

# OSI Transport Layer



## Network Fundamentals – Chapter 4

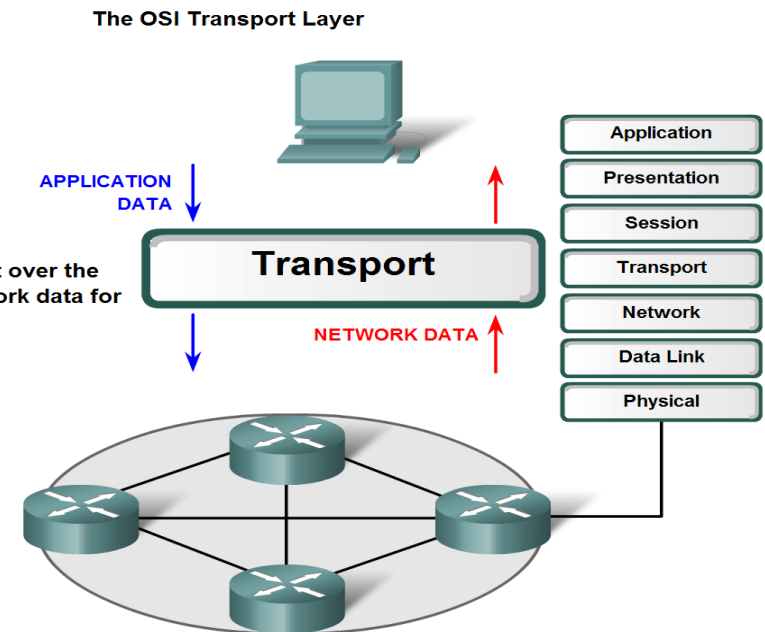
# The Transport Layer:

1. Enables multiple applications to communicate over the network at the same time on a single device.
2. Ensures (if required) reliable deliver of data.
3. Provides error checking and flow control.
  - **TCP** = Connection Transport Service
    - ✓ Reliable Delivery
  - **UDP** = Connectionless Transport Service
    - ✓ Best Effort Delivery

# Controlling the Conversation (Session)

- **TCP and UDP:**
  - ✓ Segment and Reassembly
  - ✓ Conversation Multiplexing
  
- **TCP:**
  - ✓ Establish a Session
  - ✓ Reliable Delivery
  - ✓ Same Order Delivery
  - ✓ Flow Control

The Transport layer prepares application data for transport over the network and processes network data for use by applications.



# Supporting Reliable Communication

## Transport Layer Protocols

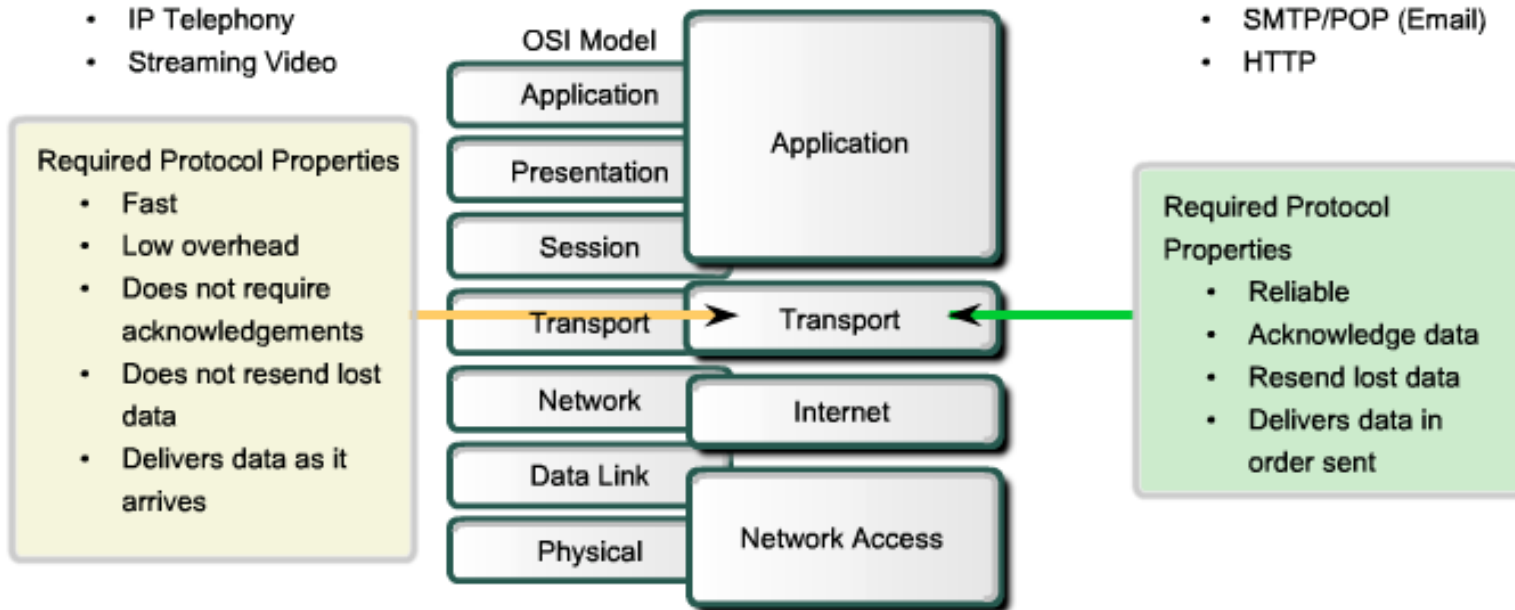


- IP Telephony
- Streaming Video



- SMTP/POP (Email)
- HTTP

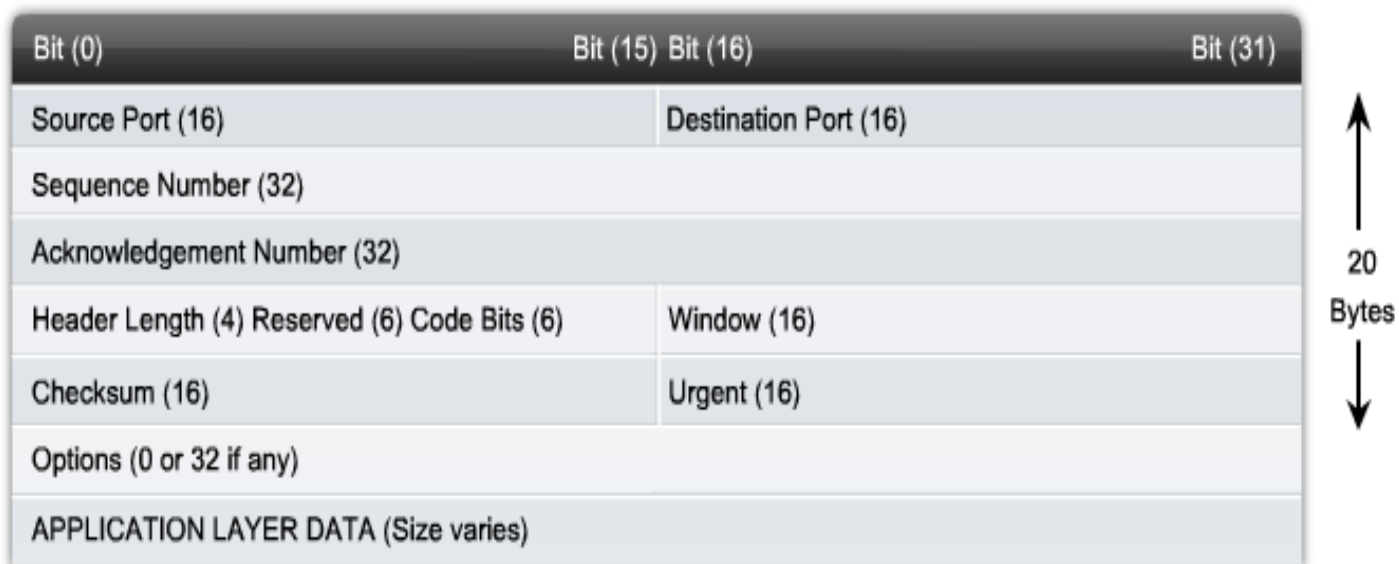
### TCP/IP Model



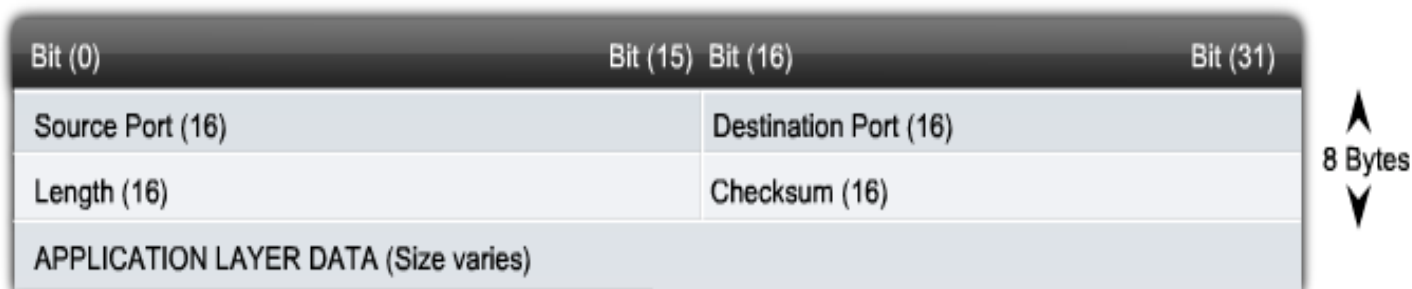
Application developers choose the appropriate Transport Layer protocol based on the nature of the application.

## TCP and UDP Headers

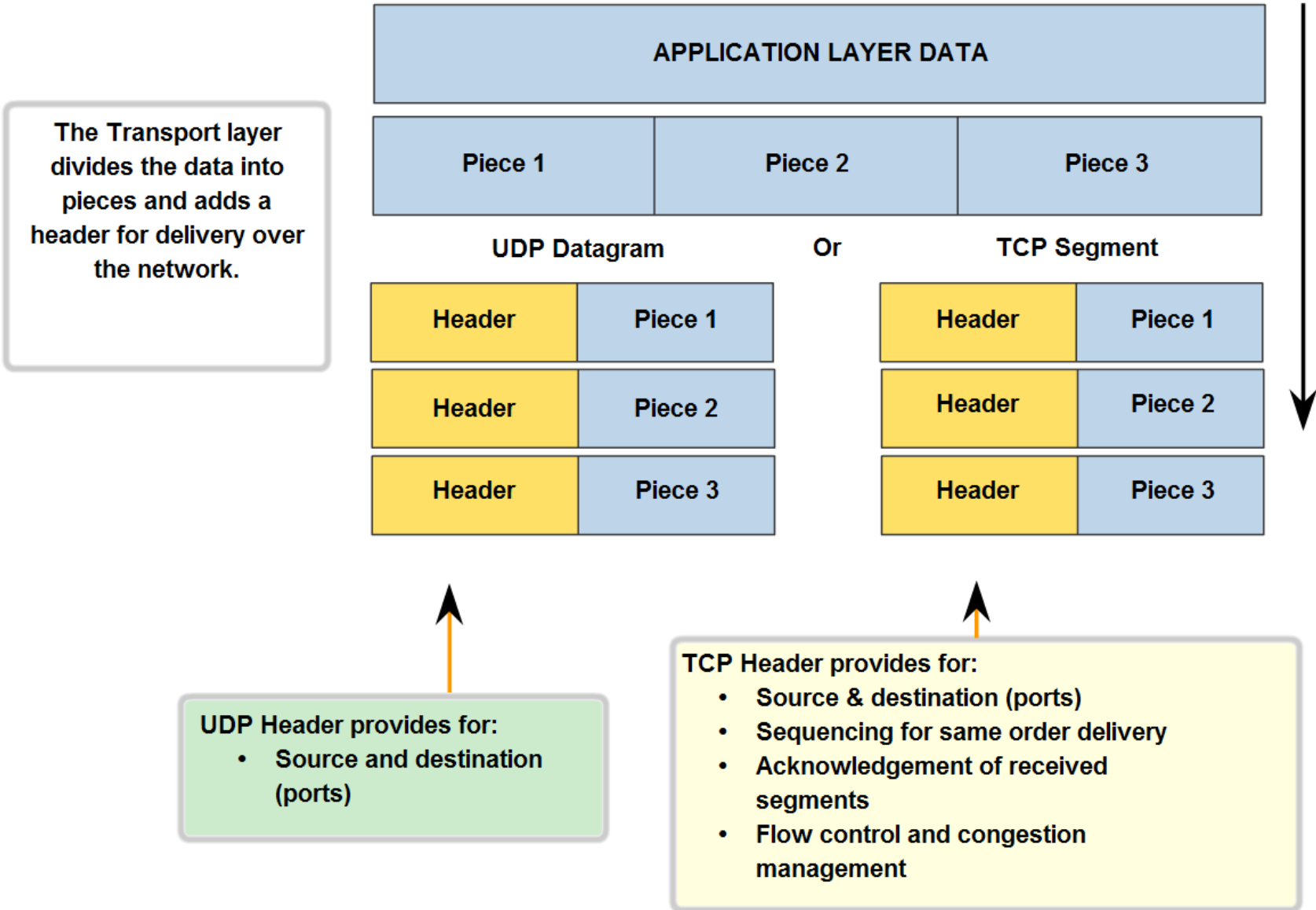
### TCP Segment



### UDP Datagram

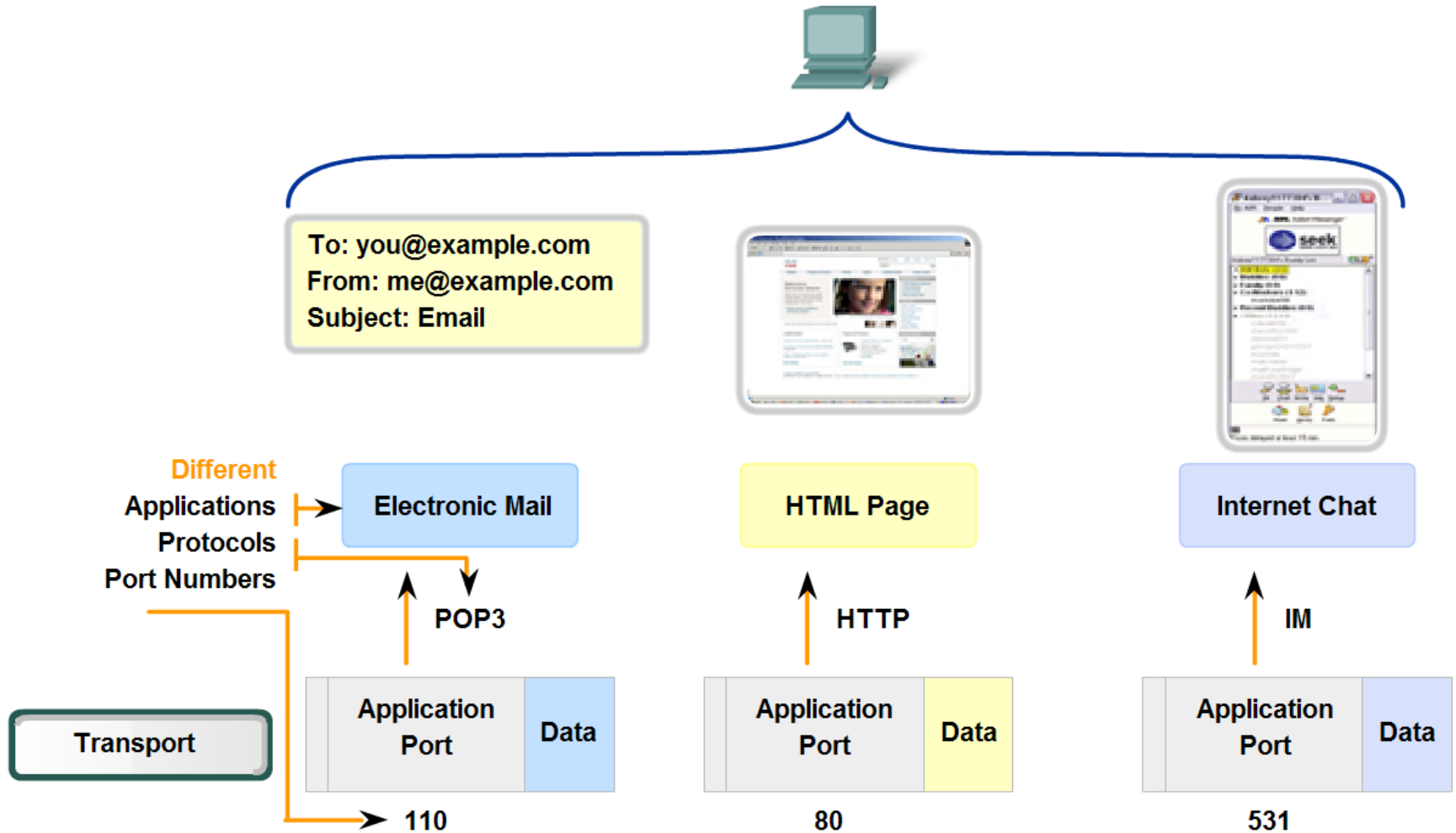


## Transport Layer Functions



## Port Addressing

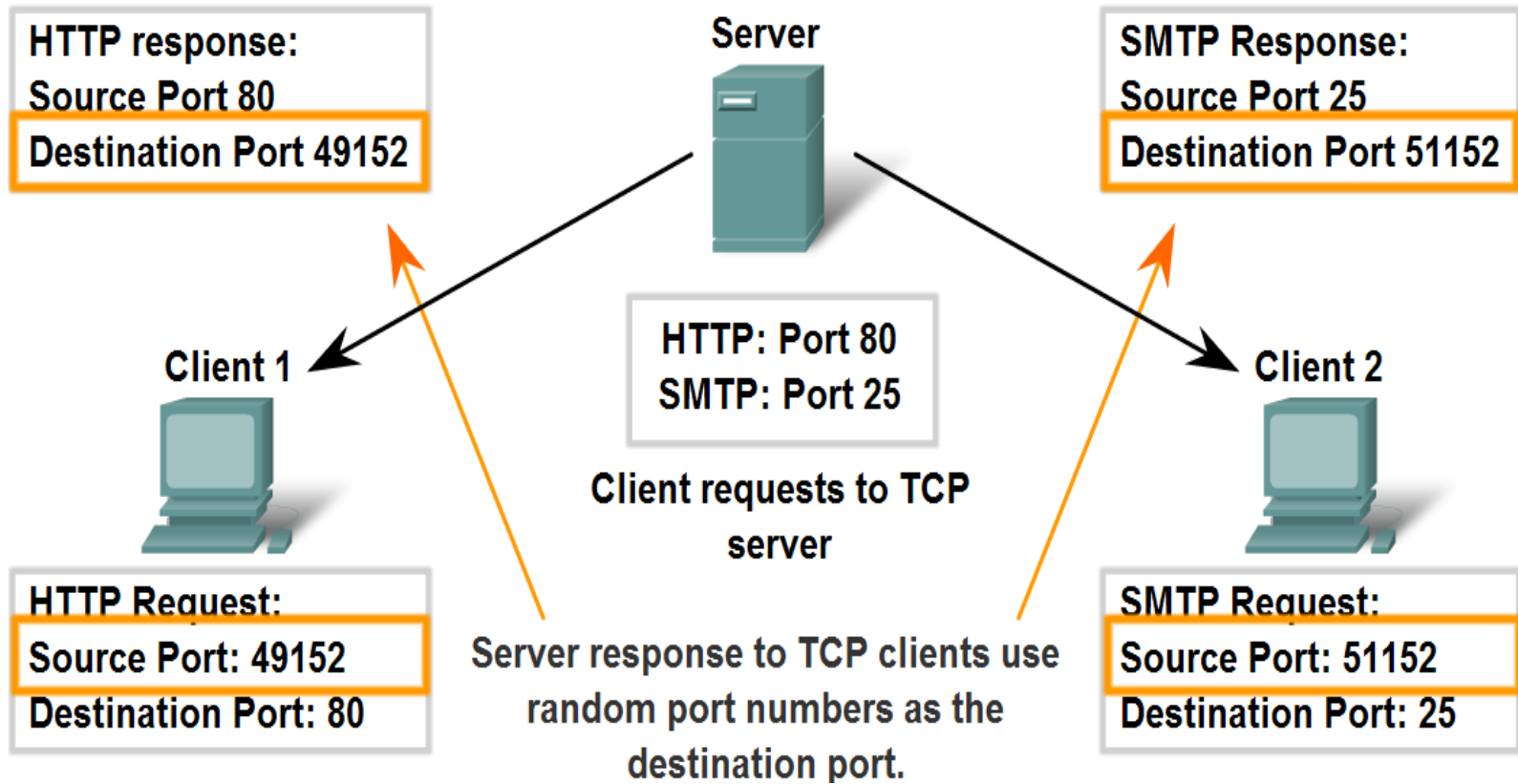
2/



Data for different applications is directed to the correct application because each application has a unique port number.



## Clients Sending TCP Requests

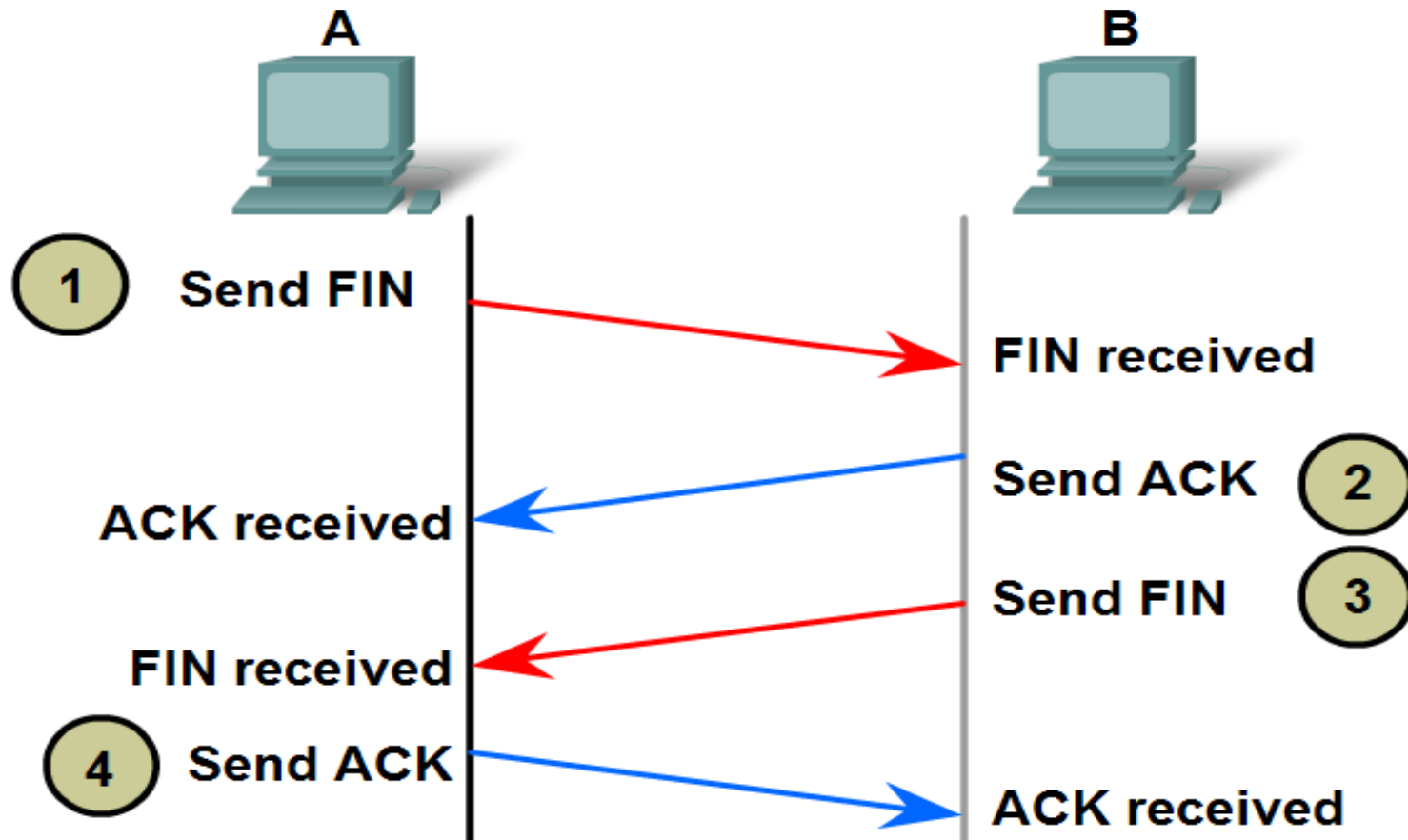


**Socket = 200.210.110.14:80    Socket = 143.116.153.19:25**



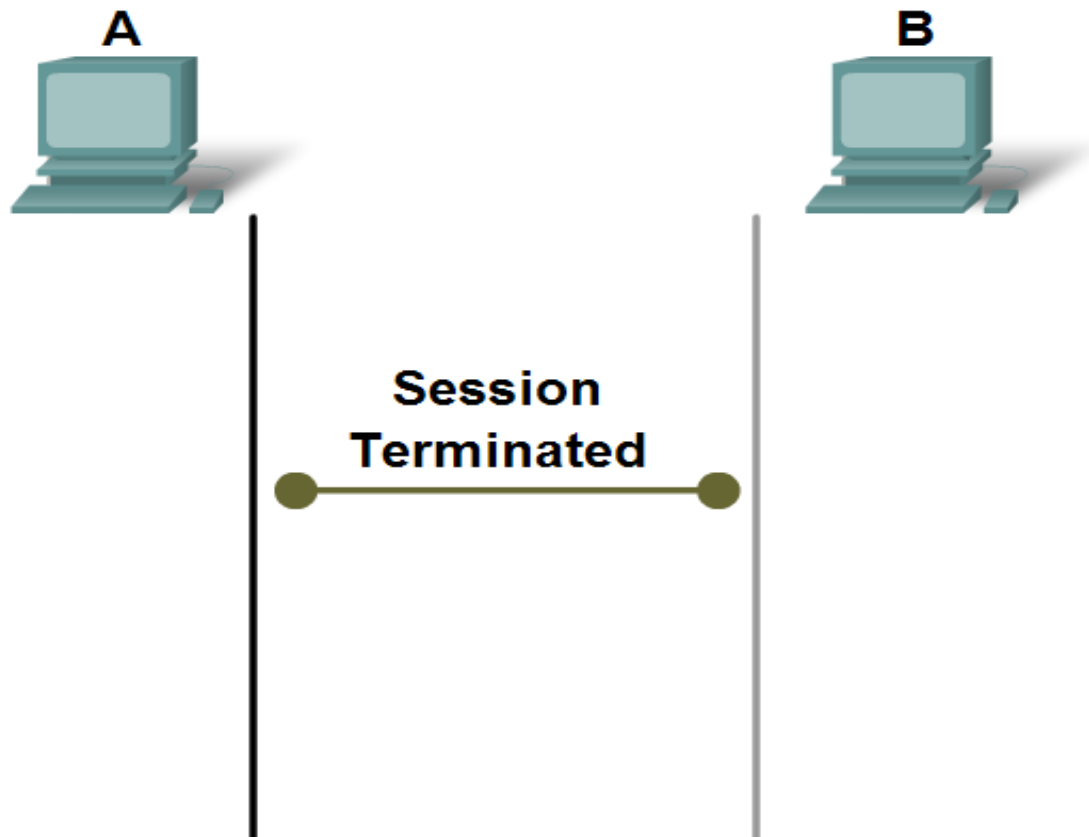
# TCP 3 Way Handshake Establishes Session

## TCP Connection Establishment and Termination

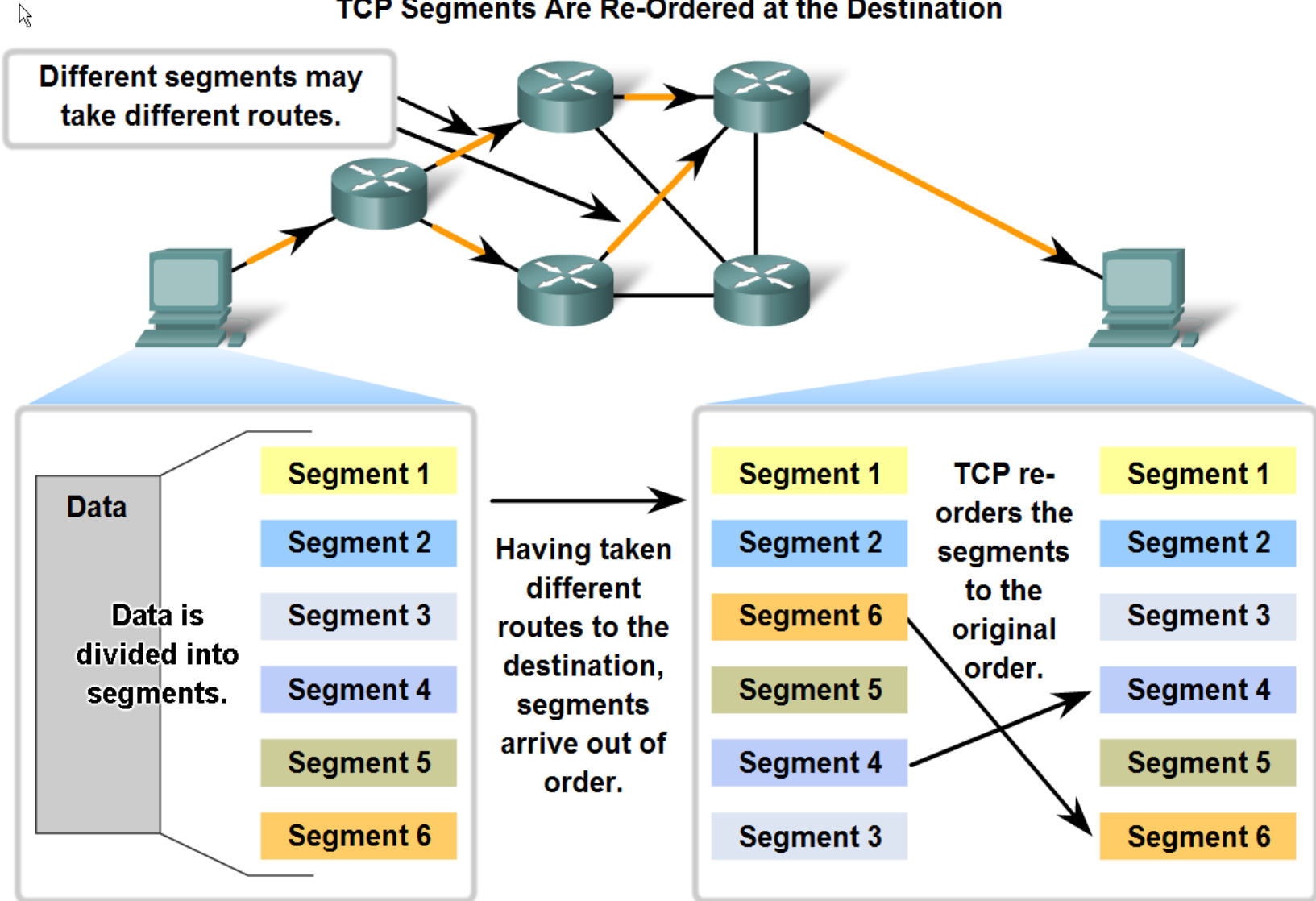


# Two Way Handshake Terminates Session

## TCP Connection Establishment and Termination

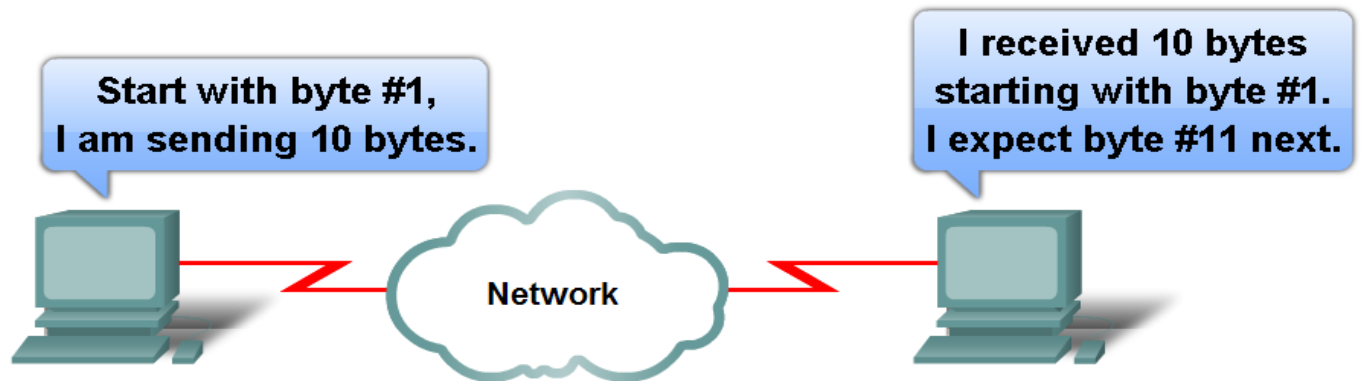


### TCP Segments Are Re-Ordered at the Destination



## Acknowledgement of TCP Segments

Source Port	Destination Port	Sequence Number	Acknowledgement Numbers	...
-------------	------------------	-----------------	-------------------------	-----



Source	Des.	Seq.	Ack.	...
1028	23	1	1	...

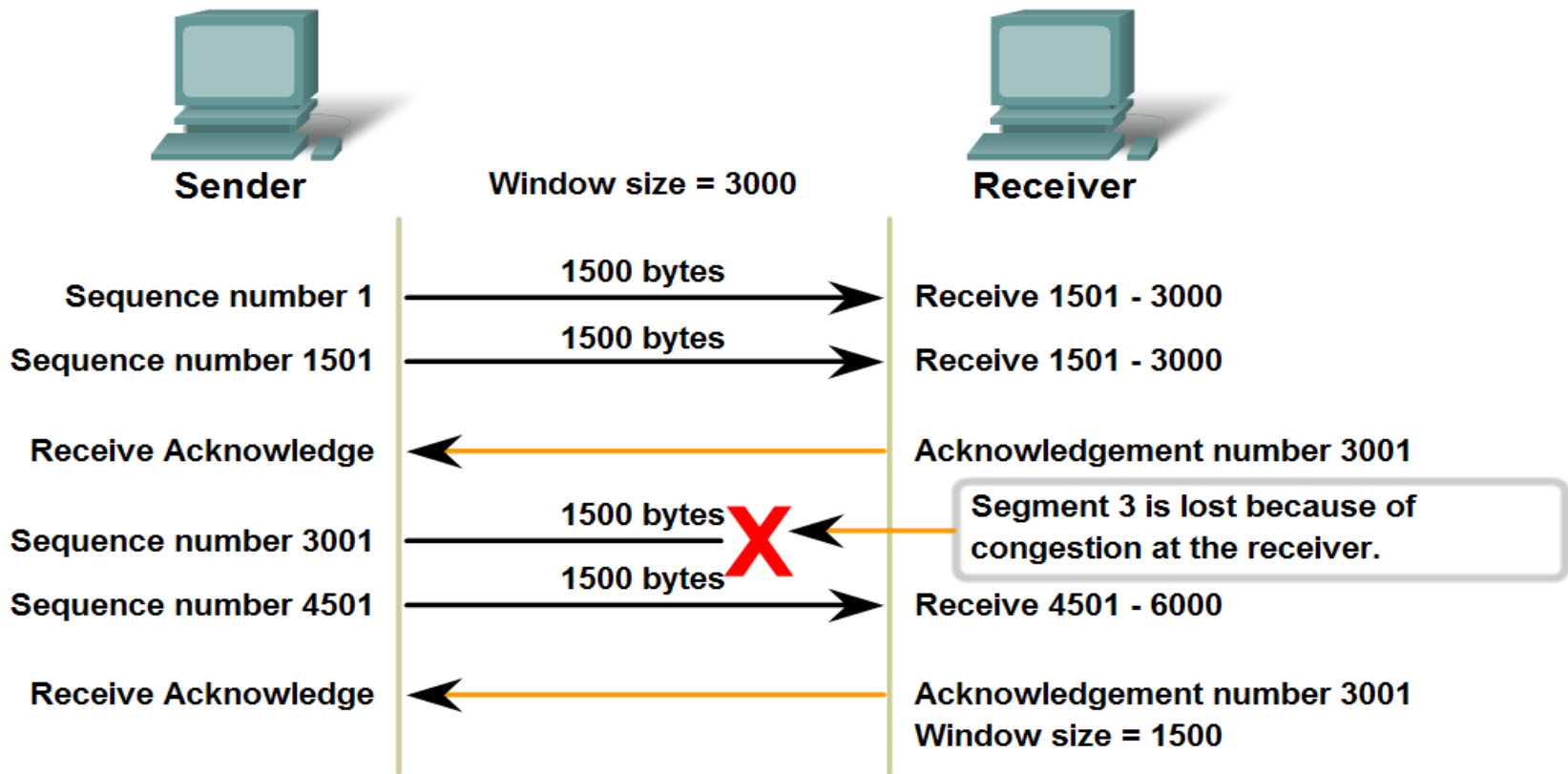
10 bytes

Source	Des.	Seq.	Ack.	...
23	1028	1	11	...

more bytes starting with byte #11

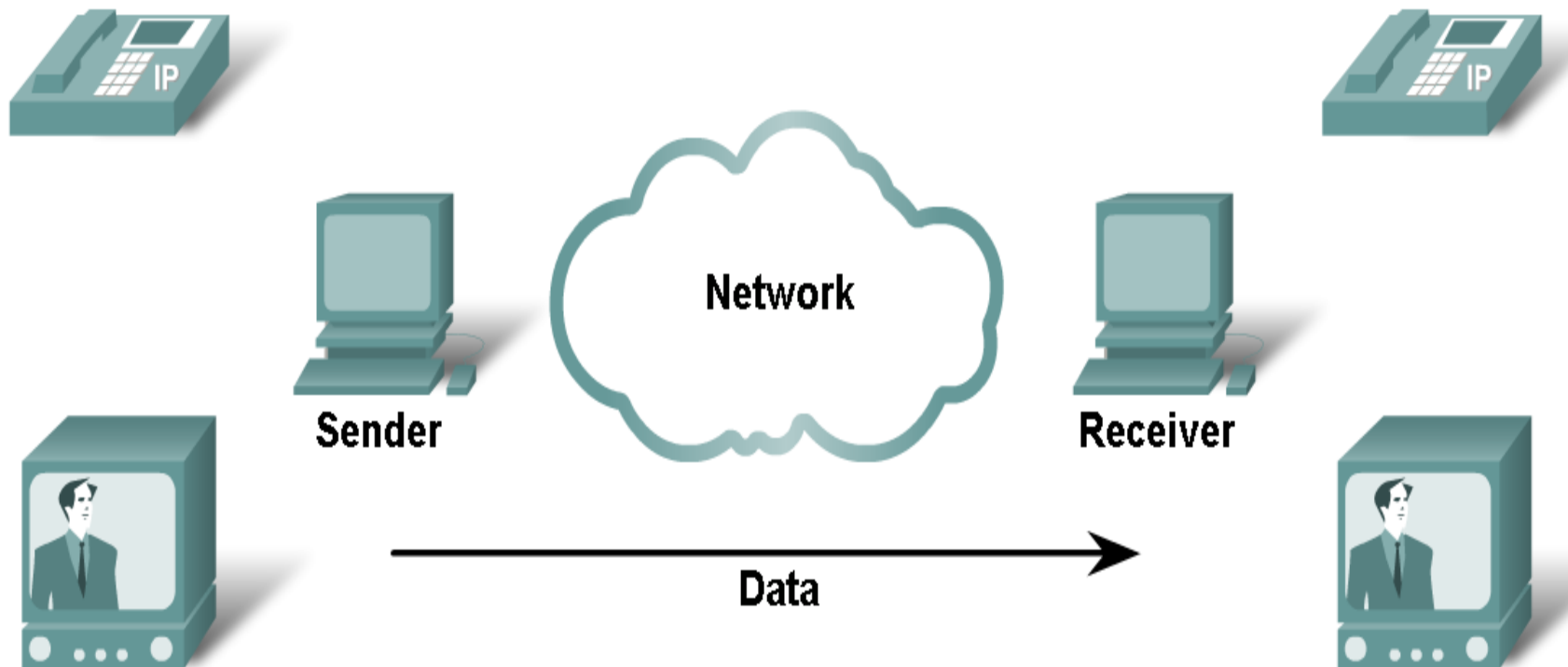
Source	Des.	Seq.	Ack.	...
1028	23	11	1	...

## TCP Congestion and Flow Control



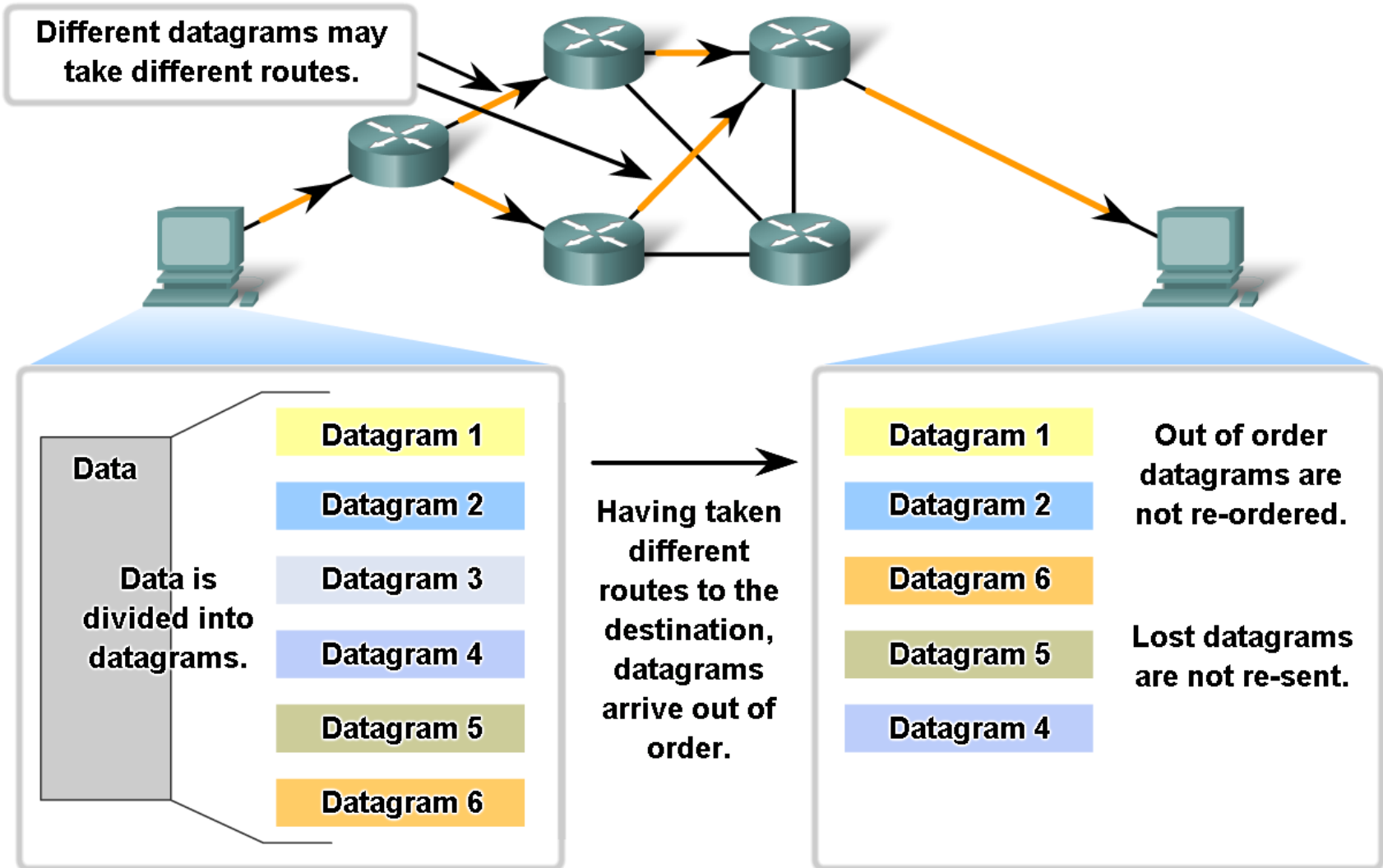
**If segments are lost because of congestion, the Receiver will acknowledge the last received sequential segment and reply with a reduced window size.**

## UDP Low Overhead Data Transport



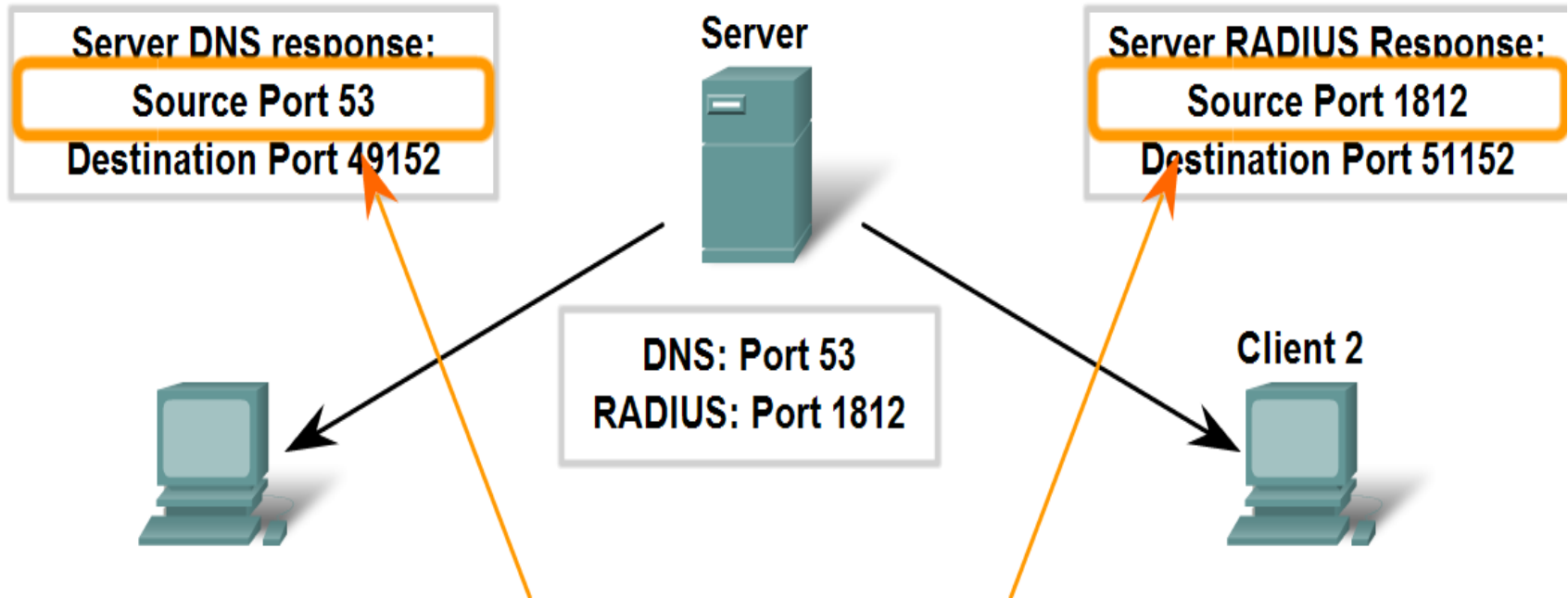
**UDP does not establish a connection before sending data.**

### UDP: Connectionless and Unreliable





### Clients Sending UDP Requests



Server response to UDP clients use well known port numbers as the source port.

Client 1 waiting for server DNS response on Port 49152

Client 2 waiting for server RADIUS response on Port 51152

# Port Numbers:

- Well Known Port Numbers: 0 - 1023
  - Use [Wikipedia](#) for Well Known Port Numbers
  - 20 = FTP (File Transfer Protocol)
  - 22 = SSH (Secure Shell)
  - 23 = Telnet
  - 25 = SMTP (Simple Mail Transfer Protocol)
  - 43 = Whois Protocol
  - 53 = DNS (Domain Name Service)
  - 80 = HTTP (Hyper Text Transfer Protocol)
- Registered Port Numbers: 1024 – 49151
- Dynamic or Private Port Numbers: 49152 - 65535

