OSI Transport Layer

Network Fundamentals – Chapter 4

Sandra Coleman, CCNA, CCAI

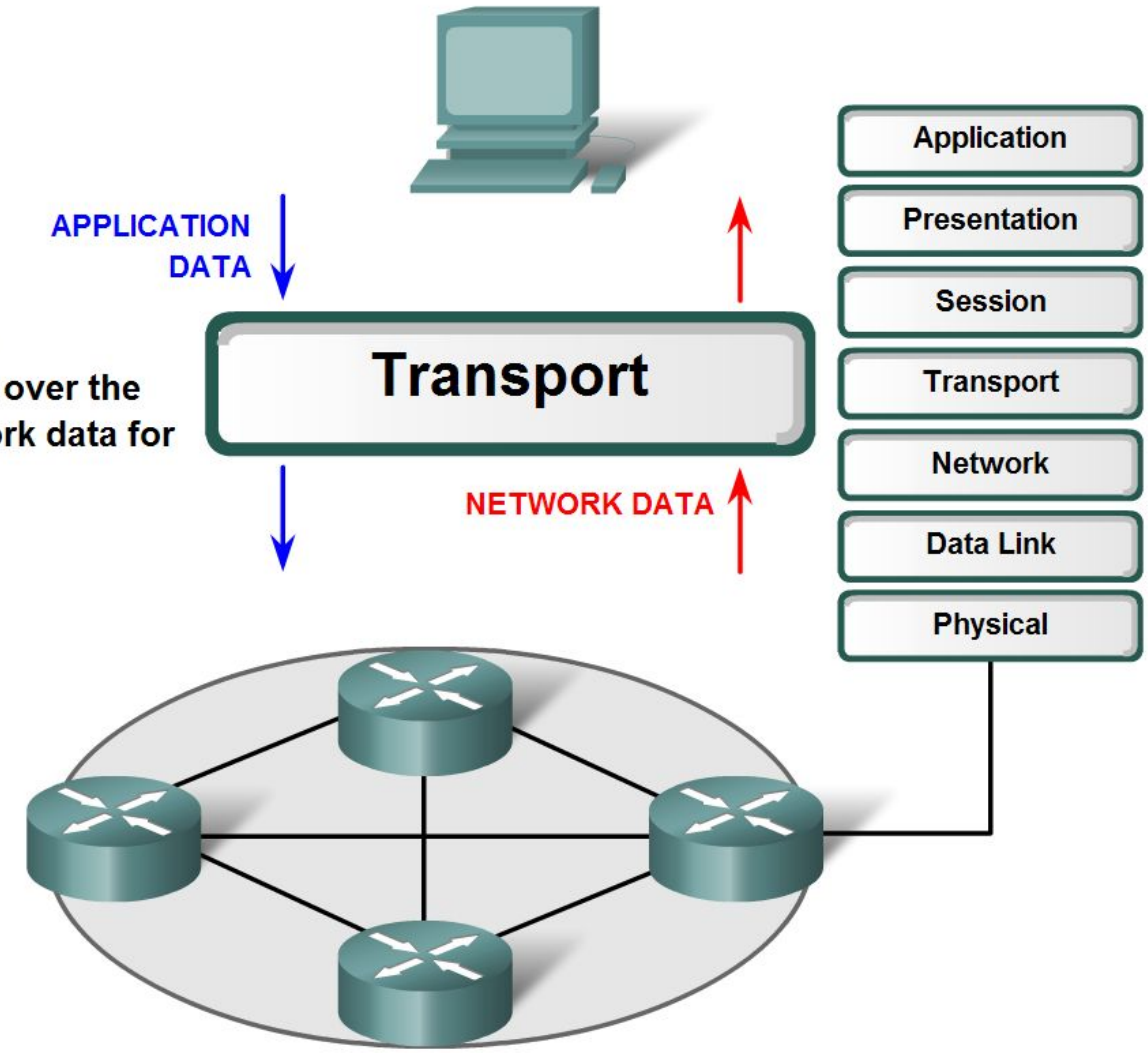
Cisco Networking Academy®
Mind Wide Open™

Objectives

- Explain the role of Transport Layer protocols and services in supporting communications across data networks.
- Analyze the application and operation of TCP mechanisms that support reliability.
- Analyze the application and operation of TCP mechanisms that support reassembly and manage data loss.
- Analyze the operation of UDP to support communicate between two processes on end devices.

Transport Layer Role and Services

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



Transport layer

- Purpose:

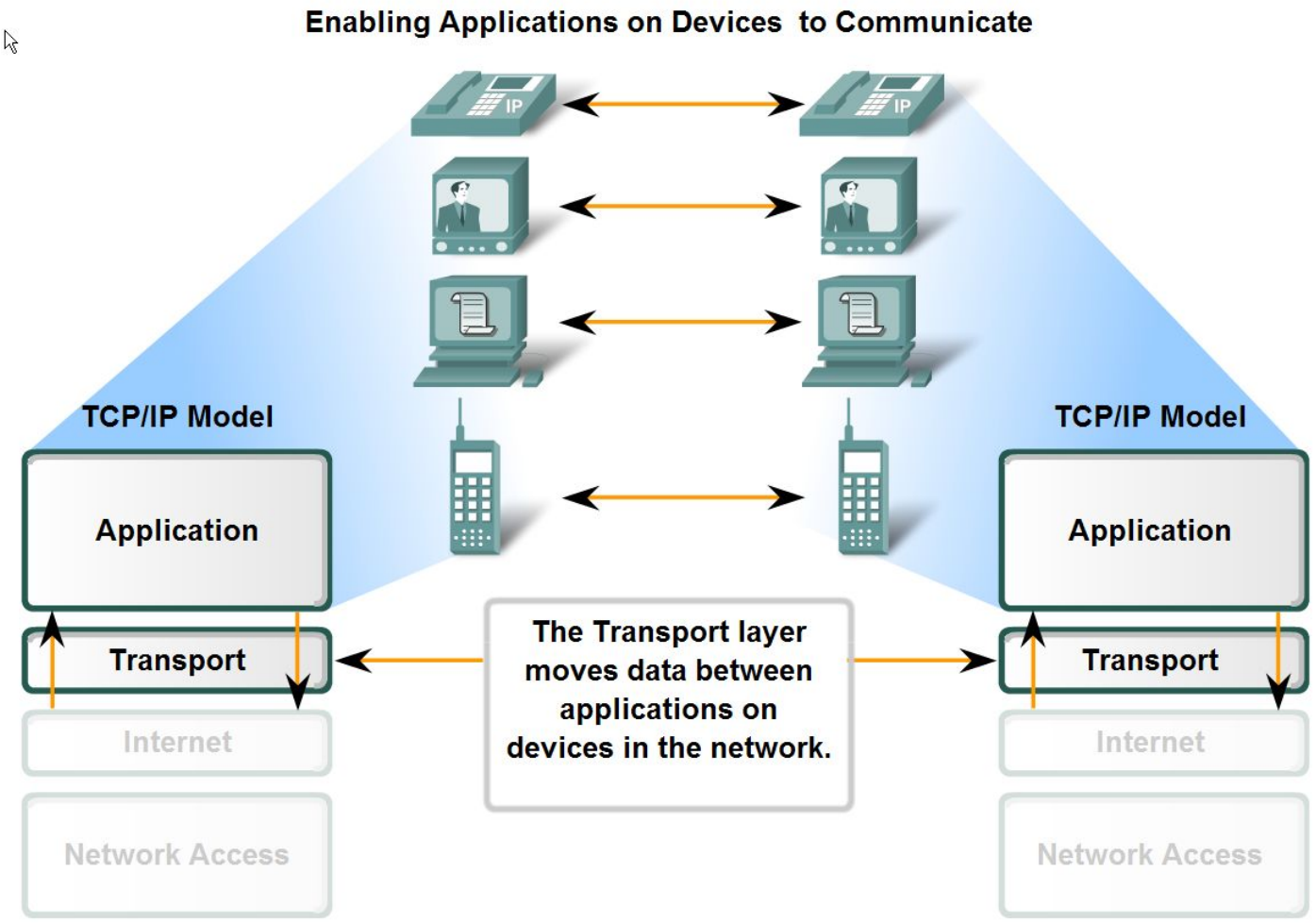
  Track individual communication between applications on source/destination hosts

  Segment data and manage each piece of data – prepares it to be sent across the network (network layer). Encapsulation is required on each piece of data including information that will allow that data to be tracked. WITHOUT SEGMENTATION, only ONE application would be able to receive data.

  Re-assemble segments back into streams of application data at the receiving host. Prepares it to be passed back to the application layer.

  Identify the different applications using port numbers. Each software process that needs to access the network is assigned a port # that is unique in that host. Indicates which application that piece of data is associated with.

Transport Layer Role and Services



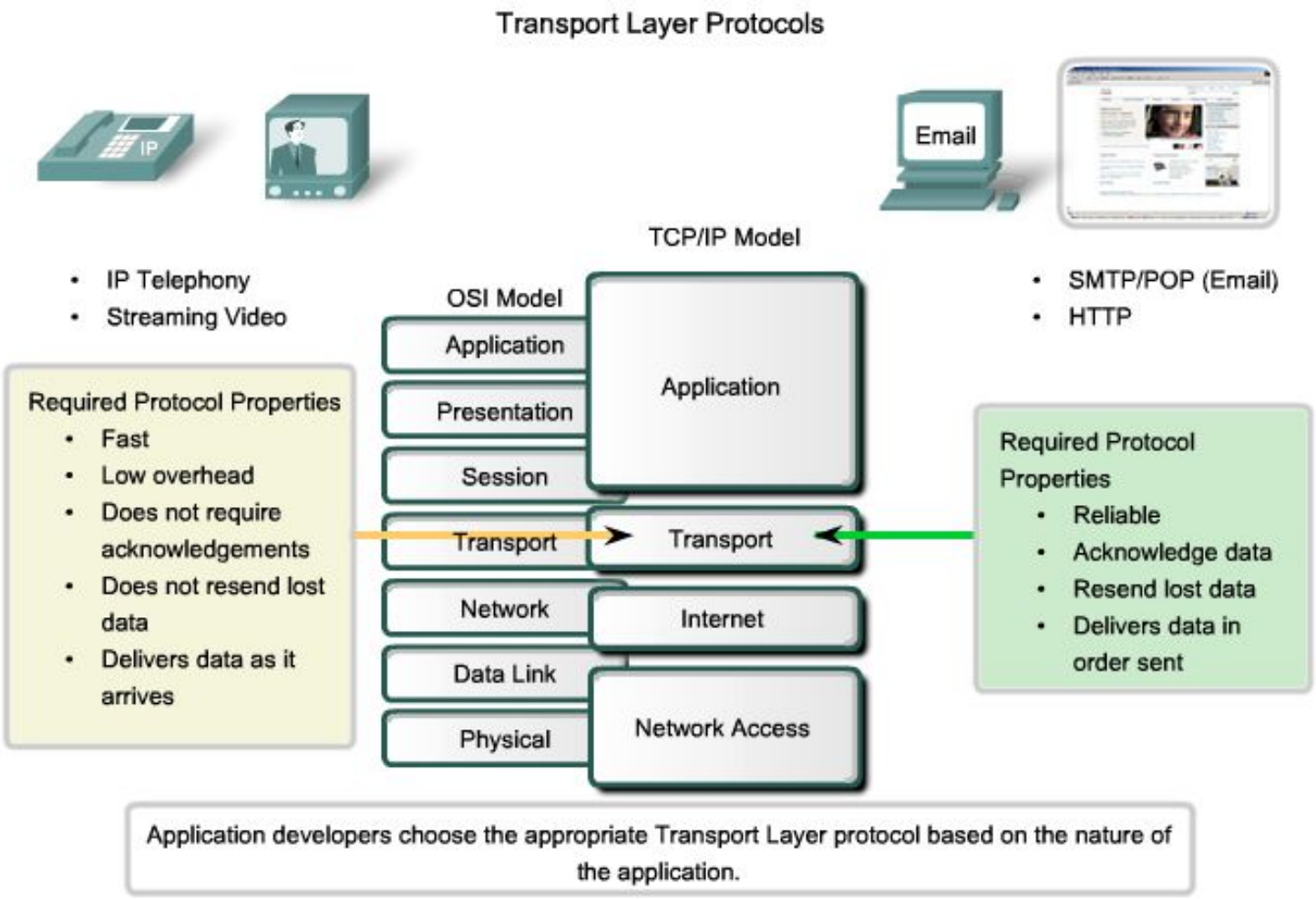
Controlling conversations

- Segmentation and reassembly – Transport layer divides application data into blocks of data that are the appropriate size. At the destination, the transport layer reassembles the data before sending it up to the application or service.
- Conversation multiplexing – Many applications or services might be running on each host. Each is assigned a port # so that the Transport layer can determine which application or service is associated with that data.
- TCP at the transport layer also provide (see next slide for details)
  - Connection-oriented conversations
  - Reliable/accurate delivery
  - Ordered data reconstruction
  - Flow control

Controlling Conversations (cont'd)

- Establish a session – connection-oriented (TCP) or connectionless (UDP)
- Reliable delivery – ensures that all pieces reach their destination by having the source device retransmit any data that is lost
- Same order delivery – numbering and sequencing segments ensures the transport layer segments are reassembled in the proper order
- Flow control – hosts have limited resources (memory, bandwidth, etc.) If these get over-taxed, transport layer can request the flow of data be slowed. Why do this? Prevent the receiver from being overwhelmed with data!

Reliable communication



TCP & UDP protocols

Web browsers

E-mail

File transfers

TCP and UDP Headers

TCP Segment

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)		
APPLICATION LAYER DATA (Size varies)		

20 Bytes

DNS

VoIP

Video streaming

UDP Datagram

Bit (0)	Bit (15) Bit (16)	Bit (31)
Source Port (16)	Destination Port (16)	
Length (16)	Checksum (16)	
APPLICATION LAYER DATA (Size varies)		

8 Bytes

TCP vs. UDP - characteristics

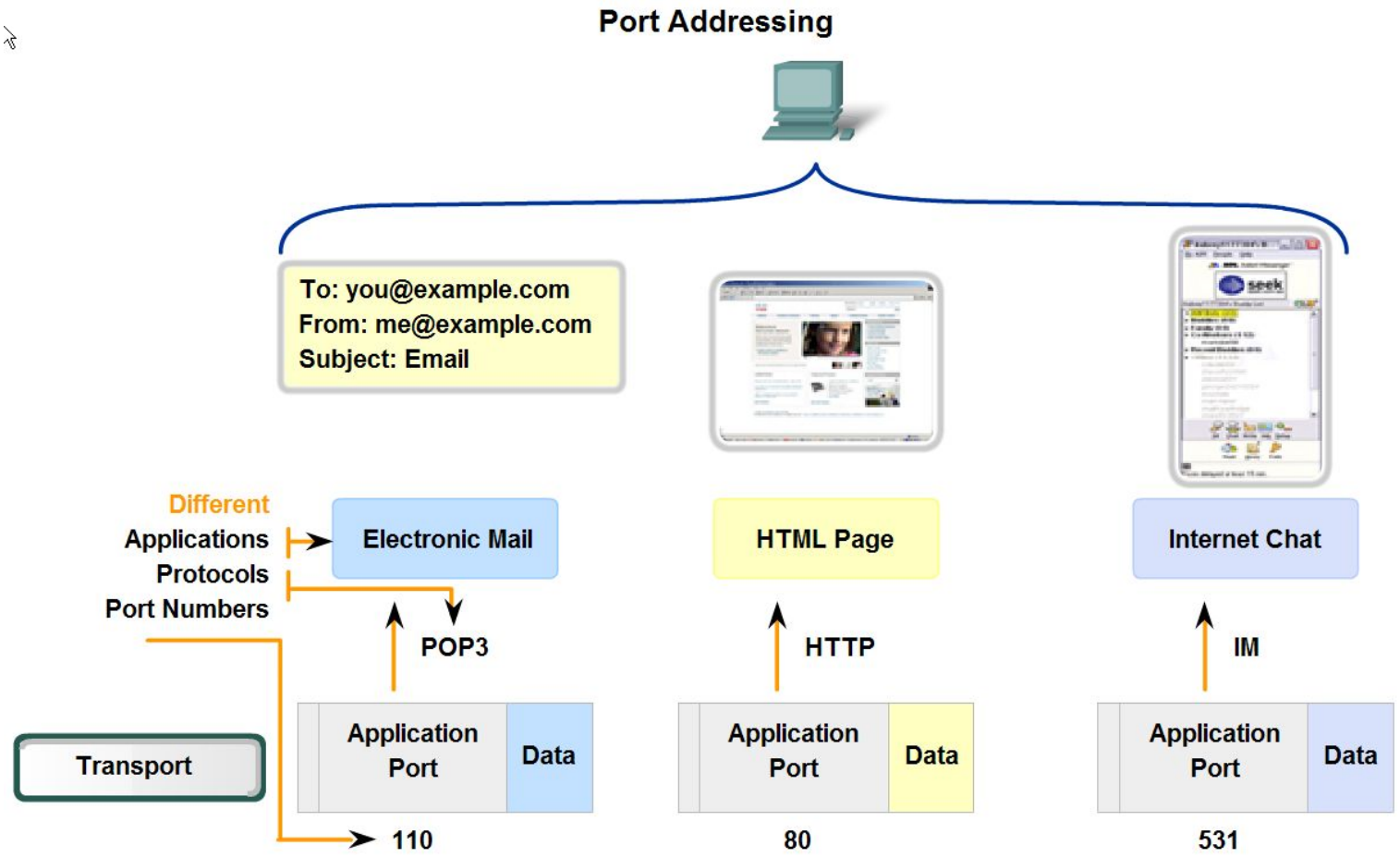
UDP – Connectionless
- Advantage – low overhead data delivery
- pieces – datagrams
- ‘Best Effort’ delivery
- Used by application that don’t require reliable delivery
- Minimal delays

TCP – connection-oriented
- More overhead
- Same order delivery
- Reliability
- flow-control

Source vs. Destination Port #'s

- Source ports –
  - Dynamically and randomly assigned by the originating device from port #'s > 1023
  - Must not conflict with other ports in use at the time
  - Acts as a 'return address' of sorts for the requesting application
- Destination port
  - Port # assigned to the service daemon running on the remote host
  - Must know which layer 4 protocol (TCP/UDP) and which application (port #)
  - Many common applications have default port # assignments
- Socket - combination of IP address and port #
  - 192.168.100.48:80 would be HTTP on that IP address

Transport Layer Role and Services



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

IANA & Port #'s

- IANA – Internet Assigned Numbers Authority – assign port #'s
- Port #'s

  0-1023 – Well know ports reserved for services & applications

  1024-49151 – registered ports assigned to user processes or applications. May be used as a dynamically selected source port

  49152-65535 – Dynamic or private ports (Ephemeral ports).

Port #'s (know these)

- TCP
  - 20&21 – FTP
  - 23 – Telnet
  - 25 – SMTP
  - 80 – HTTP
  - 110 – POP3
  - 443 - HTTPS
- UDP
  - 69 – TFTP
  - 520 – RIP
- TCP/UDP
  - 53 – DNS
  - 161 – SNMP

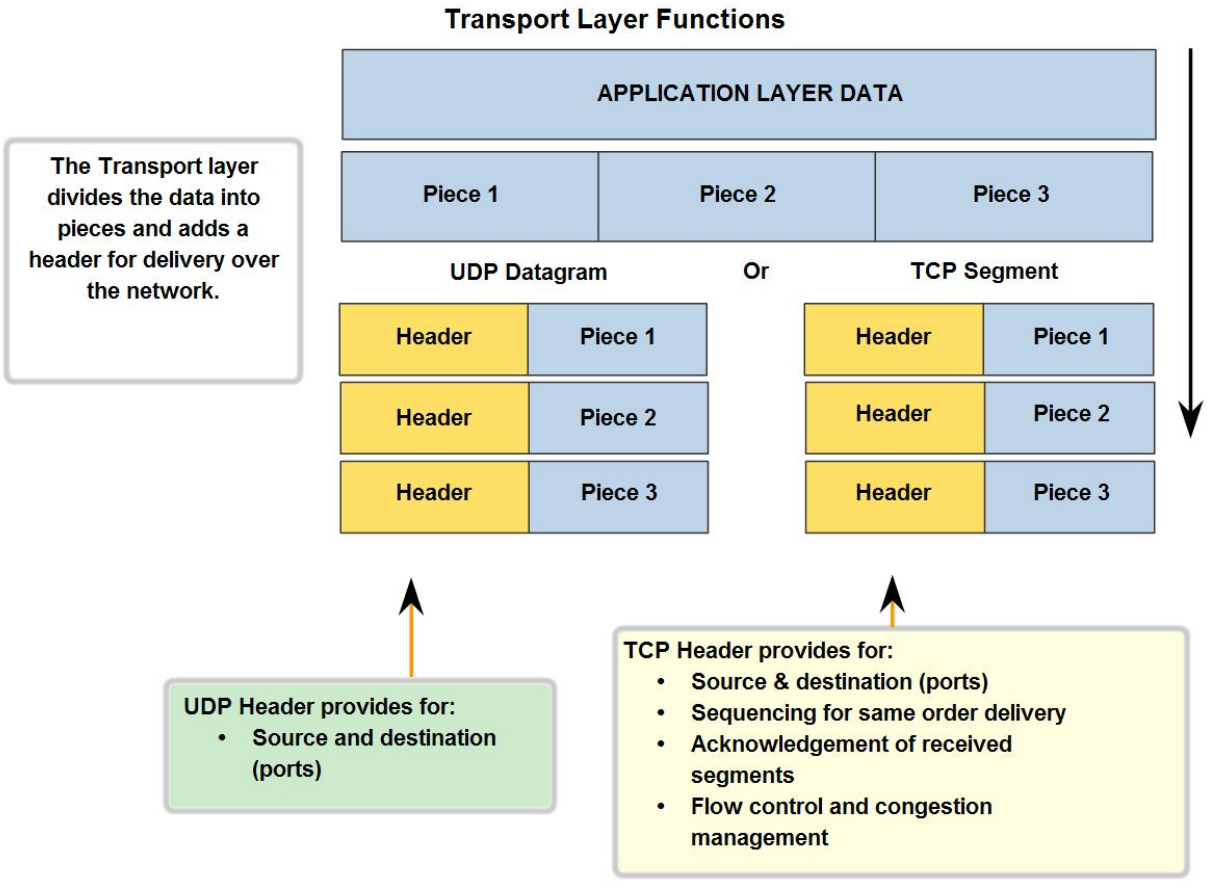
Netstat

- Utility that can be used to verify connections. Lists the protocol, the local address and port #, foreign address & port #, and the state of the connection

- Drop out to command line and try it
- >netstat
- >netstat –n (notice the port # after the : in the foreign ip address) (you will have to identify port # here on your test!)
- >netstat –e –s

Segmentation & Reassembly

- Dividing data into manageable pieces ensures data is transmitted within the limits of the media and can be multiplexed onto the media.

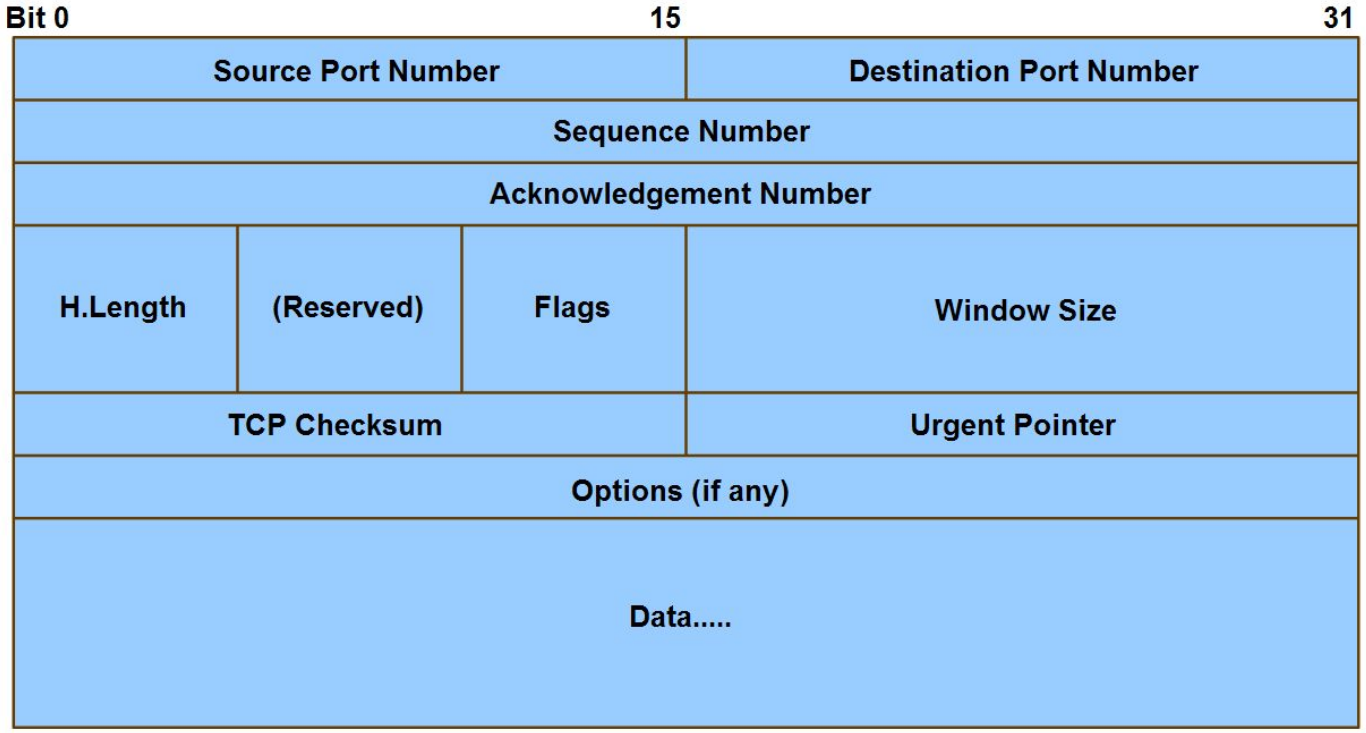


Segmentation & reassembly

- TCP & UDP do this differently
- TCP – sequence #'s are used for reassembly at the destination in the correct order. Data is ensured to be in the exact form the sender intended.
- UDP – not concerned with order or maintaining a connection. Generates less overhead which means faster data transfer. Applications that use UDP must tolerate the fact that data may not arrive in the order that it was sent. Does NOT require reliable delivery of packets.

TCP & Reliability

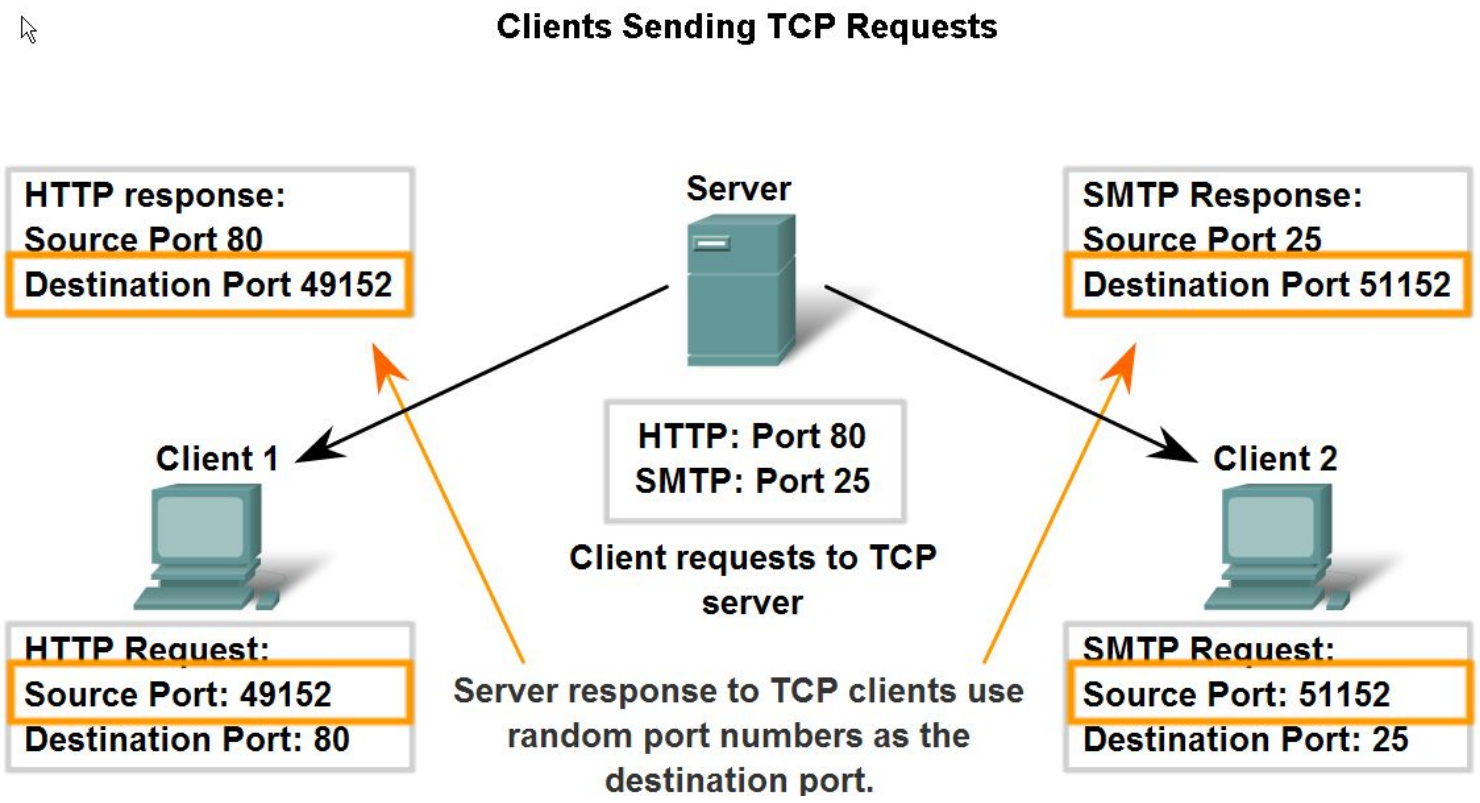
TCP Segment Header Fields



The fields of the TCP header enable TCP to provide connection-oriented, reliable data communications.

The key distinction between UDP and TCP is the reliability you get with TCP…discuss the fields.
Source/destination port #'s are on TCP and UDP Headers

TCP Server Processes



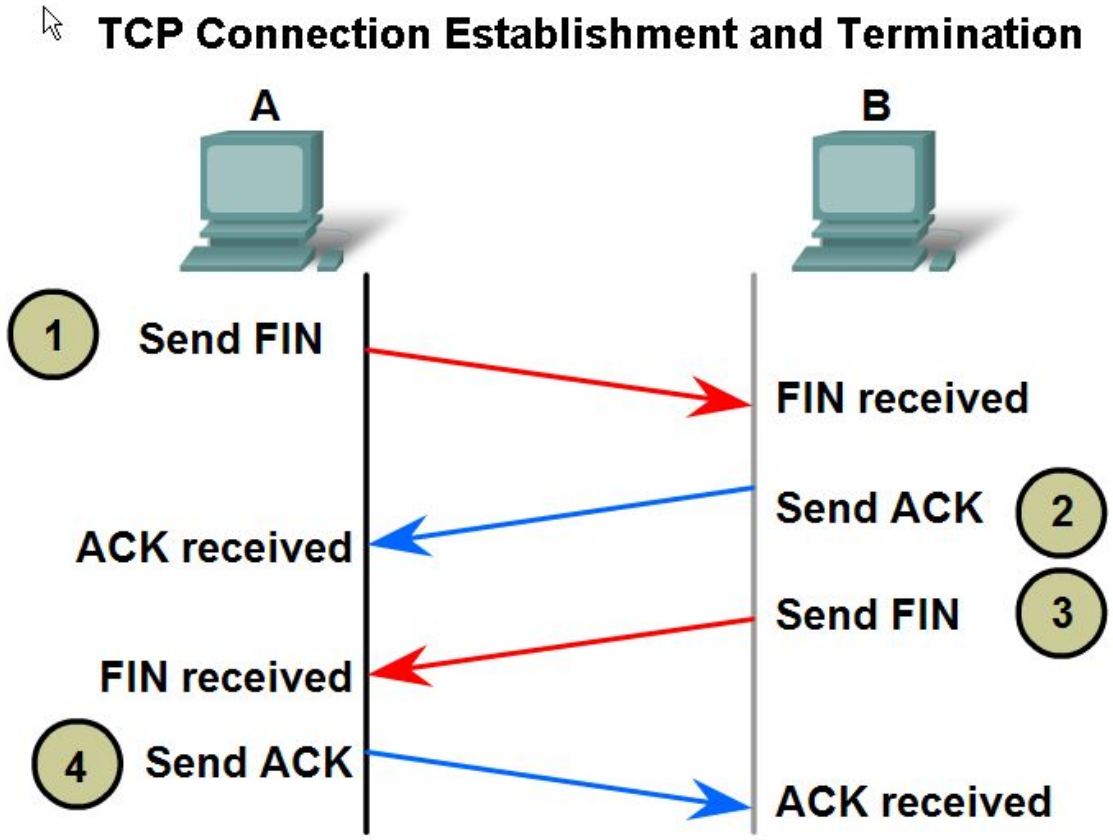
- An individual server can't have 2 services assigned to the same port # within the same transport layer services.

3-way Handshake

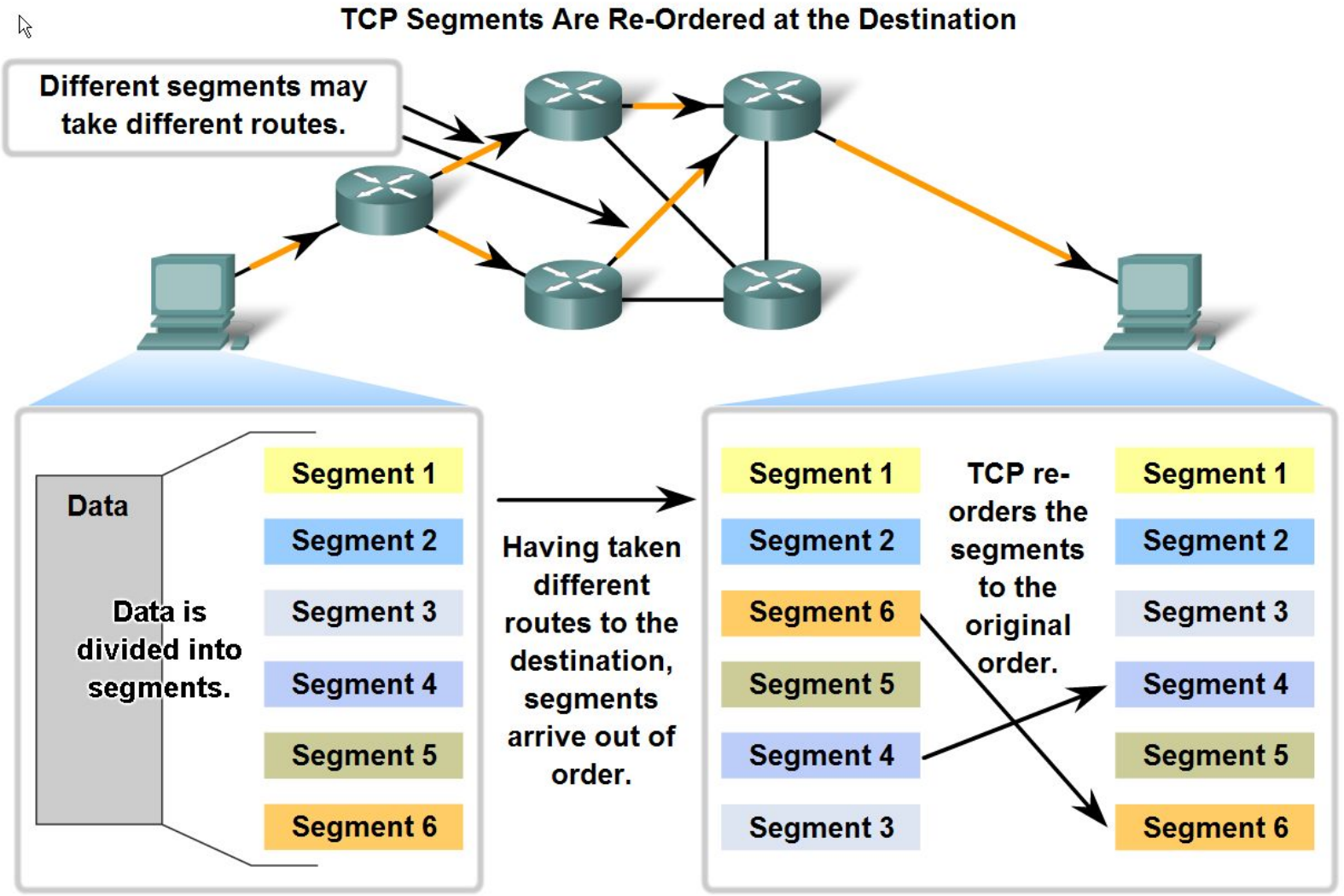
- Steps to establish a connection
  1) The sender sends an initial SEQ value (set by TCP) to begin communication!
  2) The receiver responds with an ACK value = to the SEQ value + 1. The ACK should always be the NEXT expected Byte.
  3) Sender responds with an ACK value = to SEQ value it received + 1.

1) See section 4.2.4 online for greater explanation!

3-way Handshake – Session Termination

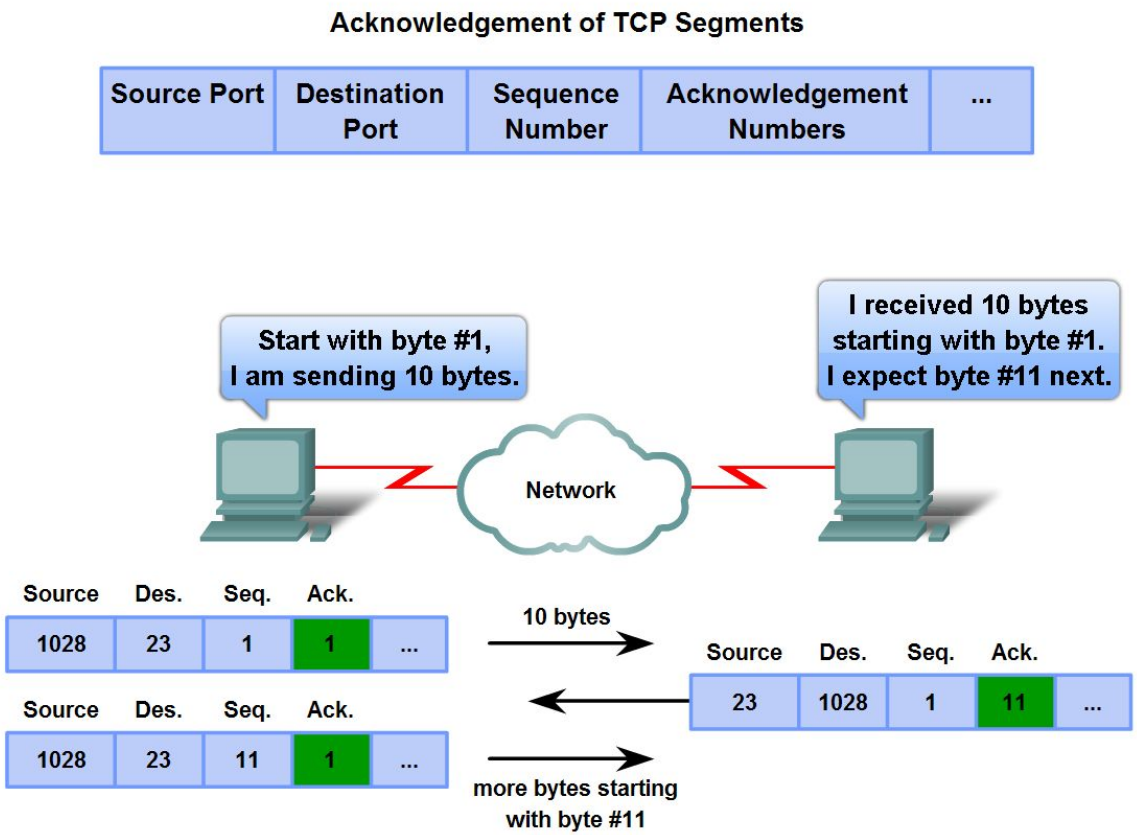


Managing TCP Sessions



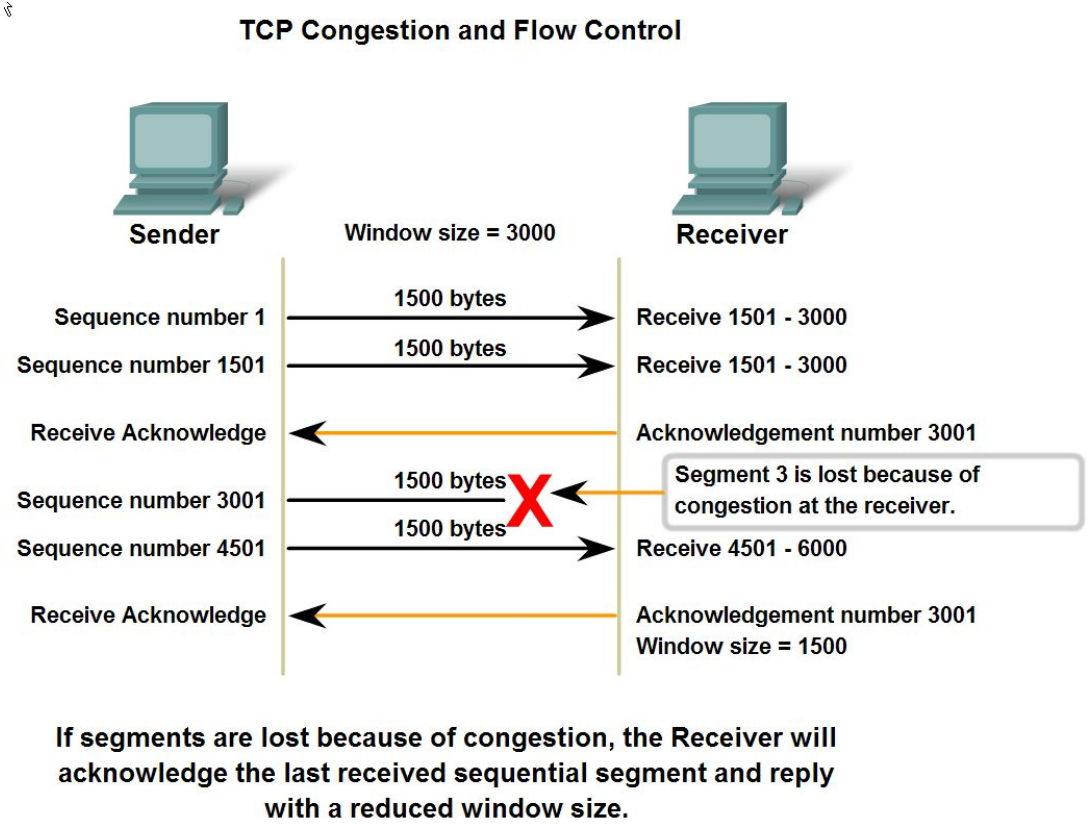
Managing TCP Sessions

If an acknowledgement isn't sent that data was received, the host will RESEND the data because it has reached a timeout.



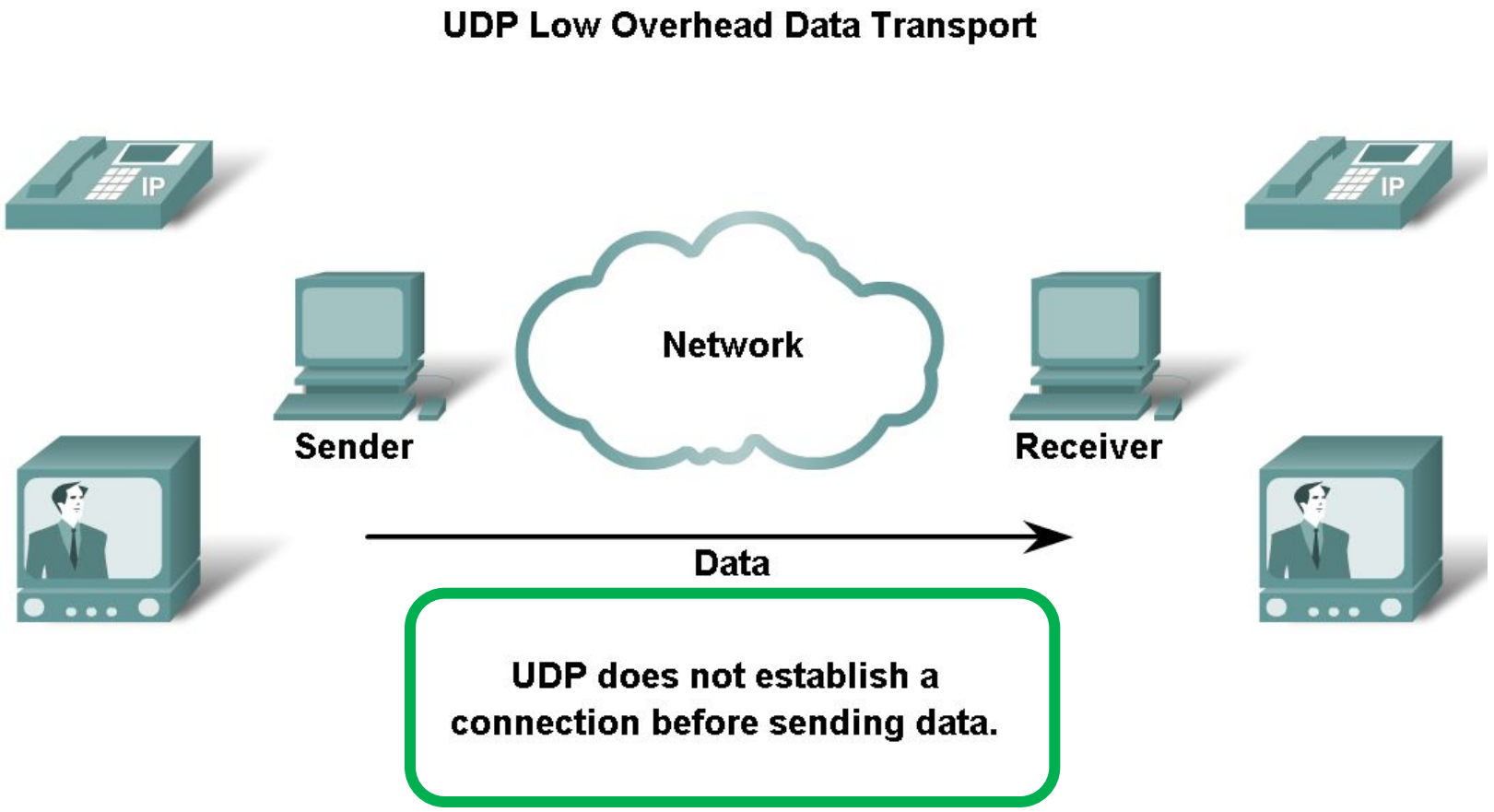
Managing TCP Sessions

- Window size – the amount of data a source can transmit before an ACK must be received. It enables the mgt. of lost data and flow control.



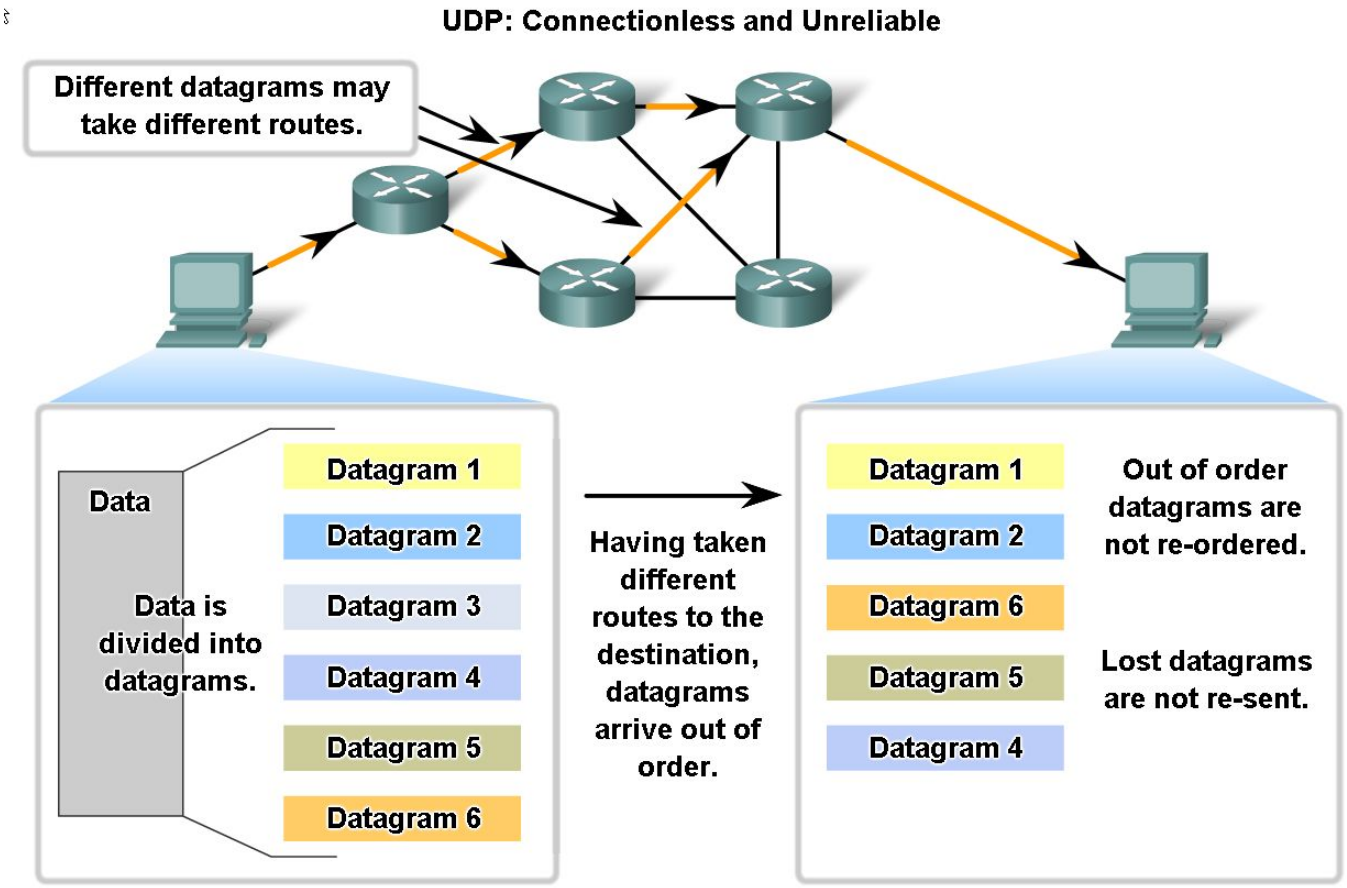
UDP Protocol

- Go over characteristics of UDP – used by DNS, SNMP, DHCP, RIP, TFTP, Online games, streaming video, etc.



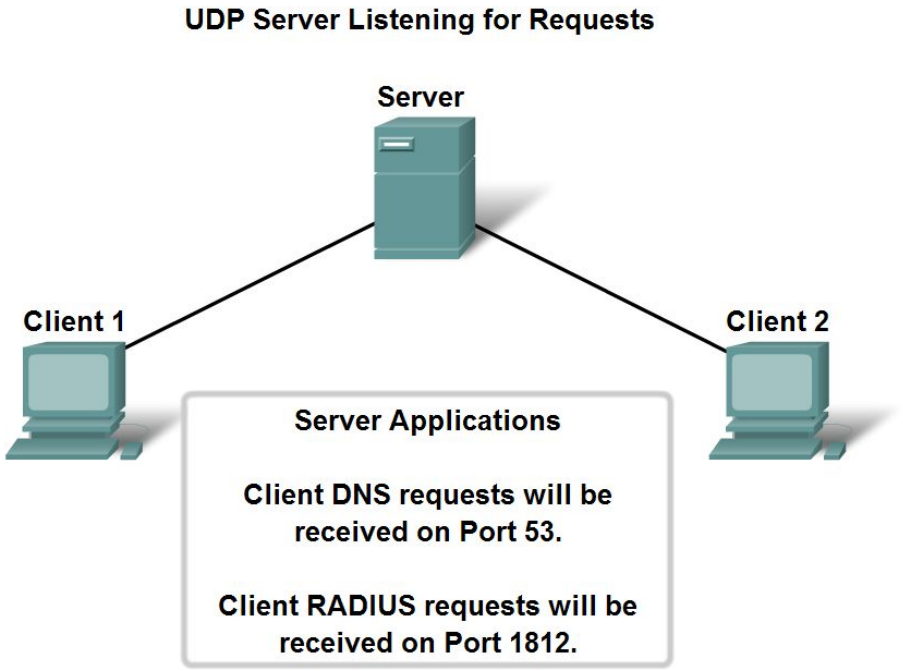
UDP Protocol

- UDP doesn't care if datagrams are out of order!



UDP Protocol

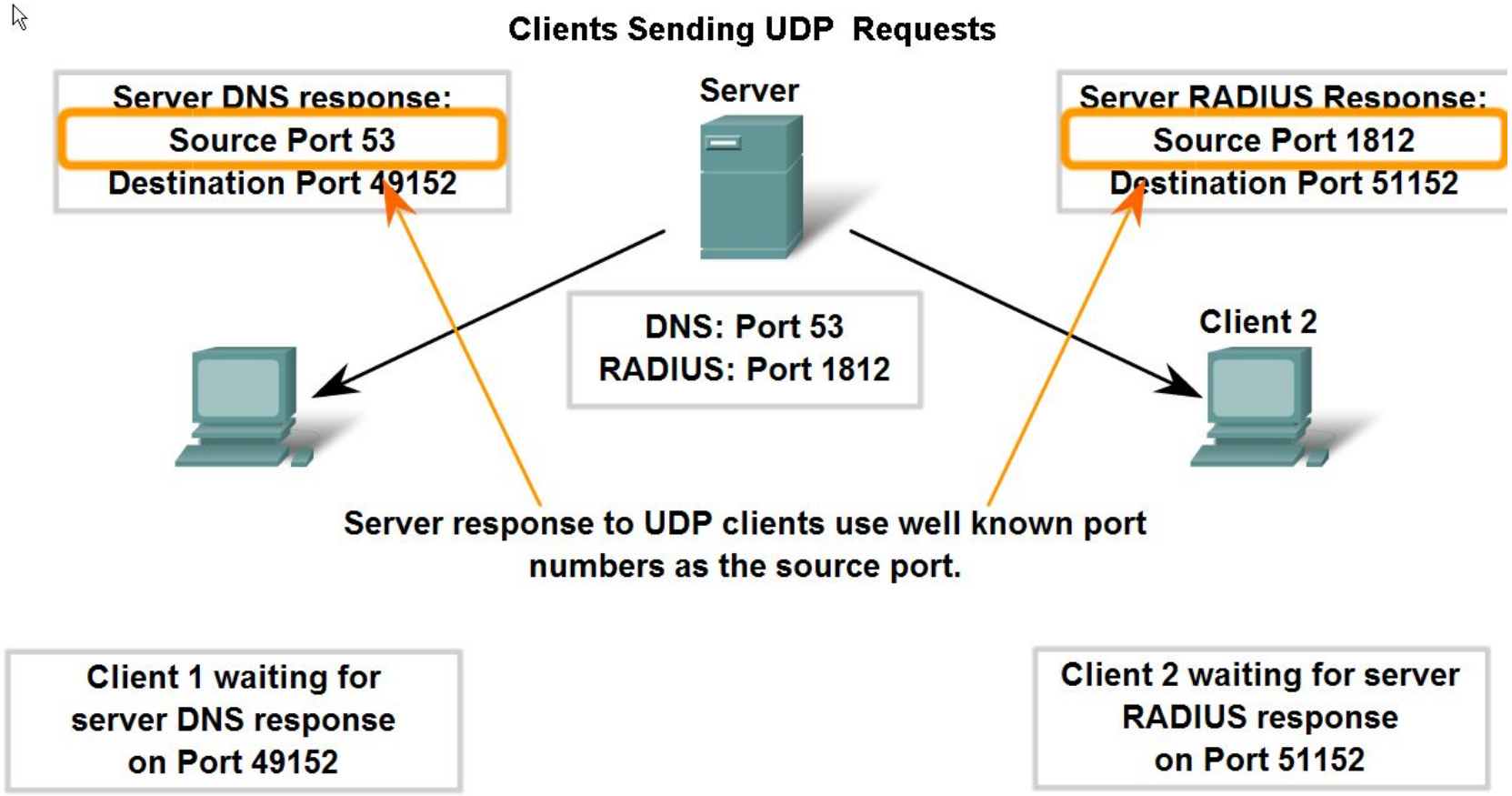
- Describe how servers use port numbers to identify a specified application layer process and direct segments to the proper service or application



Client requests to servers have well known ports numbers as the destination port.

UDP Protocol

Discuss



Summary

- Study Guide – Ch. 4 – NOW!
  - Pg. 91 - Matching
- Labs/Activities – None
- BREAK!
- Lecture on Ch. 3