CISCO CCNA1

CCNA Routing and Switching: Introduction to Networks

HOOFDSTUK 9

Transport Layer

DE HOGESCHOOL
MET HET NETWERK

Hogeschool PXL – Elfde-Liniestraat 24 – B-3500 Hasselt
www.pxl.be - www.pxl.be/facebook

CCNA1 - Overzicht

- **OSI model en de belangrijkste (LAN) protocollen.**
- Data Flow in een LAN
  (verklaring volgens het OSI model).
- IP en subnetting.
- Het toepassen en onderzoeken van bovenstaande 3 in Packettrace oefeningen.

Inhoud hoofdstuk 9

9. Transport Layer protocols

In dit hoofdstuk onderzoeken we de rol van de transport laag bij het inkapselen van applicatiegegevens voor gebruik door de netwerk laag. De transport laag omvat ook volgende functies. Zorgt ervoor dat, op een enkel apparaat, meerdere toepassingen op een netwerk kunnen communiceren op hetzelfde moment. Zorgt ervoor dat, indien nodig, alle data betrouwbaar en in juiste volgorde ontvangen wordt door de juiste toepassing. Zorg voor een foutafhandeling-mechanismen. Hoofdstuk 9 beschrijft de transport layer en de transport-layer-protocols namelijk UDP en TCP.

Doelstellingen:

- Hoe wordt een sessie opgebouwd
- Wat is TCP & UDP en wat is het verschil?
- Header informatie (UDP en TCP)
- Toepassingen TCP/UDP
- Gebruik en doel van poortnummers
- Veel gebruikte (bekende) poortnummers
- TCP handshake

Activity & PT:

- 9.1.2.10 Compare TCP and UDP characteristics
- 9.2.1.7 TCP connection and termination process
- 9.2.4.4 TCP UDP of beide?
- 9.3.1.2 PT TCP and UDP communications

Leertip:

Gebruik de PT oefening als herhaling van het hoofdstuk. Als je alles begrijpt wat er in de PT oefening besproken wordt, beheers je het hoofdstuk.

Cisco Networking Academy
Mind Wide Open

Chapter 9:

Transport Layer

Introduction to Networks v5.1

Chapter Outline

9.0 Introduction

9.1 Transport Layer Protocols

9.2 TCP and UDP

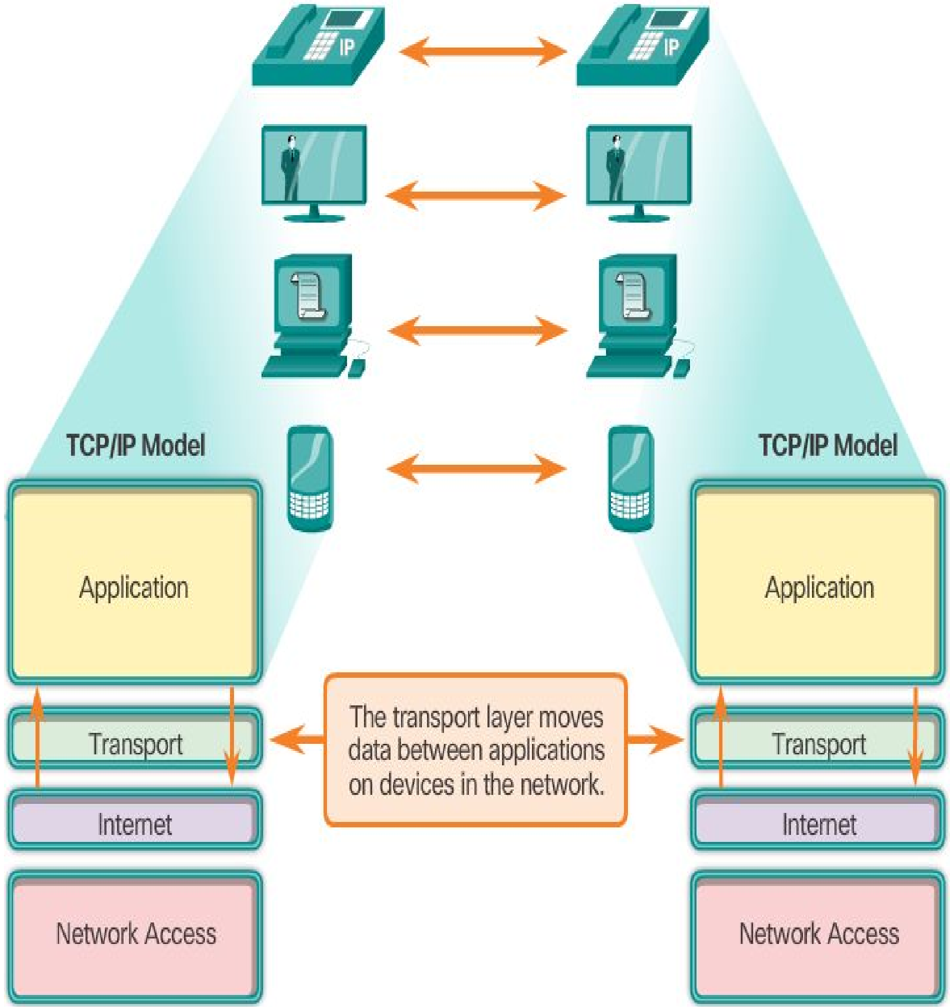
9.3 Summary

Section 9.1:
Transport Layer Protocols

9.1.1 Transportation of Data
9.1.2 TCP and UDP Overview

9.1.1.1 Role of the Transport Layer

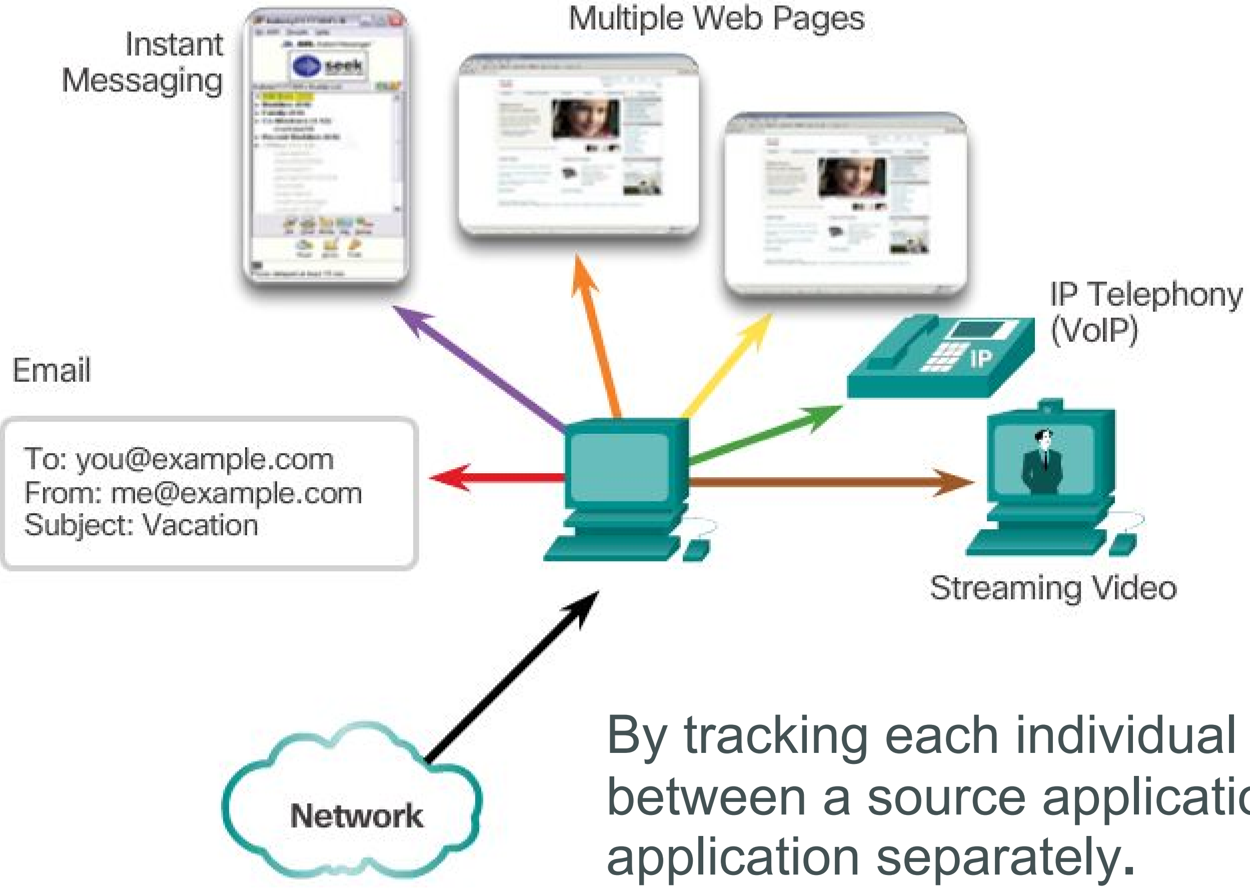
establishing a temporary communication **session** between **two applications** and delivering data between them.



The transport layer moves data between applications on devices in the network.

9.1.1.2 Transport Layer Responsibilities (1/3)

Track Individual Conversations

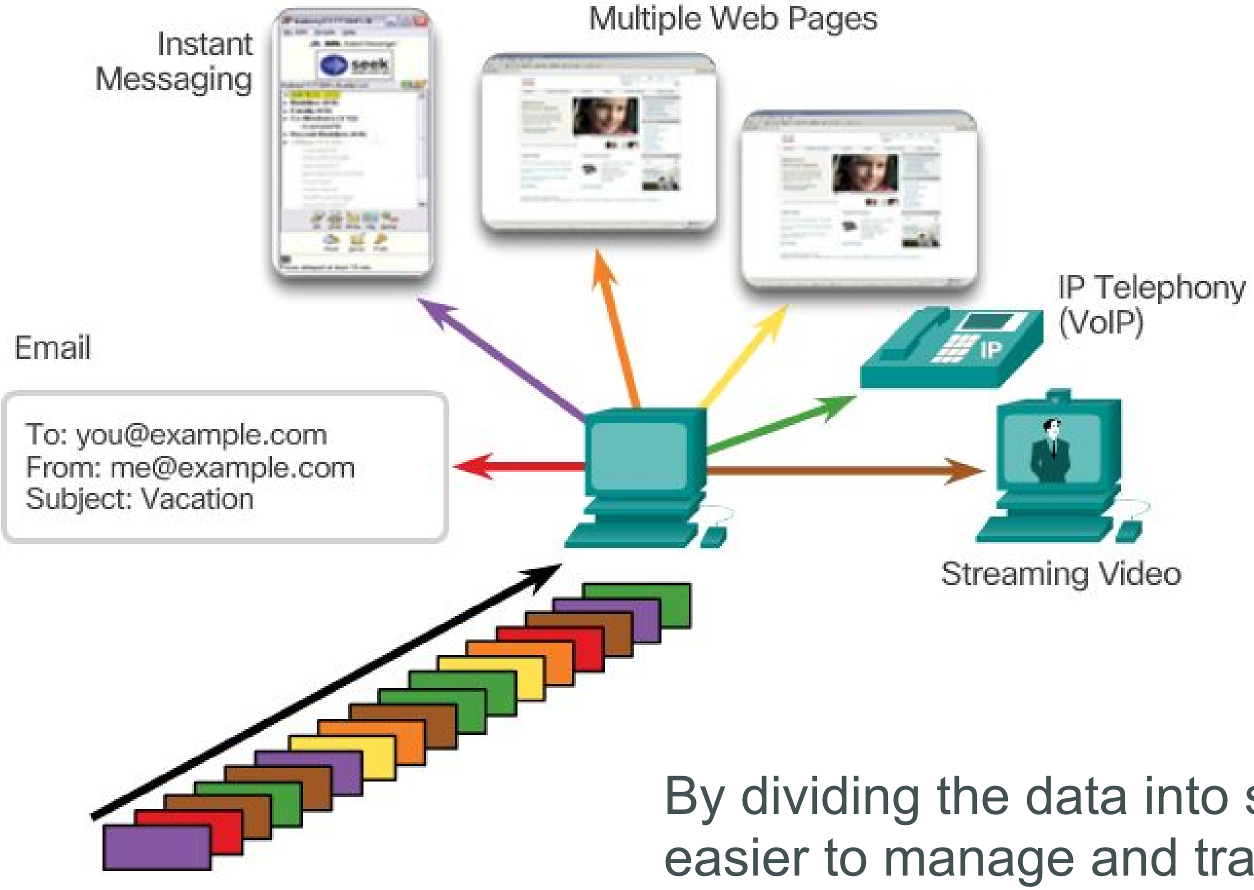


To: you@example.com
From: me@example.com
Subject: Vacation

By tracking each individual conversation flowing between a source application and a destination application separately.

9.1.1.2 Transport Layer Responsibilities (2/3)

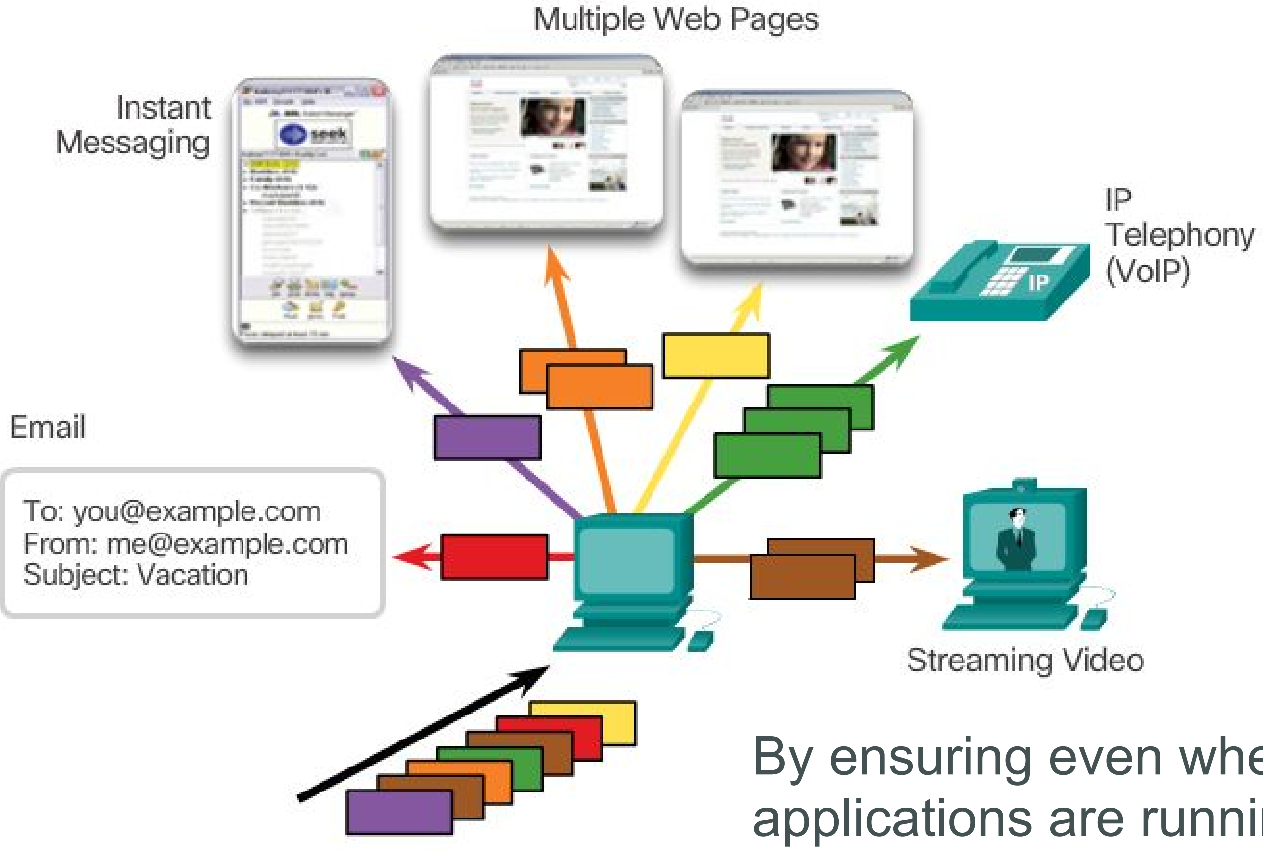
Segment Data and Reassemble Segments



By dividing the data into segments that are easier to manage and transport.

9.1.1.2 Transport Layer Responsibilities (3/3)

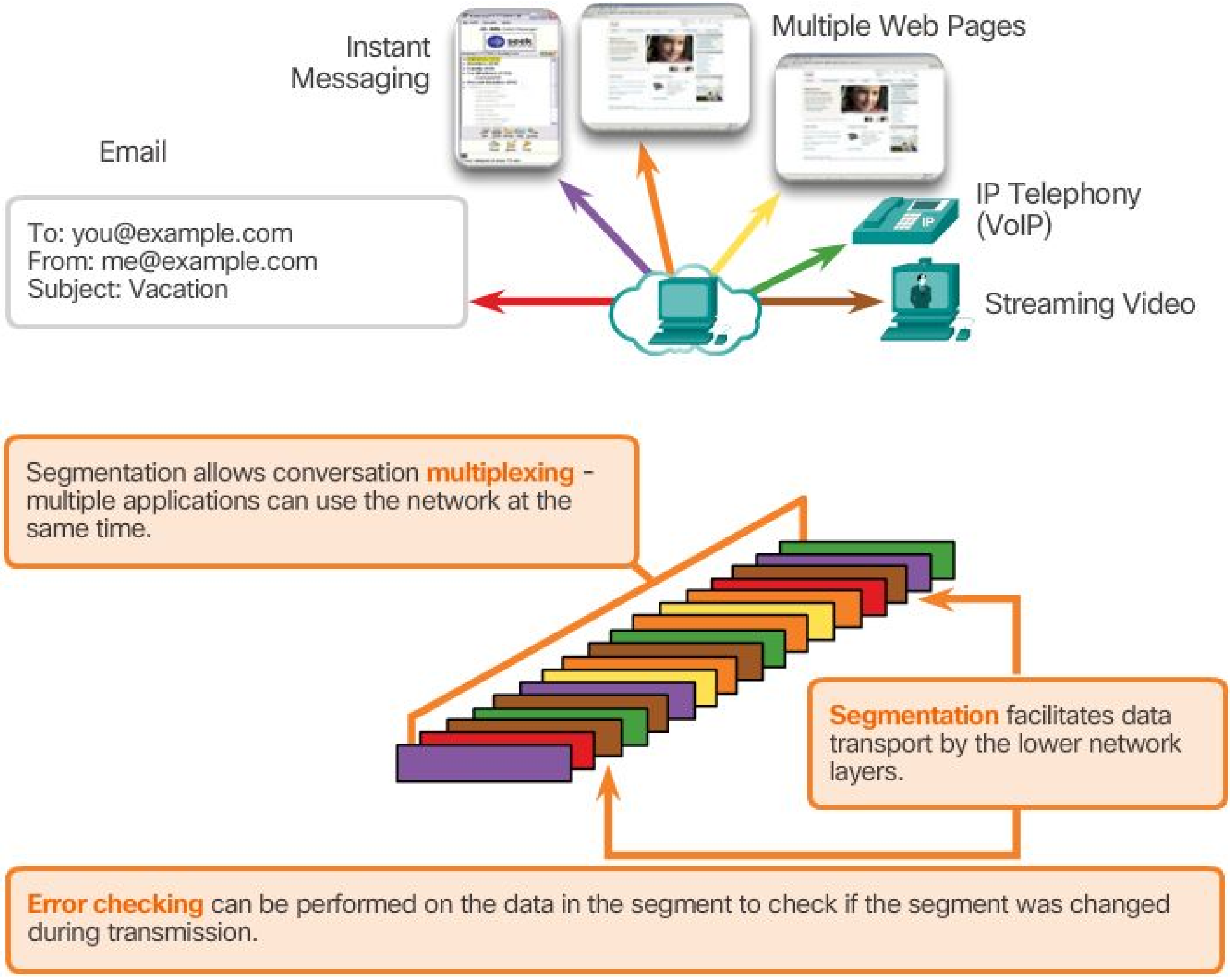
Identify the Applications



To: you@example.com
From: me@example.com
Subject: Vacation

By ensuring even when multiple applications are running on a device, all applications receive the correct data.

9.1.1.3 Conversation Multiplexing



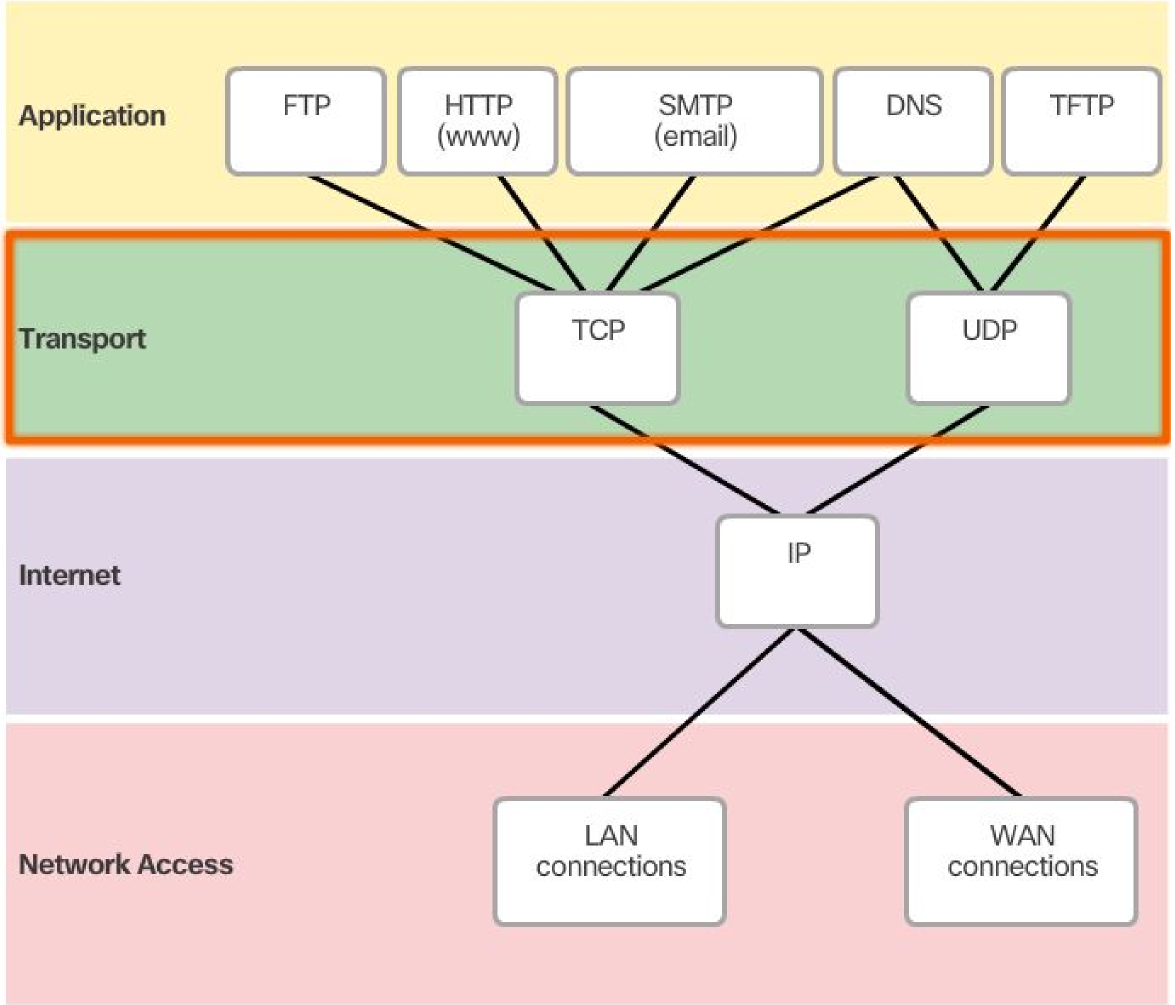
To: you@example.com
From: me@example.com
Subject: Vacation

Segmentation allows conversation **multiplexing** - multiple applications can use the network at the same time.

Segmentation facilitates data transport by the lower network layers.

Error checking can be performed on the data in the segment to check if the segment was changed during transmission.

9.1.1.4 Transport Layer Reliability



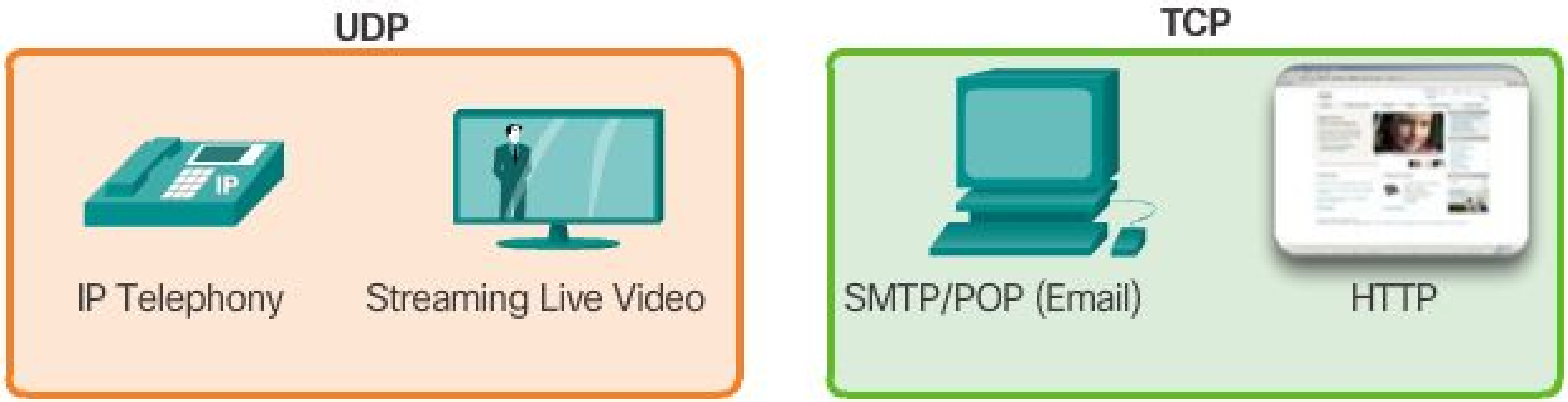
9.1.1.5 TCP

- Reliable (supports packet delivery confirmation)
- 3 basic operations (reliability):
    - Numbering and tracking data segments transmitted
    - Acknowledging received data
    - Retransmitting any unacknowledged data

9.1.1.6 UDP

- reliability is not required
- UDP provides the **basic functions** for delivering data segments between the appropriate applications, with very **little overhead** and data checking.
- Some applications do not require reliability. Reliability incurs additional overhead and possible delays in transmission.
- Adding overhead to ensure reliability for some applications could reduce the usefulness of the application and can even be detrimental.

9.1.1.7 Transport Layer Protocols



UDP

IP Telephony

Streaming Live Video

TCP

SMTP/POP (Email)

HTTP

UDP

Required protocol properties:

- Fast
- Low overhead
- Does not require acknowledgements
- Does not resend lost data
- Delivers data as it arrives

TCP

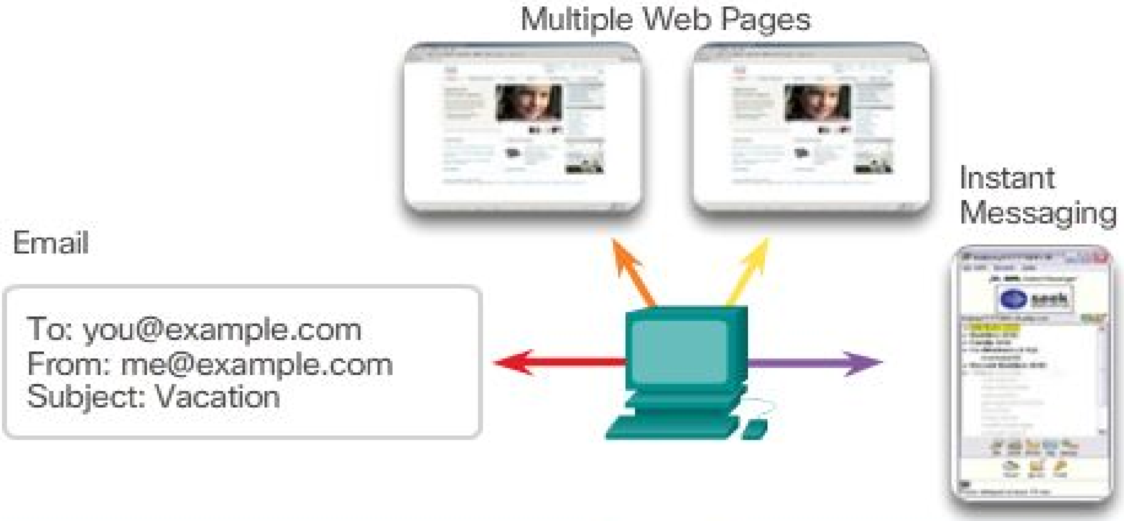
Required protocol properties:

- Reliable
- Acknowledge data
- Resends lost data
- Delivers data in order sent

Topic 9.1.2:
TCP and UDP Overview

9.1.2.1 TCP Features

TCP provides the following services:



To: you@example.com
From: me@example.com
Subject: Vacation

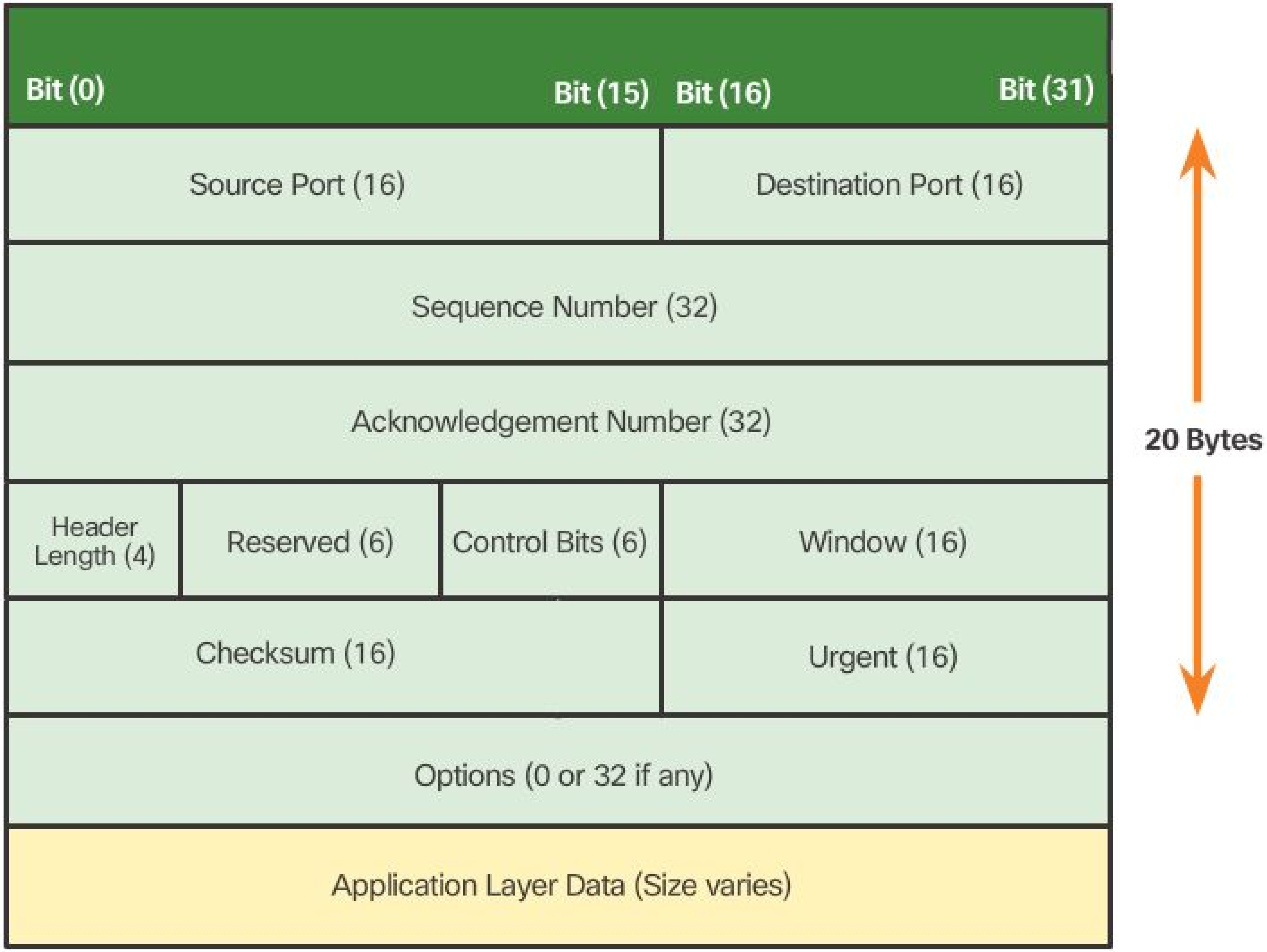
Establishing a session ensures the application is ready to receive the data.

Same order delivery ensures that the segments are reassembled into the proper order.

Reliable delivery means lost segments are resent so the data is received complete.

Flow control ensures that the receiver is able to process the data received.

9.1.2.2 TCP Header



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)

Control Bits (6)

Window (16)

Checksum (16)

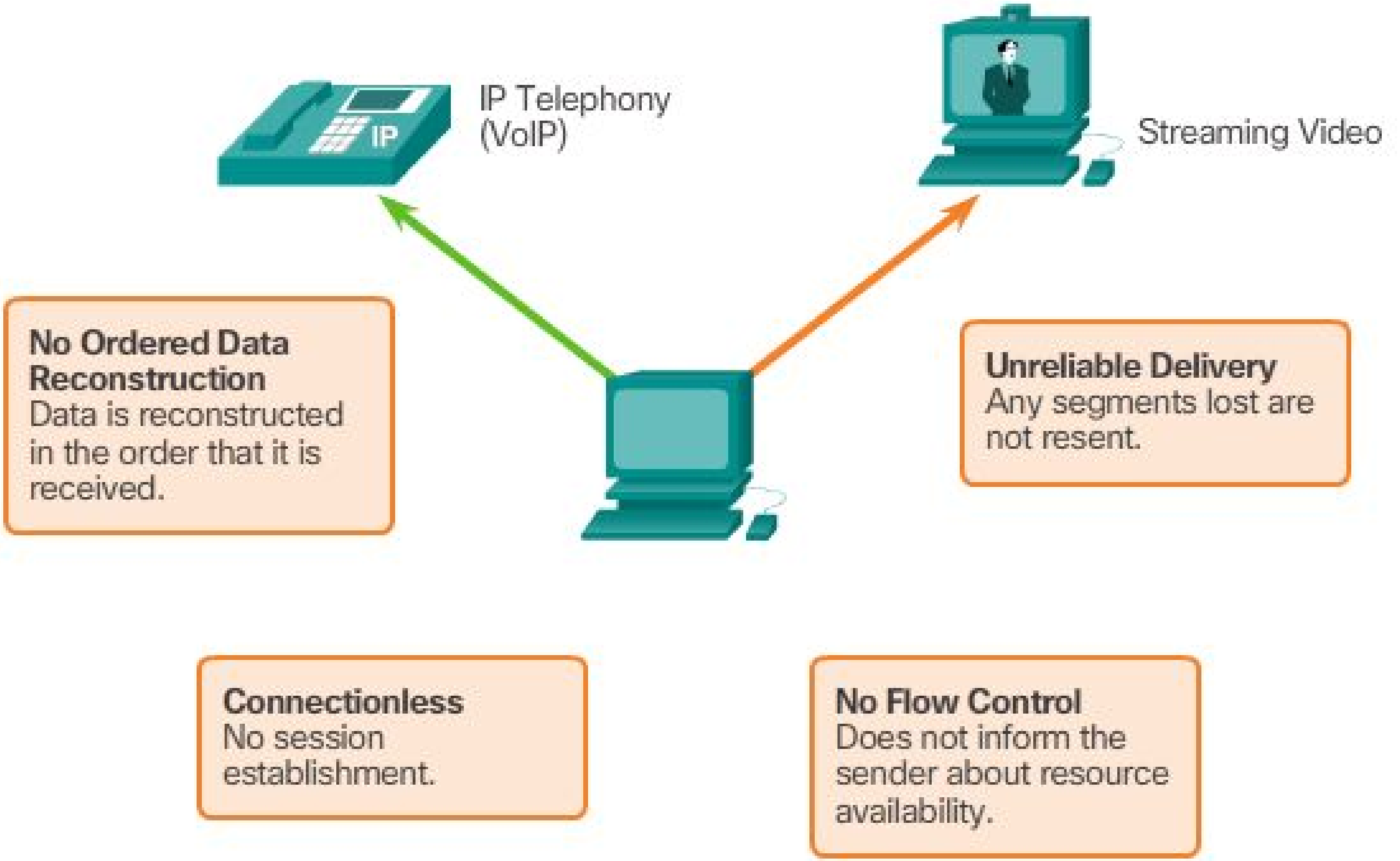
Urgent (16)

Options (0 or 32 if any)

Application Layer Data (Size varies)

20 Bytes

9.1.2.3 UDP Features



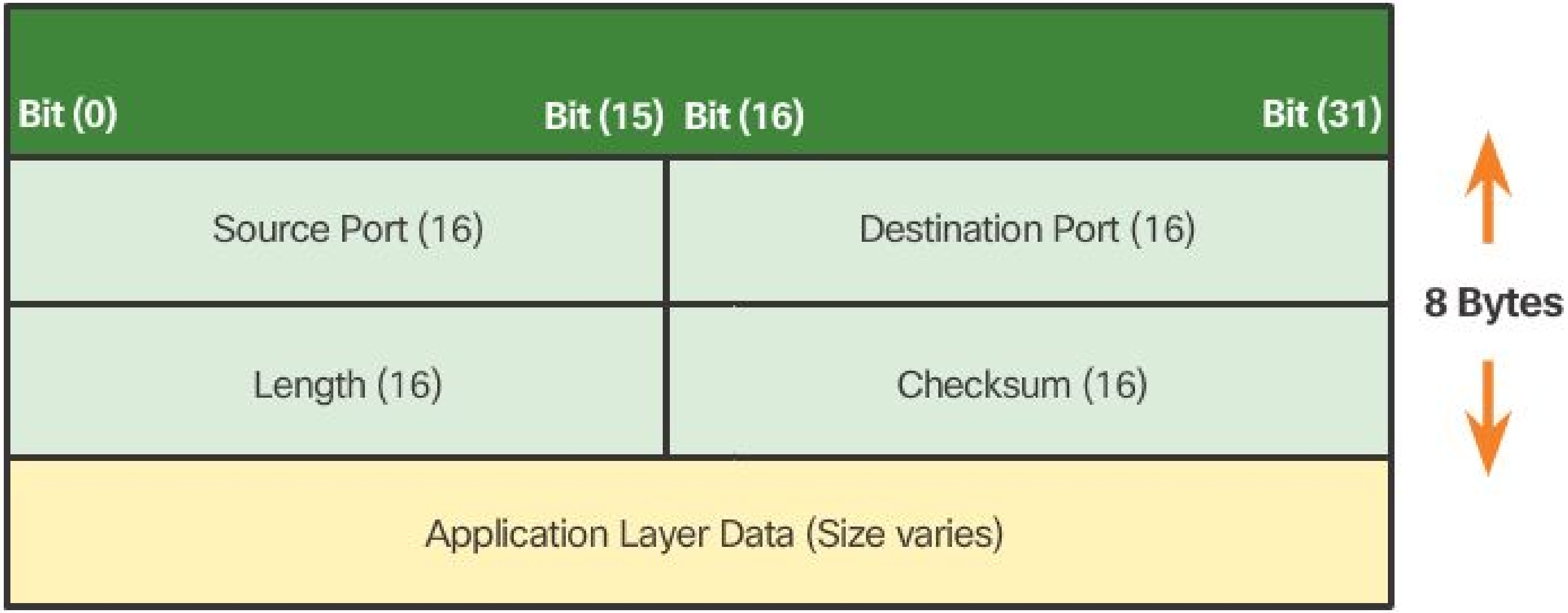
No Ordered Data Reconstruction
Data is reconstructed in the order that it is received.

Unreliable Delivery
Any segments lost are not resent.

Connectionless
No session establishment.

No Flow Control
Does not inform the sender about resource availability.

9.1.2.4 UDP Header



Bit (0) Bit (15) Bit (16) Bit (31)

Source Port (16)

Destination Port (16)

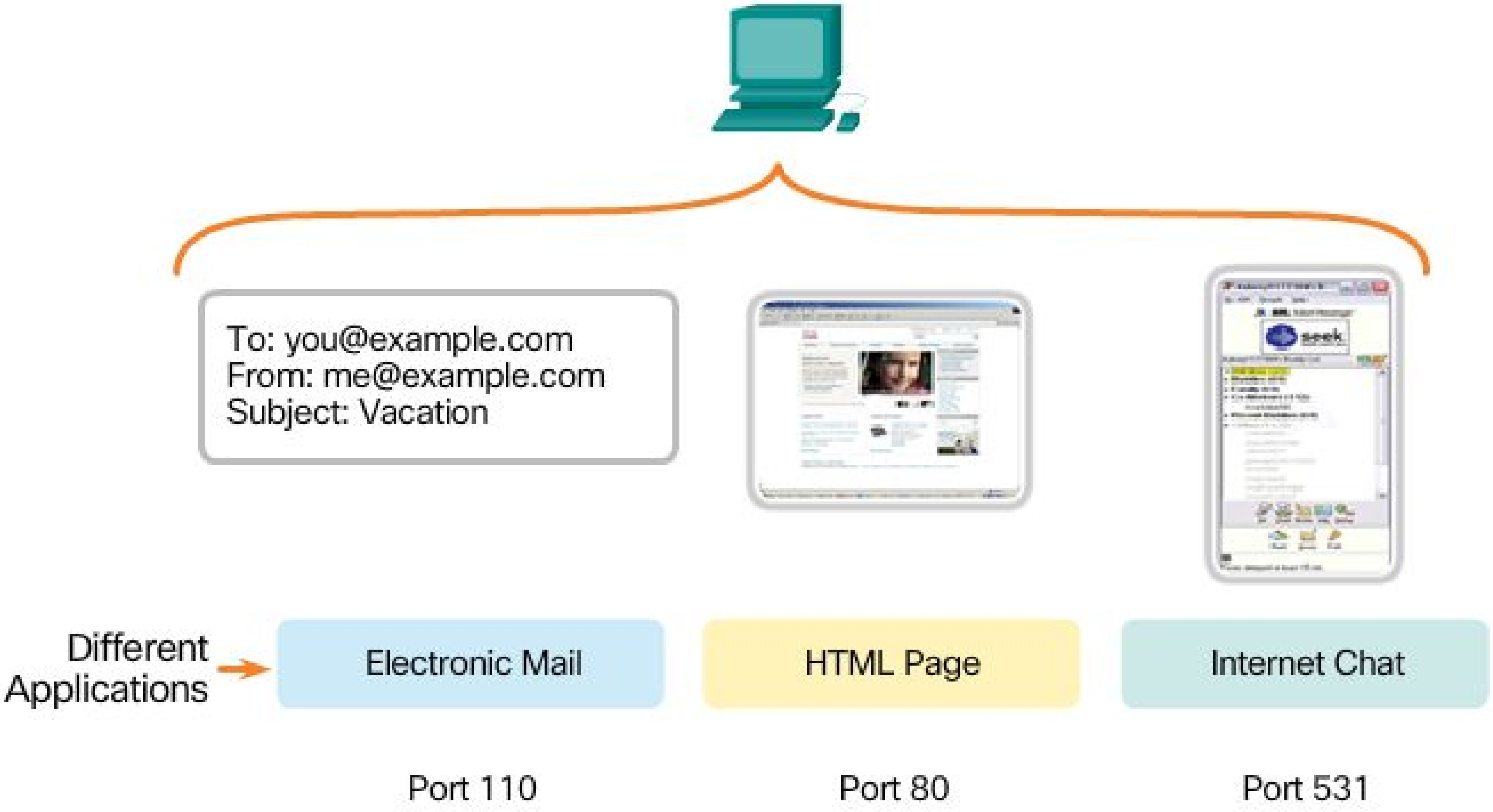
Length (16)

Checksum (16)

Application Layer Data (Size varies)

8 Bytes

9.1.2.5 Multiple Separate Conversations



To: you@example.com
From: me@example.com
Subject: Vacation

Different Applications

Electronic Mail

HTML Page

Internet Chat

Port 110

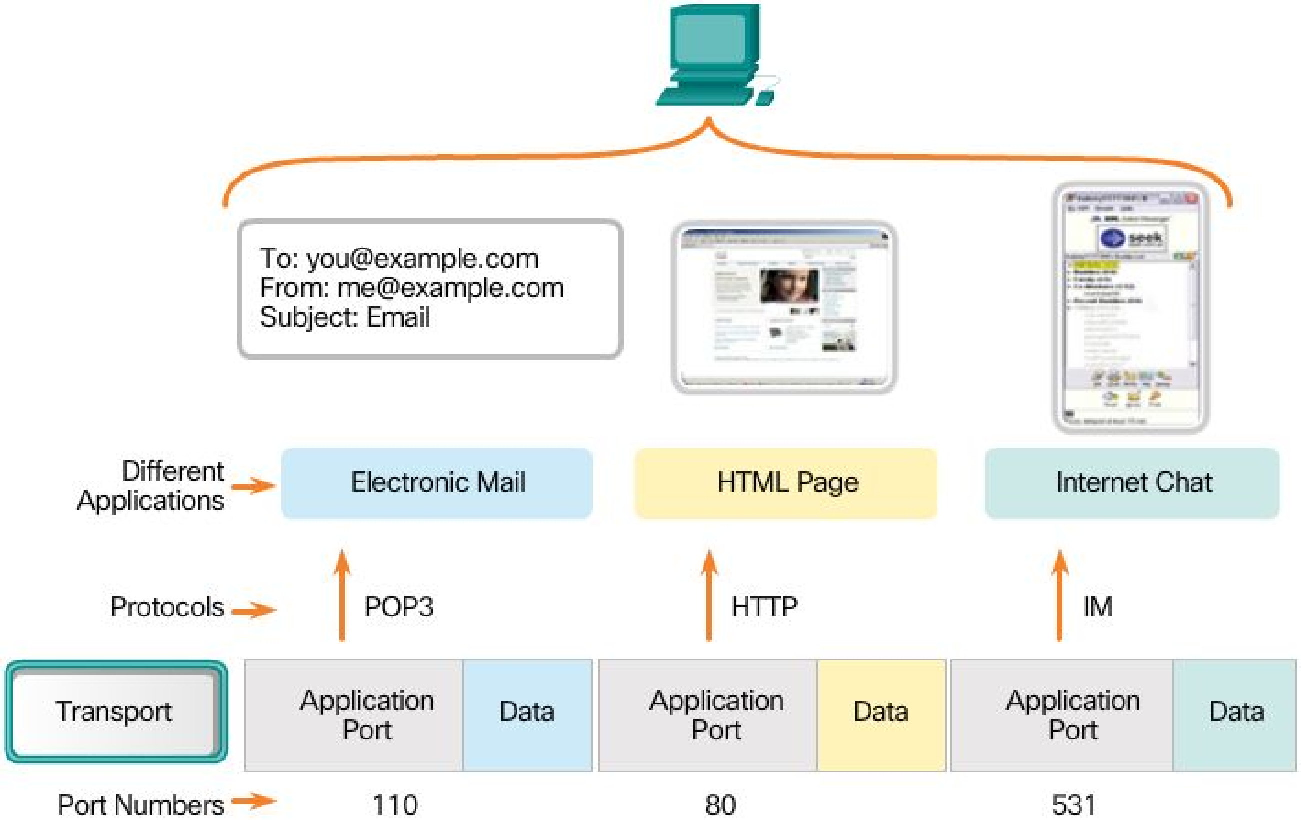
Port 80

Port 531

These unique identifiers are the port numbers.

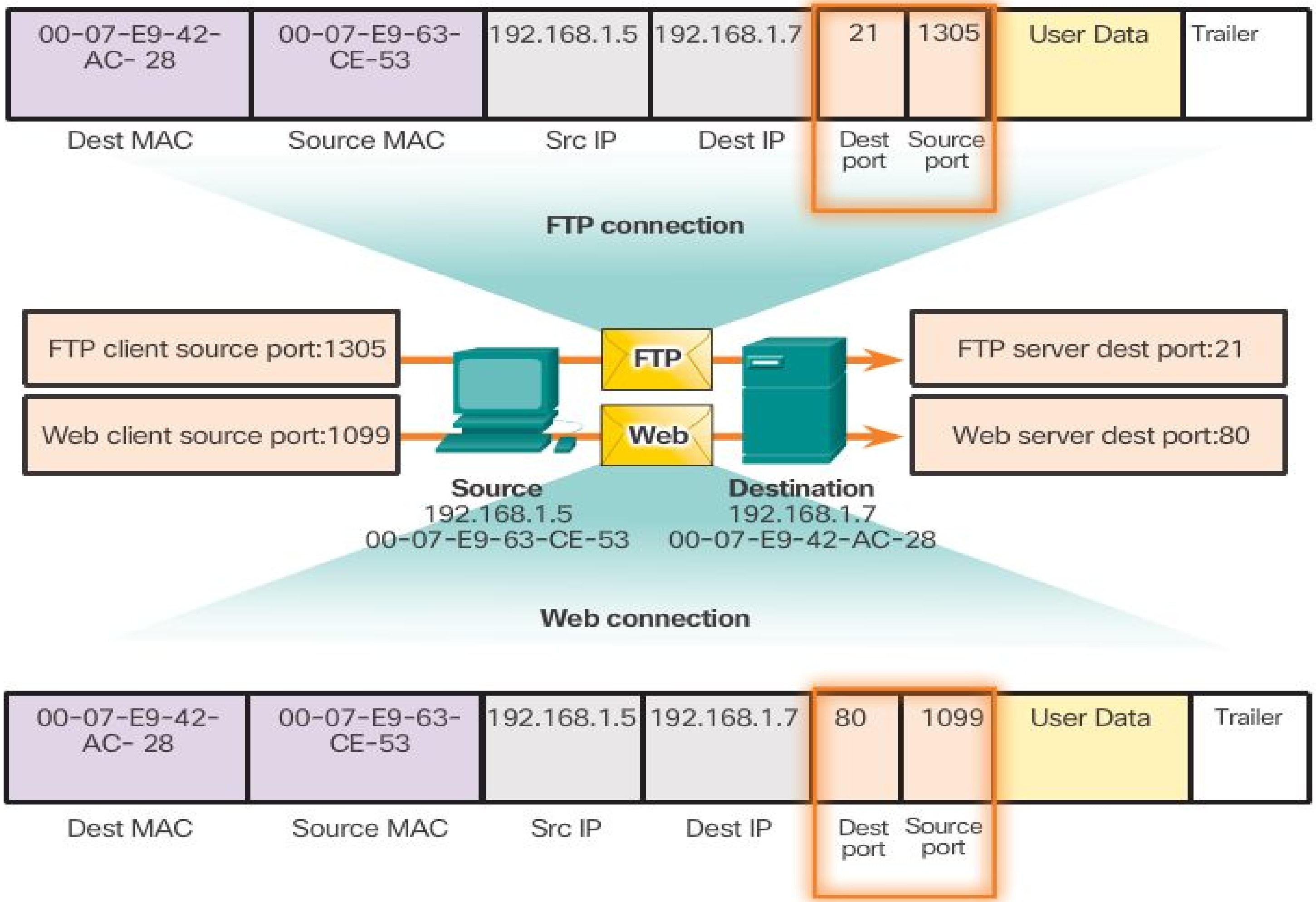
9.1.2.6 Port Numbers

- Source Port is dynamically chosen by the sending device (identify a conversation)
- Destination Port is used to identify an application or service running in the server.



To: you@example.com
From: me@example.com
Subject: Email

9.1.2.7 Socket Pairs

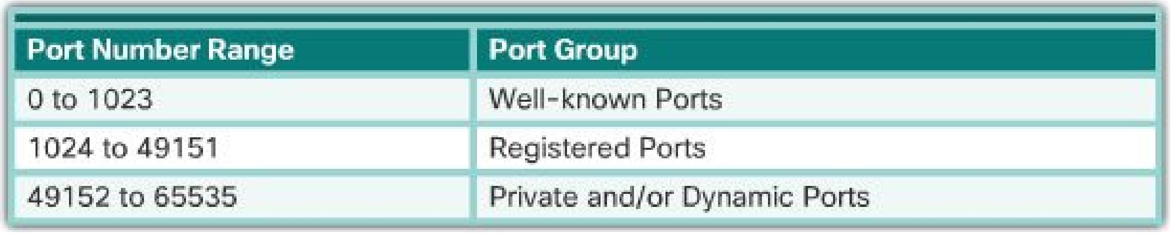


Dest MAC	Source MAC	Src IP	Dest IP	Dest port	Source port		
00-07-E9-42-AC- 28	00-07-E9-63-CE-53	192.168.1.5	192.168.1.7	21	1305	User Data	Trailer

Dest MAC	Source MAC	Src IP	Dest IP	Dest port	Source port		
00-07-E9-42-AC- 28	00-07-E9-63-CE-53	192.168.1.5	192.168.1.7	80	1099	User Data	Trailer

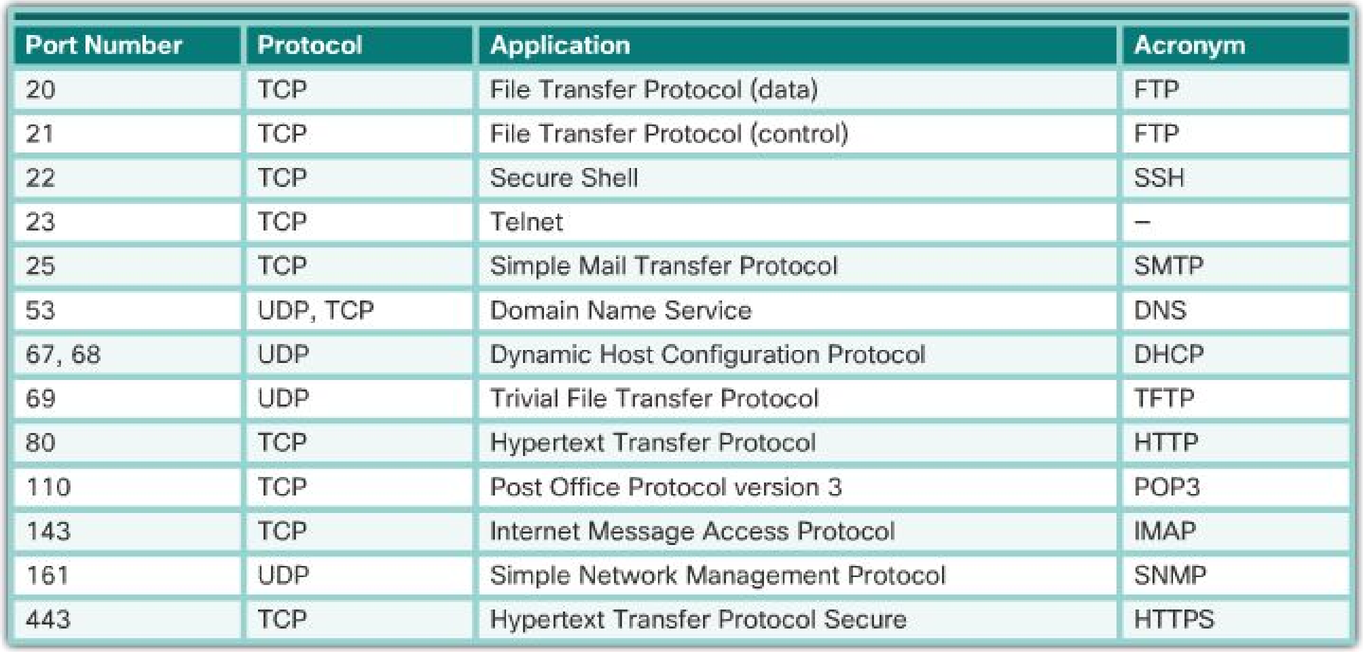
9.1.2.8 Port Number Groups

Port Numbers



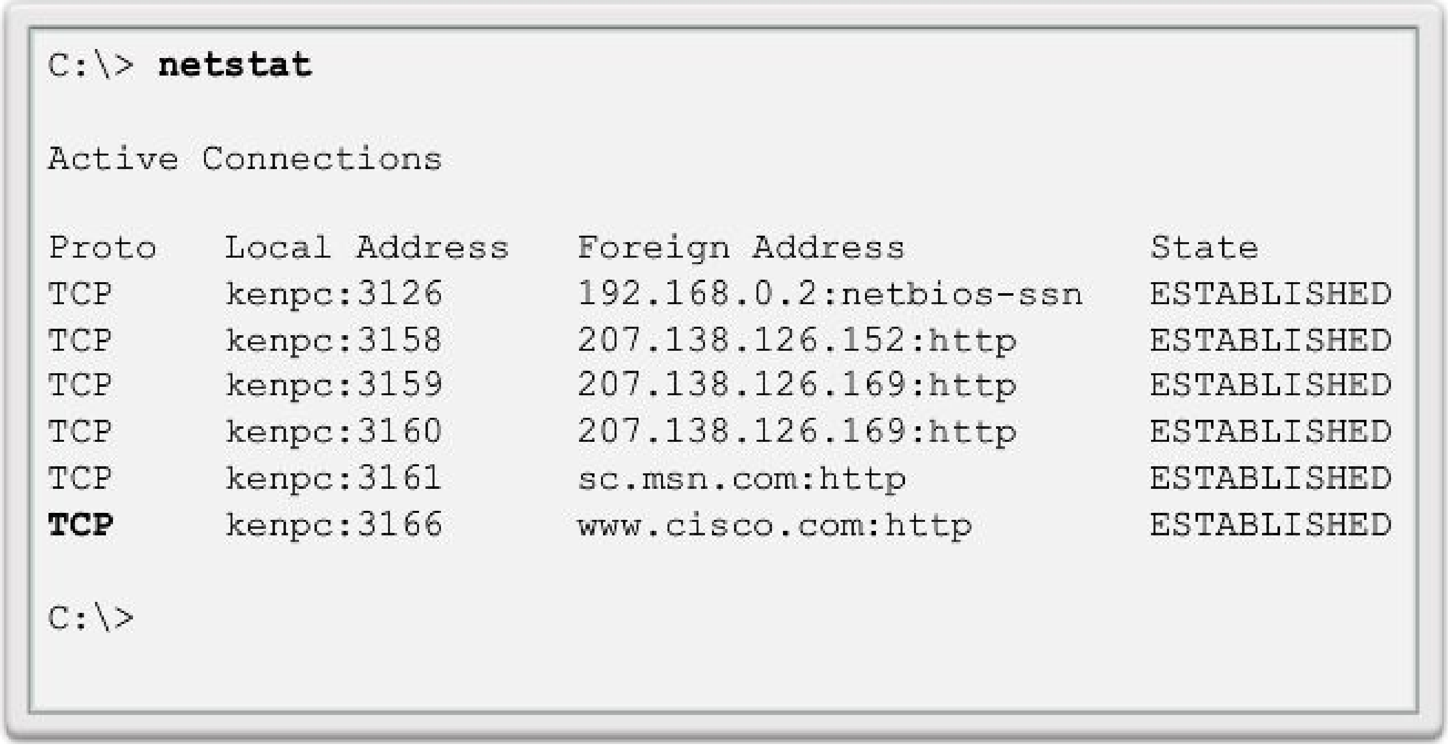
Port Number Range	Port Group
0 to 1023	Well-known Ports
1024 to 49151	Registered Ports
49152 to 65535	Private and/or Dynamic Ports

Well-Known Port Numbers



Port Number	Protocol	Application	Acronym
20	TCP	File Transfer Protocol (data)	FTP
21	TCP	File Transfer Protocol (control)	FTP
22	TCP	Secure Shell	SSH
23	TCP	Telnet	–
25	TCP	Simple Mail Transfer Protocol	SMTP
53	UDP, TCP	Domain Name Service	DNS
67, 68	UDP	Dynamic Host Configuration Protocol	DHCP
69	UDP	Trivial File Transfer Protocol	TFTP
80	TCP	Hypertext Transfer Protocol	HTTP
110	TCP	Post Office Protocol version 3	POP3
143	TCP	Internet Message Access Protocol	IMAP
161	UDP	Simple Network Management Protocol	SNMP
443	TCP	Hypertext Transfer Protocol Secure	HTTPS

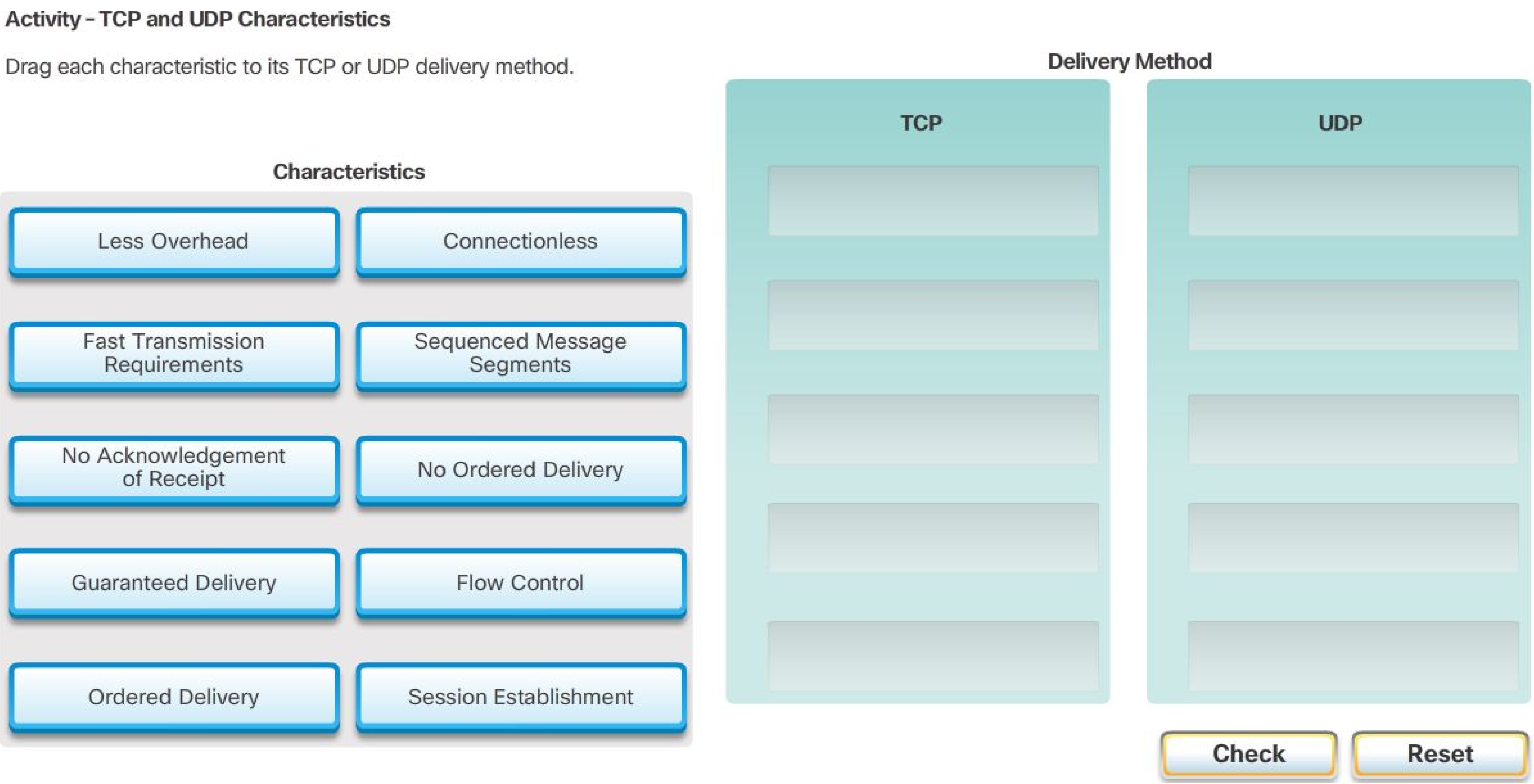
9.1.2.9 The netstat Command



```
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:\>
```

9.1.2 TCP and UDP Overview

9.1.2.10 Activity – Compare TCP and UDP Characteristics



Activity - TCP and UDP Characteristics

Drag each characteristic to its TCP or UDP delivery method.

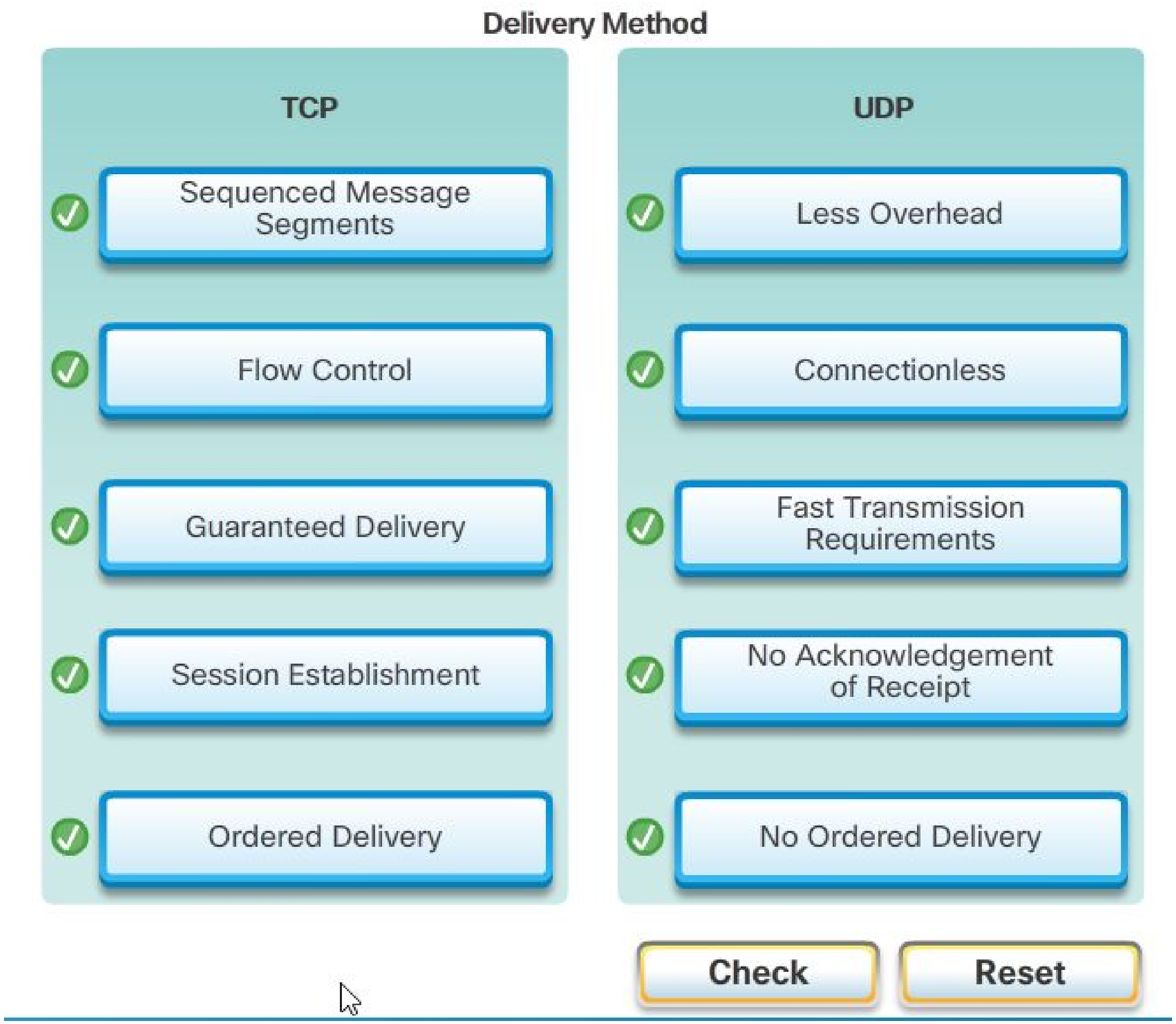
Characteristics

- Less Overhead
- Connectionless
- Fast Transmission Requirements
- Sequenced Message Segments
- No Acknowledgement of Receipt
- No Ordered Delivery
- Guaranteed Delivery
- Flow Control
- Ordered Delivery
- Session Establishment

Delivery Method

TCP	UDP

9.1.2.10 Activity – Compare TCP and UDP Characteristics



Delivery Method

TCP	UDP
✓ Sequenced Message Segments	✓ Less Overhead
✓ Flow Control	✓ Connectionless
✓ Guaranteed Delivery	✓ Fast Transmission Requirements
✓ Session Establishment	✓ No Acknowledgement of Receipt
✓ Ordered Delivery	✓ No Ordered Delivery

Section 9.2:
TCP and UPD

9.2.1 TCP Communication Process
9.2.2 Reliability and Flow Control
9.2.3 UDP Communication
9.2.4 TCP or UDP

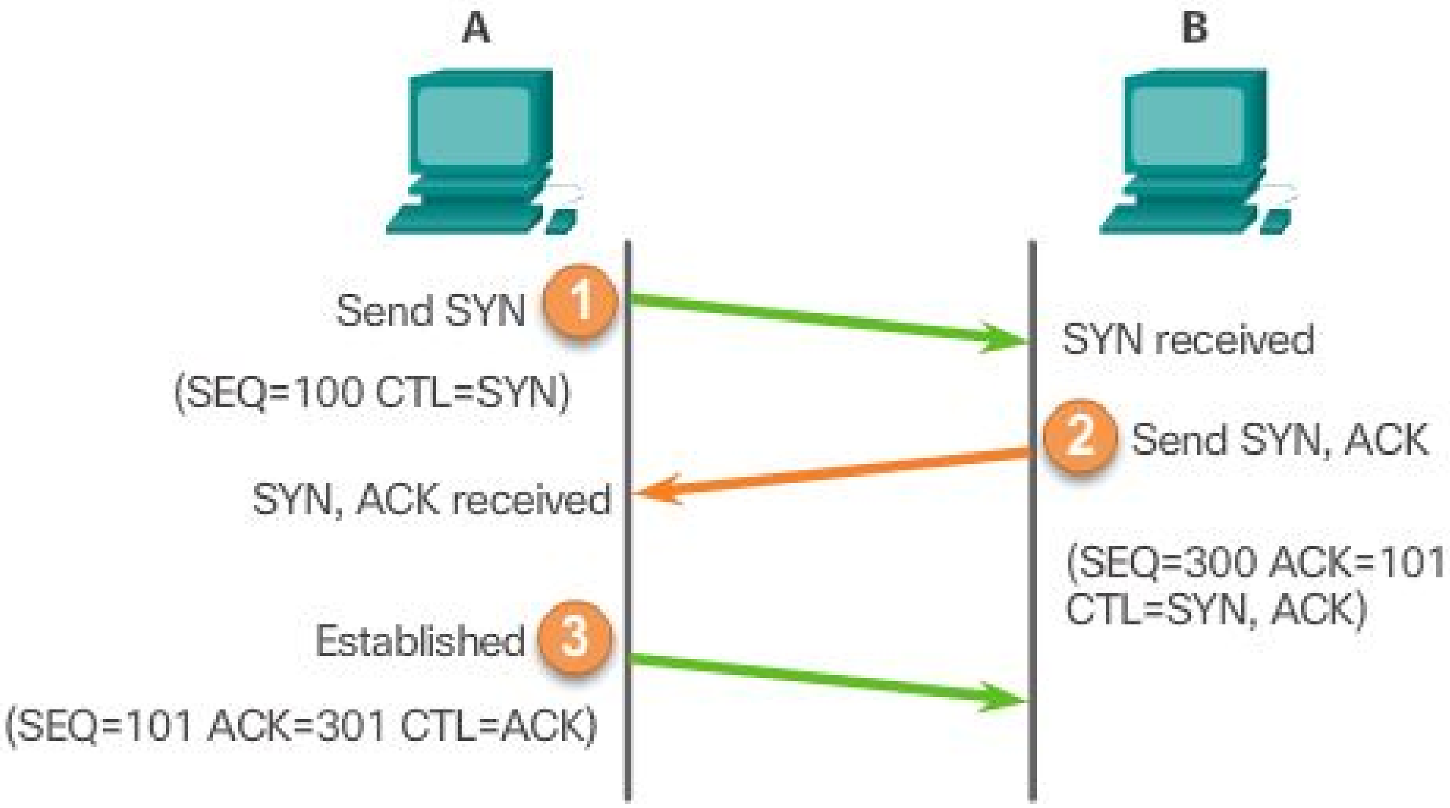
Topic 9.2.1:
TCP Communication Process

9.2.1.1 TCP Server Processes

- Each application process running on the server uses a **port number**.
- An individual server cannot have two services assigned to the same port number within the same transport layer service.
- An active server application assigned to a specific port is considered to be open.
- Any incoming client request addressed to an open port is accepted and processed by the server application bound to that port.
- There can be many ports open simultaneously on a server, one for each active server application.

9.2.1.2 TCP Connection Establishment

A TCP connection is established in three steps:

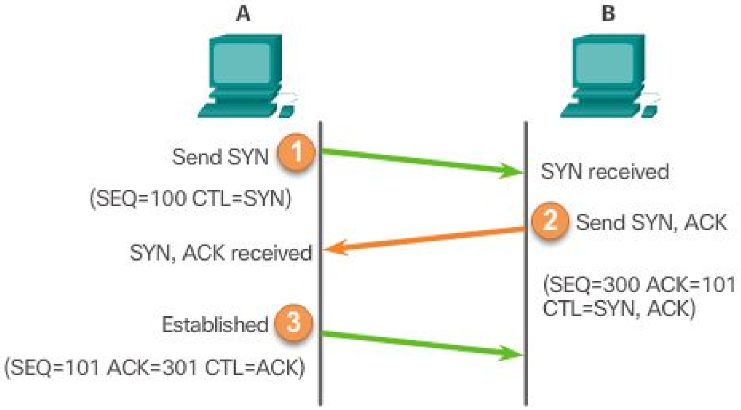


CTL = Which control bits in the TCP header are set to 1
A sends ACK response to B?

9.2.1.2 TCP Connection Establishment

A TCP connection is established in three steps:

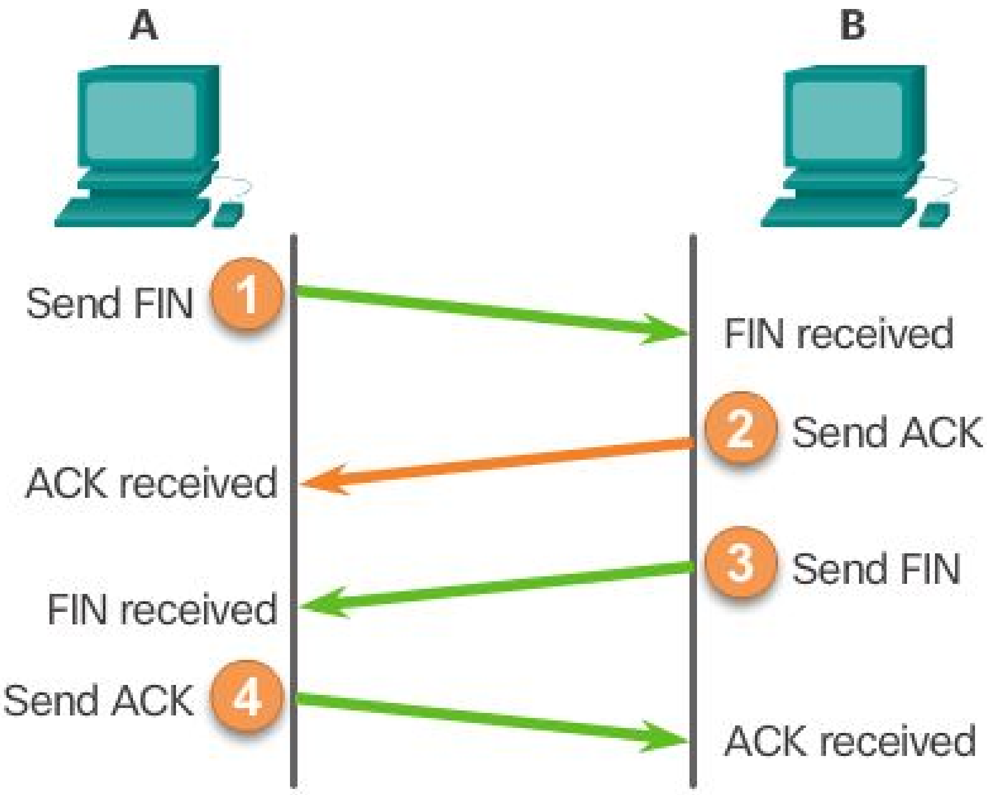
1. The initiating client **requests** a client-to-server communication session with the server.
2. The server **acknowledges** the client-to-server communication session and **requests** a server-to-client communication session.
3. The initiating client **acknowledges** the server-to-client communication session.



CTL = Which control bits in the TCP header are set to 1
A sends ACK response to B?

9.2.1.3 TCP Session Termination

The FIN TCP flag is used to terminate a TCP connection.



A sends ACK response to B.

9.2.1.3 TCP Session Termination

The FIN TCP flag is used to terminate a TCP connection.

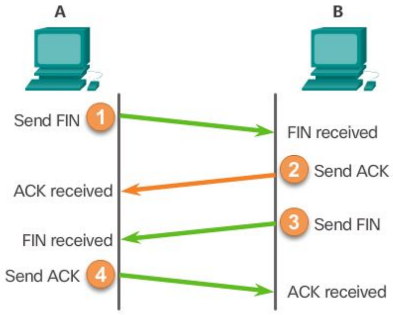
1. When the client has no more data to send in the stream, it sends a segment with the **FIN** flag set.

2. The server sends an **ACK** to acknowledge the receipt of the FIN to terminate the session from client to server.

3. The server sends a **FIN** to the client to terminate the server-to-client session.

4. The client responds with an ACK to acknowledge the FIN from the server.

5. When all segments have been acknowledged, the session is closed.

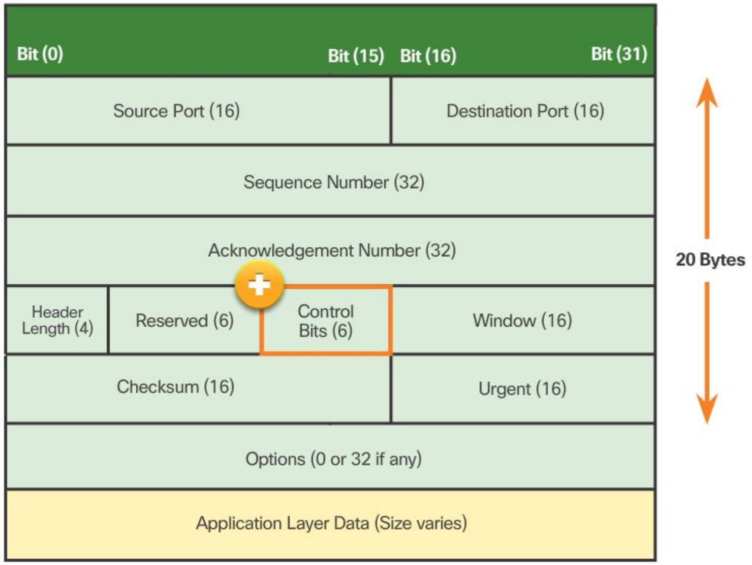


A sends ACK response to B.

9.2.1.4 TCP Three-Way Handshake Analysis

The three-way handshake:

- **Establishes** that the destination device is present on the network.
- **Verifies** that the destination device has an active service and is accepting requests on the destination port number that the initiating client intends to use
- **Informs** the destination device that the source client intends to establish a communication session on that port number.
- Video Available



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) | Control Bits (6) | Window (16)

Checksum (16) | Urgent (16)

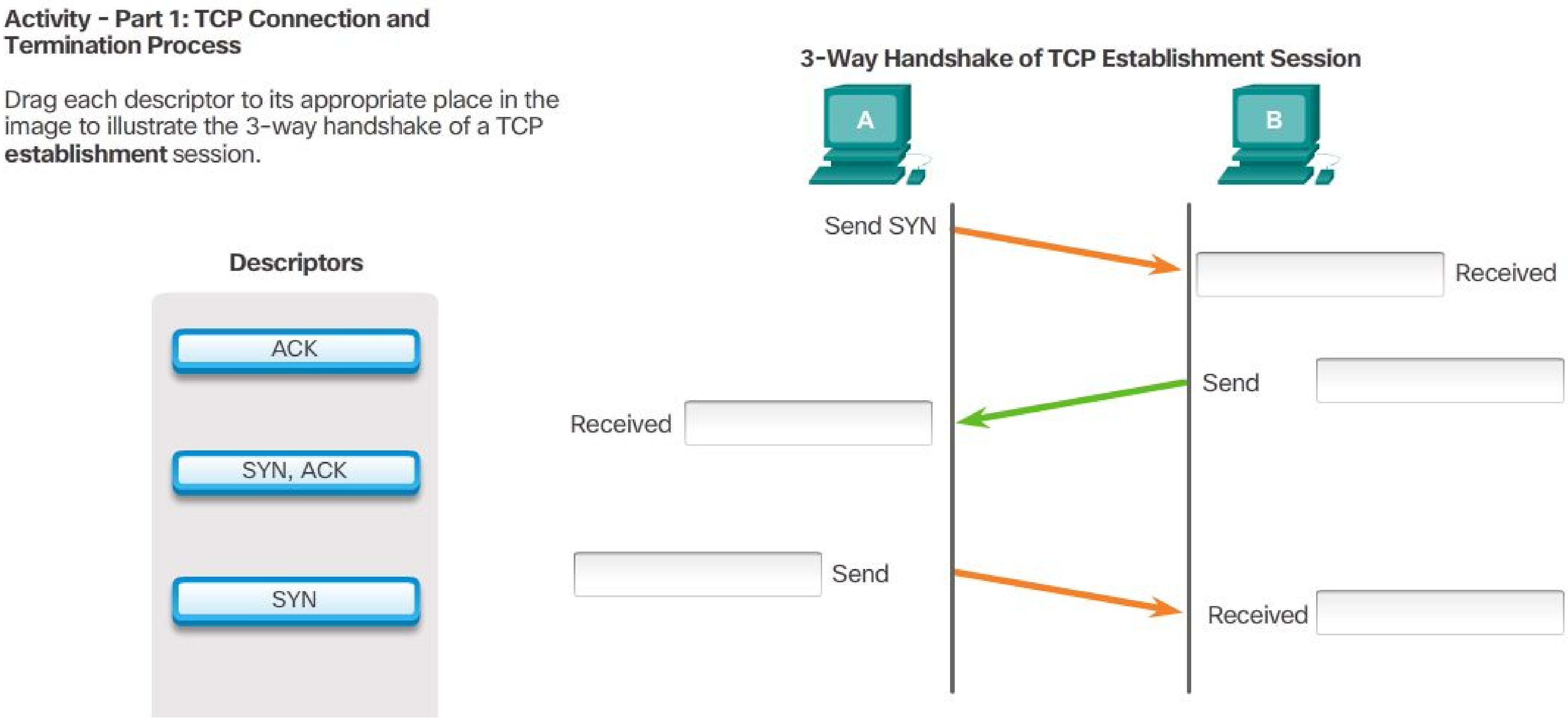
Options (0 or 32 if any)

Application Layer Data (Size varies)

20 Bytes

URG - Urgent pointer field significant
ACK - Acknowledgement field significant
PSH - Push function
RST - Reset the connection
SYN - Synchronize sequence numbers
FIN - No more data from sender

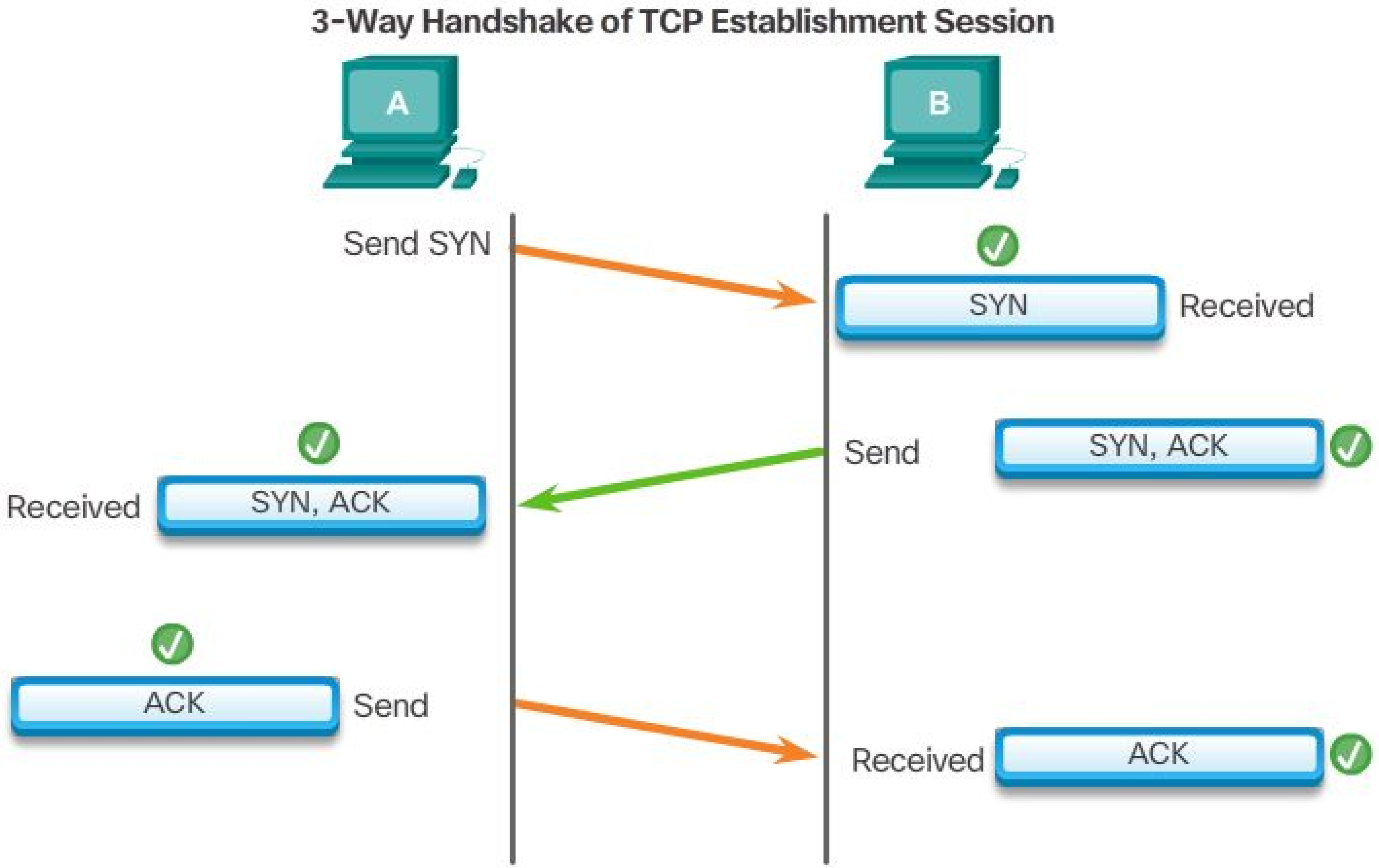
9.2.1.7 Activity - TCP Connection and Termination Process



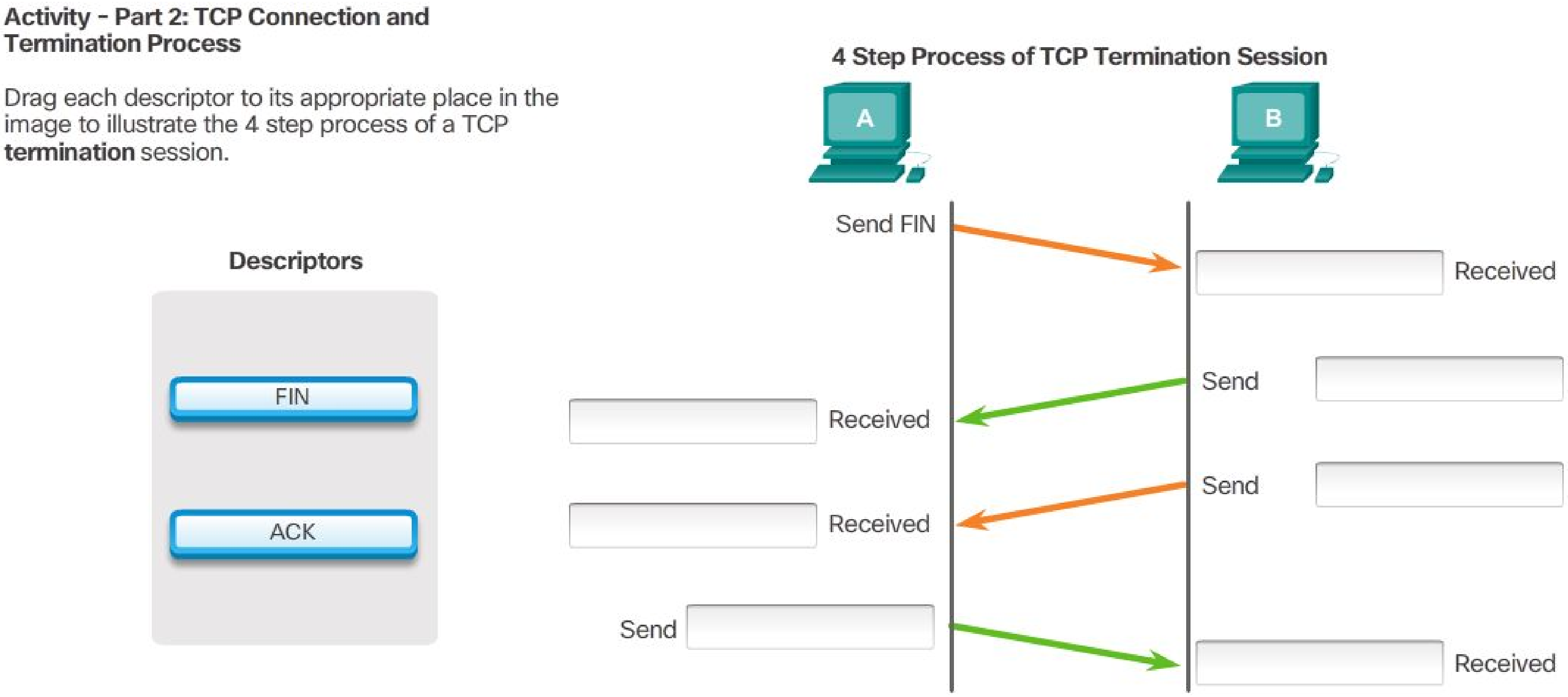
Activity - Part 1: TCP Connection and Termination Process

Drag each descriptor to its appropriate place in the image to illustrate the 3-way handshake of a TCP **establishment** session.

9.2.1.7 Activity - TCP Connection and Termination Process



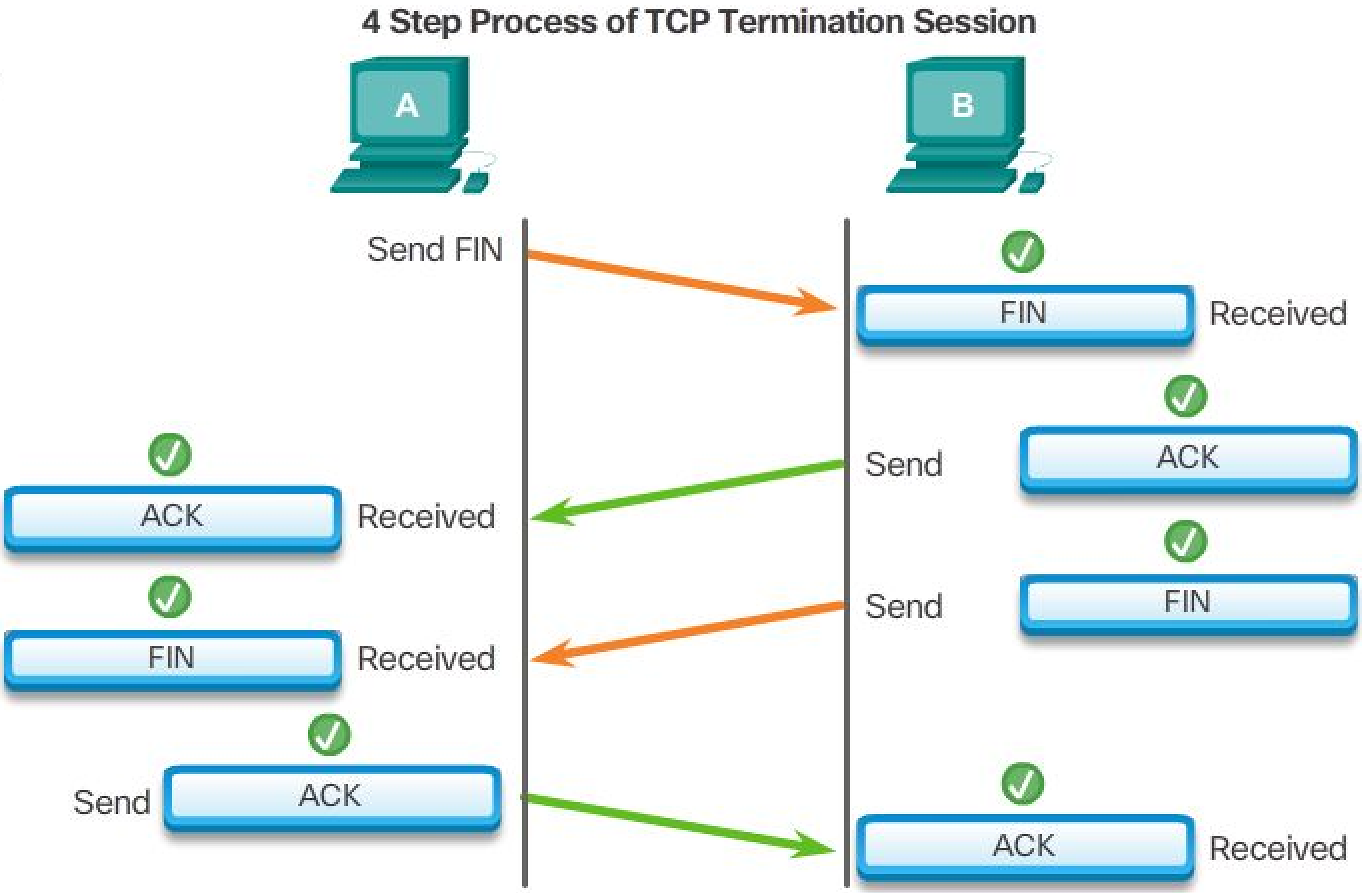
9.2.1.7 Activity - TCP Connection and Termination Process



Activity - Part 2: TCP Connection and Termination Process

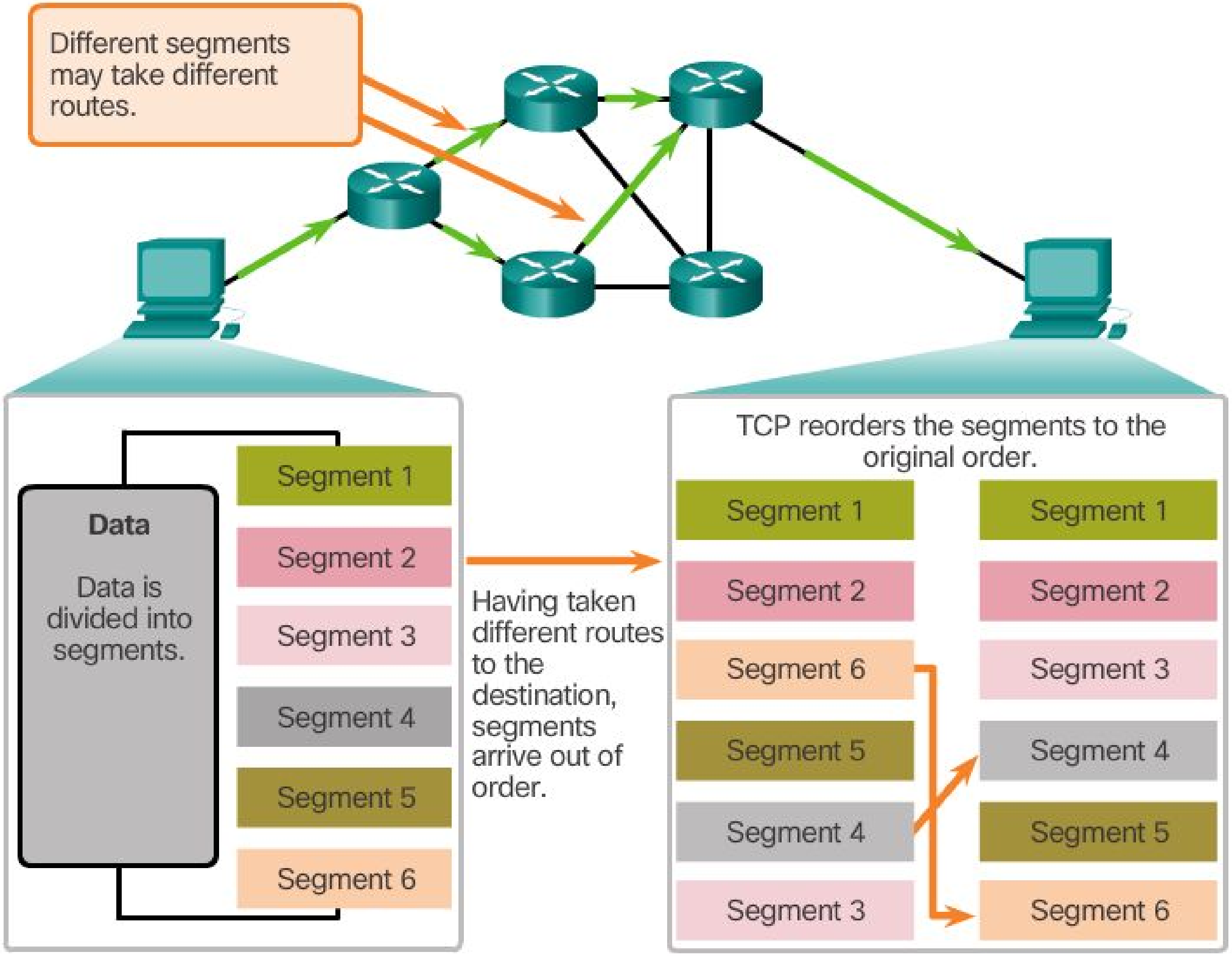
Drag each descriptor to its appropriate place in the image to illustrate the 4 step process of a TCP **termination** session.

9.2.1.7 Activity - TCP Connection and Termination Process



Topic 9.2.2:
Reliability and Flow Control

9.2.2.1 TCP Reliability – Ordered Delivery


Different segments may take different routes.

Data

Data is divided into segments.

Having taken different routes to the destination, segments arrive out of order.

TCP reorders the segments to the original order.

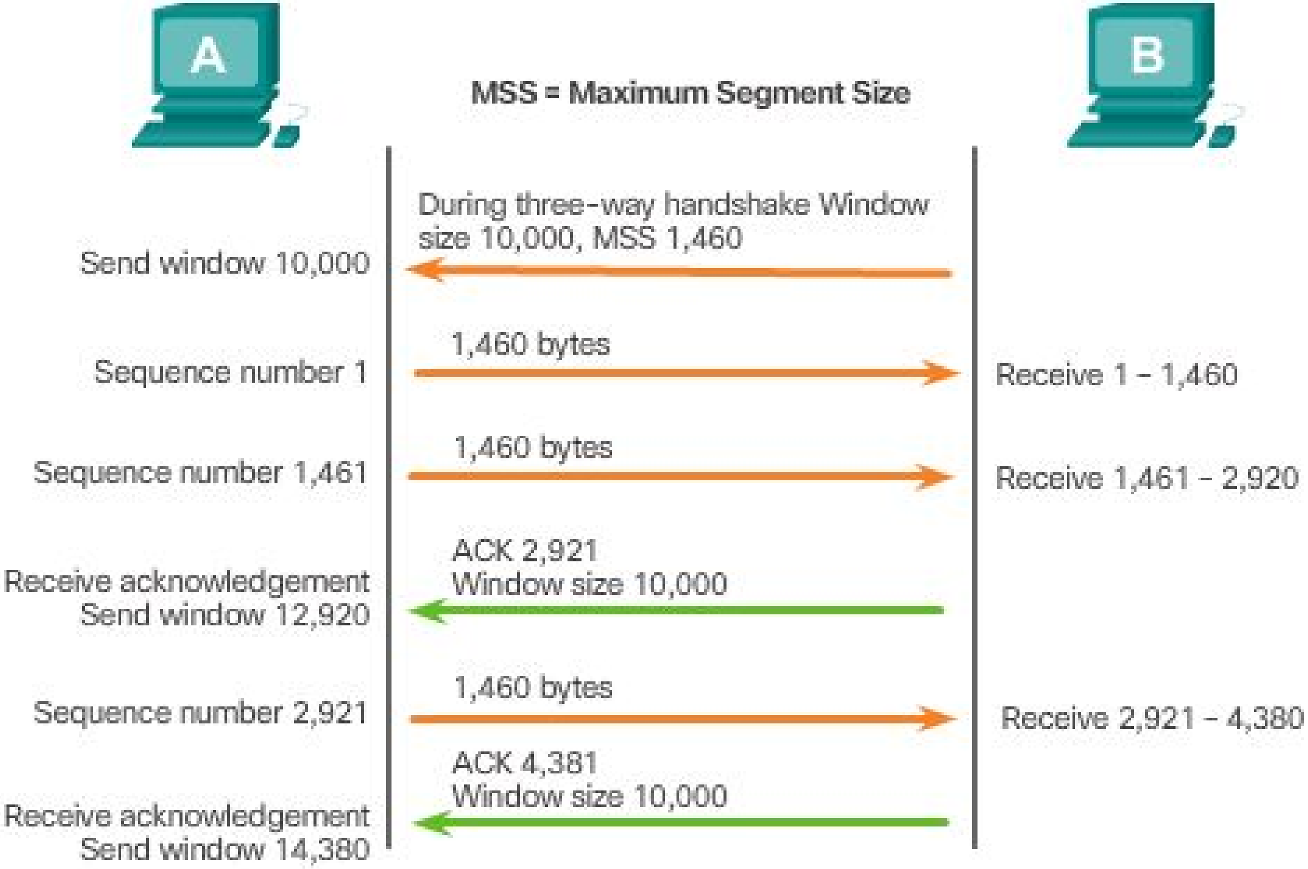
9.2.2.2 TCP Reliability - Sequence Numbers and Acknowledgements

- Confirm that each segment reached its destination.
- Ensures the destination is reachable (ready to receive data)
- Acknowledgment (data received from the source application)
- Retransmission of missed segments.
- All segments are properly re-ordered upon receipt.
- Video Available

9.2.2.3 TCP Reliability – Data Loss and Retransmission

- TCP provides methods of managing segment losses.
- Among these methods is a mechanism to retransmit segments for unacknowledged data.
- Video Available

9.2.2.4 TCP Flow Control – Window Size and Acknowledgements



A

B

MSS = Maximum Segment Size

Send window 10,000

During three-way handshake Window size 10,000, MSS 1,460

Sequence number 1

1,460 bytes

Receive 1 – 1,460

Sequence number 1,461

1,460 bytes

Receive 1,461 – 2,920

Receive acknowledgement
Send window 12,920

ACK 2,921
Window size 10,000

Sequence number 2,921

1,460 bytes

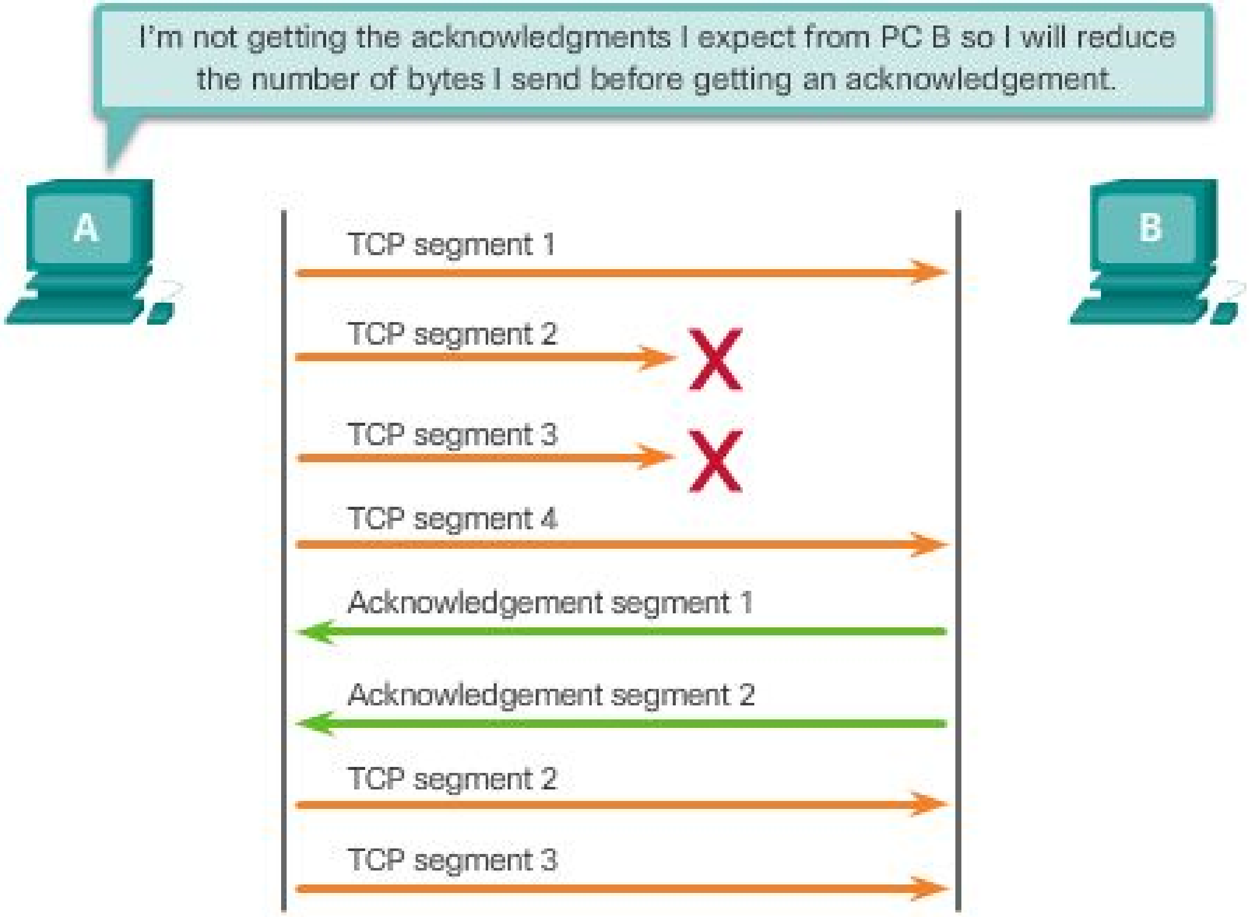
Receive 2,921 – 4,380

ACK 4,381
Window size 10,000

Receive acknowledgement
Send window 14,380

The **window size** determines the number of bytes that can be sent before expecting an acknowledgment.
The **acknowledgement** number is the number of the next expected byte.

9.2.2.5 TCP Flow Control – Congestion Avoidance

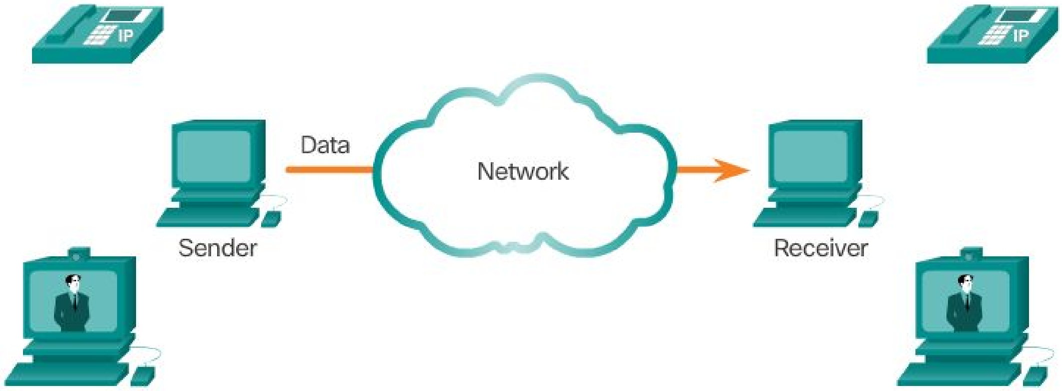

I'm not getting the acknowledgments I expect from PC B so I will reduce the number of bytes I send before getting an acknowledgement.
A
B
TCP segment 1
TCP segment 2
TCP segment 3
TCP segment 4
Acknowledgement segment 1
Acknowledgement segment 2
TCP segment 2
TCP segment 3

Acknowledgement numbers are for the next expected byte and not for a segment. Segment number are only used here for simplicity.

Topic 9.2.3: UDP Communication

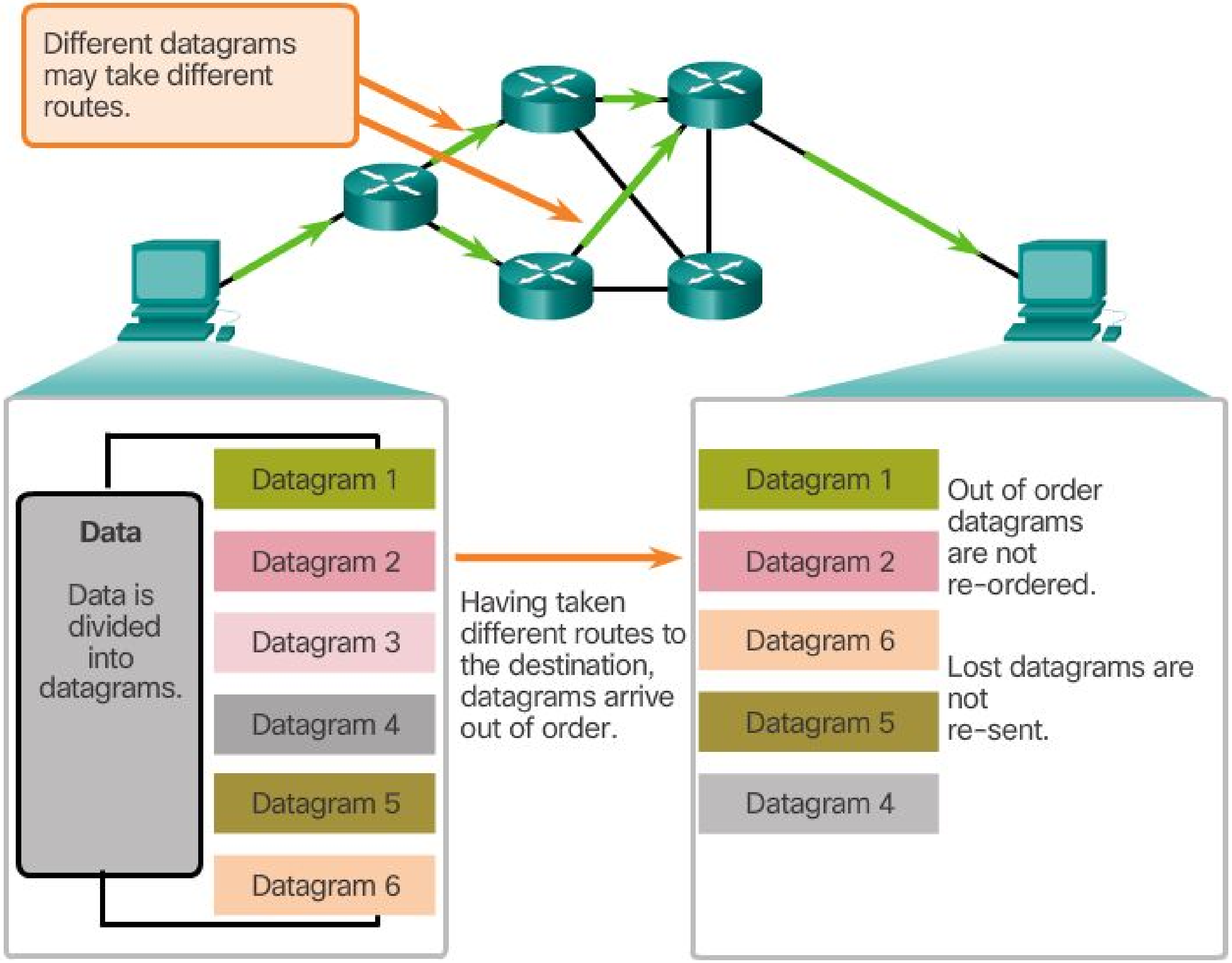
9.2.3.1 UDP Low Overhead versus Reliability

- UDP is a simple protocol.
- UDP provides the basic transport layer functions.
- UDP has much lower overhead than TCP.
- UDP is not connection-oriented and does not offer the sophisticated retransmission, sequencing, and flow control mechanisms.
- Applications running UDP can still use reliability, but it must be implemented in the application layer.
- However, UDP is not inferior. It is designed to be simpler and faster than TCP at the expense of reliability.

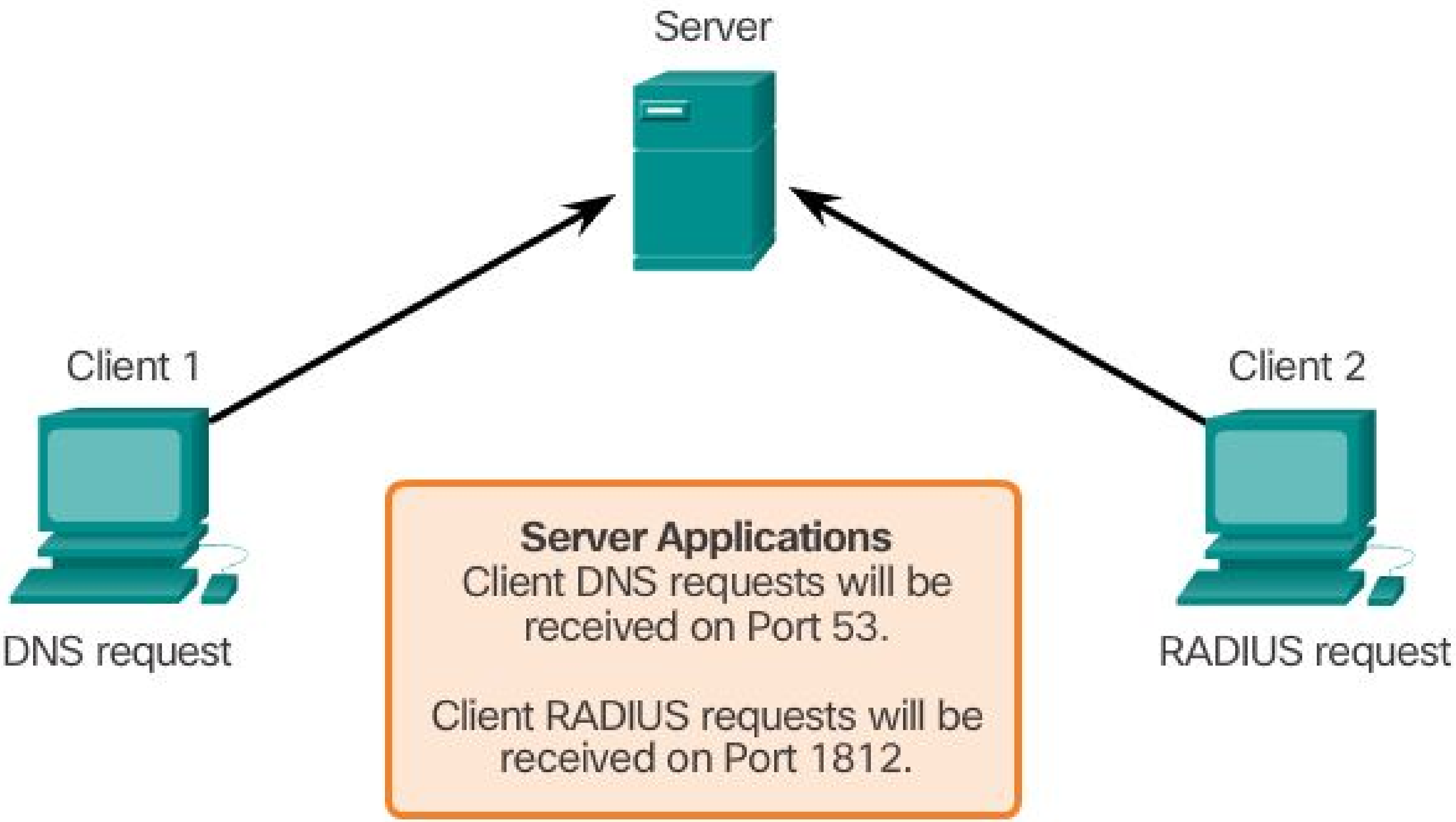


UDP does not establish a connection before sending data.

9.2.3.2 UDP Datagram Reassembly


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.

9.2.3.3 UDP Server Processes



Server Applications
Client DNS requests will be received on Port 53.

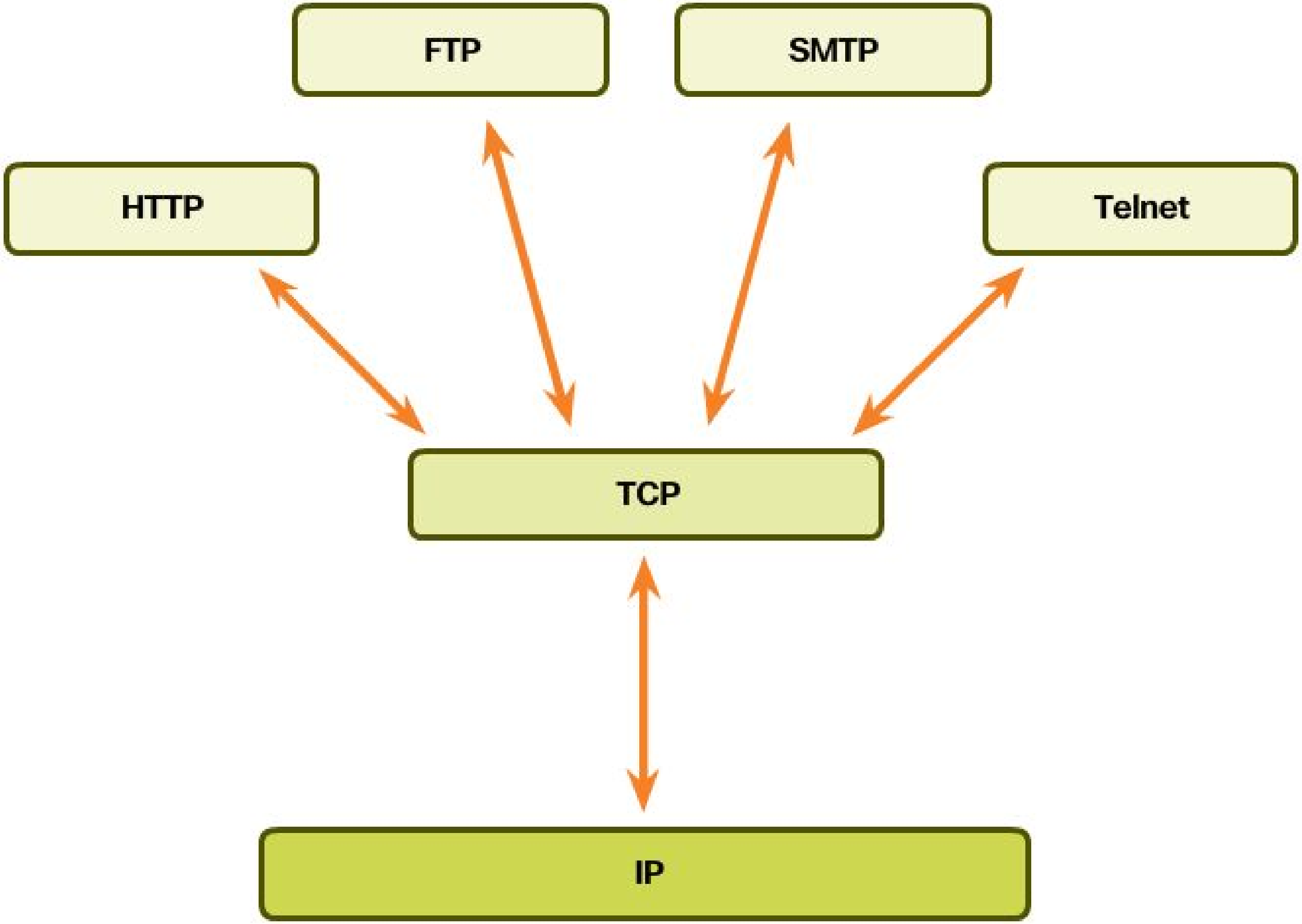
Client RADIUS requests will be received on Port 1812.

9.2.3.4 UDP Server Processes

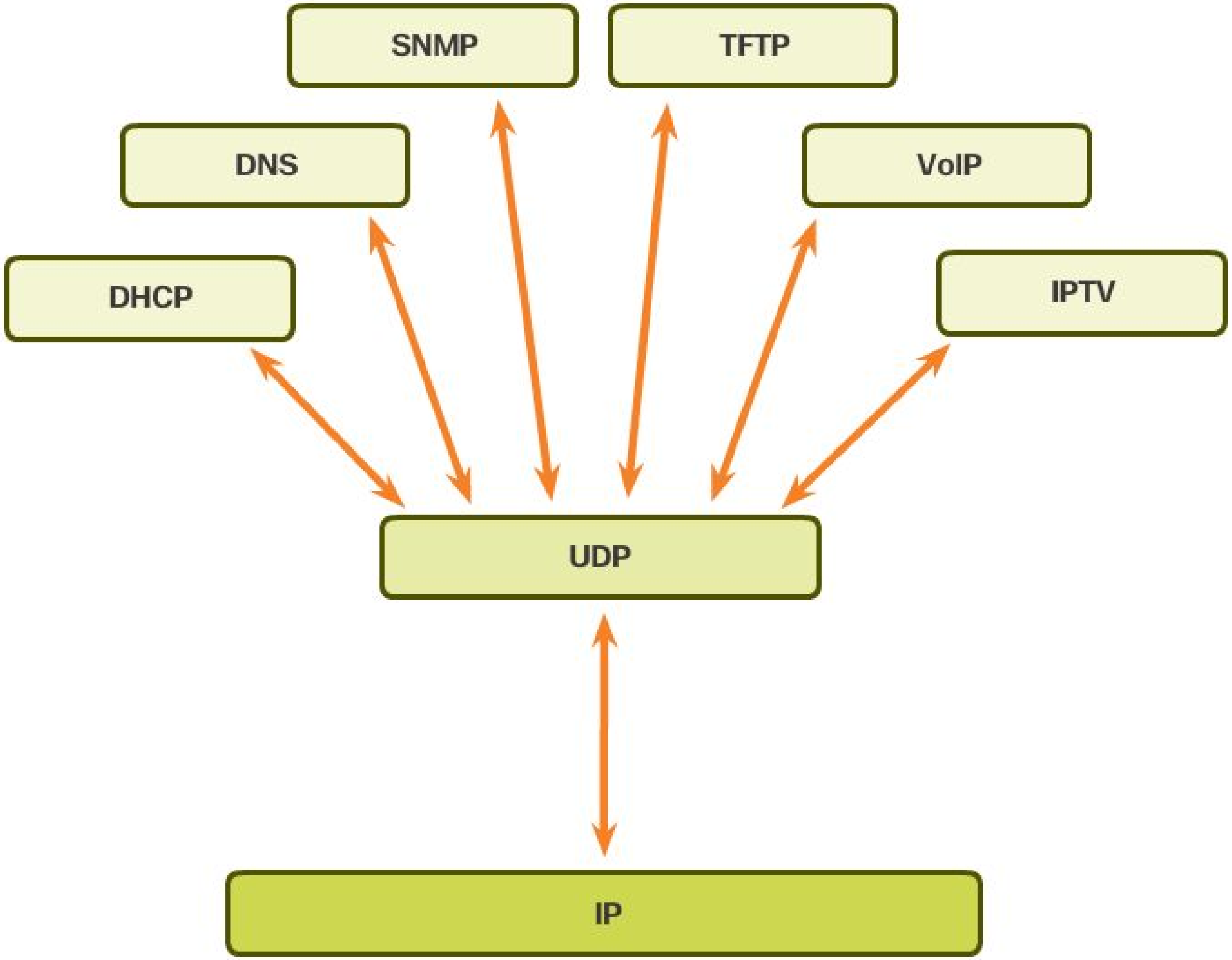
- UDP client-server communication is also initiated by a client application.
- The UDP client process dynamically selects a port number and uses this as the source port.
- The destination port is usually the well-known or registered port number assigned to the server process.
- The same source-destination pair of ports is used in the header of all datagrams used in the transaction.
- Data returning to the client from the server uses a flipped source and destination port numbers in the datagram header.

Topic 9.2.4:
TCP or UDP

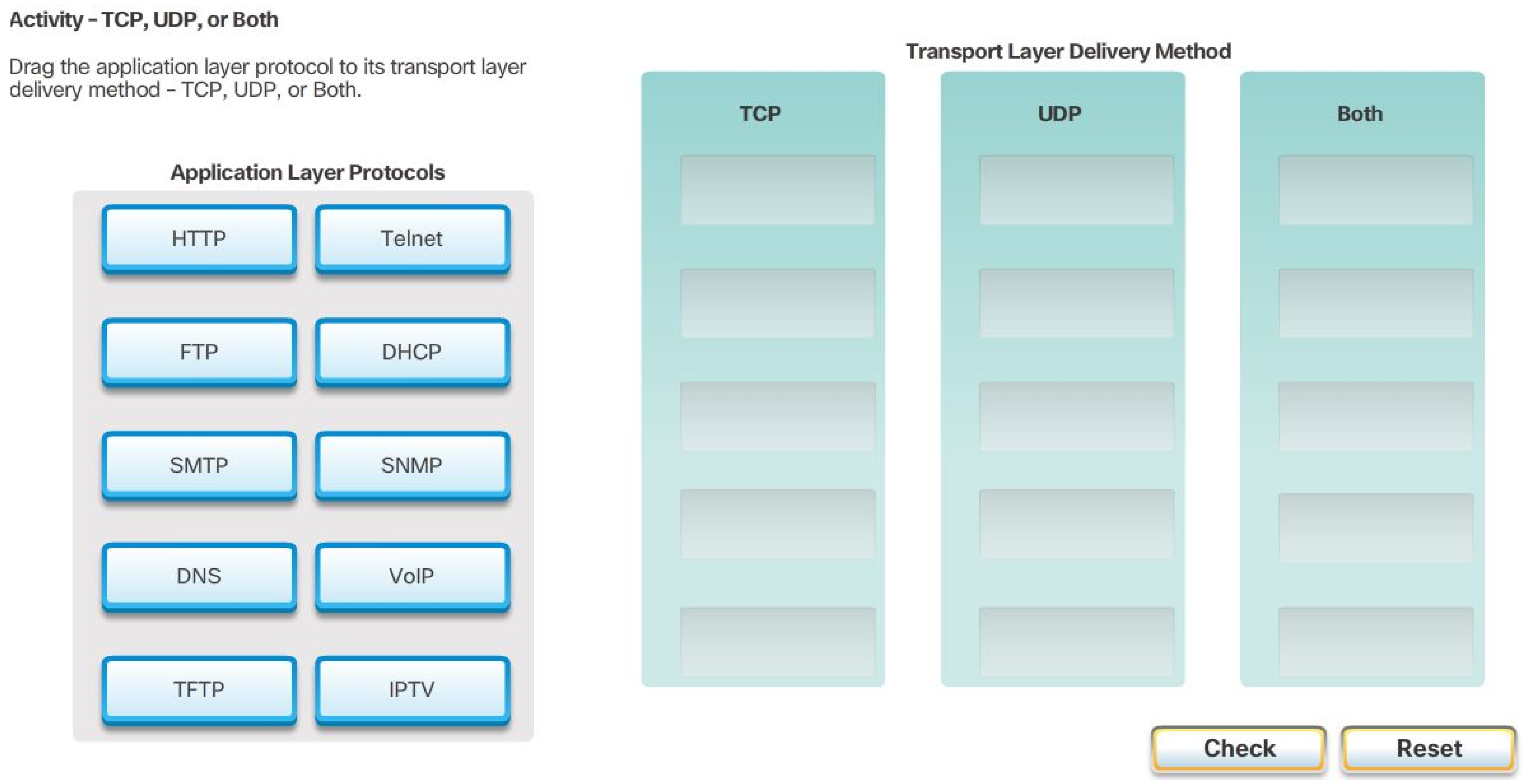
9.2.4.1 Applications that use TCP



9.2.4.2 Applications that use UDP



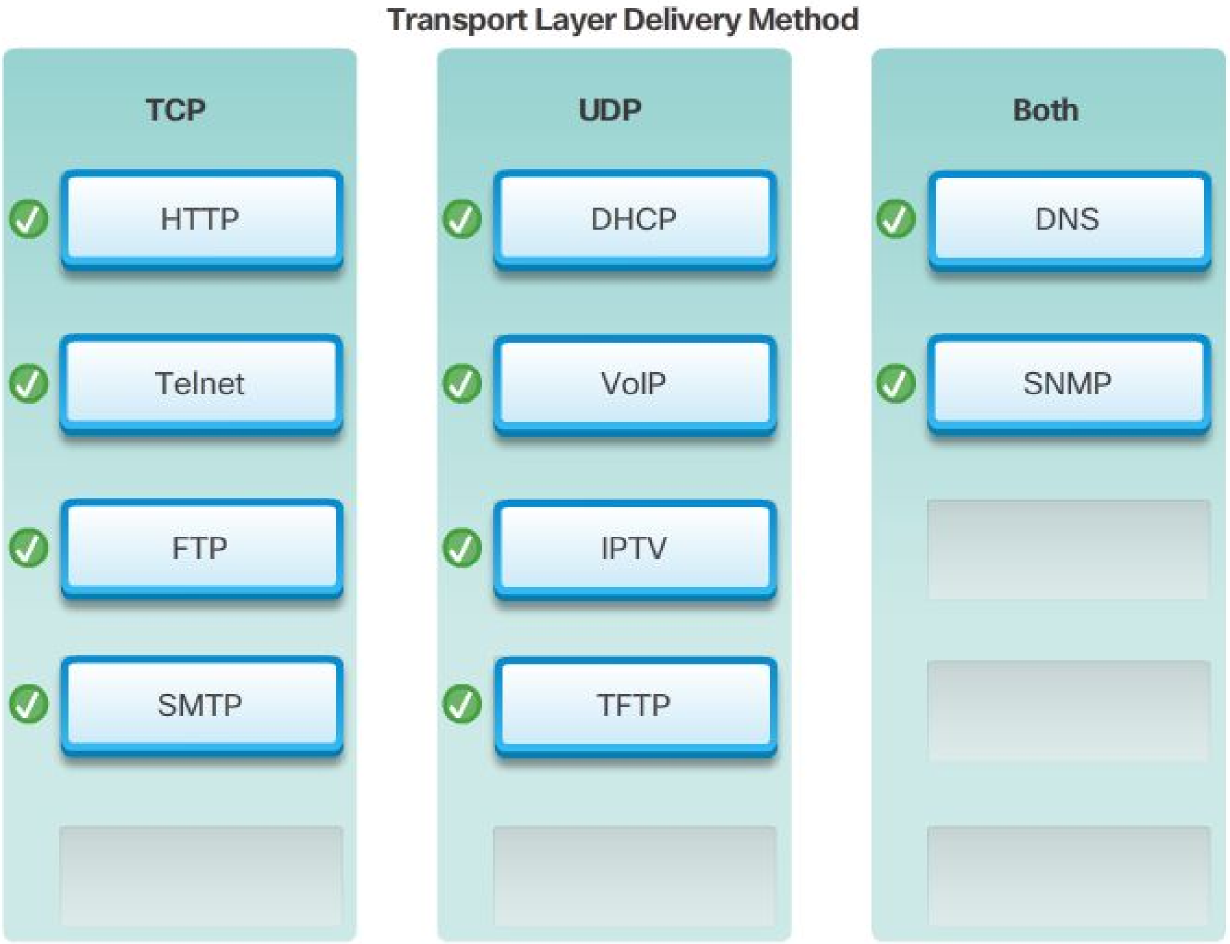
9.2.4.4 Activity TCP, UDP or Both



Activity - TCP, UDP, or Both

Drag the application layer protocol to its transport layer delivery method - TCP, UDP, or Both.

9.2.4.4 Activity TCP, UDP or Both



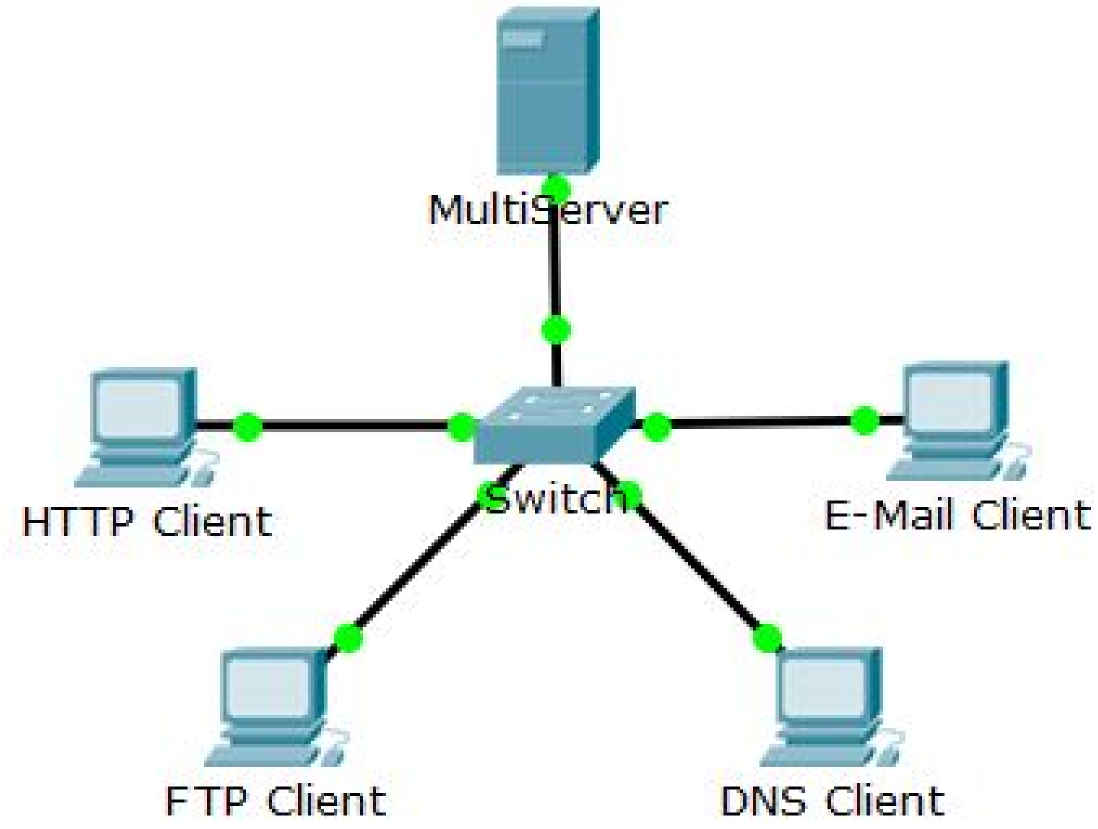
Transport Layer Delivery Method

TCP	UDP	Both
✓ HTTP	✓ DHCP	✓ DNS
✓ Telnet	✓ VoIP	✓ SNMP
✓ FTP	✓ IPTV	
✓ SMTP	✓ TFTP	

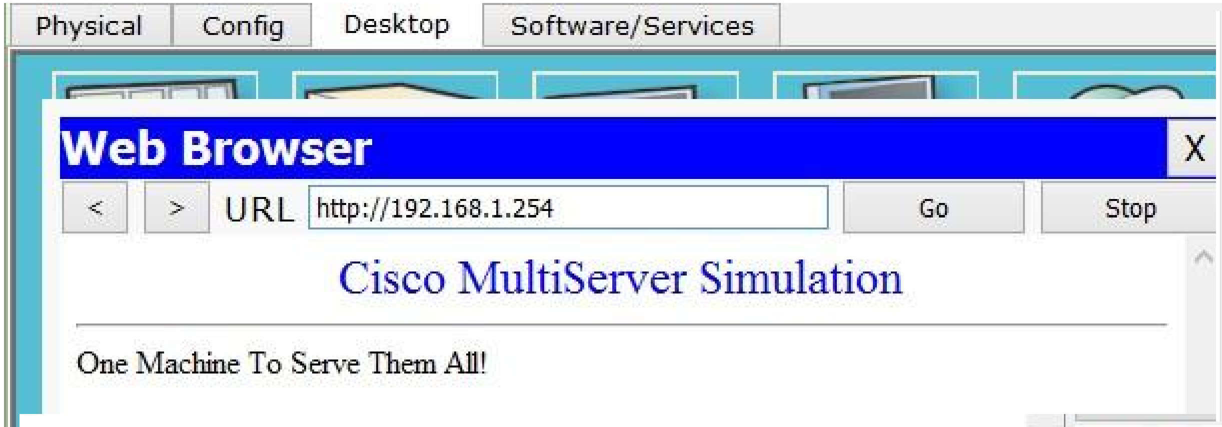
Topic 9.3.1: Conclusion

9.2.4 Conclusion

9.3.1.2 Packet Tracer – TCP and UDP Communications



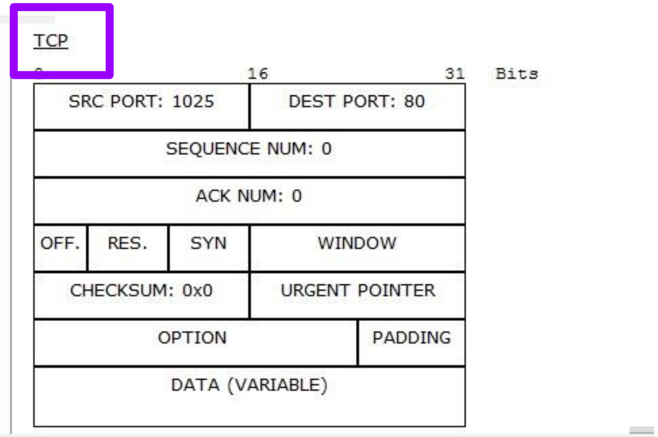
9.3.1.2 Packet Tracer – TCP and UDP Communications



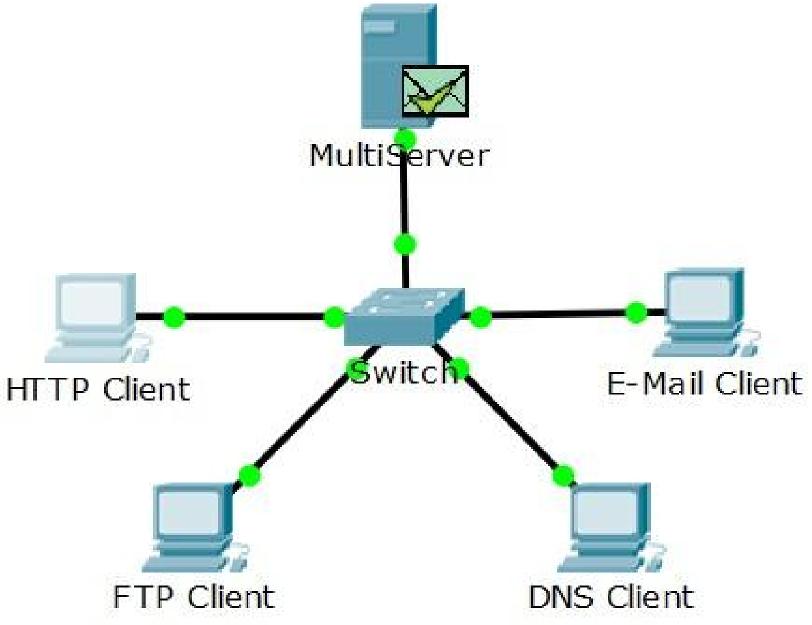
URL http://192.168.1.254

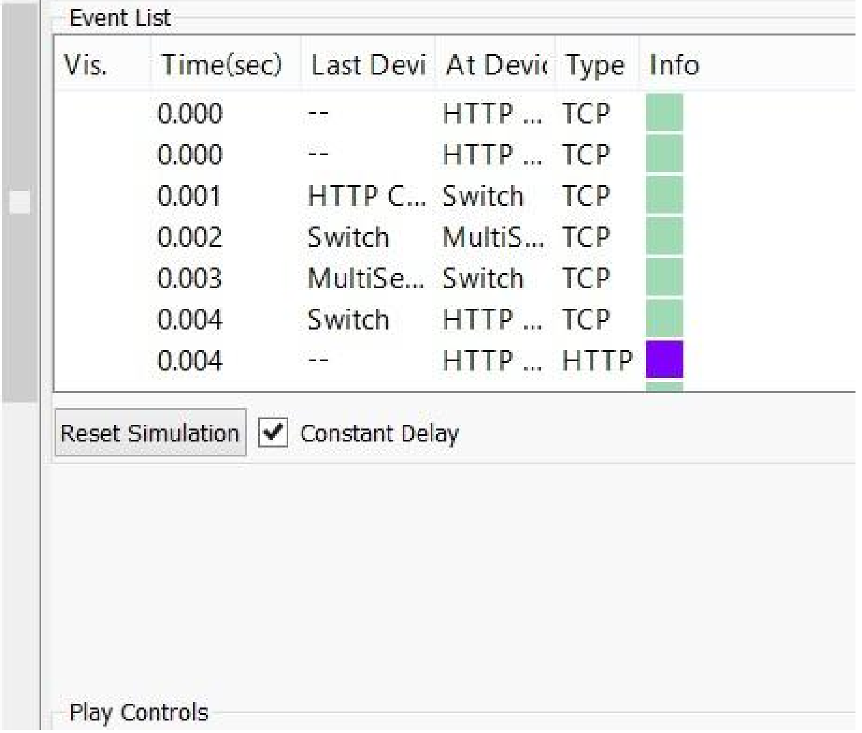
Cisco MultiServer Simulation

One Machine To Serve Them All!



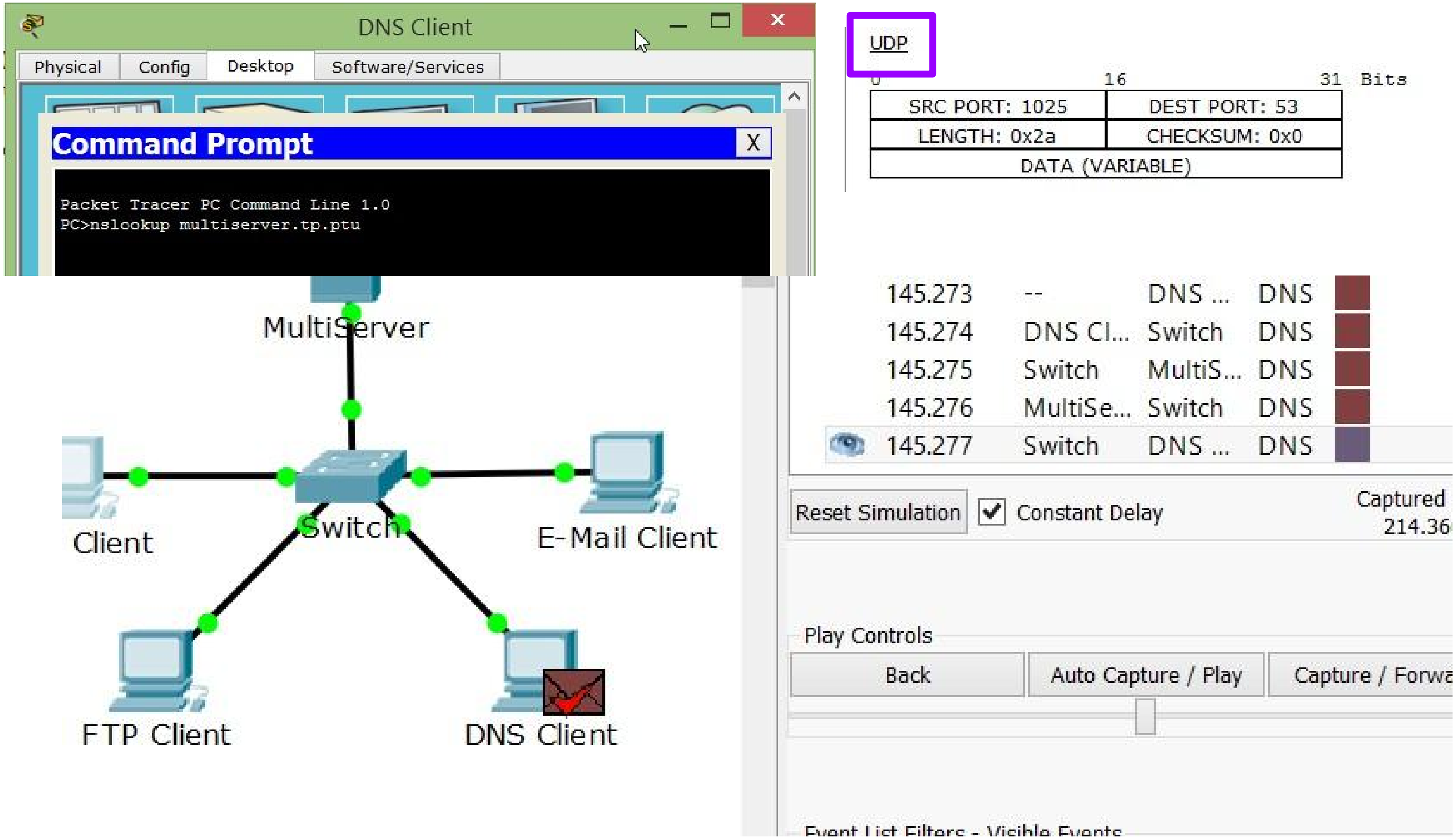
SRC PORT: 1025		DEST PORT: 80	
SEQUENCE NUM: 0			
ACK NUM: 0			
OFF.	RES.	SYN	WINDOW
CHECKSUM: 0x0		URGENT POINTER	
OPTION			PADDING
DATA (VARIABLE)			





Vis.	Time(sec)	Last Devi	At Devi	Type	Info
	0.000	--	HTTP ...	TCP	🟩
	0.000	--	HTTP ...	TCP	🟩
	0.001	HTTP C...	Switch	TCP	🟩
	0.002	Switch	MultiS...	TCP	🟩
	0.003	MultiSe...	Switch	TCP	🟩
	0.004	Switch	HTTP ...	TCP	🟩
	0.004	--	HTTP ...	HTTP	🟪

9.3.1.2 Packet Tracer – TCP and UDP Communications



```
Packet Tracer PC Command Line 1.0
PC>nslookup multiserver.tp.ptu
```

0	16	31 Bits
SRC PORT: 1025	DEST PORT: 53	
LENGTH: 0x2a	CHECKSUM: 0x0	
DATA (VARIABLE)		

145.273	--	DNS ...	DNS
145.274	DNS Cl...	Switch	DNS
145.275	Switch	MultiS...	DNS
145.276	MultiSe...	Switch	DNS
145.277	Switch	DNS ...	DNS

Reset Simulation ☑ Constant Delay

Captured
214.36

Thank you.

Cisco Networking Academy
Mind Wide Open