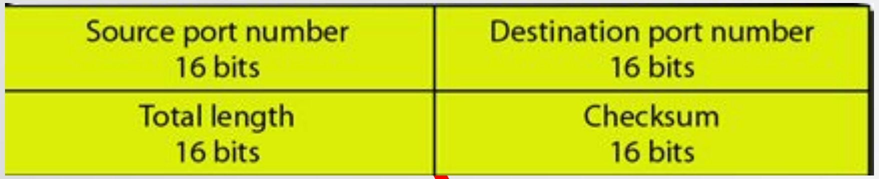

TCP Segments Are Re-Ordered at the Destination
Different segments may take different routes.
Data
Data is divided into segments.
Segment 1
Segment 2
Segment 3
Segment 4
Segment 5
Segment 6
Having taken different routes to the destination, segments arrive out of order.
Segment 1
Segment 2
Segment 6
Segment 5
Segment 4
Segment 3
TCP re-orders the segments to the original order.
Segment 1
Segment 2
Segment 3
Segment 4
Segment 5
Segment 6

UDP

TCP

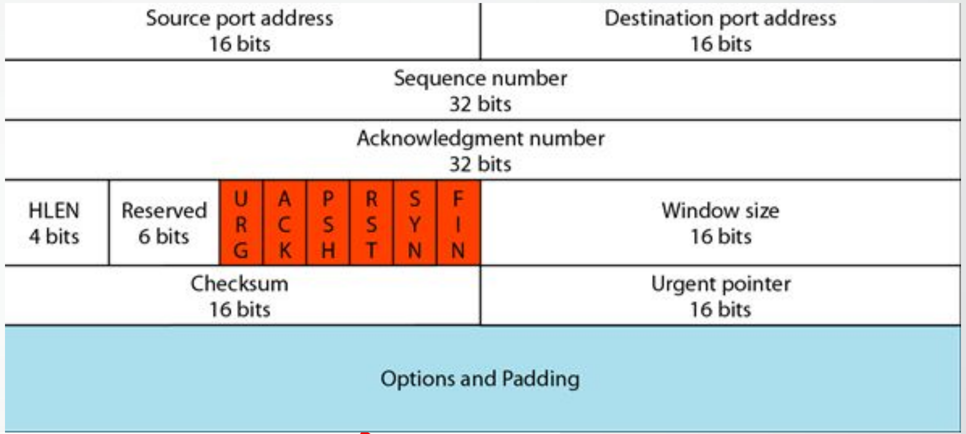
TCP and UDP Headers

UDP HEADER

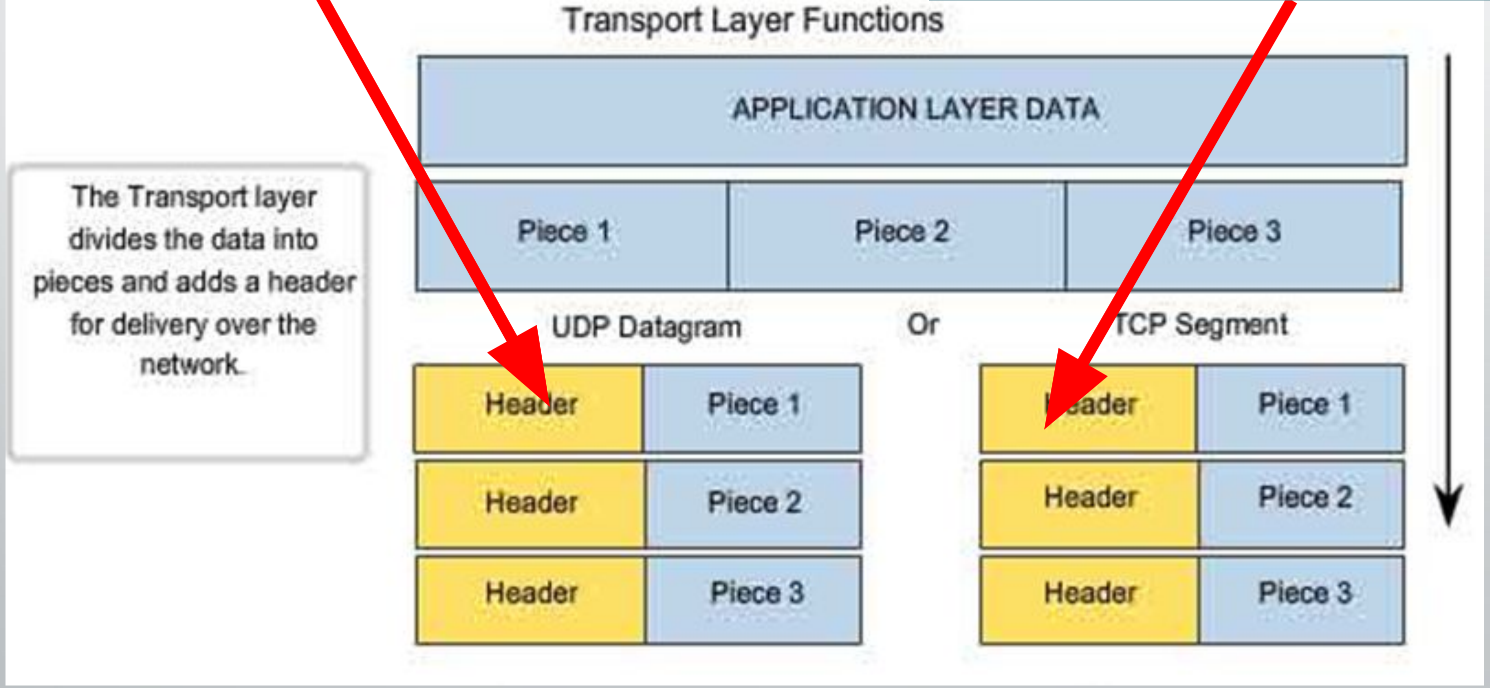


Source port number 16 bits	Destination port number 16 bits
Total length 16 bits	Checksum 16 bits

TCP HEADER

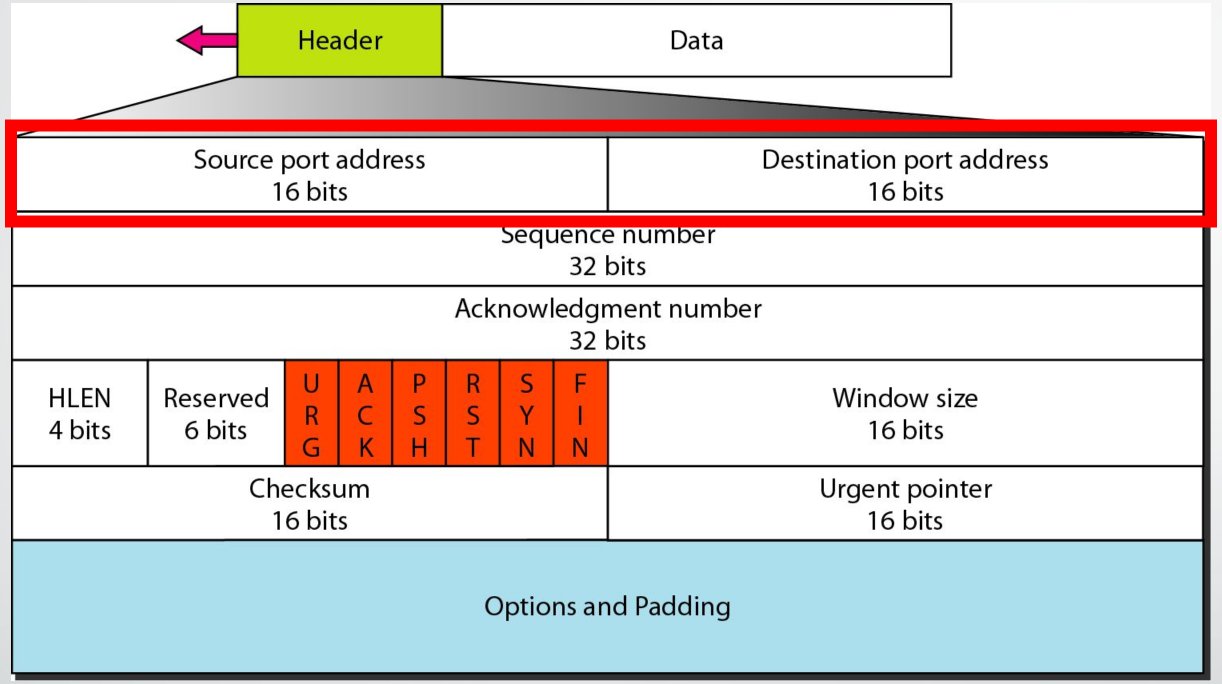


<table>
<tr><td colspan="8">Source port address
16 bits</td><td>Destination port address
16 bits</td></tr>
<tr><td colspan="9">Sequence number
32 bits</td></tr>
<tr><td colspan="9">Acknowledgment number
32 bits</td></tr>
<tr><td>HLEN
4 bits</td><td>Reserved
6 bits</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td>Window size
16 bits</td></tr>
<tr><td colspan="8">Checksum
16 bits</td><td>Urgent pointer
16 bits</td></tr>
<tr><td colspan="9">Options and Padding</td></tr>
</table>

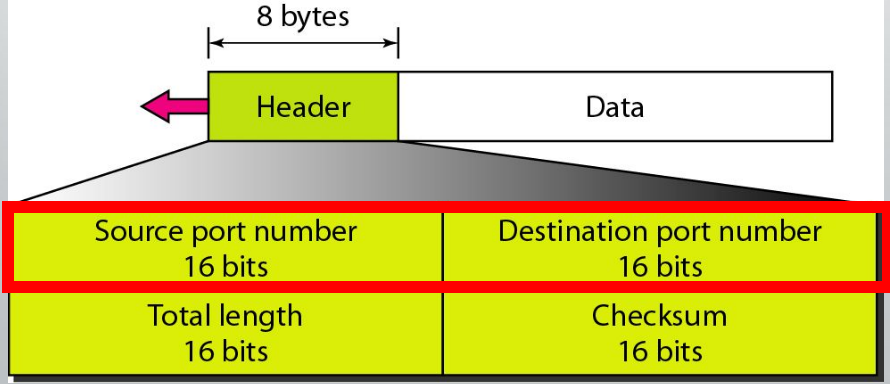


The Transport layer divides the data into pieces and adds a header for delivery over the network.

TCP and UDP Headers



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



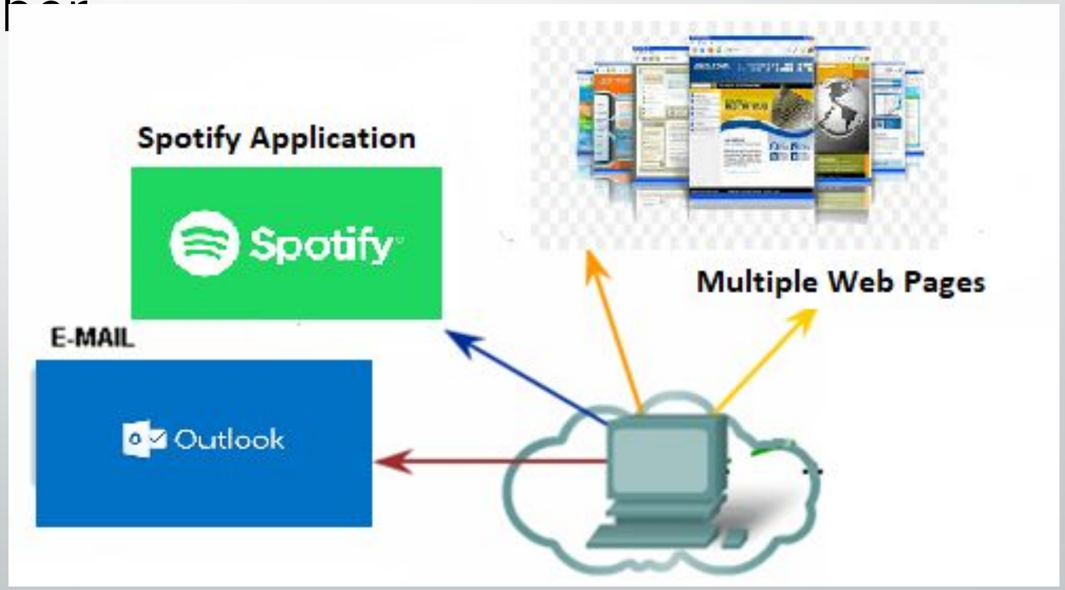
Source port number 16 bits	Destination port number 16 bits
Total length 16 bits	Checksum 16 bits

- TCP Header

- UDP Header

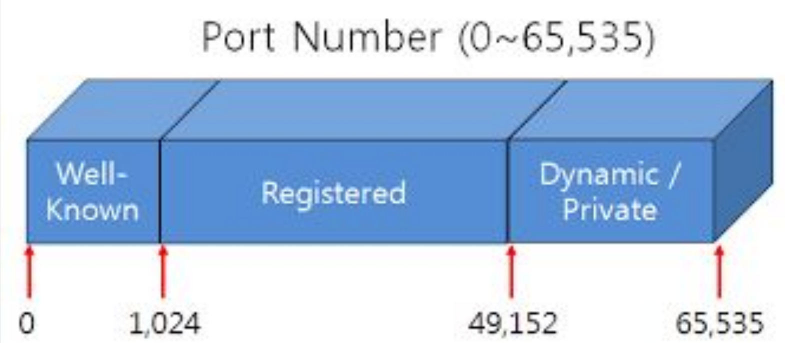
Function 3 – Identifying Different Applications

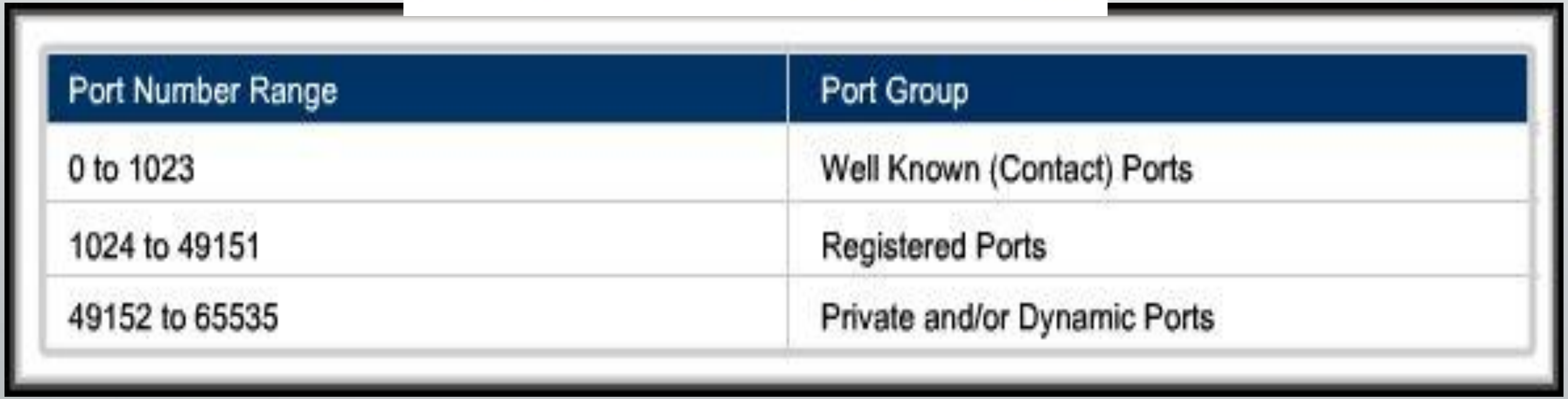
- Port Numbers/Addresses are used to identify different applications/processes running in a computer
- 16-bits in length
  - Represented as one single decimal numb
  - Range **0 - 65535**
  - e.g. **80 – Web**
  - **25 – SMTP**
  - **4070 – Spotify**



Port Numbers

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

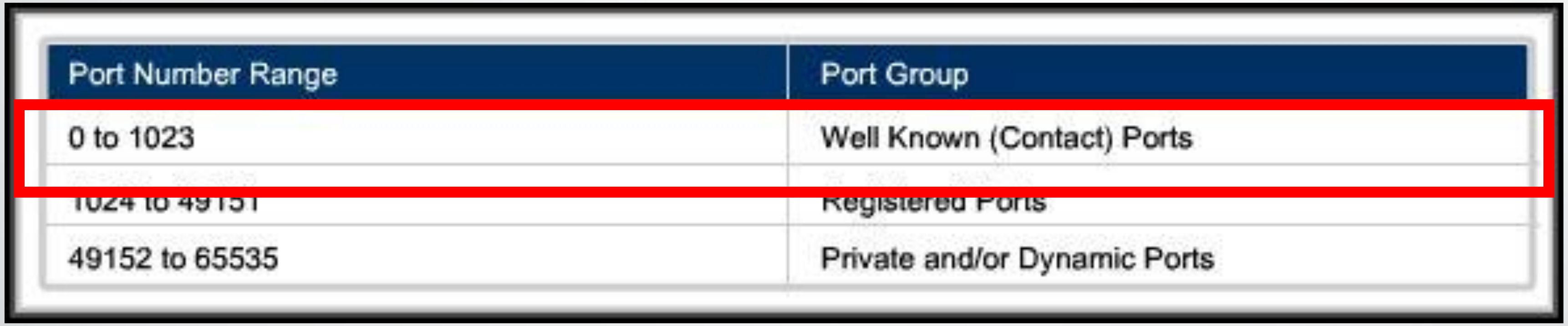




Port Number Range	Port Group
0 to 1023	Well Known (Contact) Ports
1024 to 49151	Registered Ports
49152 to 65535	Private and/or Dynamic Ports

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

443 – HTTPS

25 – SMTP

123 – NTP

80 – HTTP

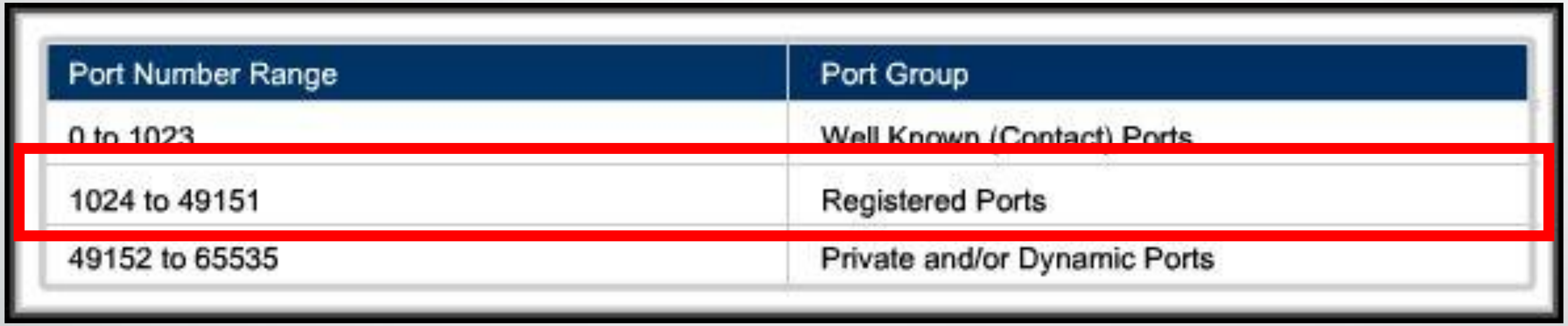
110 – POP3

143 – IMAP

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

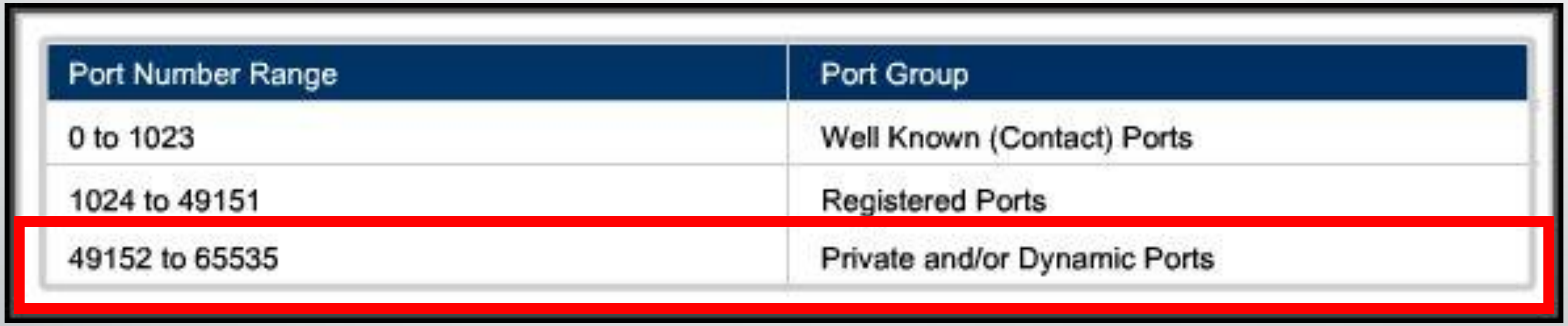
4070 – Spotify

5060 – SIP (VoIP)

3306 – MySQL

Port Number Types

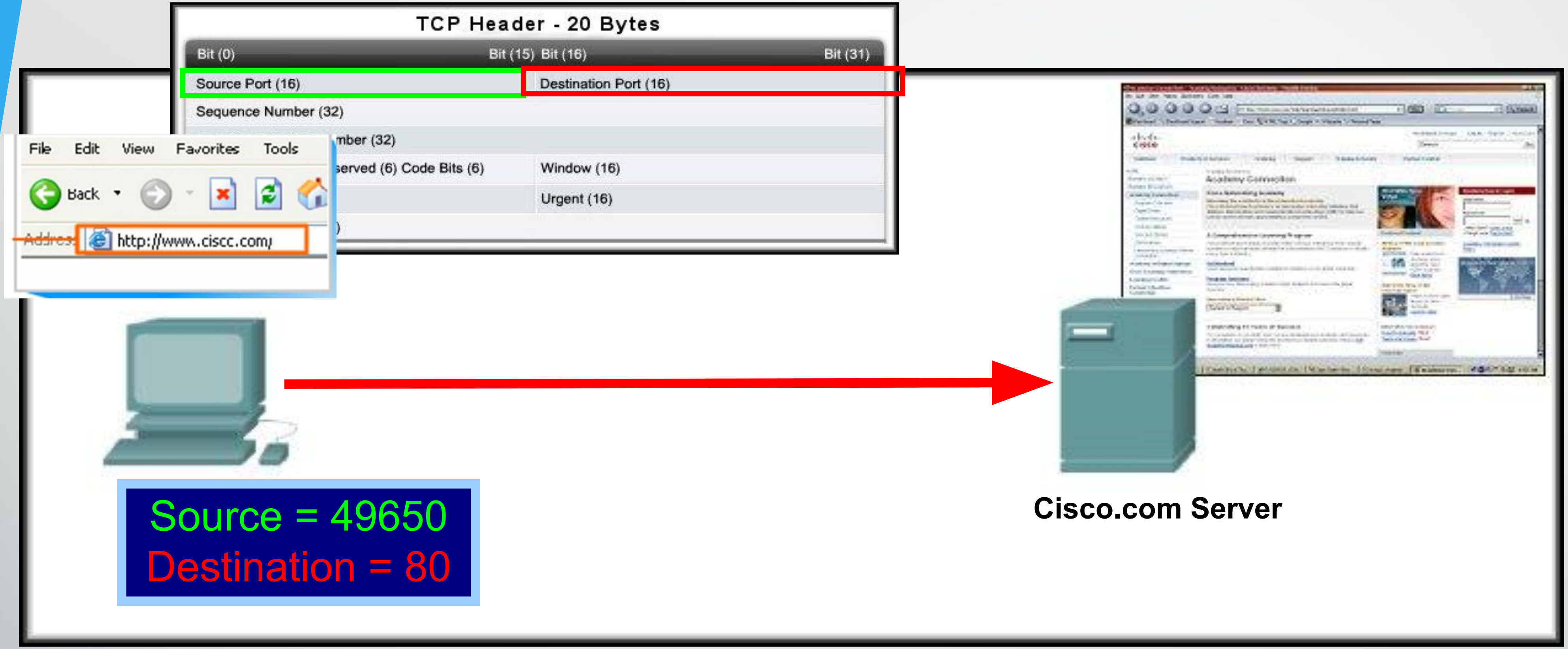
- **Dynamic Ports:**
  - Also known as private or ephemeral ports
  - Never assigned or controlled by IANA.



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



TCP Header - 20 Bytes

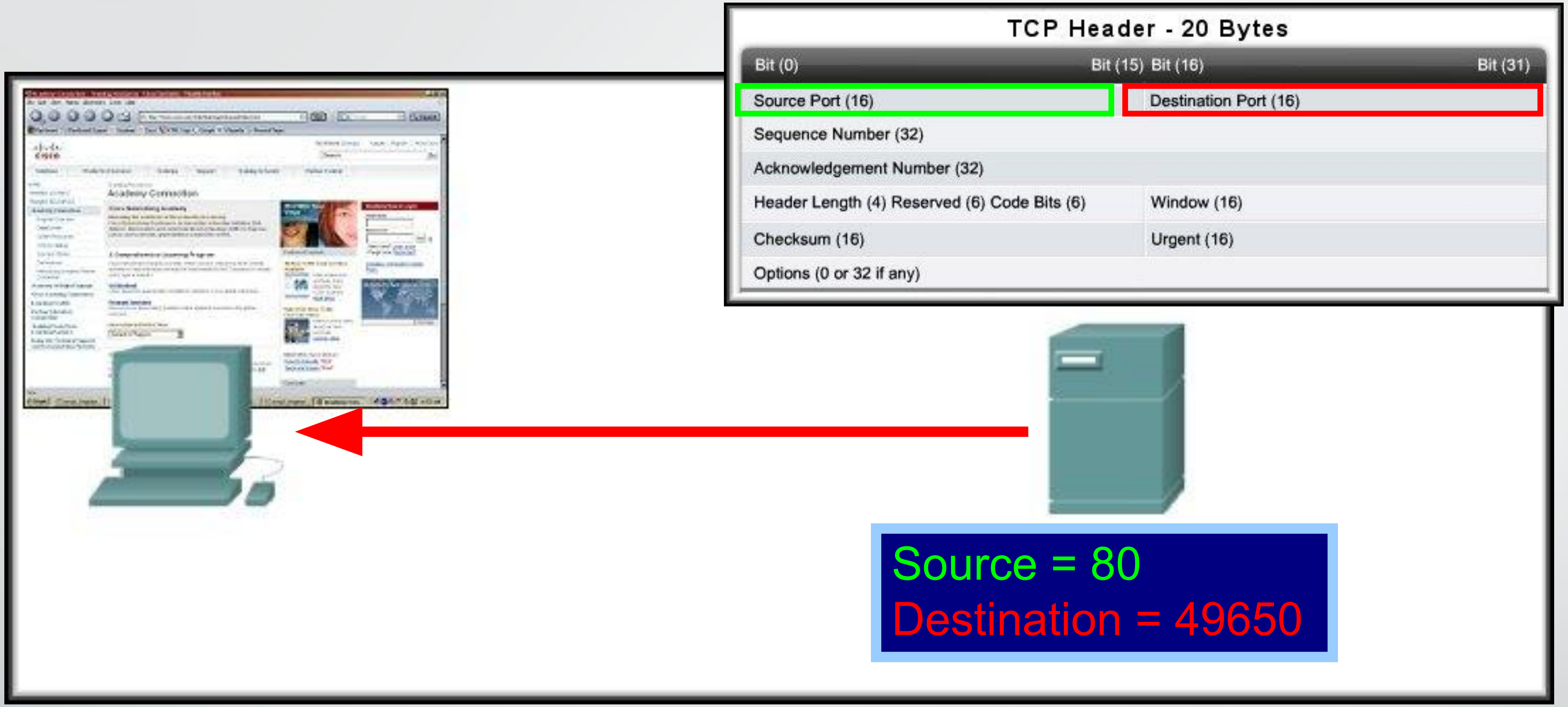
Bit (0)	Bit (15) Bit (16)	Bit (31)
Source Port (16)	Destination Port (16)	
Sequence Number (32)		
mber (32)		
served (6) Code Bits (6)	Window (16)	
	Urgent (16)	

Source = 49650
Destination = 80

Cisco.com Server

- 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



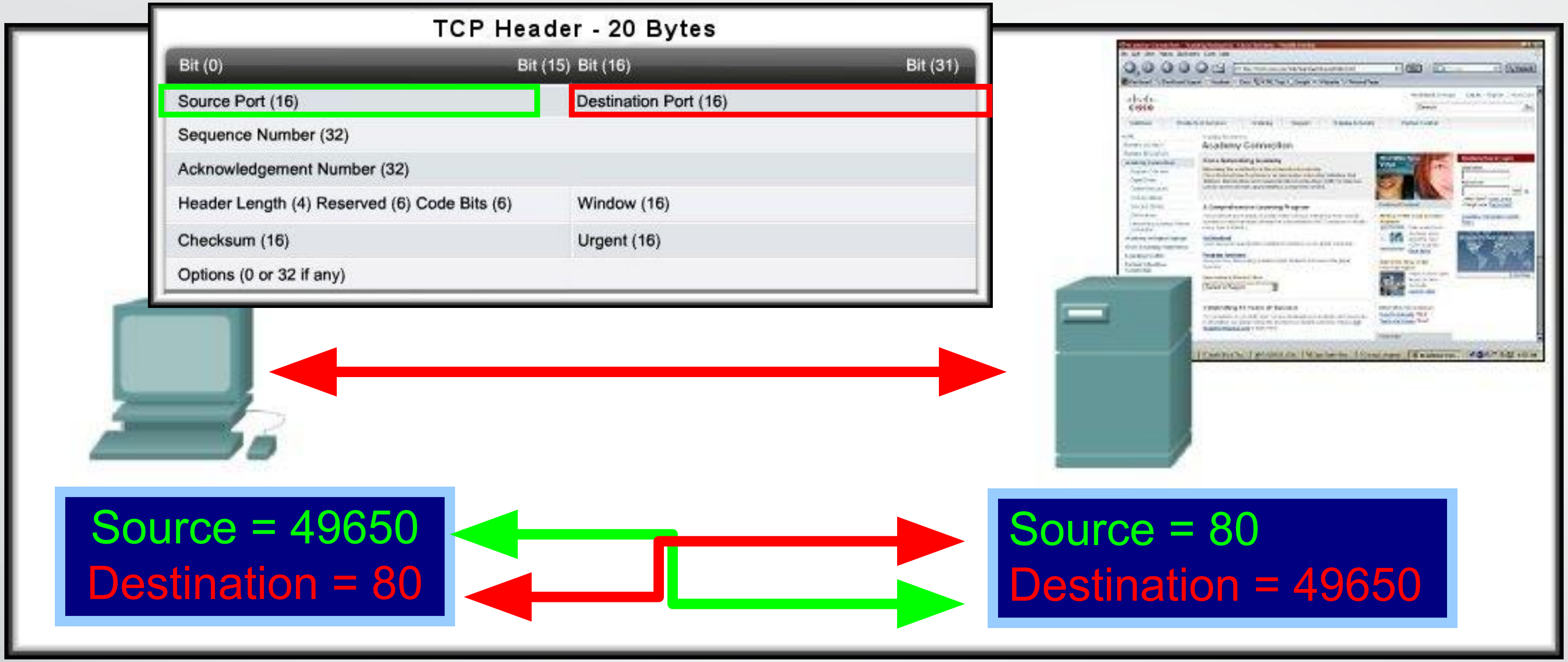
TCP Header - 20 Bytes

Bit (0) Bit (15)	Bit (16) Bit (31)
Source Port (16)	Destination Port (16)
Sequence Number (32)	
Acknowledgement Number (32)	
Header Length (4) Reserved (6) Code Bits (6)	Window (16)
Checksum (16)	Urgent (16)
Options (0 or 32 if any)	

Source = 80
Destination = 49650

- 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



TCP Header - 20 Bytes

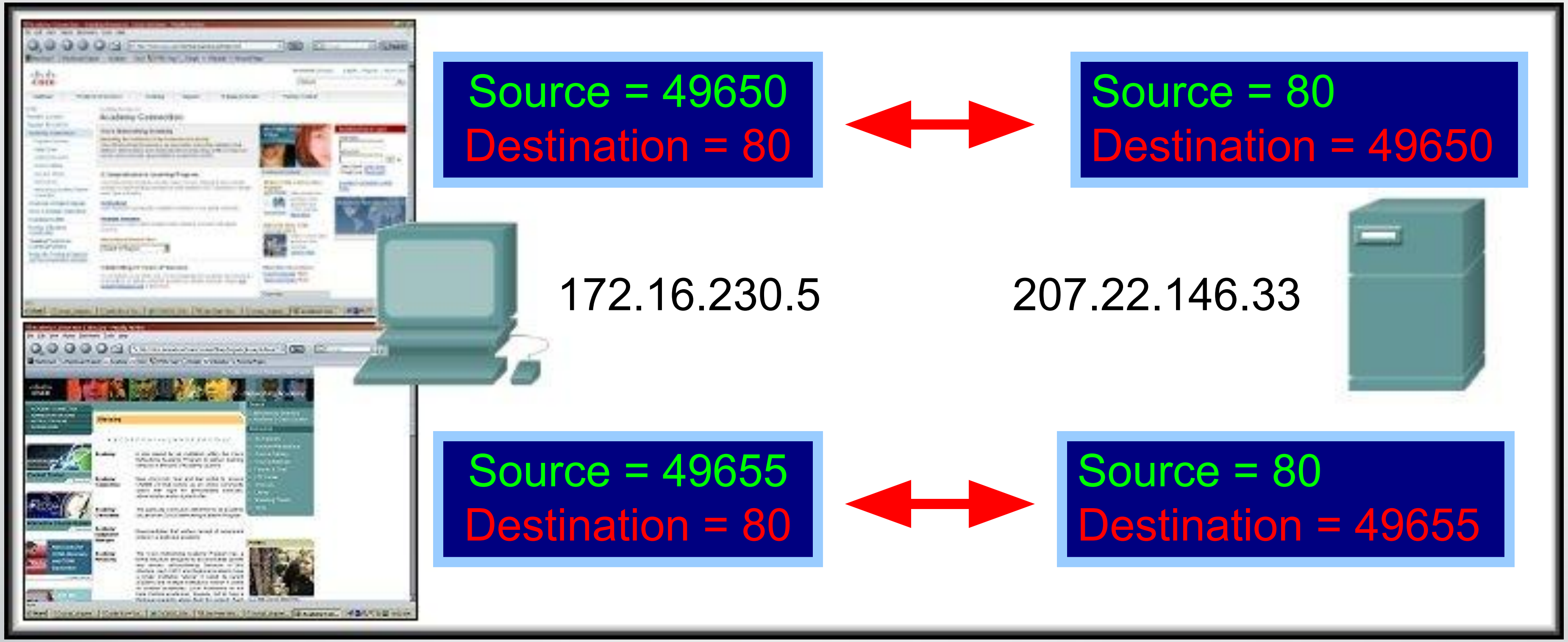
Bit (0) Bit (15)	Bit (16) Bit (31)
Source Port (16)	Destination Port (16)
Sequence Number (32)	
Acknowledgement Number (32)	
Header Length (4) Reserved (6) Code Bits (6)	Window (16)
Checksum (16)	Urgent (16)
Options (0 or 32 if any)	

Source = 49650
Destination = 80

Source = 80
Destination = 49650

- 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



Source = 49650
Destination = 80

Source = 80
Destination = 49650

172.16.230.5

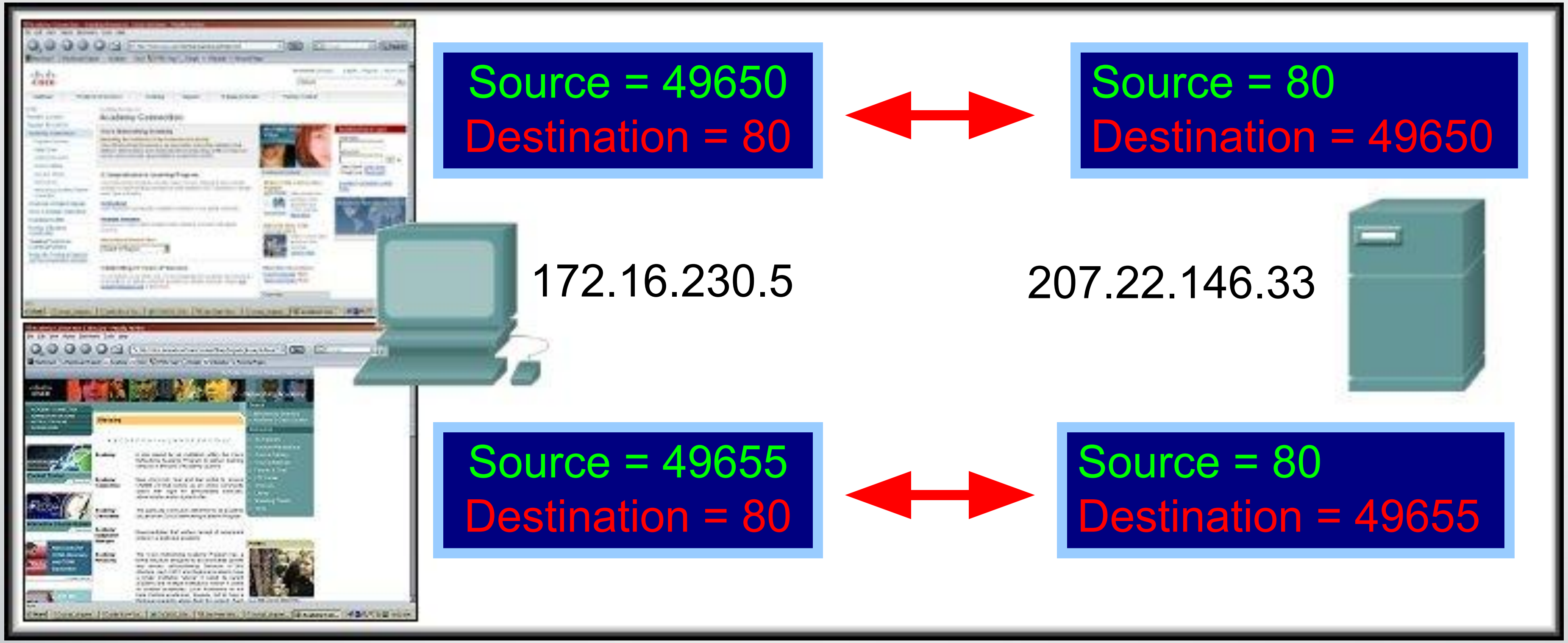
207.22.146.33

Source = 49655
Destination = 80

Source = 80
Destination = 49655

- 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



Source = 49650
Destination = 80

Source = 80
Destination = 49650

172.16.230.5

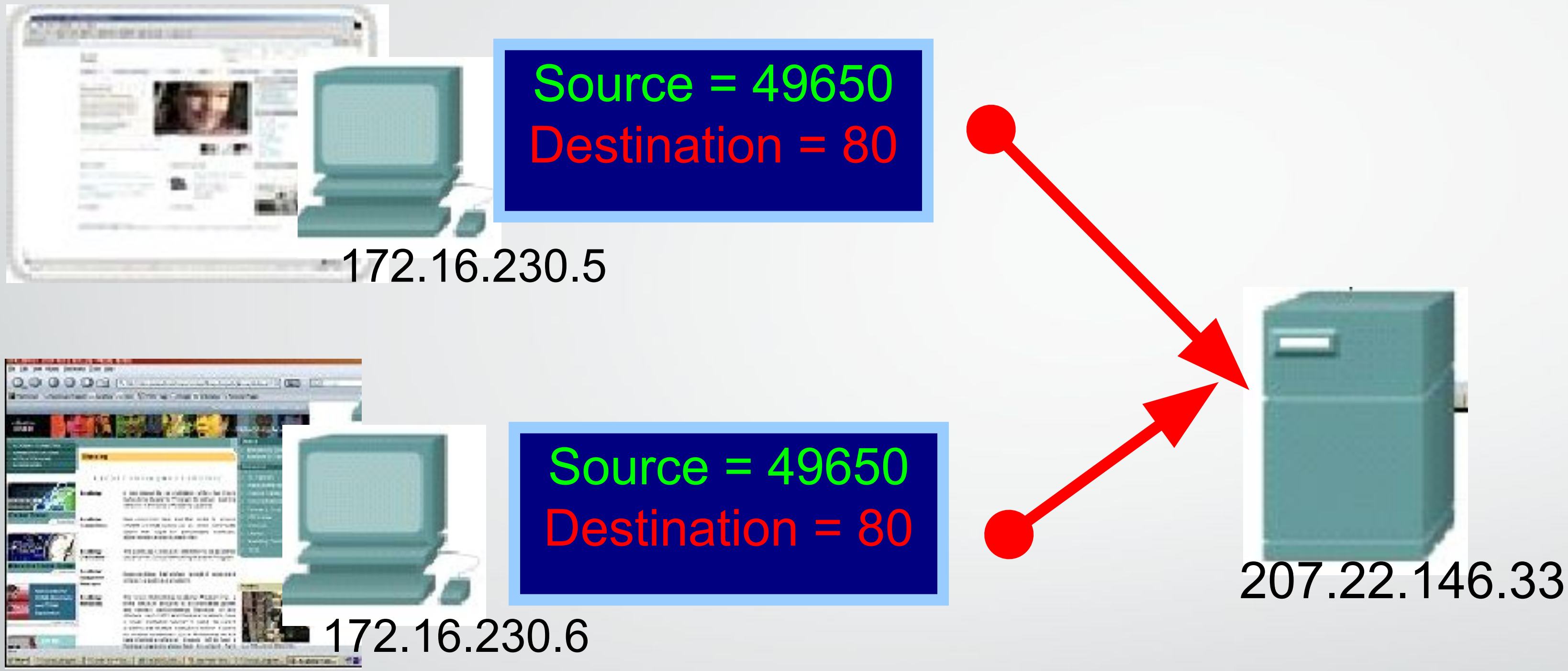
207.22.146.33

Source = 49655
Destination = 80

Source = 80
Destination = 49655

- 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



Source = 49650
Destination = 80

172.16.230.5

Source = 49650
Destination = 80

172.16.230.6

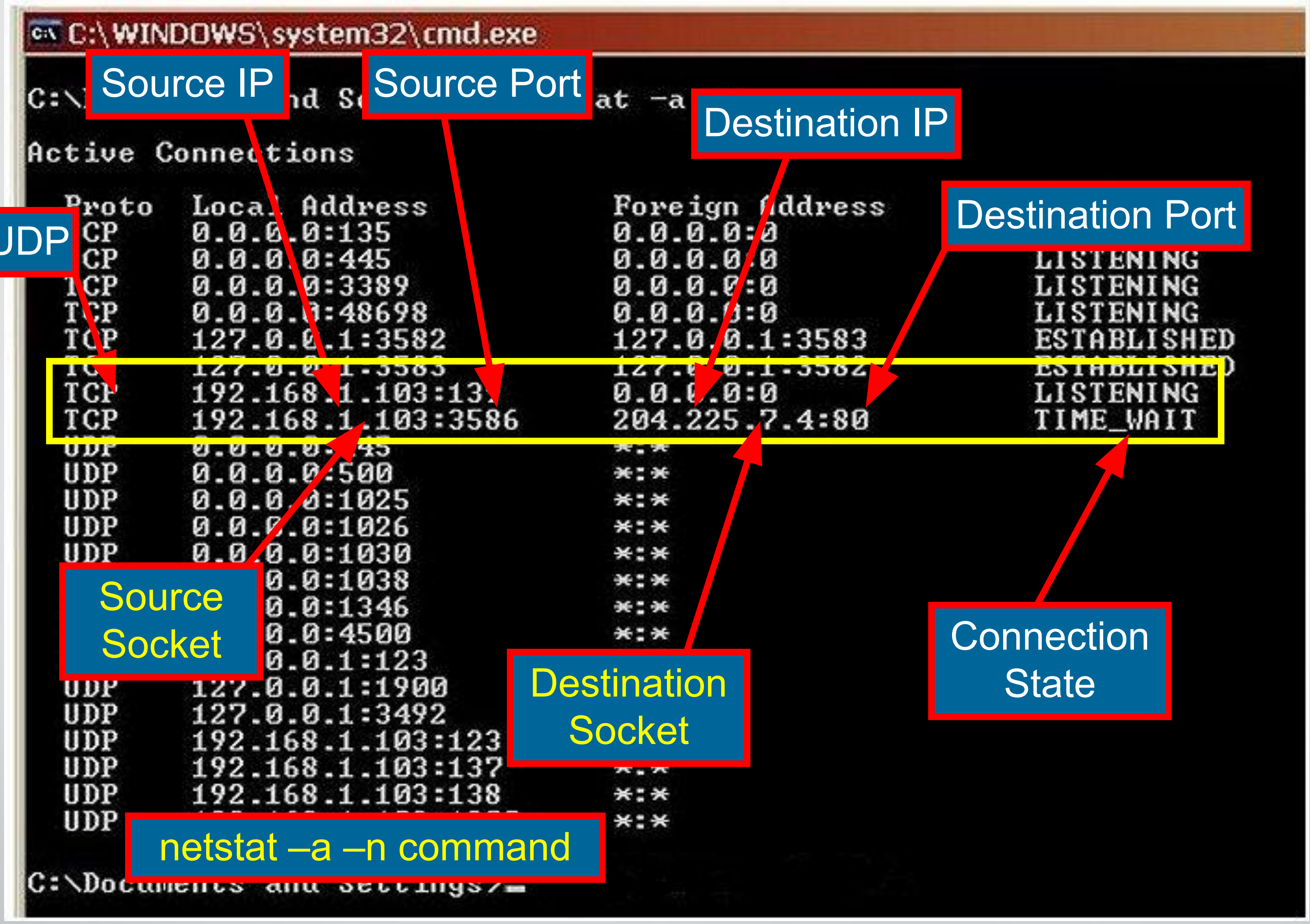
207.22.146.33

- 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



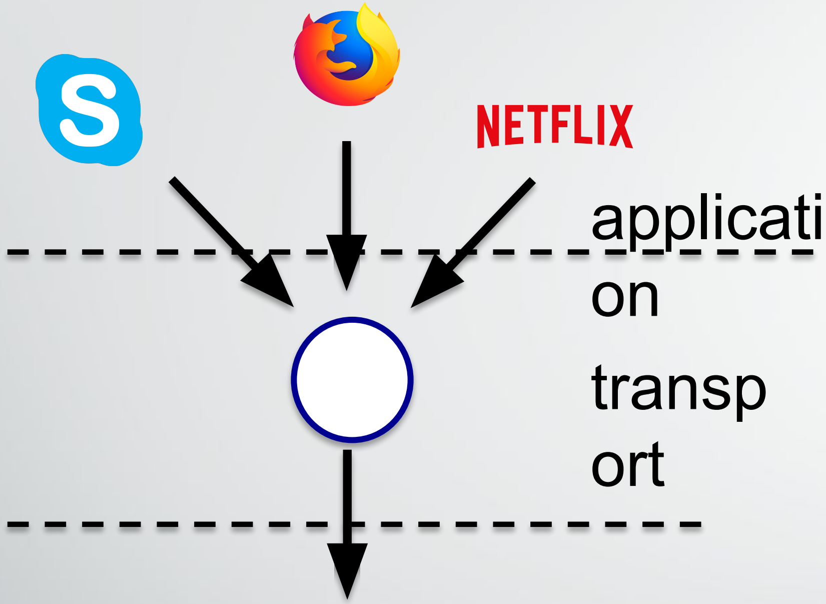
```
C:\      nd S          at -a

Active Connections

  Proto  Local Address          Foreign Address        State
   CP    0.0.0.0:135            0.0.0.0:0
   CP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:3389           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:48698          0.0.0.0:0              LISTENING
  TCP    127.0.0.1:3582         127.0.0.1:3583         ESTABLISHED
  TCP    127.0.0.1:3583         127.0.0.1:3582         ESTABLISHED
  TCP    192.168.1.103:139      0.0.0.0:0              LISTENING
  TCP    192.168.1.103:3586     204.225.7.4:80         TIME_WAIT
  UDP    0.0.0.0:445            *:*
  UDP    0.0.0.0:500            *:*
  UDP    0.0.0.0:1025           *:*
  UDP    0.0.0.0:1026           *:*
  UDP    0.0.0.0:1030           *:*
             0.0:1038           *:*
             0.0:1346           *:*
             0.0:4500           *:*
             0.0.1:123          *:*
  UDP    127.0.0.1:1900
  UDP    127.0.0.1:3492
  UDP    192.168.1.103:123
  UDP    192.168.1.103:137
  UDP    192.168.1.103:138      *:*
  UDP                           *:*

C:\Documents and Settings>
```

Function 4 –Multiplexing

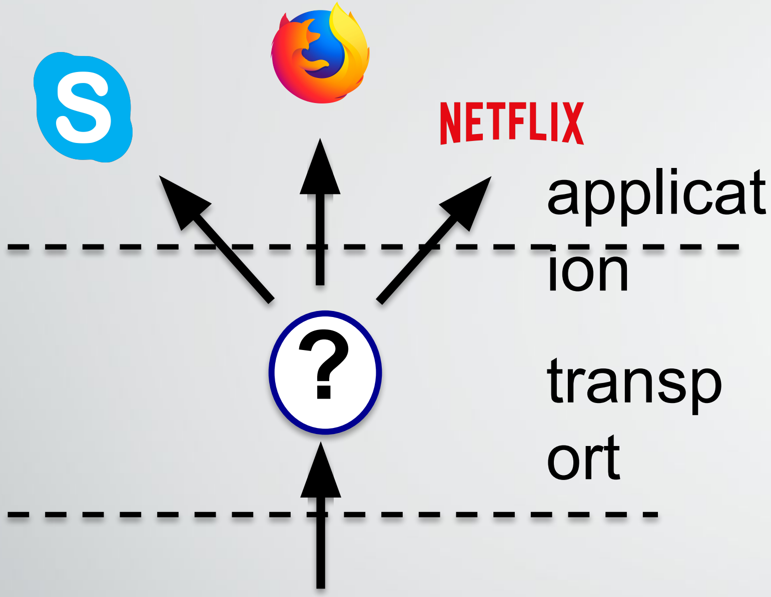


application

transport

multiplexing

Function 4 –DeMultiplexing



application

transport

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!