

# EXPLORATION 1

---

Chapter 4

OSI Transport Layer

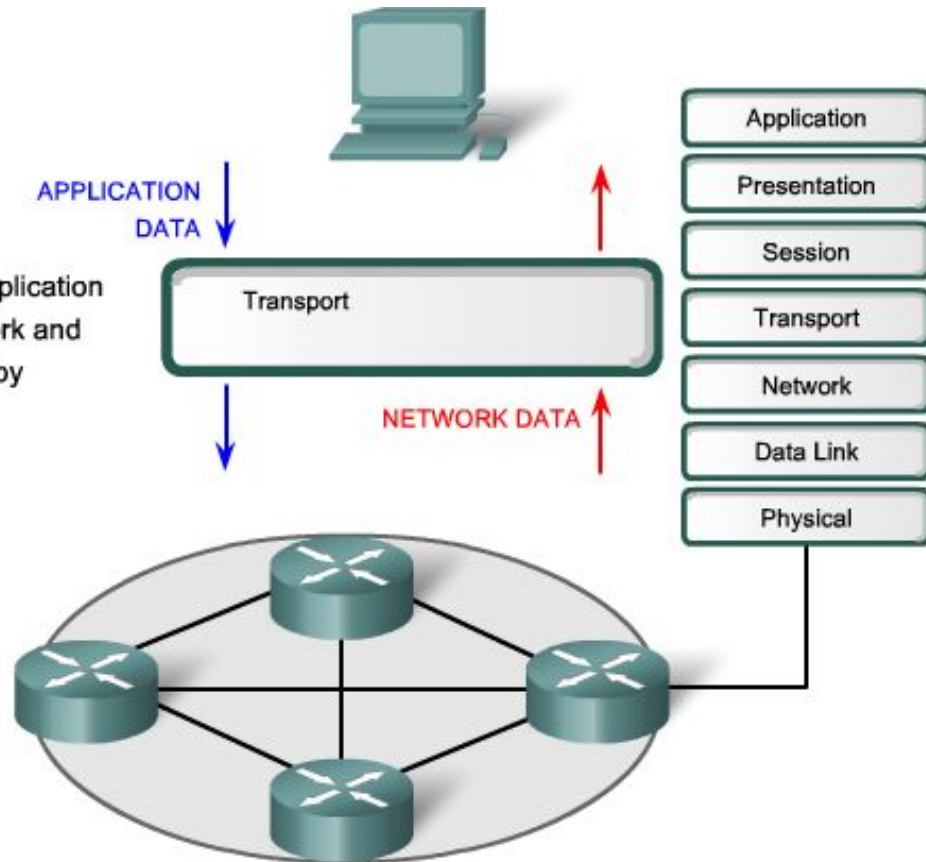
By Zhamanov Azamat

# Upon completion of this chapter, you will be able to:

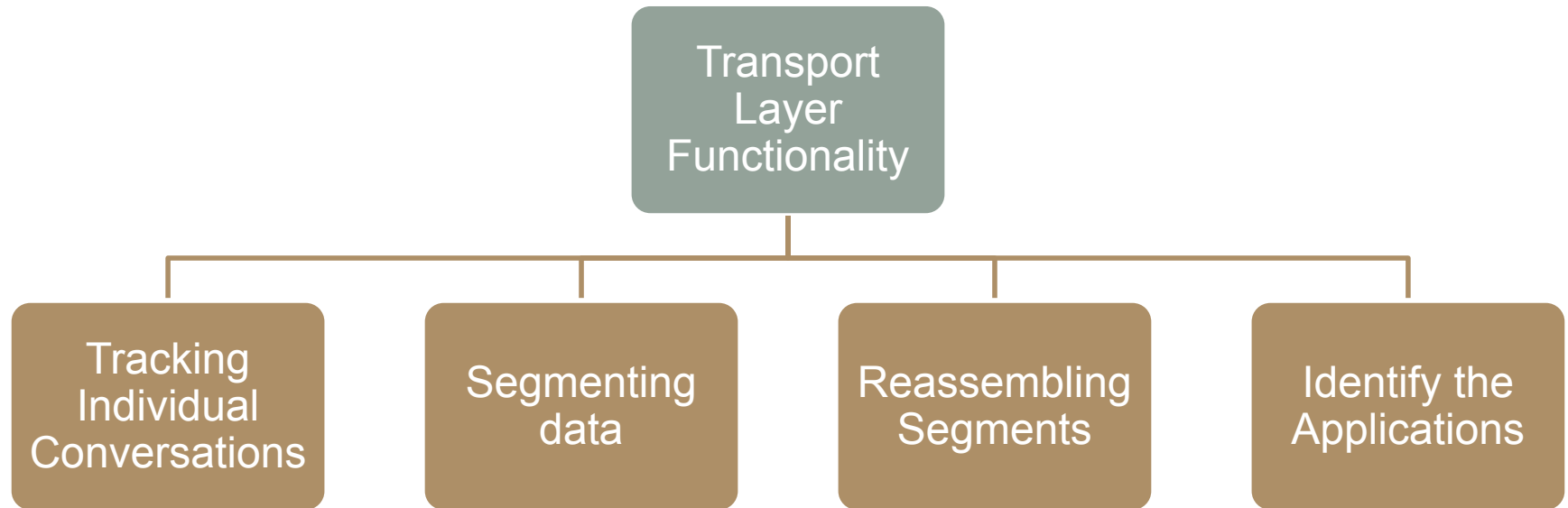
- Explain the need for the Transport layer.
- Identify the role of the Transport layer as it provides the end-to-end transfer of data between applications.
- Describe the role of two TCP/IP Transport layer protocols: TCP and UDP.
- Explain the key functions of the Transport layer, including reliability, port addressing, and segmentation.
- Explain how TCP and UDP each handle key functions.
- Identify when it is appropriate to use TCP or UDP and provide examples of applications that use each protocol.

# Transport Layer

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



# Purpose of Transport Layer



# Tracking Individual Conversations

- Any host may have multiple applications that are communicating across the network. Each of these applications will be communicating with one or more applications on remote hosts. It is the responsibility of the Transport layer to maintain the multiple communication streams between these applications.

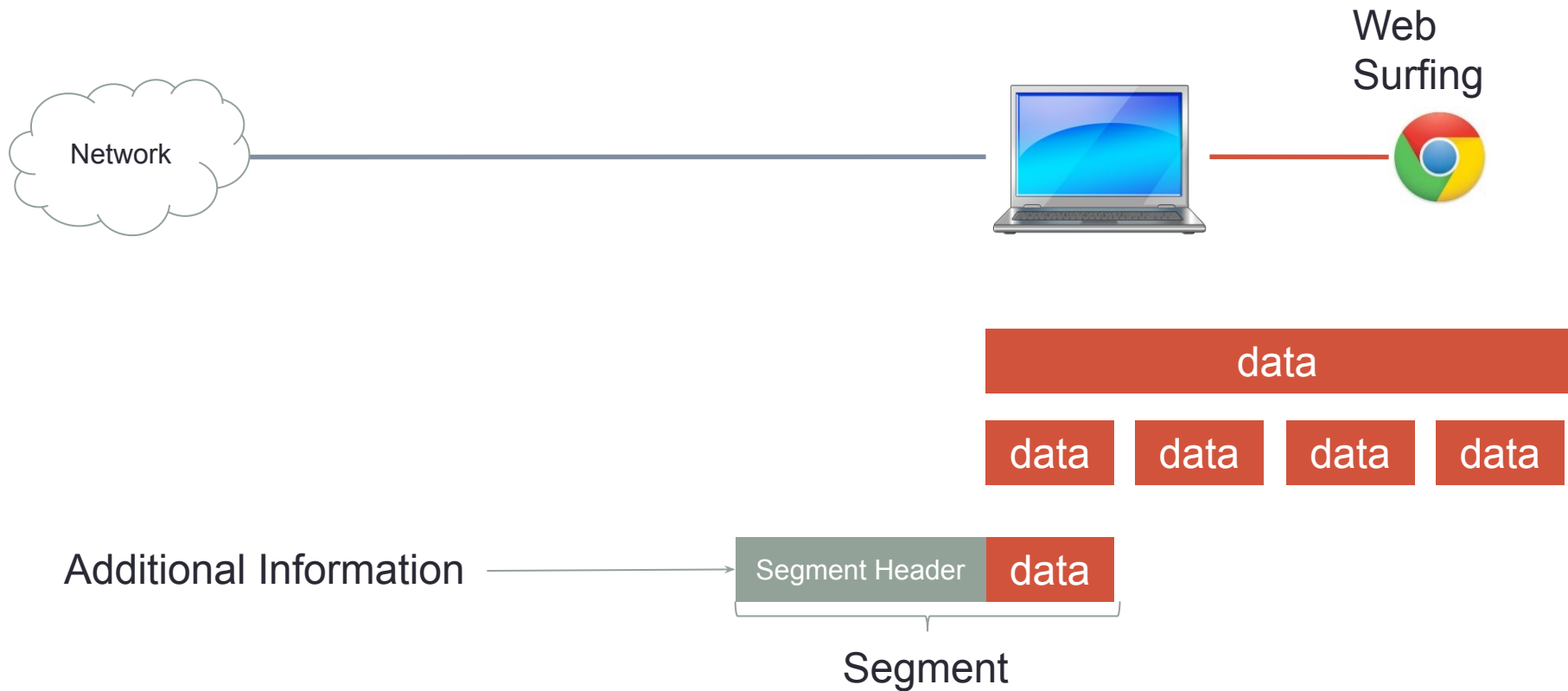
# Tracking Individual Conversations



# Segmenting Data

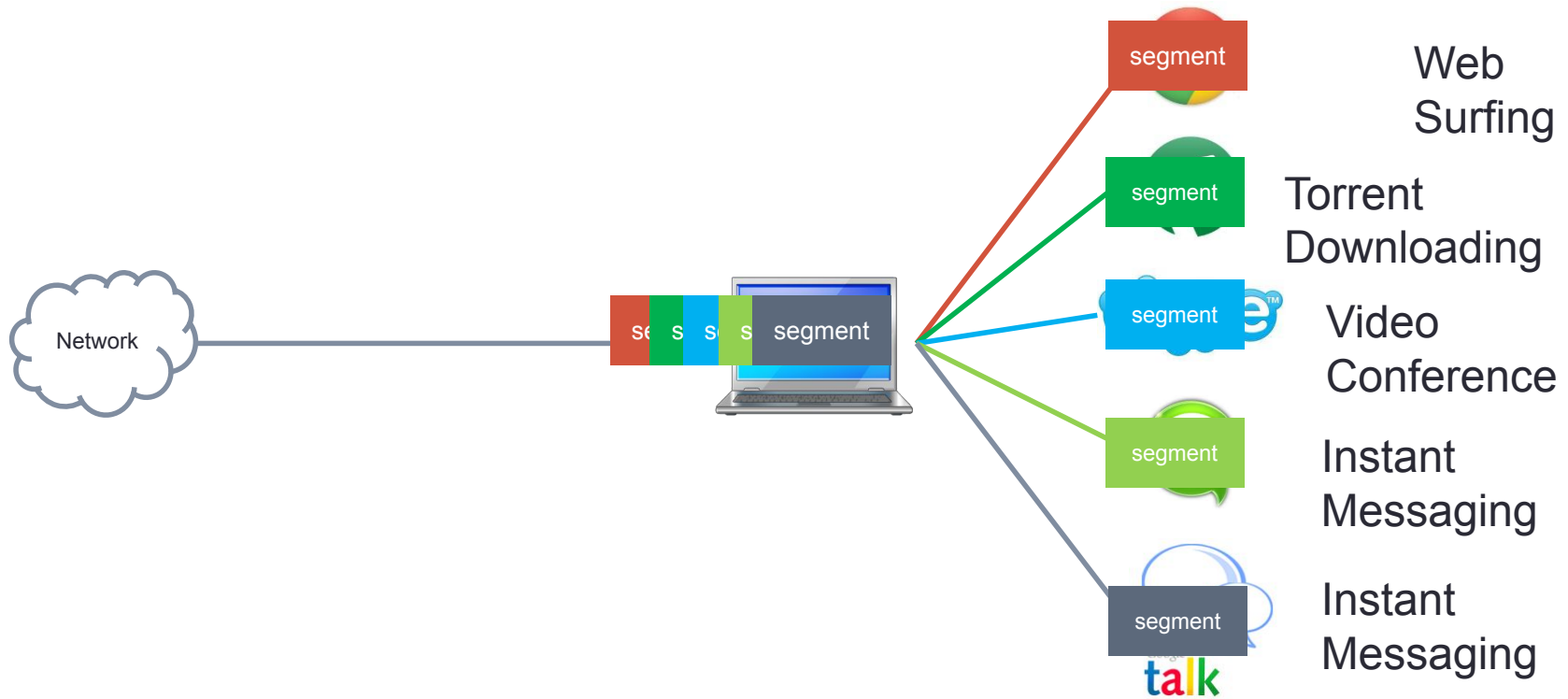
- The Transport layer protocols describe services that segment data from the Application layer.
- This includes the encapsulation required on each piece of data.
- Each piece of application data requires headers to be added at the Transport layer to indicate to which communication it is associated.

# Segmentation





# Segmentation



# Reassembling Segments

- At the receiving host, each piece of data may be directed to the appropriate application.
- Additionally, these individual pieces of data must also be reconstructed into a complete data stream that is useful to the Application layer.
- The protocols at the Transport layer describe the how the Transport layer header information is used to reassemble the data pieces into streams to be passed to the Application layer.

# Reassembling Segments, Application Identification

Web  
Surfing



Network

data

data

data

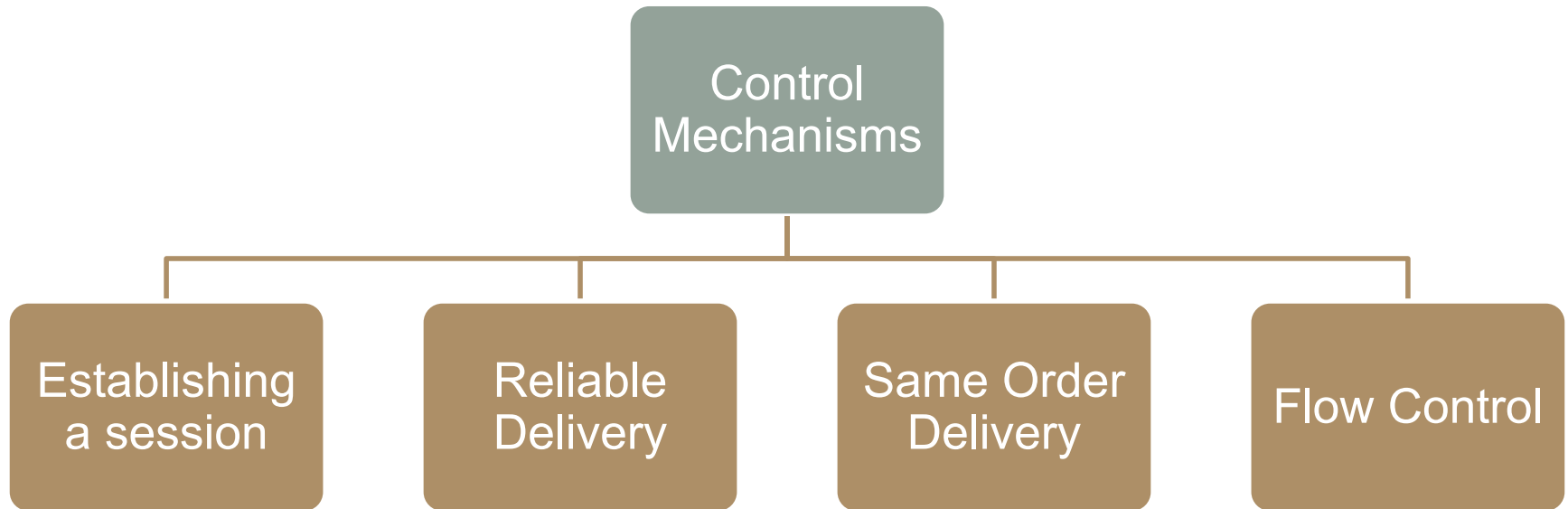
data

data

Segment Header

data

# Controlling the Conversations



# Establishing a Session

- The Transport layer can provide this connection orientation by creating a sessions between the applications. These connections prepare the applications to communicate with each other before any data is transmitted. Within these sessions, the data for a communication between the two applications can be closely managed.

# Reliable Delivery

- For many reasons, it is possible for a piece of data to become corrupted, or lost completely, as it is transmitted over the network. The Transport layer can ensure that all pieces reach their destination by having the source device to retransmit any data that is lost.

# Same Order Delivery

- Because networks may provide multiple routes that can have different transmission times, data can arrive in the wrong order. By numbering and sequencing the segments, the Transport layer can ensure that these segments are reassembled into the proper order.

# Flow Control Mechanism

- Network hosts have limited resources, such as memory or bandwidth. When Transport layer is aware that these resources are overtaxed, some protocols can request that the sending application reduce the rate of data flow. This is done at the Transport layer by regulating the amount of data the source transmits as a group. Flow control can prevent the loss of segments on the network and avoid the need for retransmission.



# Transport Layer Protocols

- Computers use different types of Transport layer protocols for some reasons.
- What is the reason for it?

# Question

- What kind of Computer Network Communications do you know?

# Different kinds of Computer Networks

- Web Browsing
- File sharing
- IP Telephony
- Video Streaming (conference or online lesson)
- Working with mail

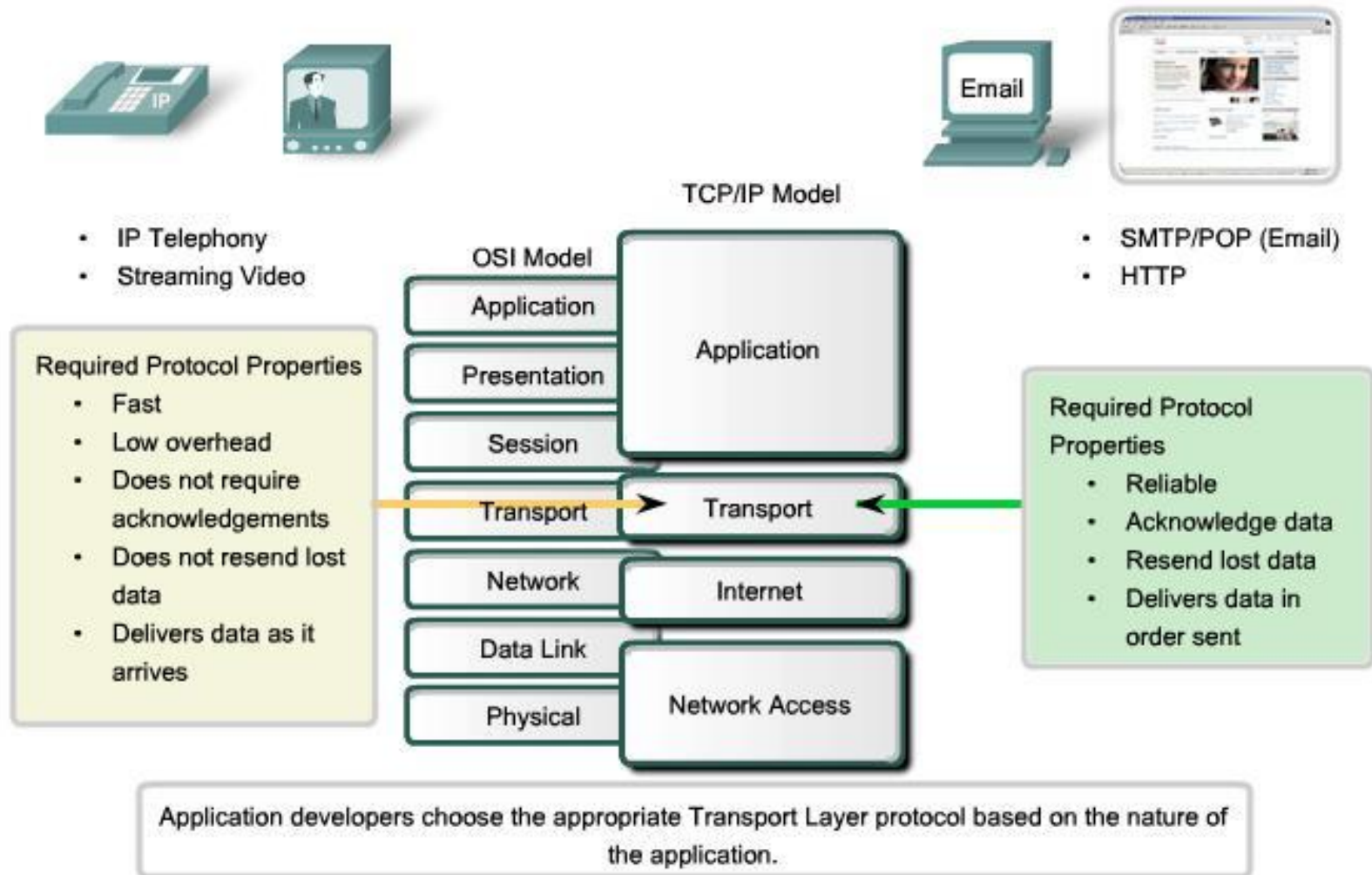
# Question

- For which of them we need reliability in delivery?

# Question

- For which of them we need Speed in delivery?

# Transport Layer Protocols



# TCP - Transmission Control Protocol and UDP – User Datagram Protocol

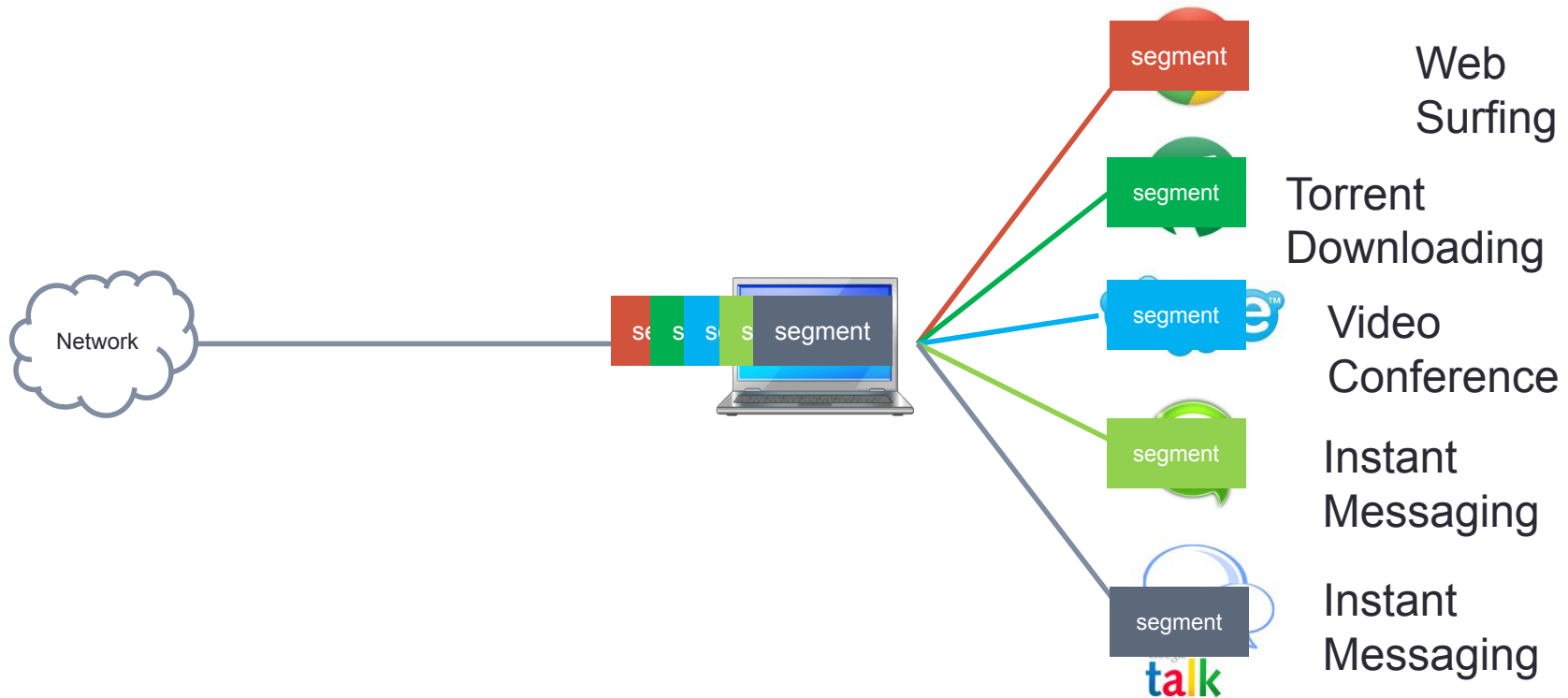
- TCP

- Web Browsers
- E-mail
- File Transfers

- UDP

- Video Streaming
- Voice over IP (VoIP)

# Identifying Conversations





# Question

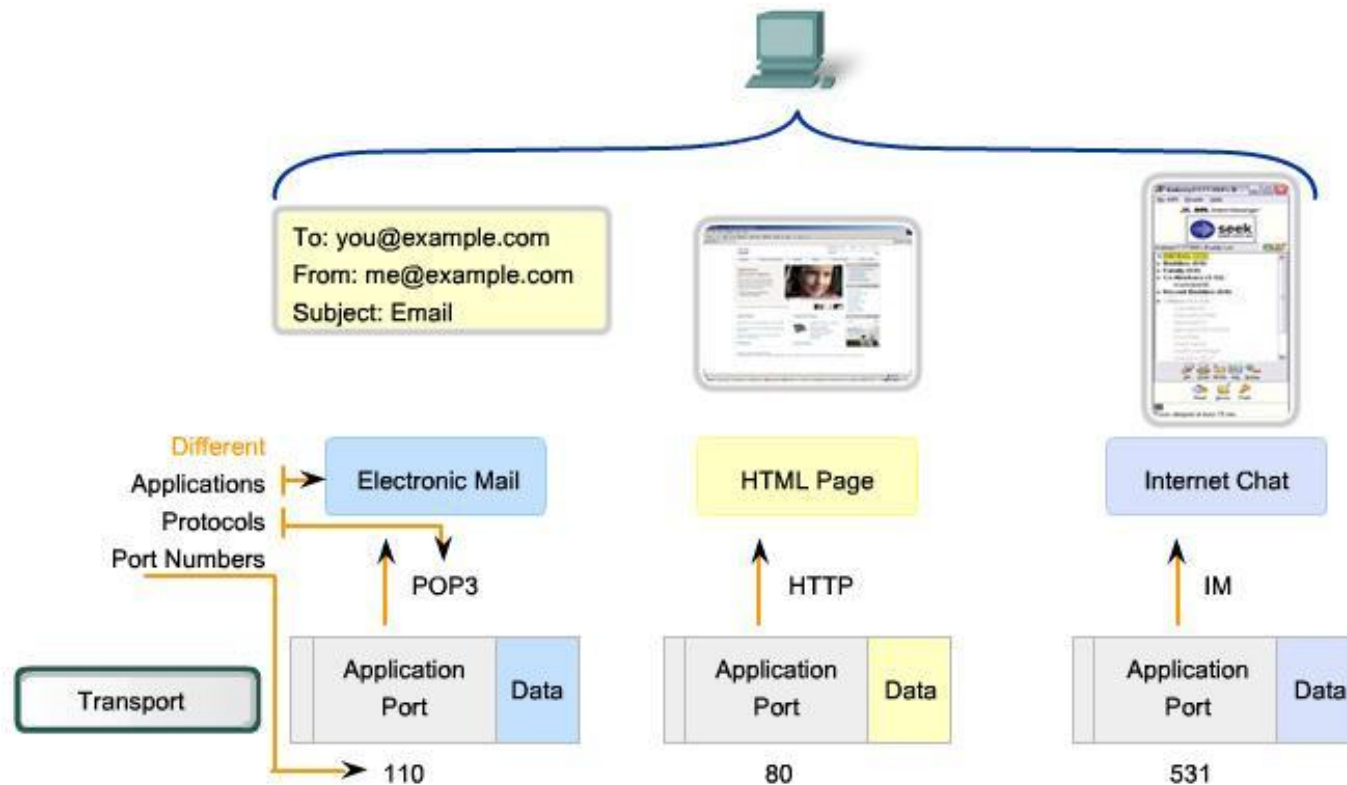
- We can see visual different colors of segments, by using colors we separate segments.
- How computers make separation between segments?

# Answer

- Computers use ***Port Addresses***

# Port Addressing

- Ports are used to identify conversations



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

# Port Addresses

- Well Known Ports (Numbers 0 to 1023)
  - Registered Ports (Numbers 1024 to 49151)
  - Dynamic or Private Ports (Numbers 49152 to 65535)
- 
- NETSTAT is an important network utility that can be used to verify current connections

# Well Known Ports

- Are served for services and applications.
- They are commonly used for applications such as HTTP (web server) POP3/SMTP (e-mail server) and Telnet.



# Registered Ports

- Are assigned to user processes or applications.
- These processes are primarily individual applications that a user has chosen to install rather than common applications that would receive a Well Known Port.
- When not used for a server resource, these ports may also be used dynamically selected by a client as its source port.



# Dynamic or Private Ports

- Also known as Ephemeral Ports, these are usually assigned dynamically to client applications when initiating a connection.
- It is not very common for a client to connect to a service using a Dynamic or Private Port (although some peer-to-peer file sharing programs do).

# Question

- How to view states of connection in your computer?



# Answer

- Use *Netstat*

# Netstat Snapshot

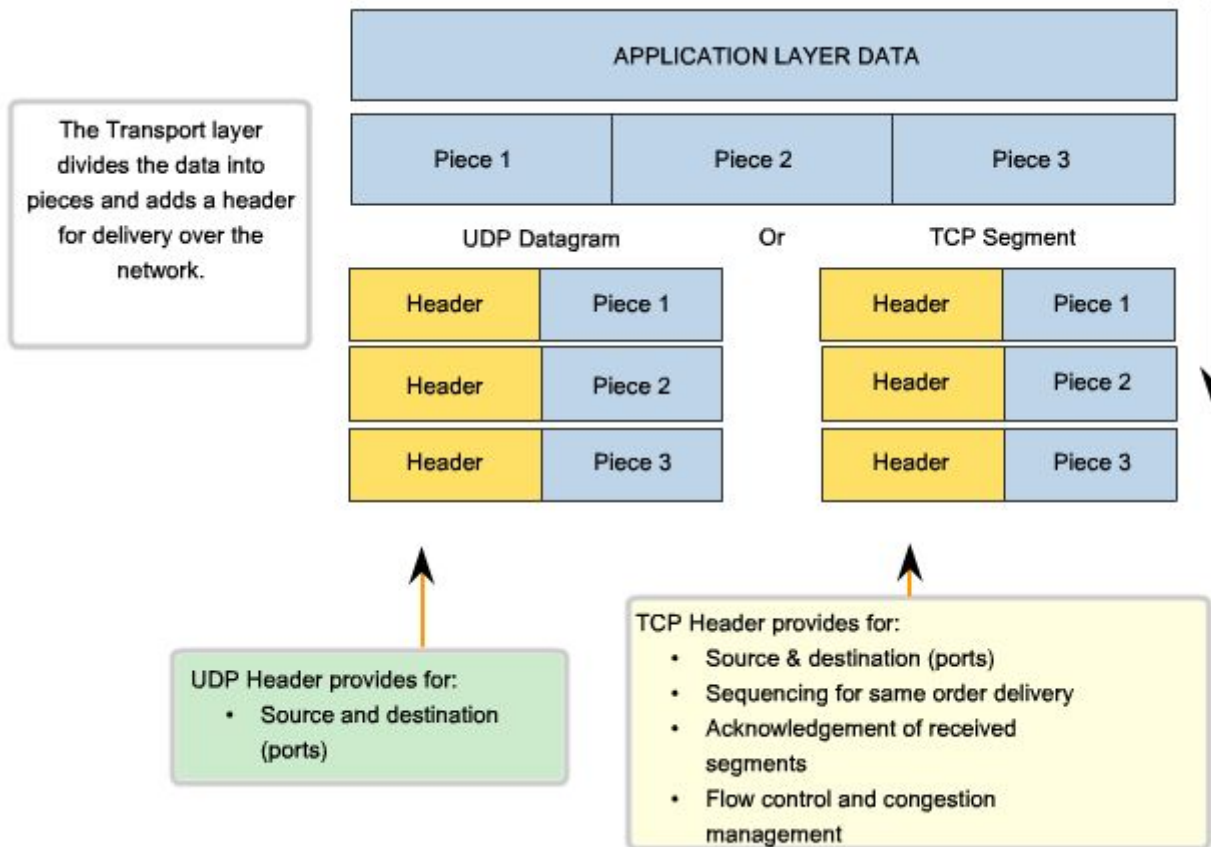
```
C:\>netstat

Active Connections

Proto  Local Address           Foreign Address         State
TCP    kenpc:3126              192.168.0.2:netbios-ssn ESTABLISHED
TCP    kenpc:3158              207.138.126.152:http   ESTABLISHED
TCP    kenpc:3159              207.138.126.169:http   ESTABLISHED
TCP    kenpc:3160              207.138.126.169:http   ESTABLISHED
TCP    kenpc:3161              sc.msn.com:http        ESTABLISHED
TCP    kenpc:3166              www.cisco.com:http     ESTABLISHED

C:\>
```

# TCP and UDP Handle Segmentation Differently.



# TCP Header

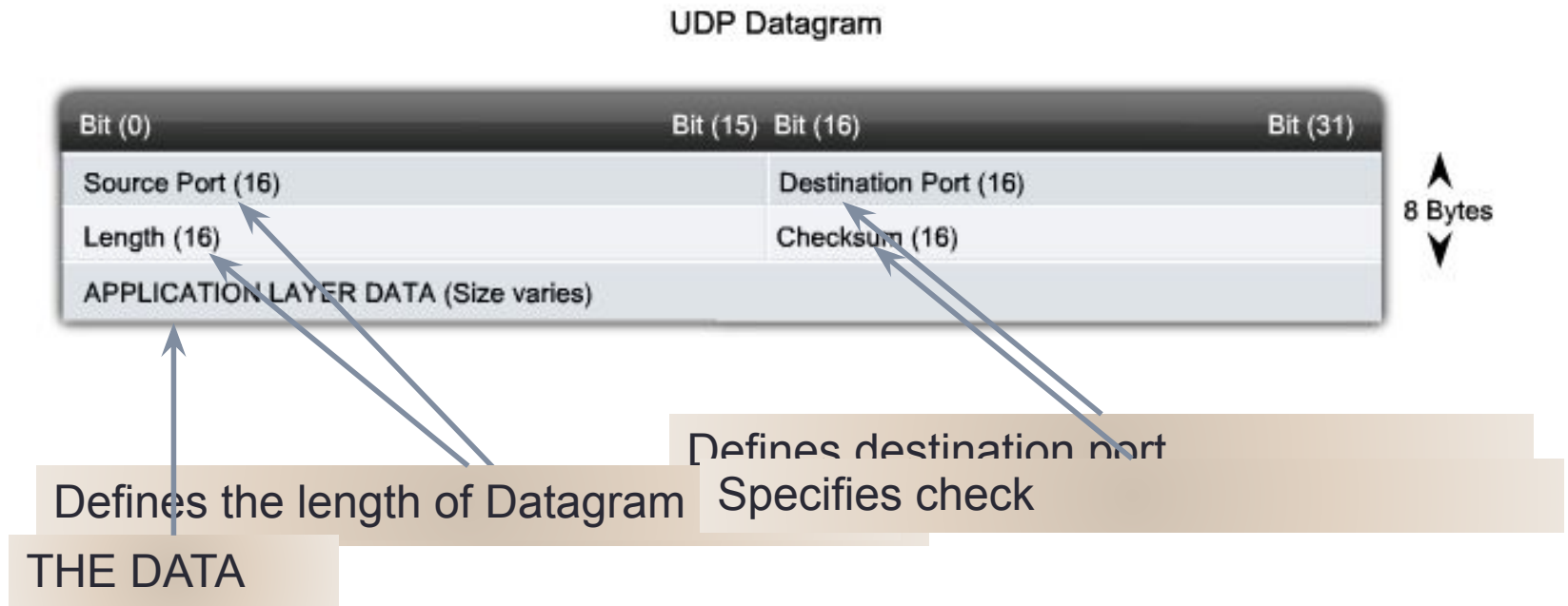
Defines the order of segments  
 Defines to which port data have to be  
 Defines expectation of next sequence



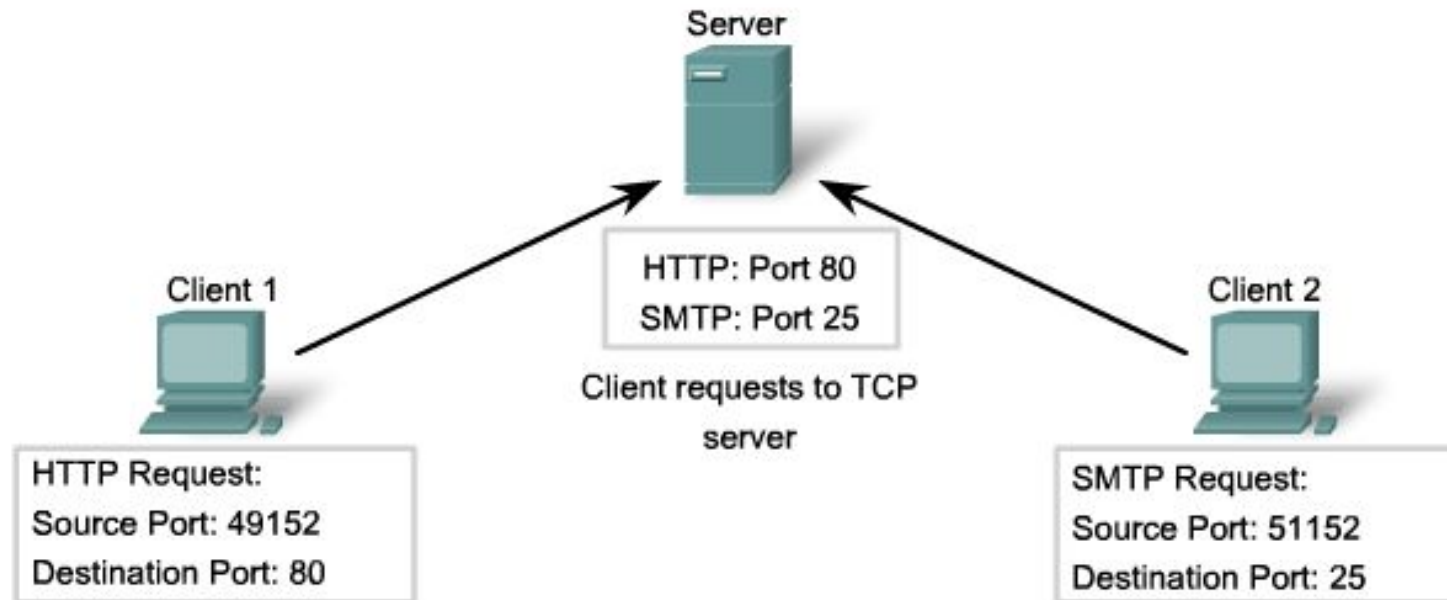
Specifies Reserved for Use in Window size defines number of bits  
 Used for error checking of Header  
 Used for defining that this Segment must be send in urgent

THE DATA

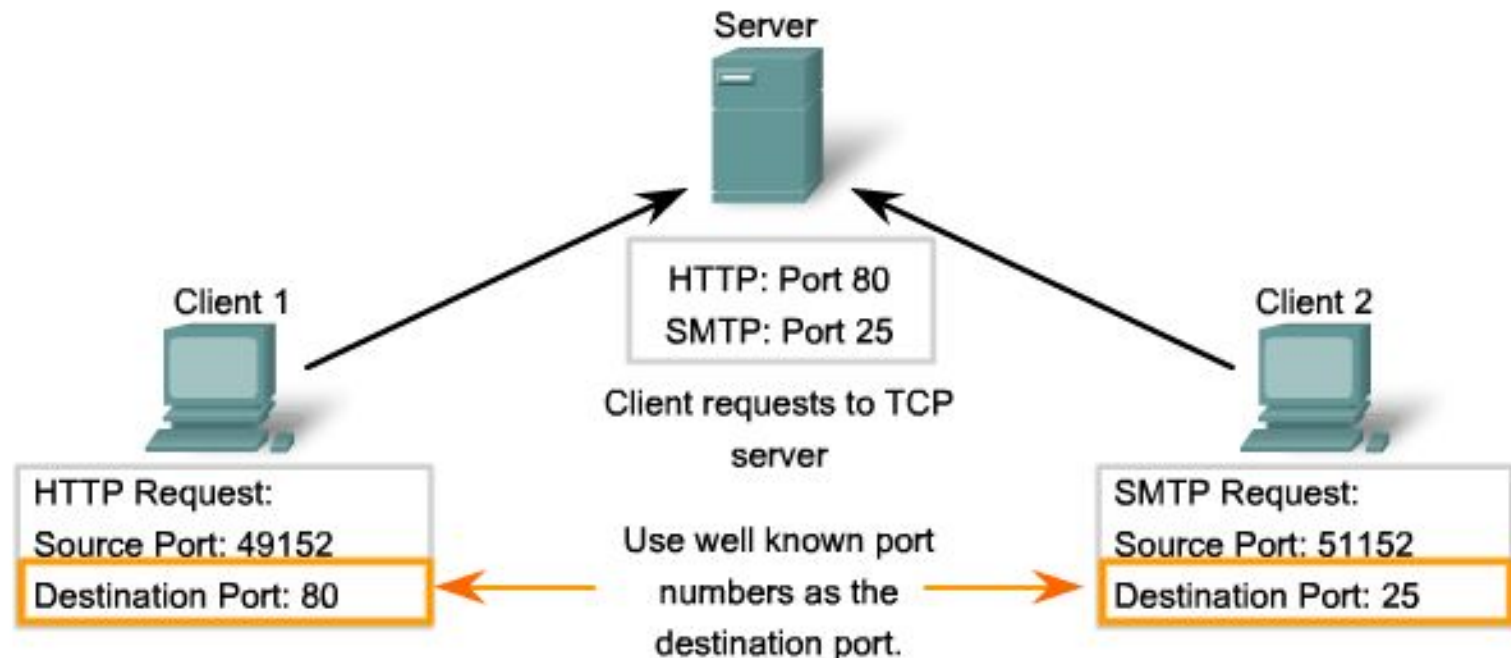
# UDP Header



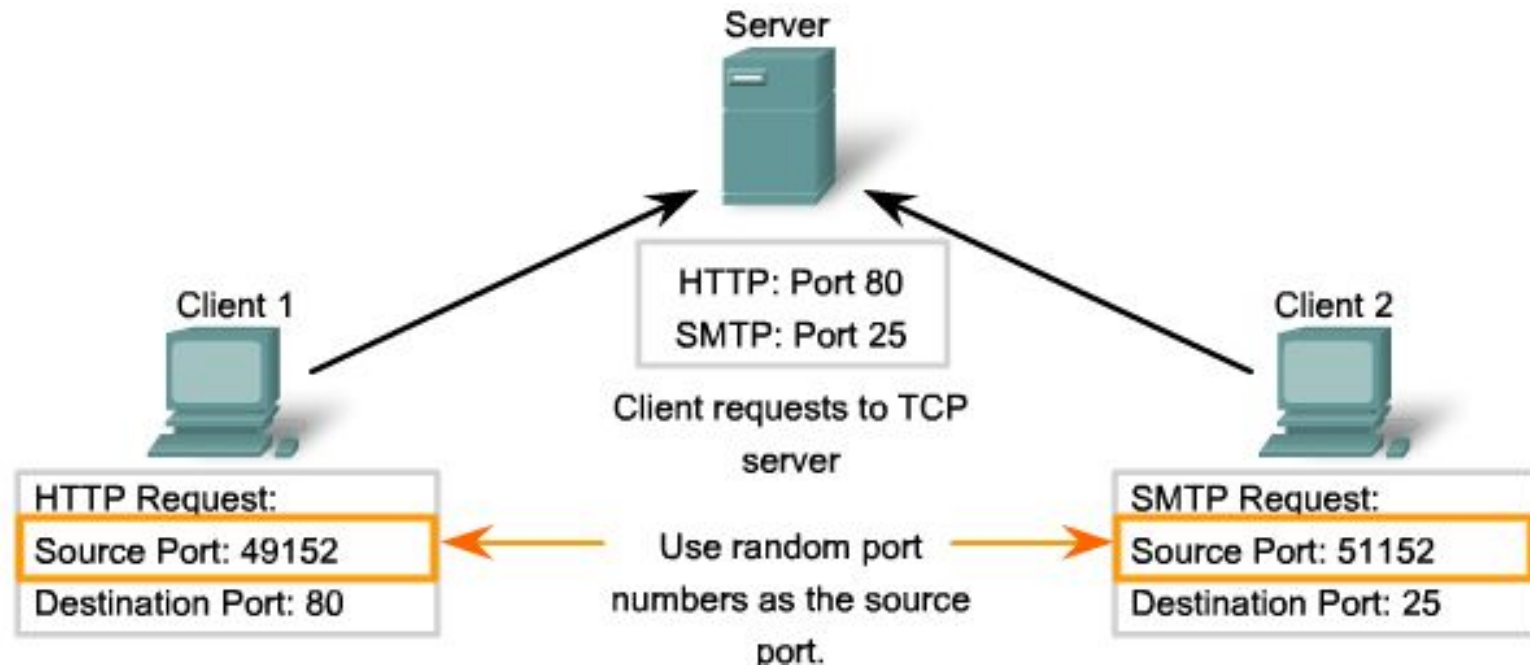
# TCP in process



# Clients Define Destination Ports

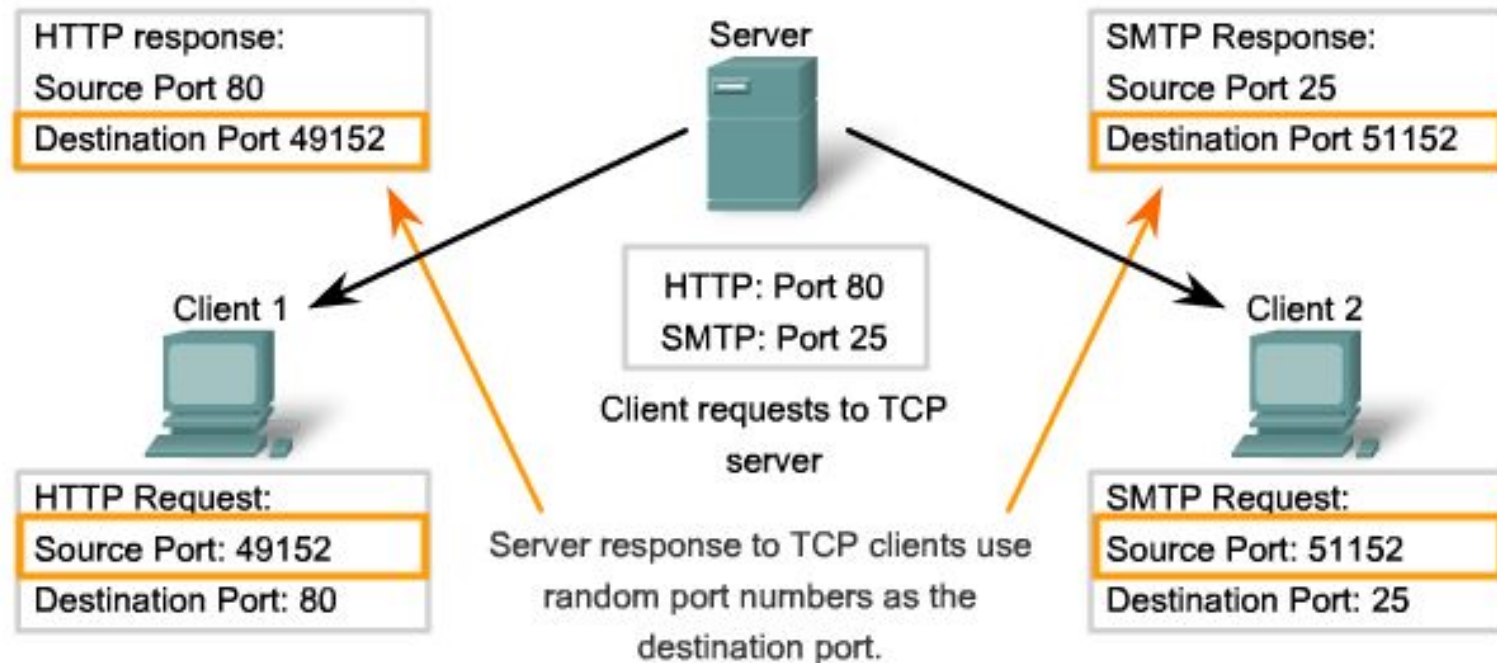


# Clients Define Source Ports

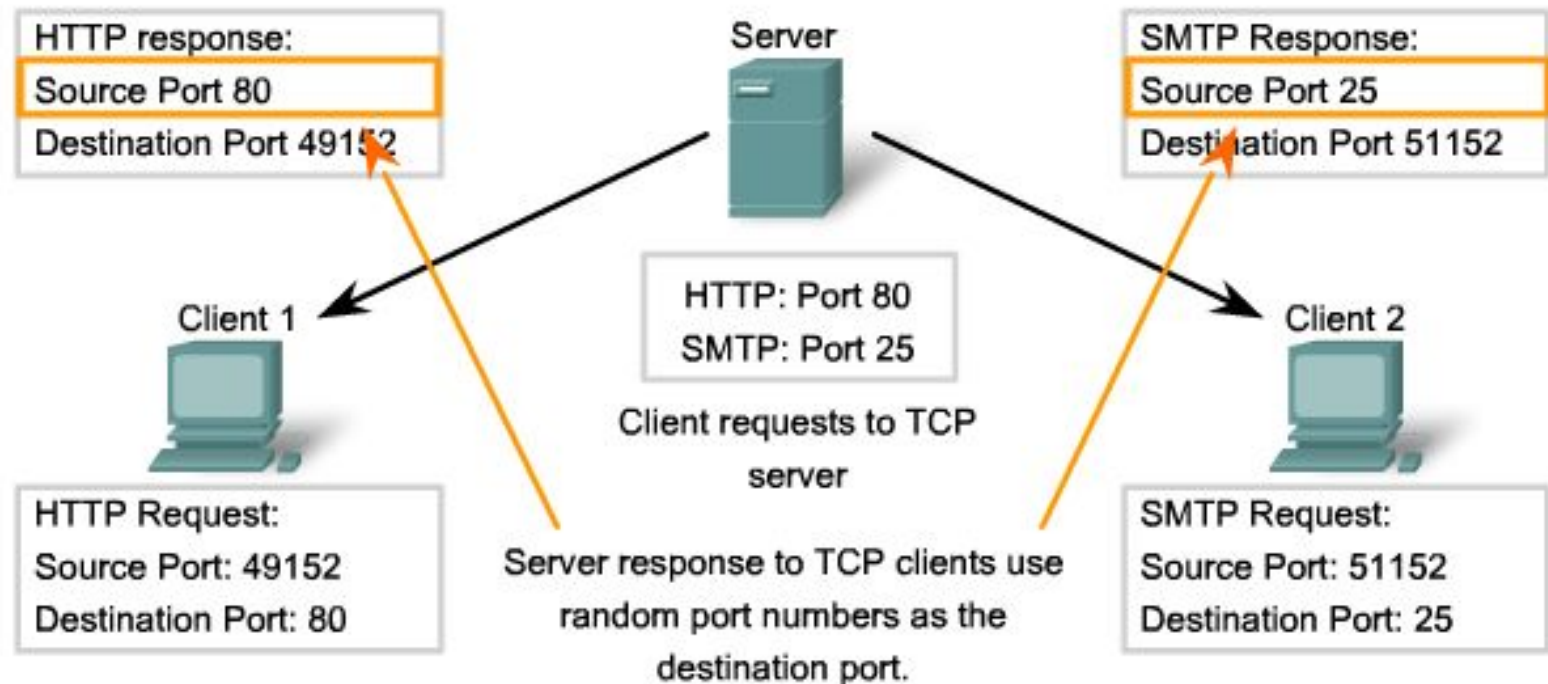




# Server Defines Destination Ports



# Server Defines Source Ports



# TCP CONNECTION ESTABLISHMENT AND TERMINATION

---

# Question

- Establishment?

# Answer



# Question

- How me make handshake?

# Answer



# Question

- How many way do we have in handshake process?



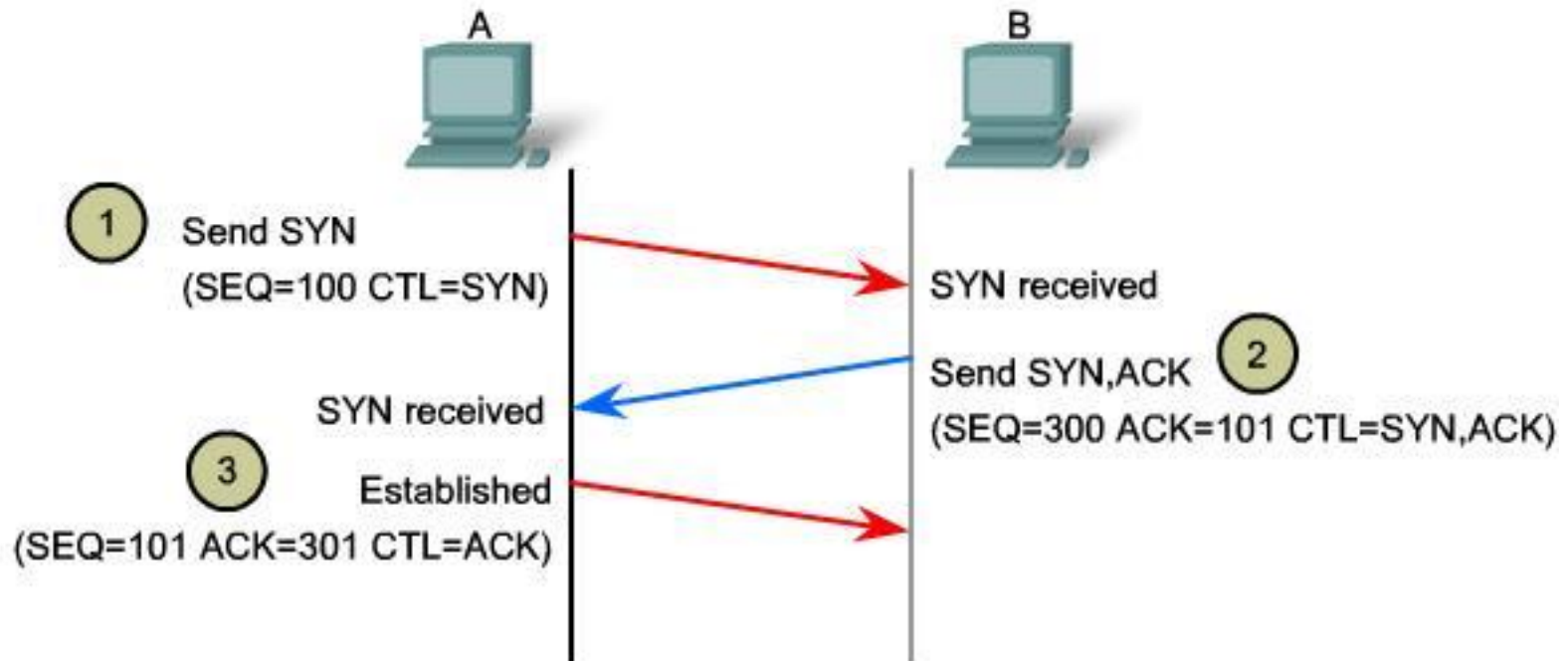
# Answer



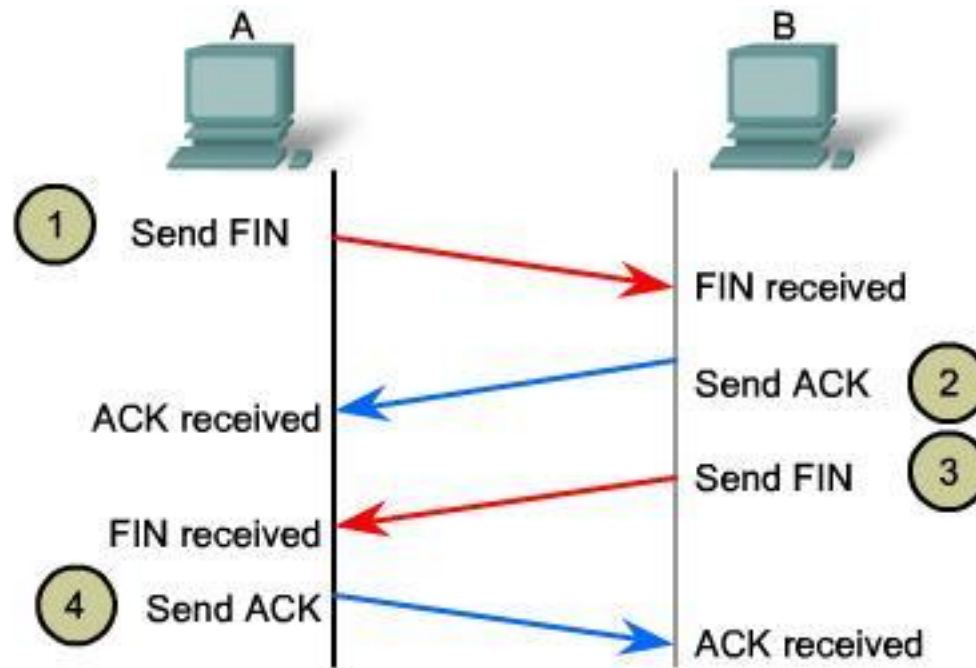
# TCP Have to do Establishment before it Starts Transmission

- This process is called 3 way handshake

# 3 way handshake



# TCP Session Termination



# Question

- How people say goodbye?

# Answer



# REORDERING SEGMENTS AFTER RECEIVING

---

# Question

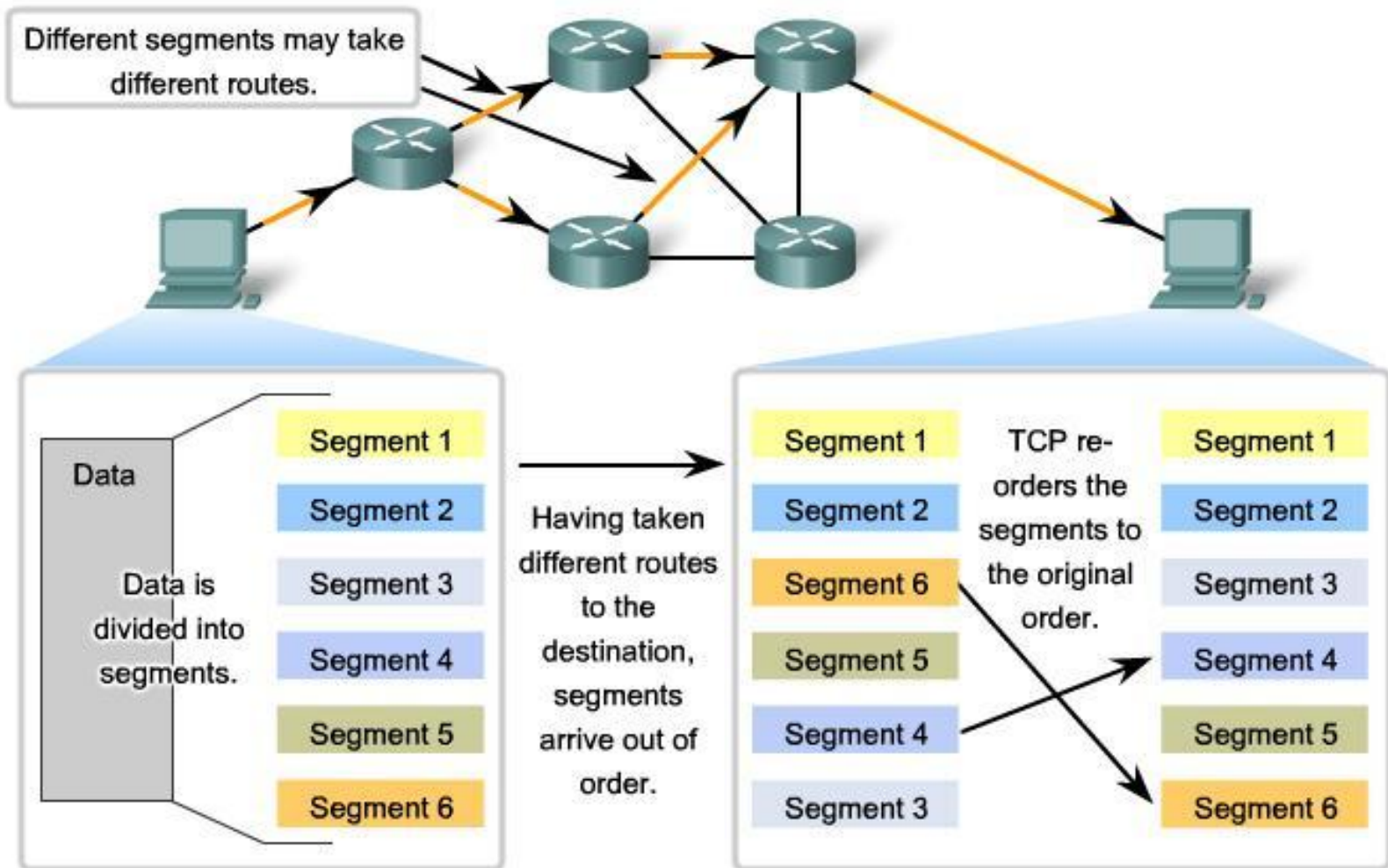
- Why it is needed?



# Answer

- Segments can be delivered in wrong order

# Re-ordering segments



# Acknowledgement

• is?



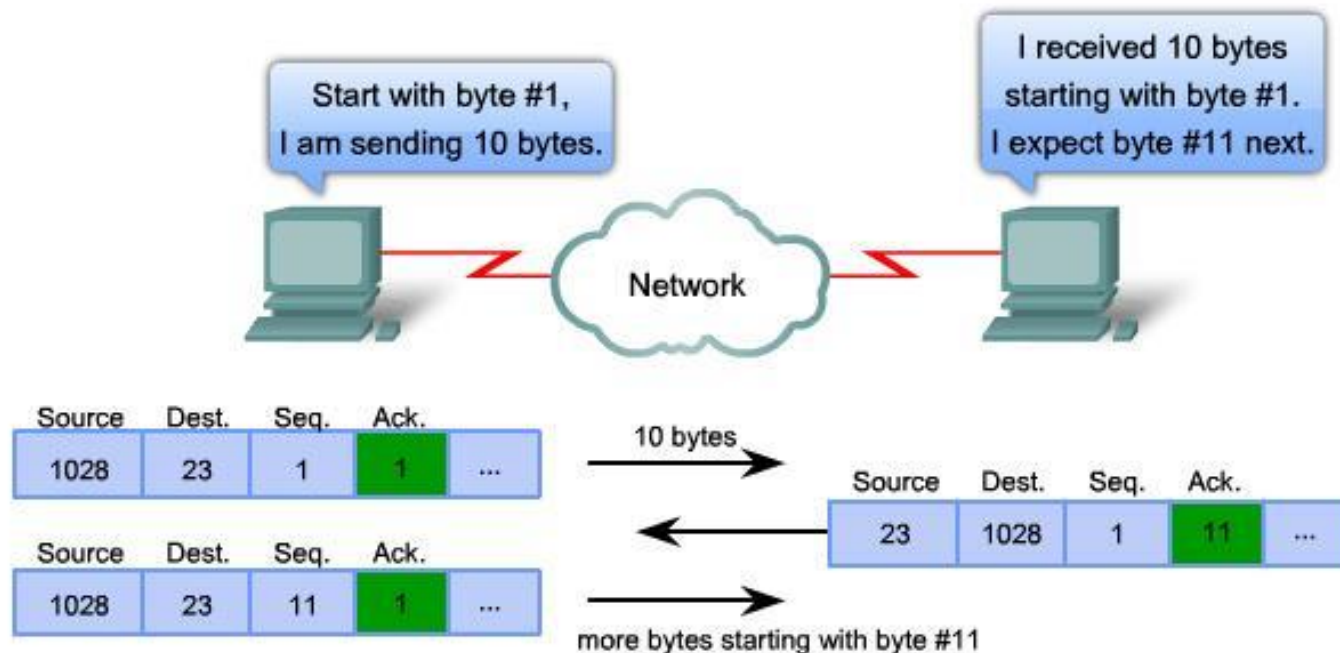
# Computer Acknowledgment

- Computers often sends acknowledgement to each other, to confirm that they have received data.

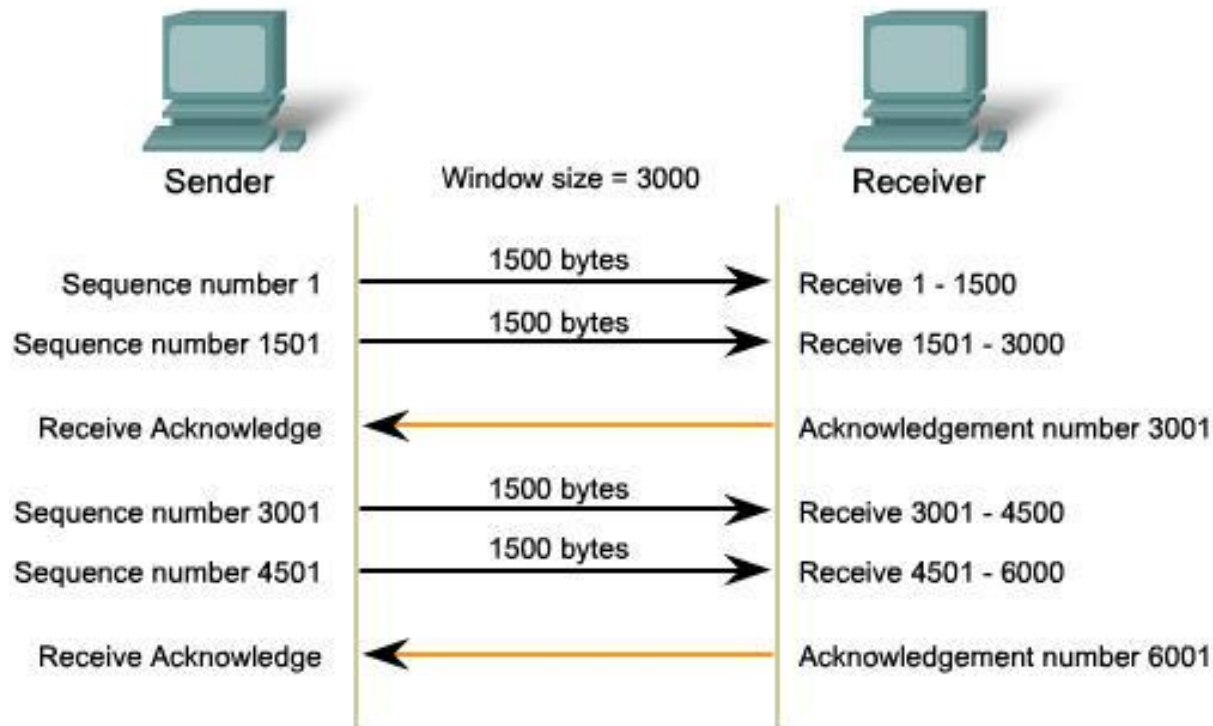
# Clients retransmit data if they don't get acknowledgement

Acknowledgement of TCP Segments

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



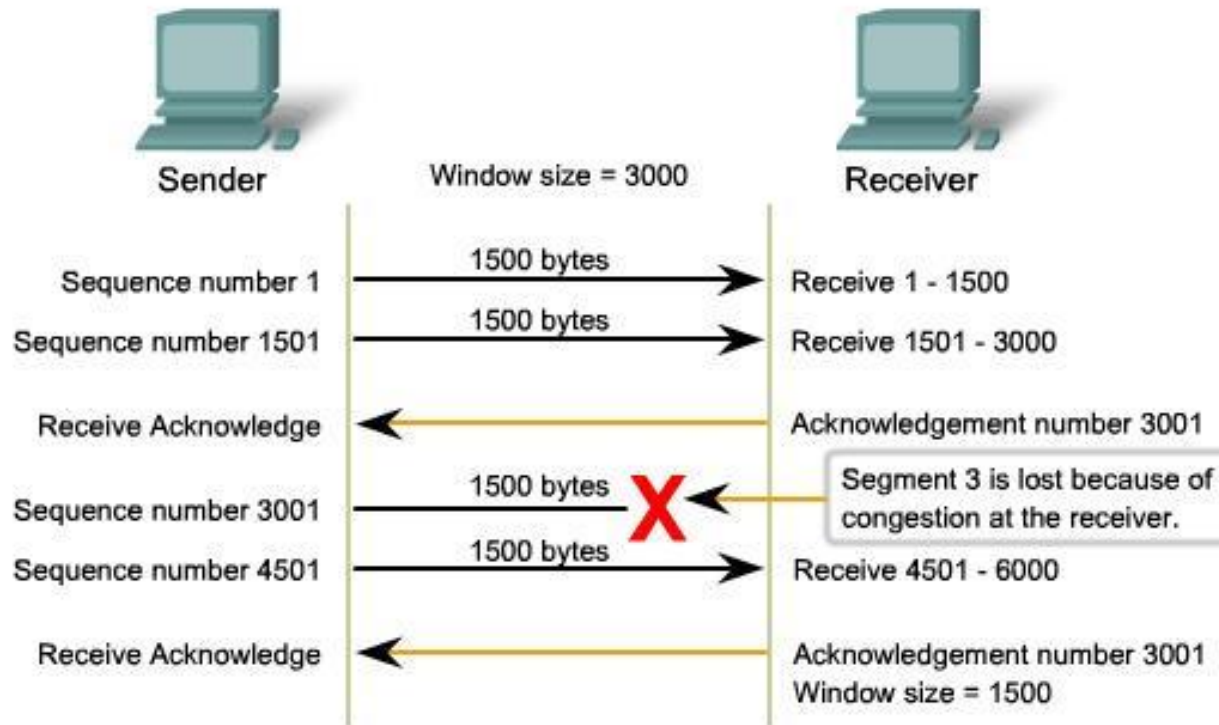
# Flow Control



The **window size** determines the number of bytes sent before an acknowledgment is expected.

The **acknowledgement** number is the number of the next expected byte.

# Flow Control with problems



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

# UDP – USER DATAGRAM PROTOCOL

---



# UDP

- UDP protocol is the protocol of OSI model's 4<sup>th</sup> layer (transport layer), is not reliable but very very very fast!!!

# QUESTIONS

---