Transport Layer

Cisco Networking Academy®
Mind Wide Open™

7.0 Introduction

7.1 Transport Layer Protocols

7.2 TCP and UDP

Chapter 7: Objectives

- Describe the purpose of the transport layer in managing the transportation of data in end-to-end communication.
- Describe characteristics of the TCP and UDP protocols, including port numbers and their uses.
- Explain how TCP session establishment and termination processes facilitate reliable communication.
- Explain how TCP protocol data units are transmitted and acknowledged to guarantee delivery.
- Explain the UDP client processes to establish communication with a server.
- Determine whether high-reliability TCP transmissions, or non-guaranteed UDP transmissions, are best suited for common applications.

7.1: Transport Layer Protocols

Cisco | Networking Academy®
Mind Wide Open™

Transportation of Data

Role of the Transport Layer

The transport layer is responsible for establishing a temporary communication session between two applications and delivering data between them.

TCP/IP uses two protocols to achieve this:

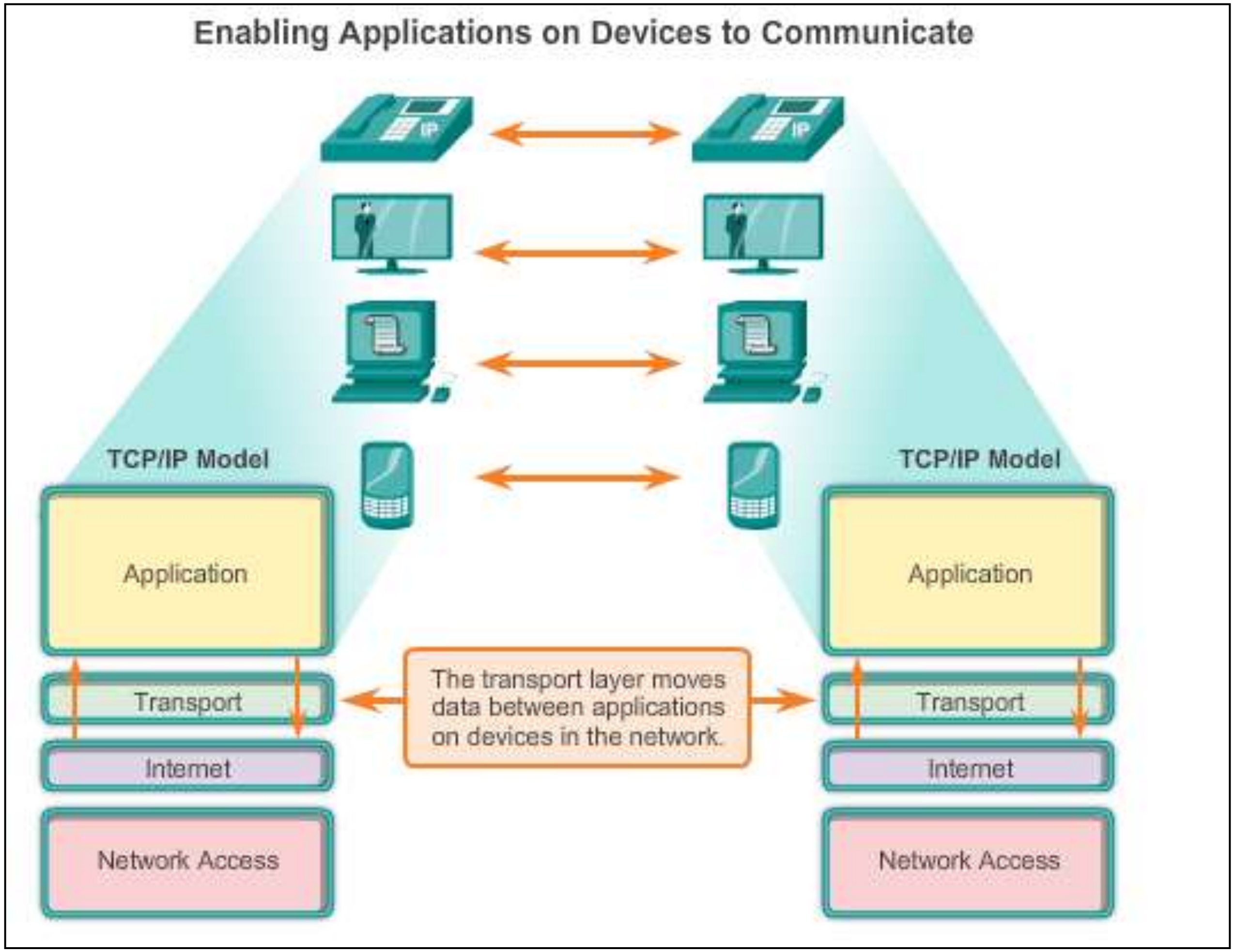
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

Primary Responsibilities of Transport Layer Protocols

- Tracking the individual communication between applications on the source and destination hosts
- Segmenting data for manageability and reassembling segmented data into streams of application data at the destination
- Identifying the proper application for each communication stream

Transportation of Data

Role of the Transport Layer (Cont.)



Enabling Applications on Devices to Communicate

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

Transportation of Data

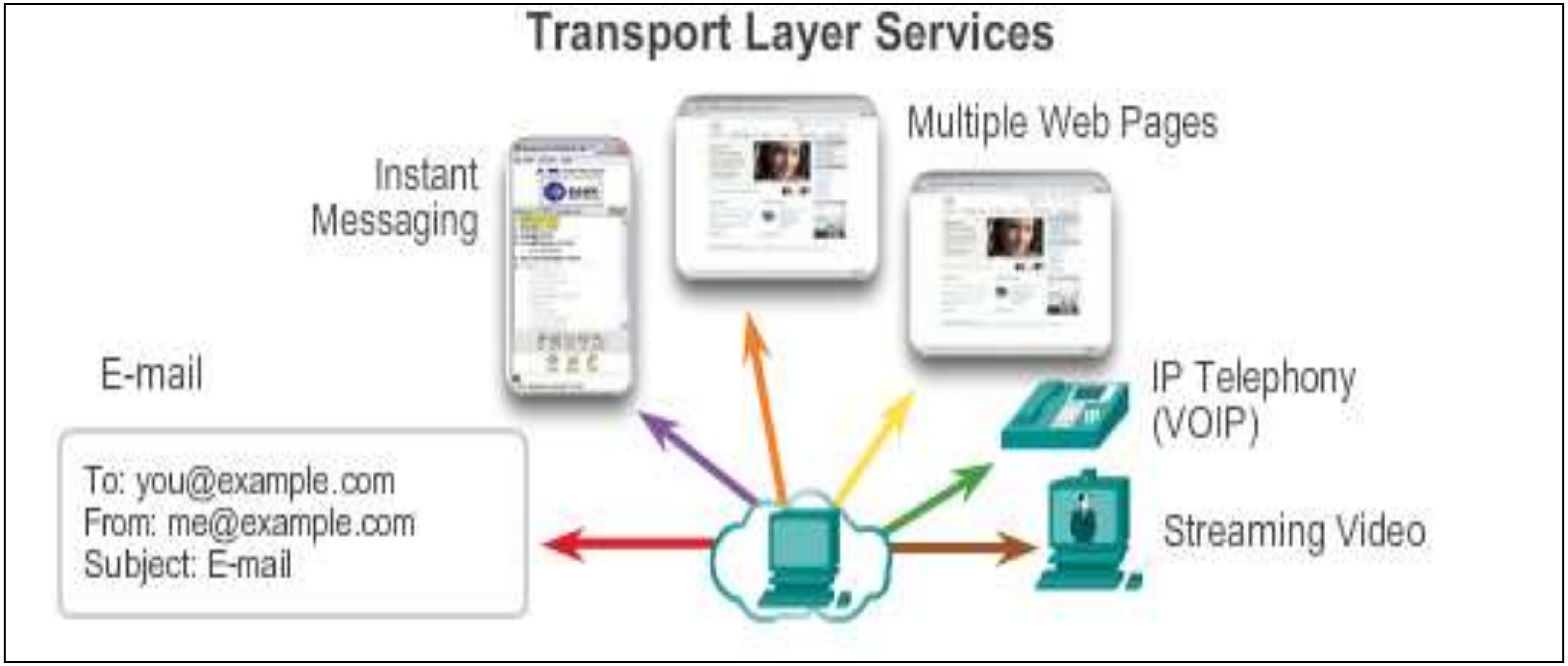
Conversation Multiplexing

Segmenting the Data

- Enables many different communications, from many different users, to be interleaved (multiplexed) on the same network, at the same time.
- Provides the means to both send and receive data when running multiple applications.
- Header added to each segment to identify it.

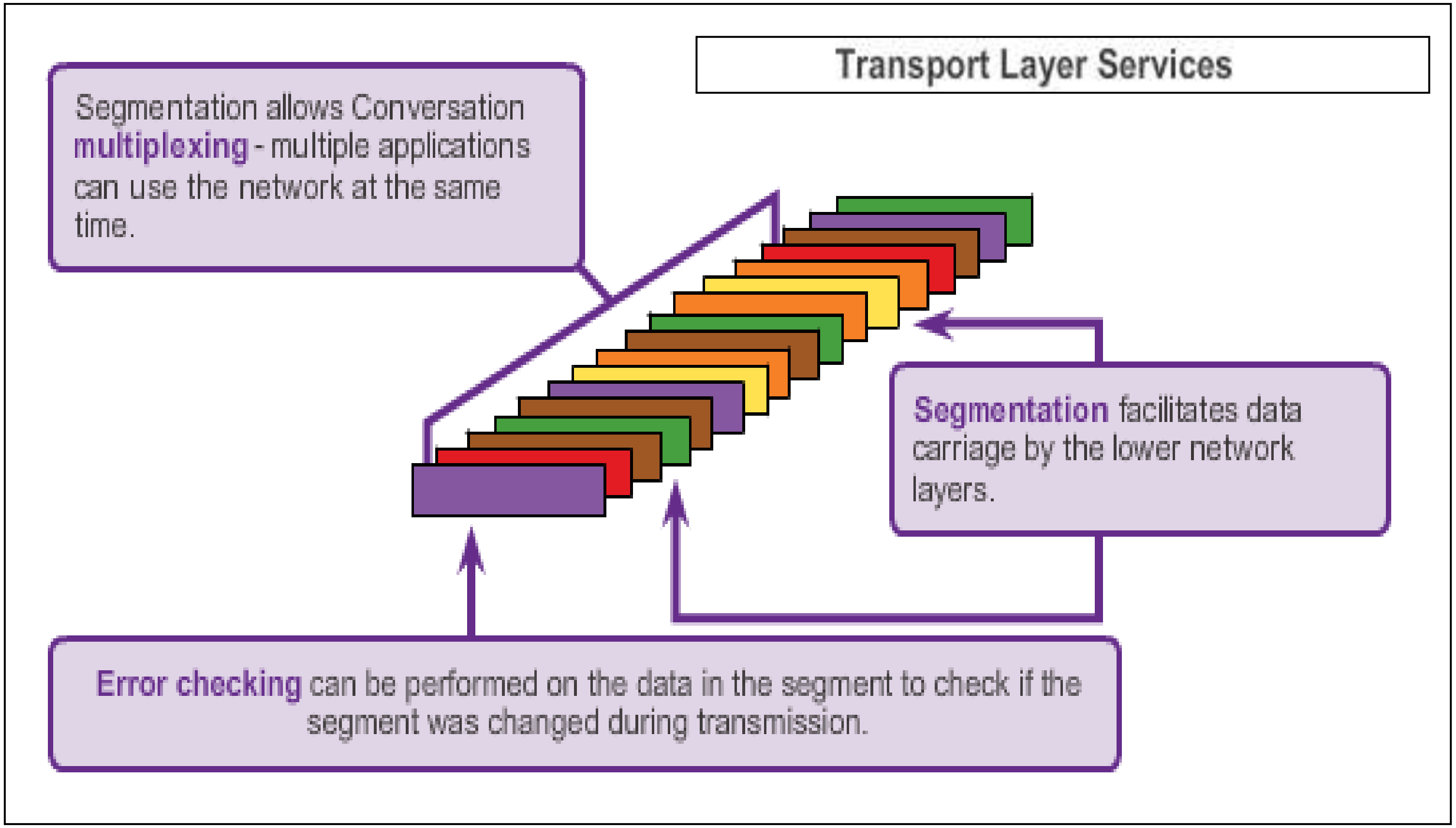
Transportation of Data

Conversation Multiplexing (Cont.)



To: you@example.com
From: me@example.com
Subject: E-mail

Conversation Multiplexing (Cont.)



Transport Layer Services

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

Segmentation facilitates data carriage 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.

Transportation of Data

Transport Layer Reliability

Different applications have different transport reliability requirements.

TCP/IP provides two transport layer protocols, **TCP and UDP.**

TCP

- Provides reliable delivery ensuring that all of the data arrives at the destination.
- Uses acknowledged delivery and other processes to ensure delivery
- Makes larger demands on the network – more overhead.

UDP

- Provides just the basic functions for delivery – no reliability.
- Less overhead.

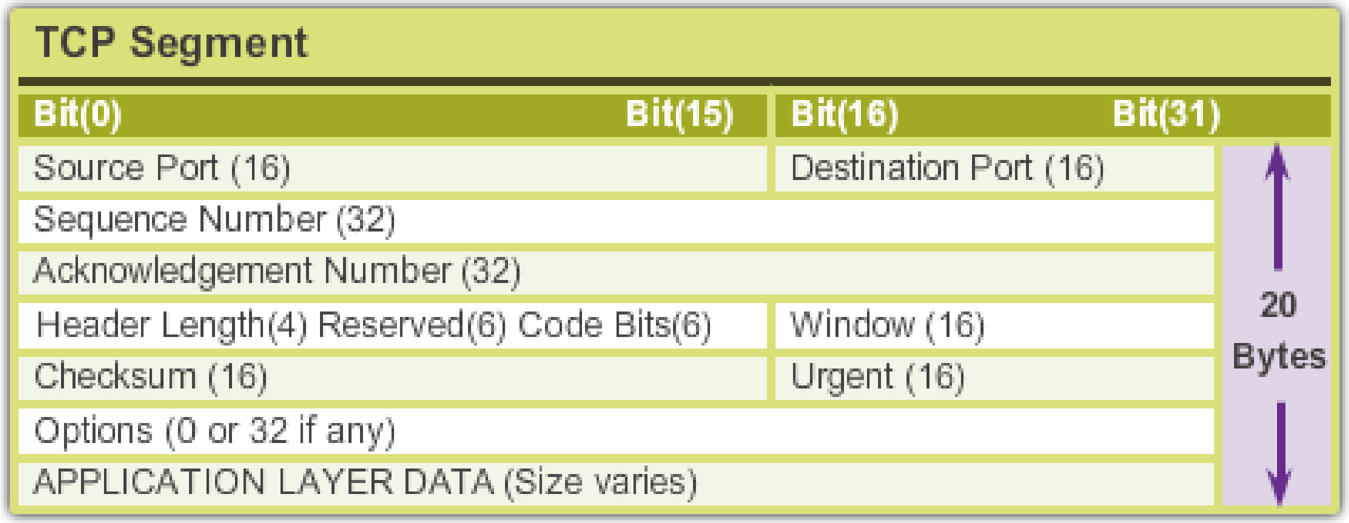
TCP or UDP

- There is a trade-off between the value of reliability and the burden it places on the network.
- Application developers choose the transport protocol based on the requirements of their applications.

Introducing TCP and UDP

Introducing TCP

- Defined in RFC 793
- Connection-oriented – Creates a session between the source and destination
- Reliable delivery – Retransmits lost or corrupt data
- Ordered data reconstruction – Reconstructs numbering and sequencing of segments
- Flow control – Regulates the amount of data transmitted
- Stateful protocol – Tracks the session



TCP Segment

Bit(0) Bit(15)	Bit(16) Bit(31)	
Source Port (16)	Destination Port (16)	20 Bytes
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)		

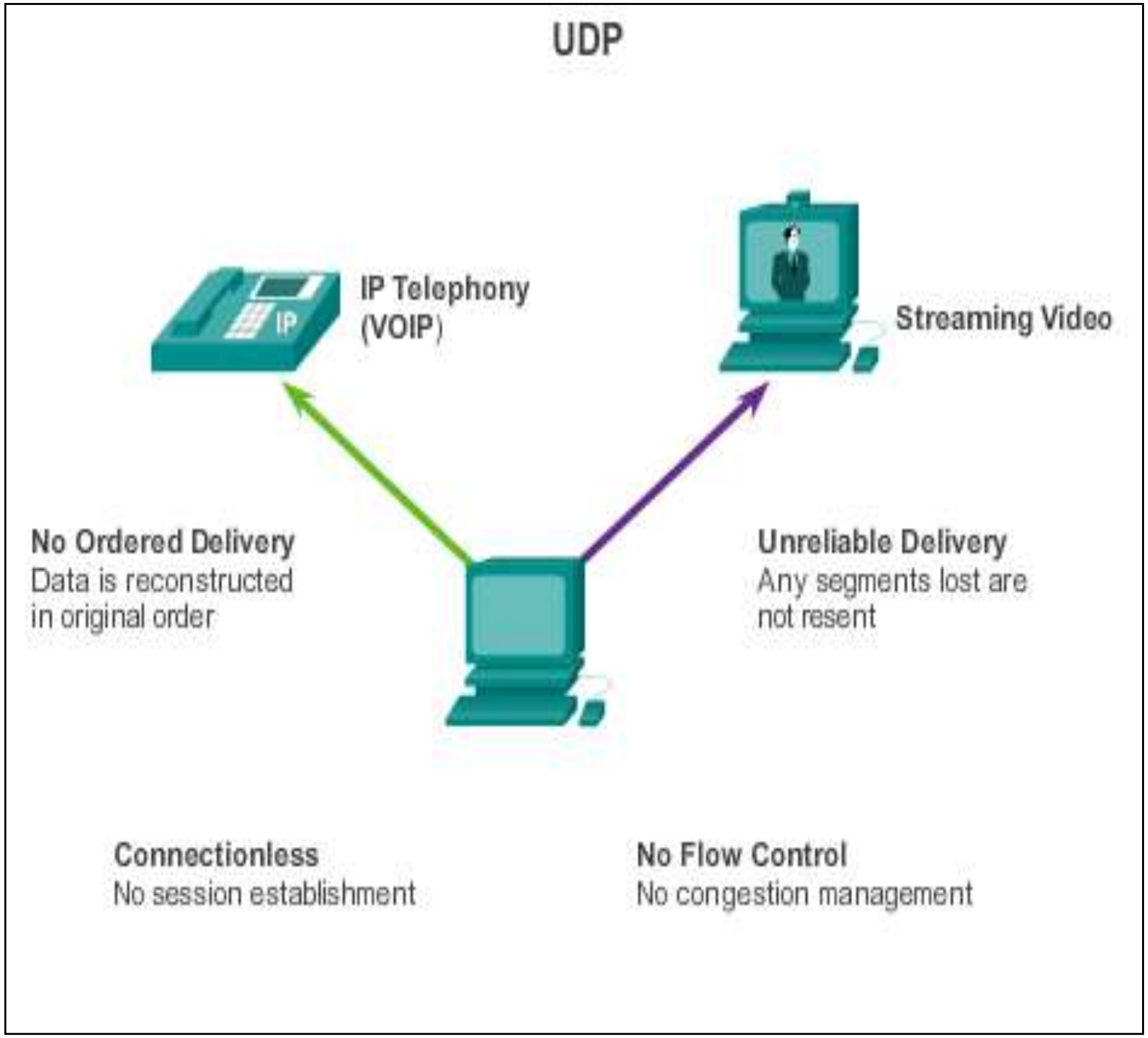
Introducing TCP and UDP

Introducing UDP

- RFC 768
- Connectionless
- Unreliable delivery
- No ordered data reconstruction
- No flow control
- Stateless protocol

Applications that use UDP:

- Domain Name System (DNS)
- Video Streaming
- VoIP

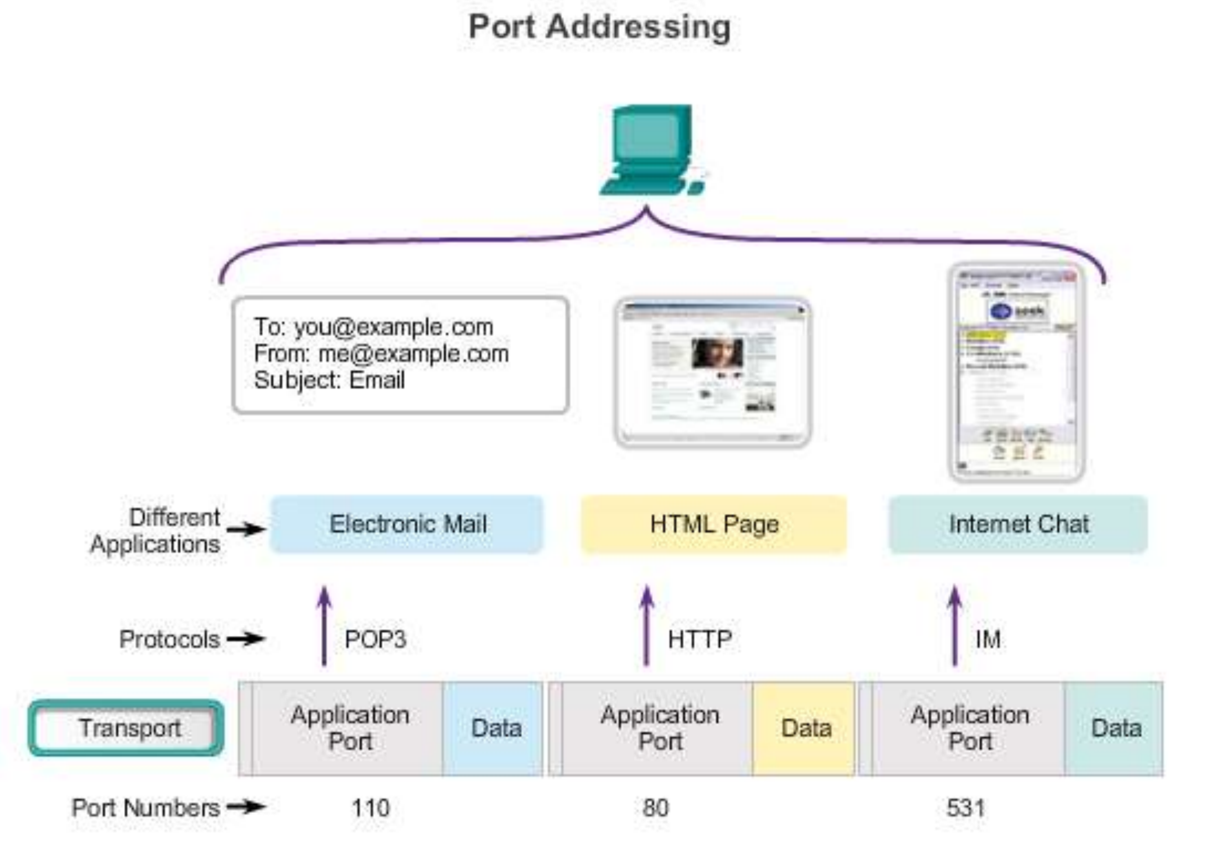


Introducing TCP and UDP

Separating Multiple Communications

TCP and UDP use port numbers to differentiate between applications.

.



Port Addressing

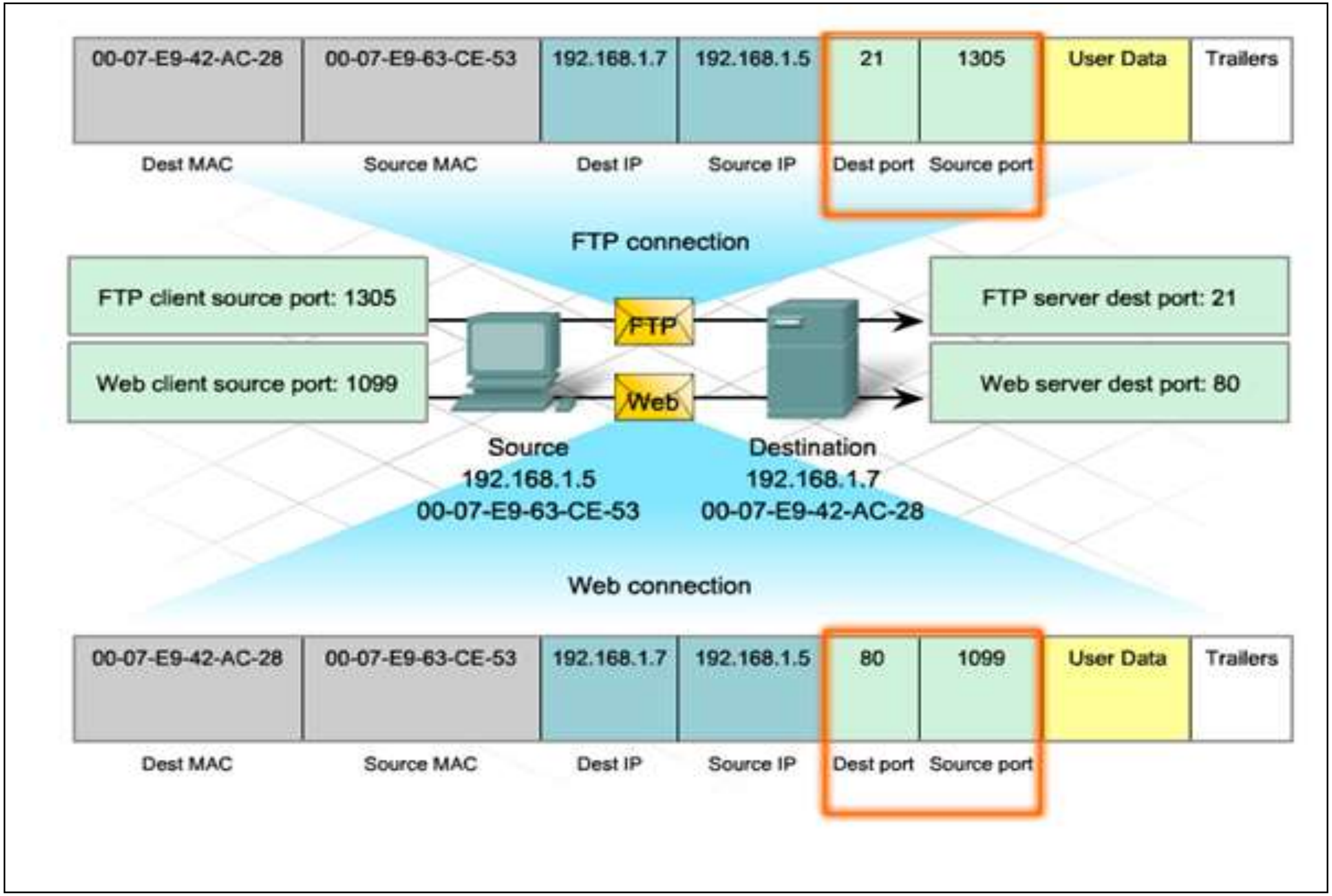
To: you@example.com
From: me@example.com
Subject: Email

Different Applications →	Electronic Mail	HTML Page	Internet Chat
Protocols →	POP3	HTTP	IM
Transport	Application Port | Data	Application Port | Data	Application Port | Data
Port Numbers →	110	80	531

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

Introducing TCP and UDP

TCP and UDP Port Addressing



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

FTP connection

FTP client source port: 1305

Web client source port: 1099

FTP server dest port: 21

Web server dest port: 80

Source
192.168.1.5
00-07-E9-63-CE-53

Destination
192.168.1.7
00-07-E9-42-AC-28

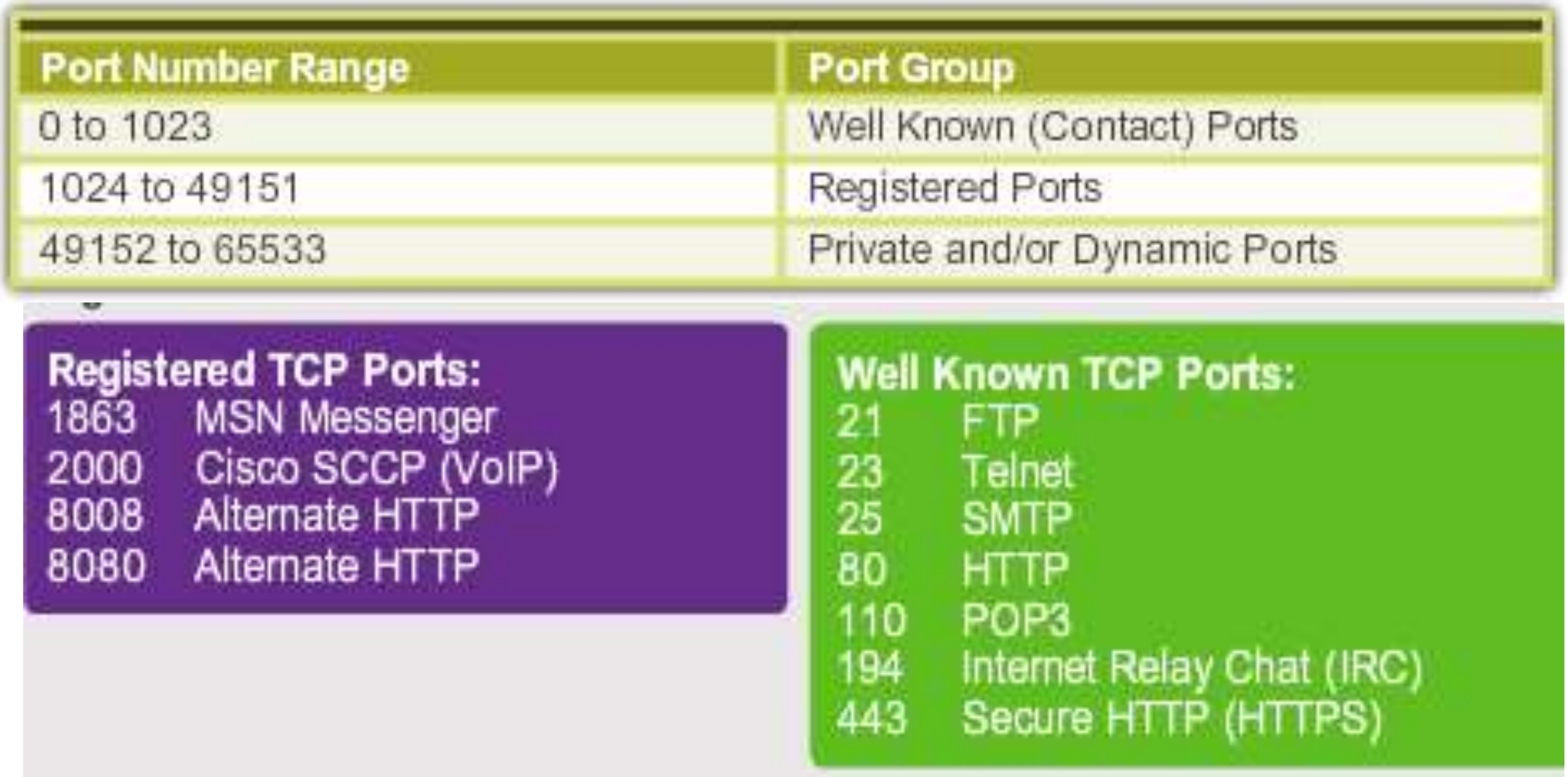
Web connection

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

Introducing TCP and UDP

TCP and UDP Port Addressing (Cont.)

Port Numbers



Port Number Range	Port Group
0 to 1023	Well Known (Contact) Ports
1024 to 49151	Registered Ports
49152 to 65533	Private and/or Dynamic Ports

Registered TCP Ports:

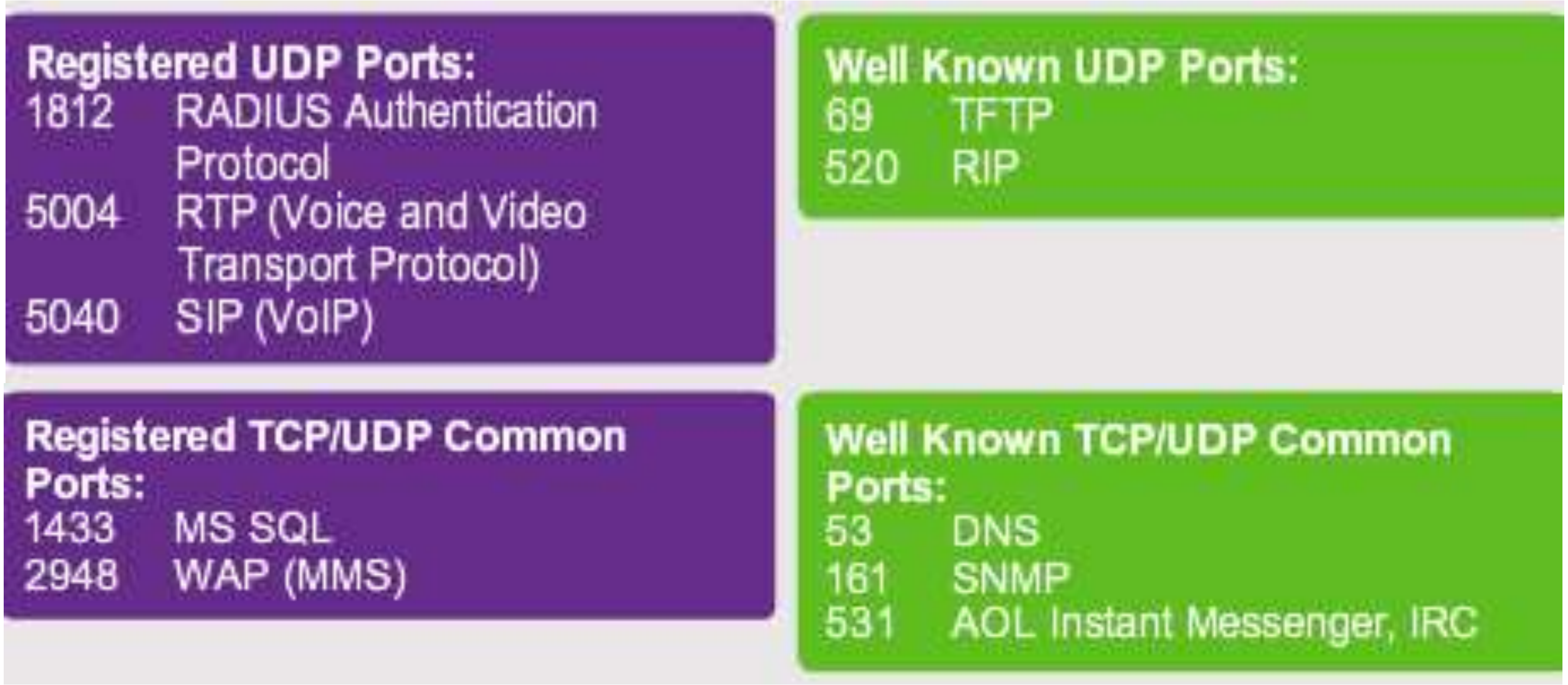
1863	MSN Messenger
2000	Cisco SCCP (VoIP)
8008	Alternate HTTP
8080	Alternate HTTP

Well Known TCP Ports:

21	FTP
23	Telnet
25	SMTP
80	HTTP
110	POP3
194	Internet Relay Chat (IRC)
443	Secure HTTP (HTTPS)

Introducing TCP and UDP

TCP and UDP Port Addressing (Cont.)



Registered UDP Ports:

1812	RADIUS Authentication Protocol
5004	RTP (Voice and Video Transport Protocol)
5040	SIP (VoIP)

Well Known UDP Ports:

69	TFTP
520	RIP

Registered TCP/UDP Common Ports:

1433	MS SQL
2948	WAP (MMS)

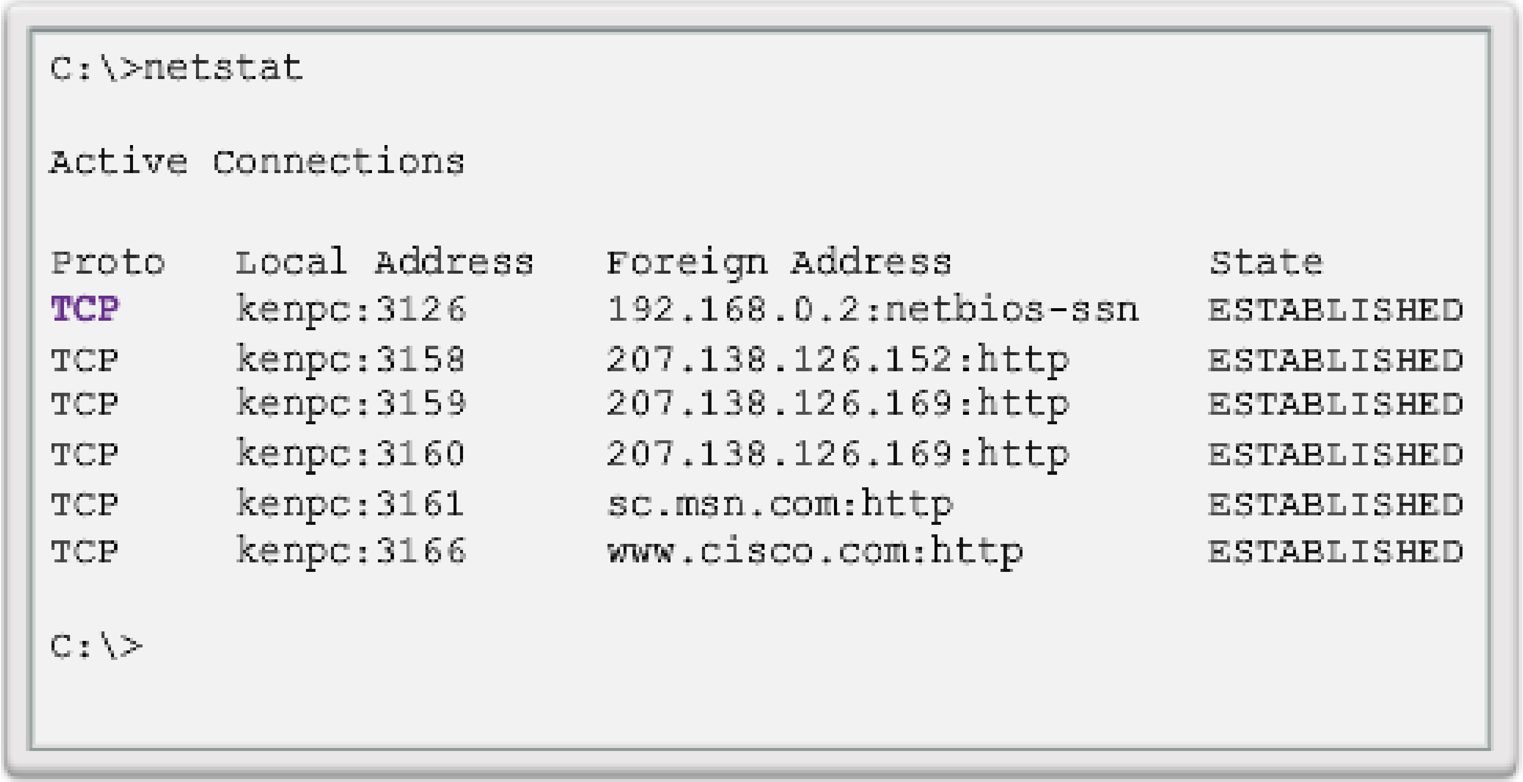
Well Known TCP/UDP Common Ports:

53	DNS
161	SNMP
531	AOL Instant Messenger, IRC

Introducing TCP and UDP

TCP and UDP Port Addressing (Cont.)

Netstat is used to examine TCP connections that are open and running on a networked host.

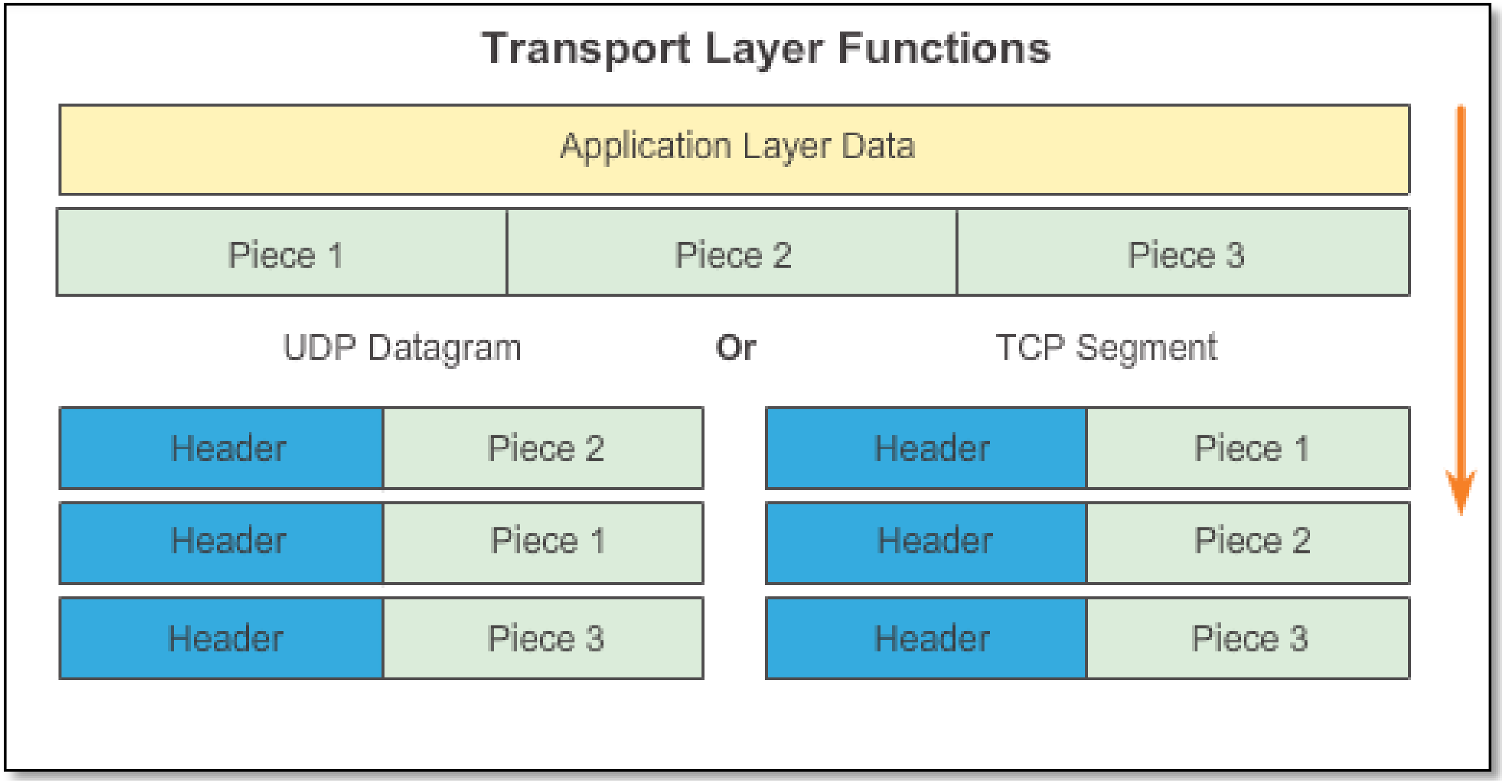


```
C:\>netstat

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:\>
```

Introducing TCP and UDP

TCP and UDP Segmentation

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



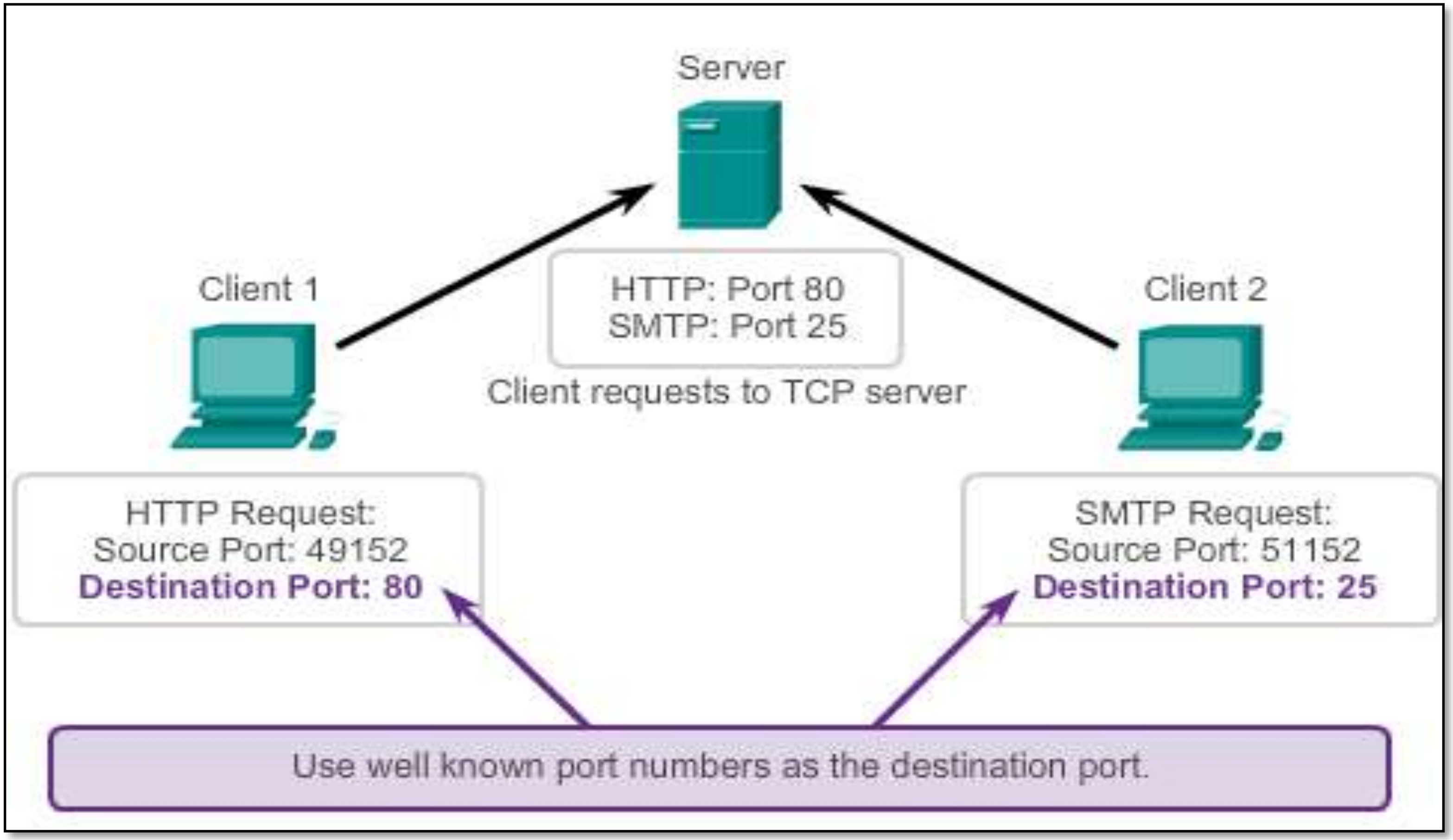
7.2 TCP and UDP

Cisco | Networking Academy®
Mind Wide Open™

TCP Communication

TCP Server Processes

Request Destination Ports



Server

HTTP: Port 80
SMTP: Port 25

Client requests to TCP server

Client 1

HTTP Request:
Source Port: 49152
Destination Port: 80

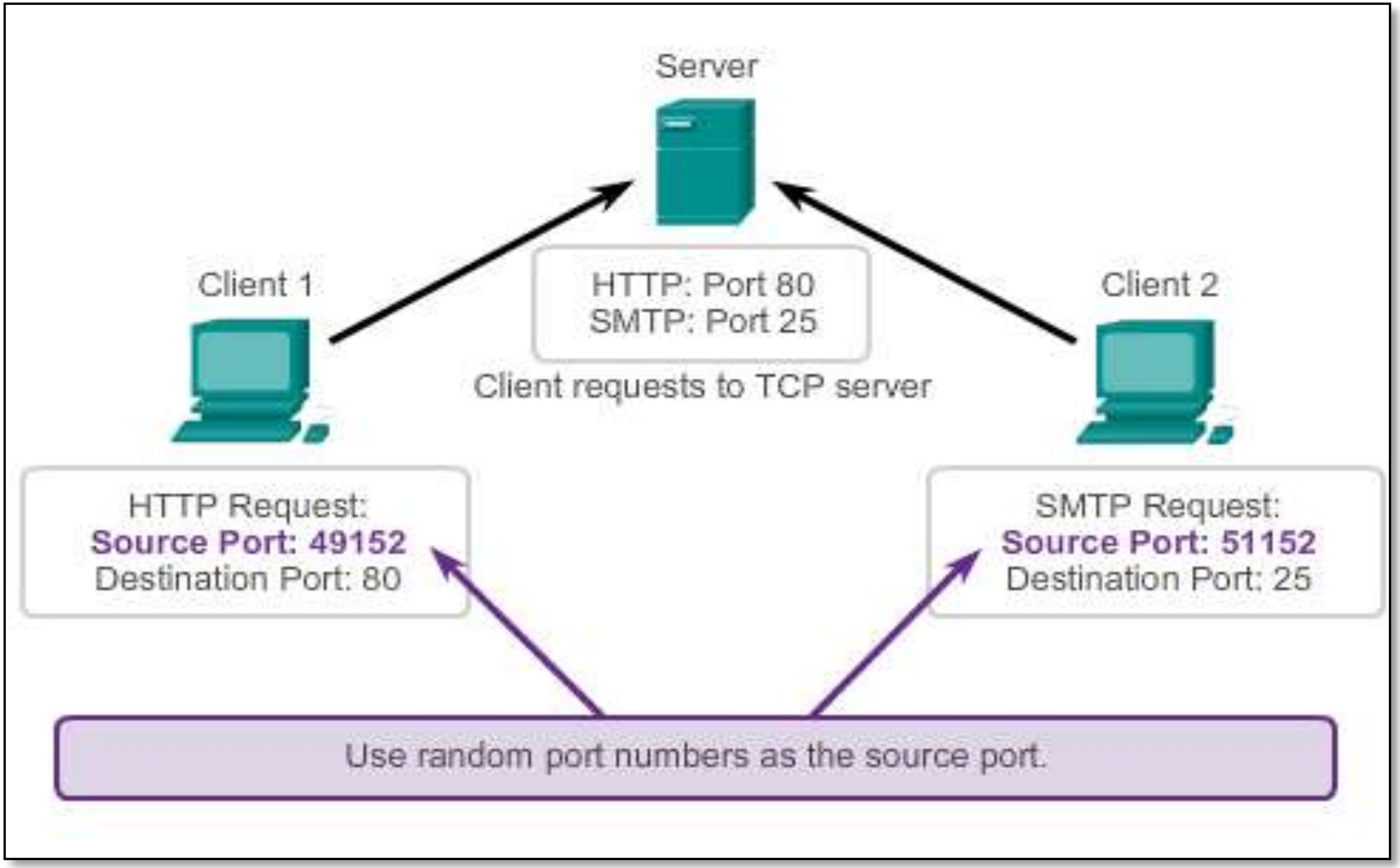
Client 2

SMTP Request:
Source Port: 51152
Destination Port: 25

Use well known port numbers as the destination port.

TCP Communication

TCP Server Processes (Cont.)



Server

HTTP: Port 80
SMTP: Port 25

Client requests to TCP server

Client 1

Client 2

HTTP Request:
Source Port: 49152
Destination Port: 80

SMTP Request:
Source Port: 51152
Destination Port: 25

Use random port numbers as the source port.

TCP Communication
TCP Connection, Establishment and Termination

Three-Way Handshake

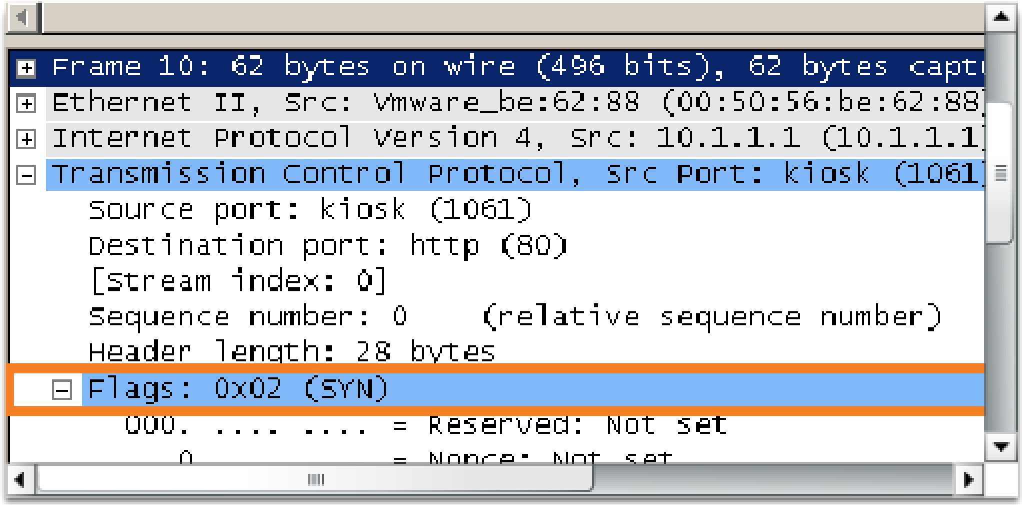
- Establishes that the destination device is present on the network
- Verifies that the destination device has an active service and is accepting requests on the destination port number that the initiating client intends to use for the session
- Informs the destination device that the source client intends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

TCP Communication

TCP Three-Way Handshake – Step 1

Step 1: The initiating client requests a client-to-server communication session with the server

TCP 3-Way Handshake (SYN)



```
⊞ Frame 10: 62 bytes on wire (496 bits), 62 bytes captu
⊞ Ethernet II, Src: Vmware_be:62:88 (00:50:56:be:62:88)
⊞ Internet Protocol Version 4, Src: 10.1.1.1 (10.1.1.1)
⊟ Transmission Control Protocol, Src Port: kiosk (1061)
    Source port: kiosk (1061)
    Destination port: http (80)
    [Stream index: 0]
    Sequence number: 0    (relative sequence number)
    Header length: 28 bytes
  ⊟ Flags: 0x02 (SYN)
      000. .... .... = Reserved: Not set
         0           = Nonce: Not set
```

A protocol analyzer shows initial client request for session in frame 10

TCP segment in this frame shows:

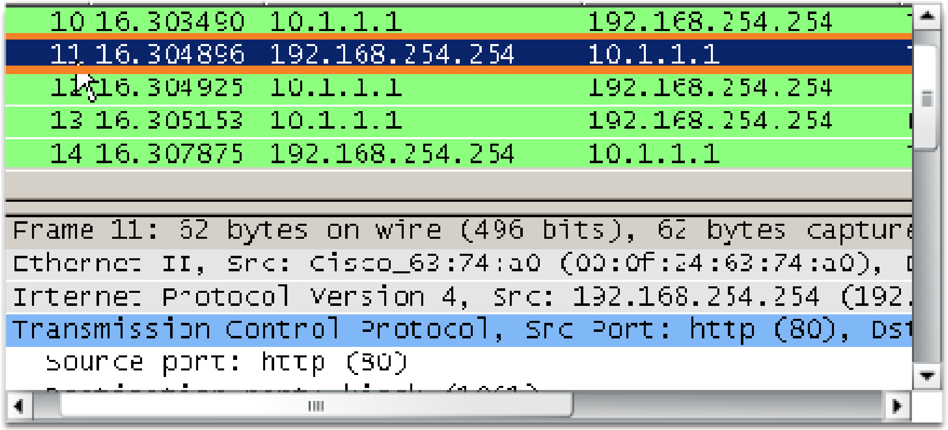
- SYN flag set to validate an Initial Sequence Number
- Randomized sequence number valid (relative value is 0)
- Random source port 1061
- Well-known destination port is 80 (HTTP port) indicates web server (httpd)

TCP Communication

TCP Three-Way Handshake – Step 2

Step 2: The server acknowledges the client-to-server communication session and requests a server-to-client communication session.

TCP 3-Way Handshake (SYN, ACK)



```
10 16.303490  10.1.1.1            192.168.254.254
11 16.304896  192.168.254.254     10.1.1.1
12 16.304925  10.1.1.1            192.168.254.254
13 16.305153  10.1.1.1            192.168.254.254
14 16.307875  192.168.254.254     10.1.1.1

Frame 11: 62 bytes on wire (496 bits), 62 bytes capture
Ethernet II, Src: Cisco_63:74:a0 (00:0f:24:63:74:a0), D
Internet Protocol Version 4, Src: 192.168.254.254 (192.
Transmission Control Protocol, Src Port: http (80), Dst
  Source port: http (80)
```

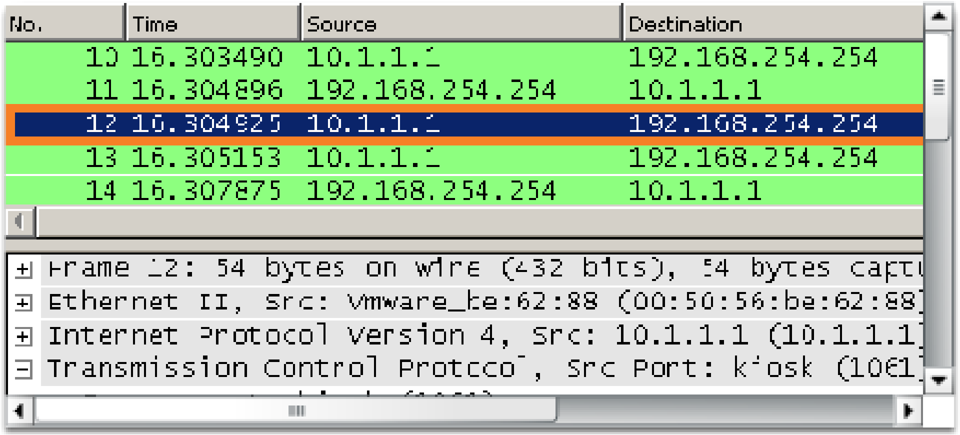
A protocol analyzer shows server response in frame 11

- ACK flag set to indicate a valid Acknowledgement number
- Acknowledgement number response to initial sequence number as relative value of 1
- SYN flag set to indicate the Initial Sequence Number for the server to client session
- Destination port number of 1061 to corresponding to the clients source port
- Source port number of 80 (HTTP) indicating the web server service (httpd)

TCP Three-Way Handshake – Step 3

Step 3: The initiating client acknowledges the server-to-client communication session.

TCP 3-Way Handshake (ACK)



```
No.     Time       Source            Destination
     10 16.303490  10.1.1.1          192.168.254.254
     11 16.304896  192.168.254.254   10.1.1.1
     12 16.304925  10.1.1.1          192.168.254.254
     13 16.305153  10.1.1.1          192.168.254.254
     14 16.307875  192.168.254.254   10.1.1.1

+ Frame 12: 54 bytes on wire (432 bits), 54 bytes captu
+ Ethernet II, Src: Vmware_be:62:88 (00:50:56:be:62:88)
+ Internet Protocol Version 4, Src: 10.1.1.1 (10.1.1.1)
- Transmission Control Protocol, Src Port: kiosk (1061)
```

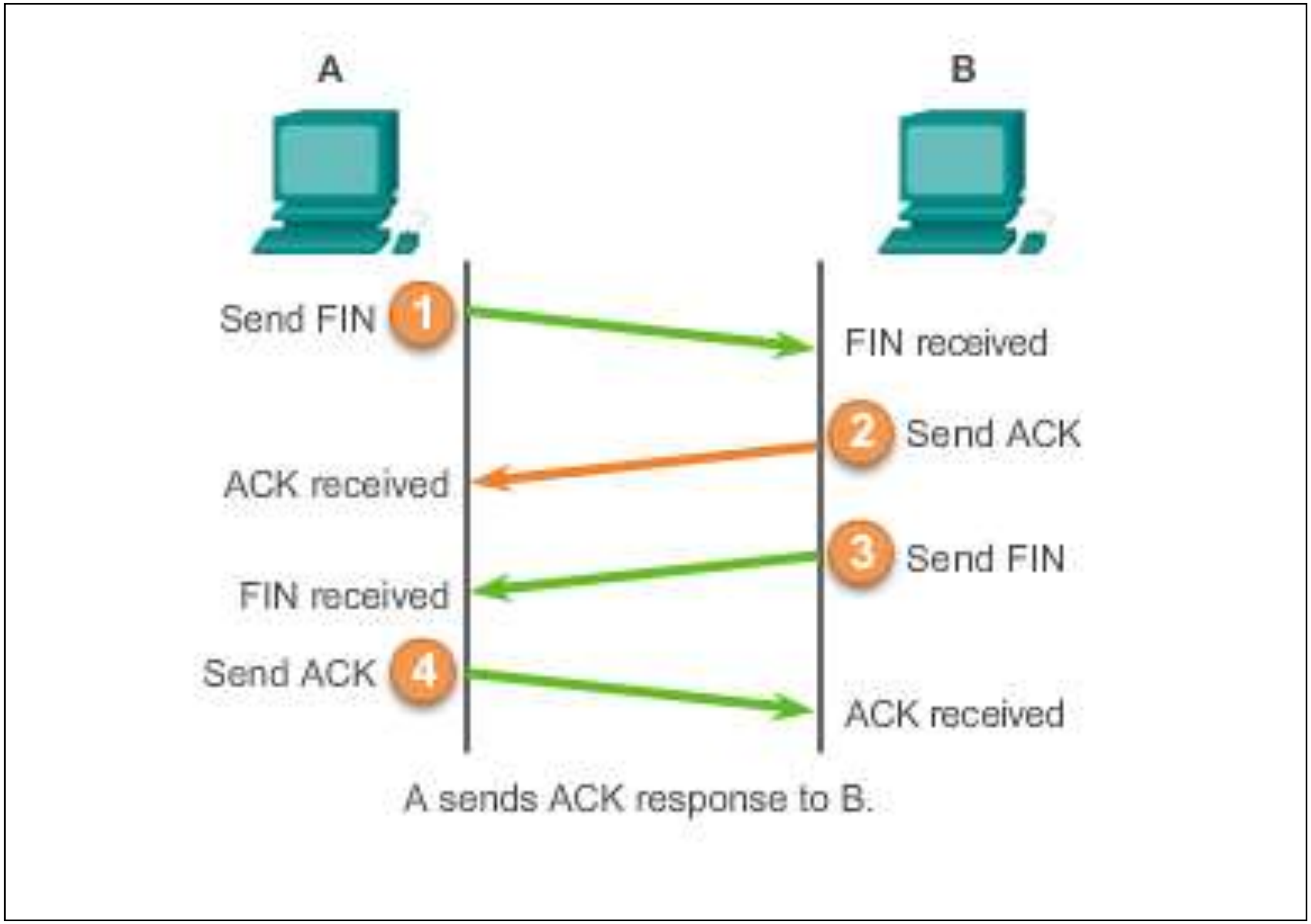
A protocol analyzer shows client response to session in frame 12

The TCP segment in this frame shows:

- ACK flag set to indicate a valid Acknowledgement number
- Acknowledgement number response to initial sequence number as relative value of 1
- Source port number of 1061 to corresponding
- Destination port number of 80 (HTTP) indicating the web server service (httpd)

TCP Communication

TCP Session Termination

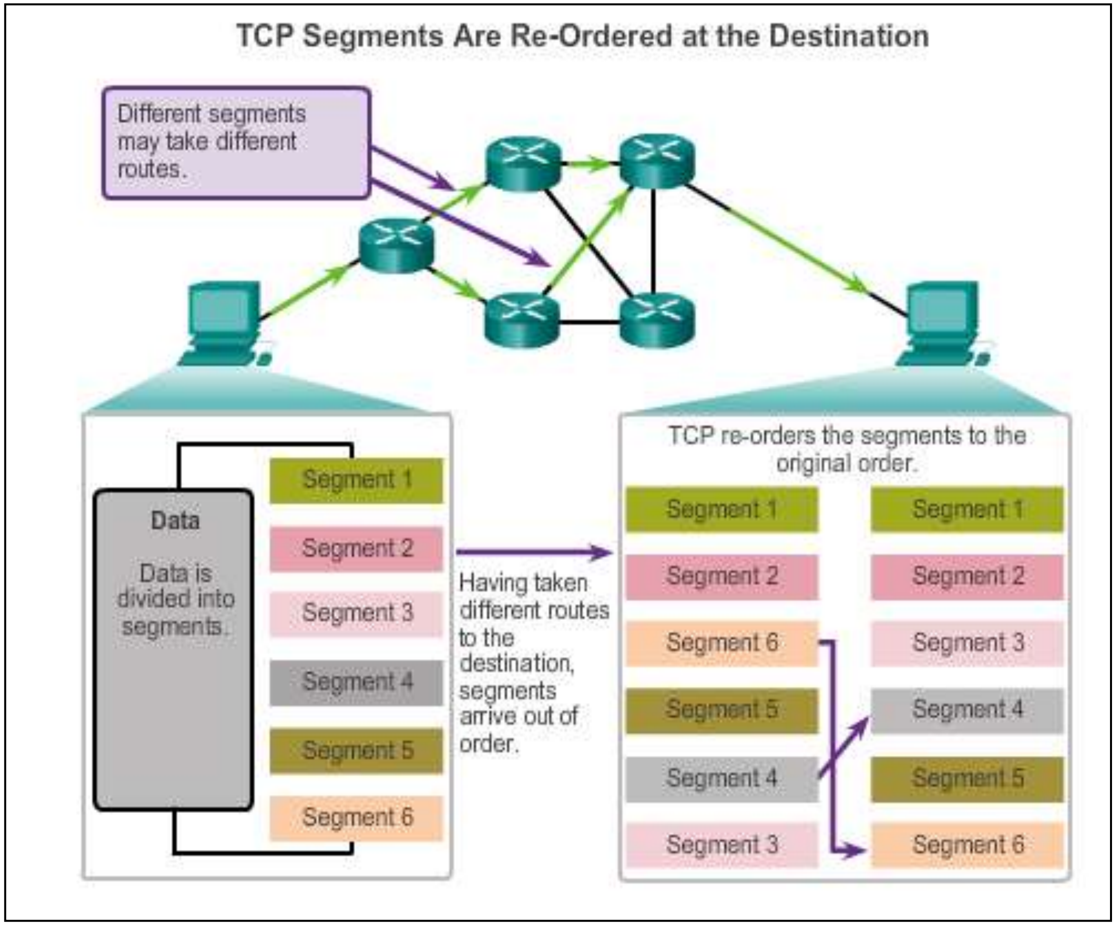


A sends ACK response to B.

Reliability and Flow Control

TCP Reliability – Ordered Delivery

Sequence numbers are used to reassemble segments into their original order.



TCP Segments Are Re-Ordered at the Destination

Different segments may take different routes.

Data

Data is divided into segments.

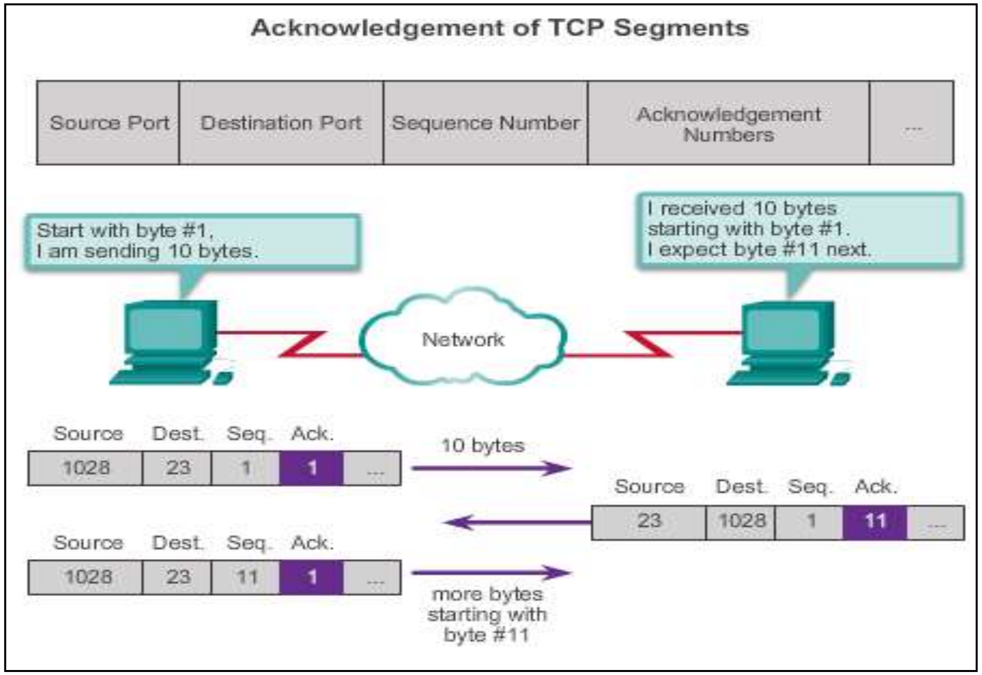
Having taken different routes to the destination, segments arrive out of order.

TCP re-orders the segments to the original order.

Reliability and Flow Control

Acknowledgement and Window Size

The sequence number and acknowledgement number are used together to confirm receipt.



Acknowledgement of TCP Segments

Source Port	Destination Port	Sequence Number	Acknowledgement Numbers	...

Start with byte #1,
I am sending 10 bytes.

I received 10 bytes
starting with byte #1.
I expect byte #11 next.

Source	Dest.	Seq.	Ack.	
1028	23	1	1	...

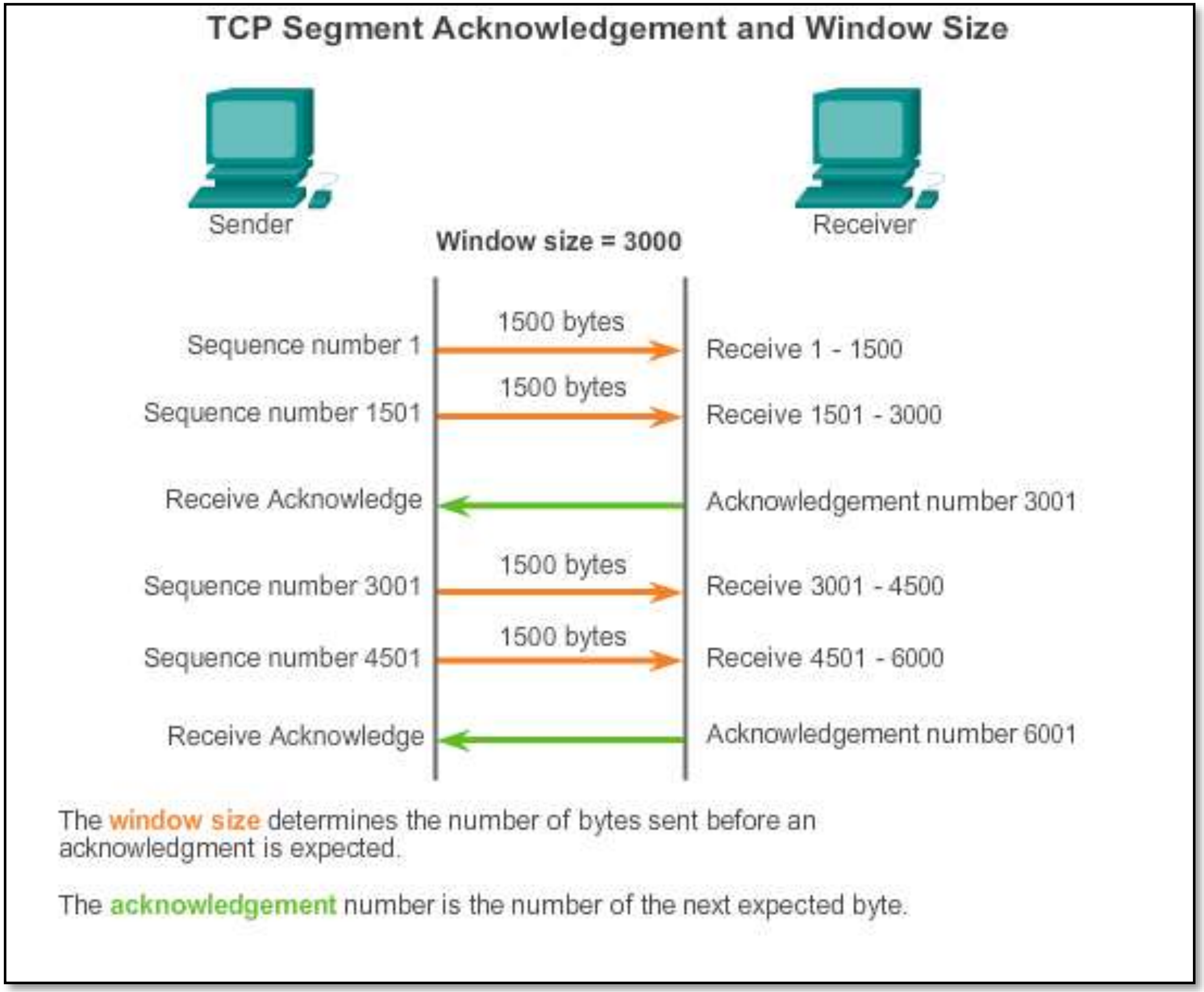
Source	Dest.	Seq.	Ack.	
23	1028	1	11	...

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

The window size is the amount of data that a source can transmit before an acknowledgement must be received.

Reliability and Flow Control

Window Size and Acknowledgements



TCP Segment Acknowledgement and Window Size

Sender

Receiver

Window size = 3000

Sequence number 1 — 1500 bytes — Receive 1 - 1500

Sequence number 1501 — 1500 bytes — Receive 1501 - 3000

Receive Acknowledge — Acknowledgement number 3001

Sequence number 3001 — 1500 bytes — Receive 3001 - 4500

Sequence number 4501 — 1500 bytes — Receive 4501 - 6000

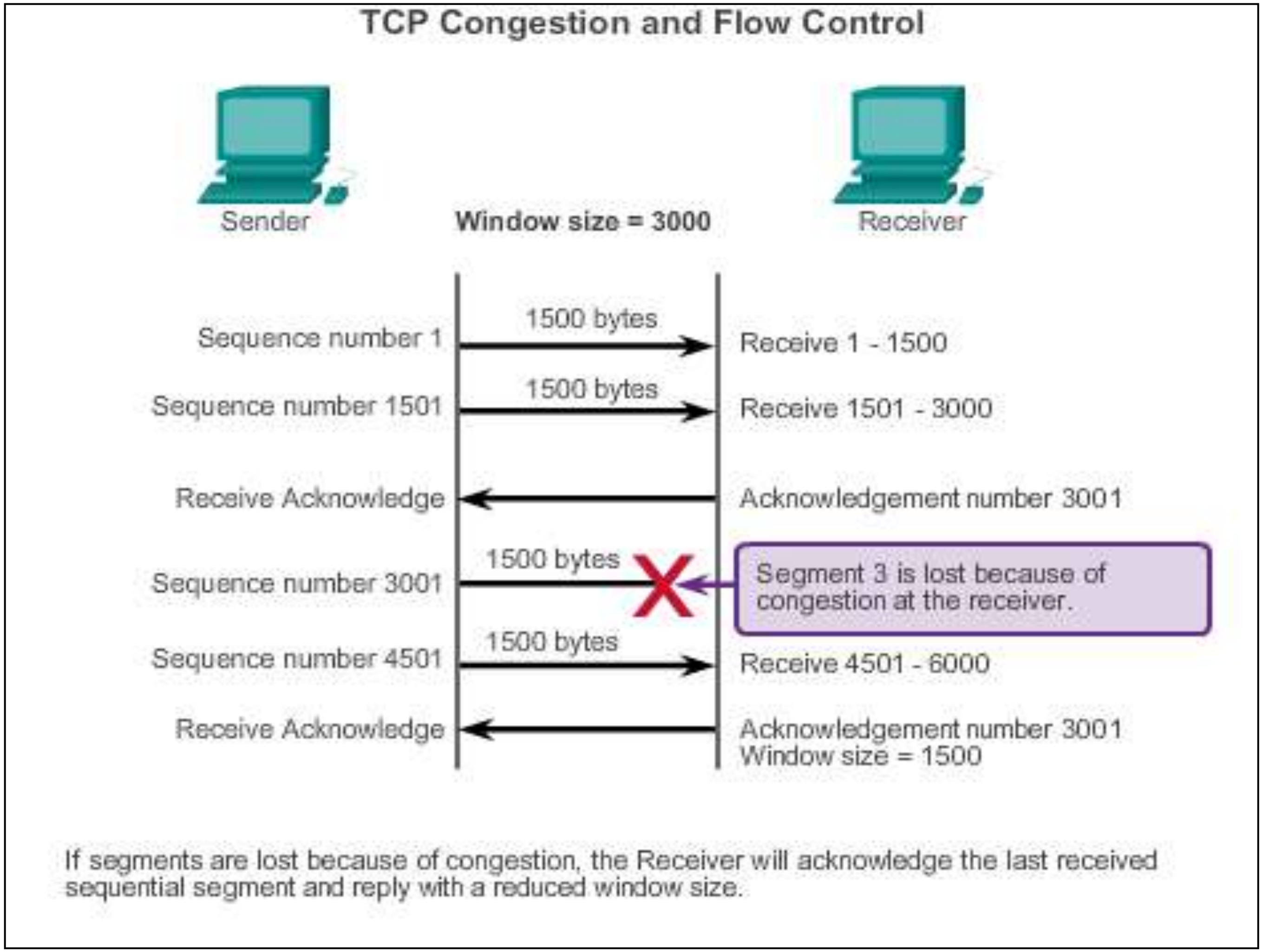
Receive Acknowledge — Acknowledgement number 6001

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.

Reliability and Flow Control

TCP Flow Control – Congestion Avoidance



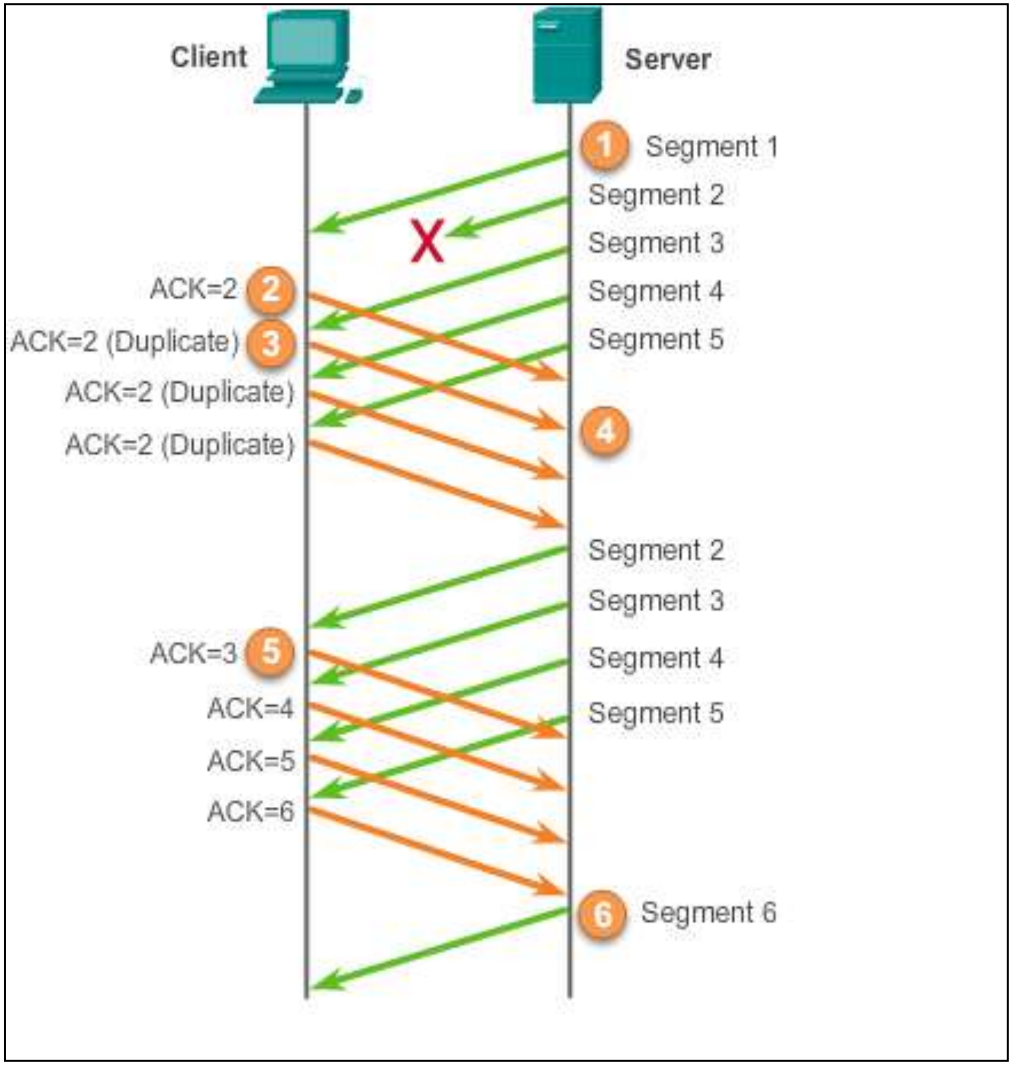
TCP Congestion and Flow Control

Segment 3 is lost because of congestion at the receiver.

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

Reliability and Flow Control

TCP Reliability - Acknowledgements



UDP Communication

UDP Low Overhead vs. Reliability

UDP

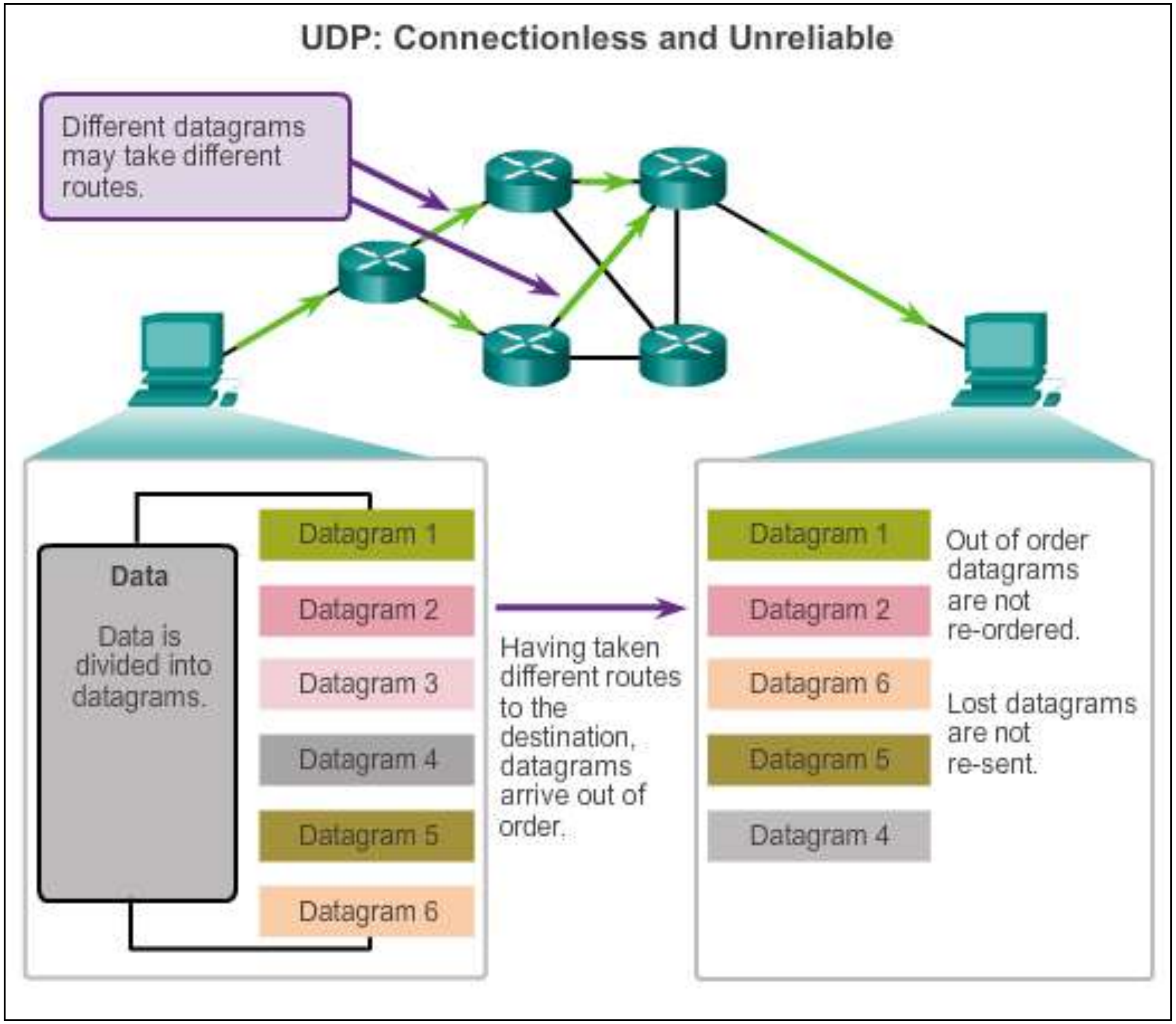
- Simple protocol that provides the basic transport layer function
- Used by applications that can tolerate small loss of data
- Used by applications that cannot tolerate delay

Used by

- DNS
- Simple Network Management Protocol (SNMP)
- Dynamic Host Configuration Protocol (DHCP)
- Trivial File Transfer Protocol (TFTP)
- IP telephony or VoIP
- Online games

UDP Communication

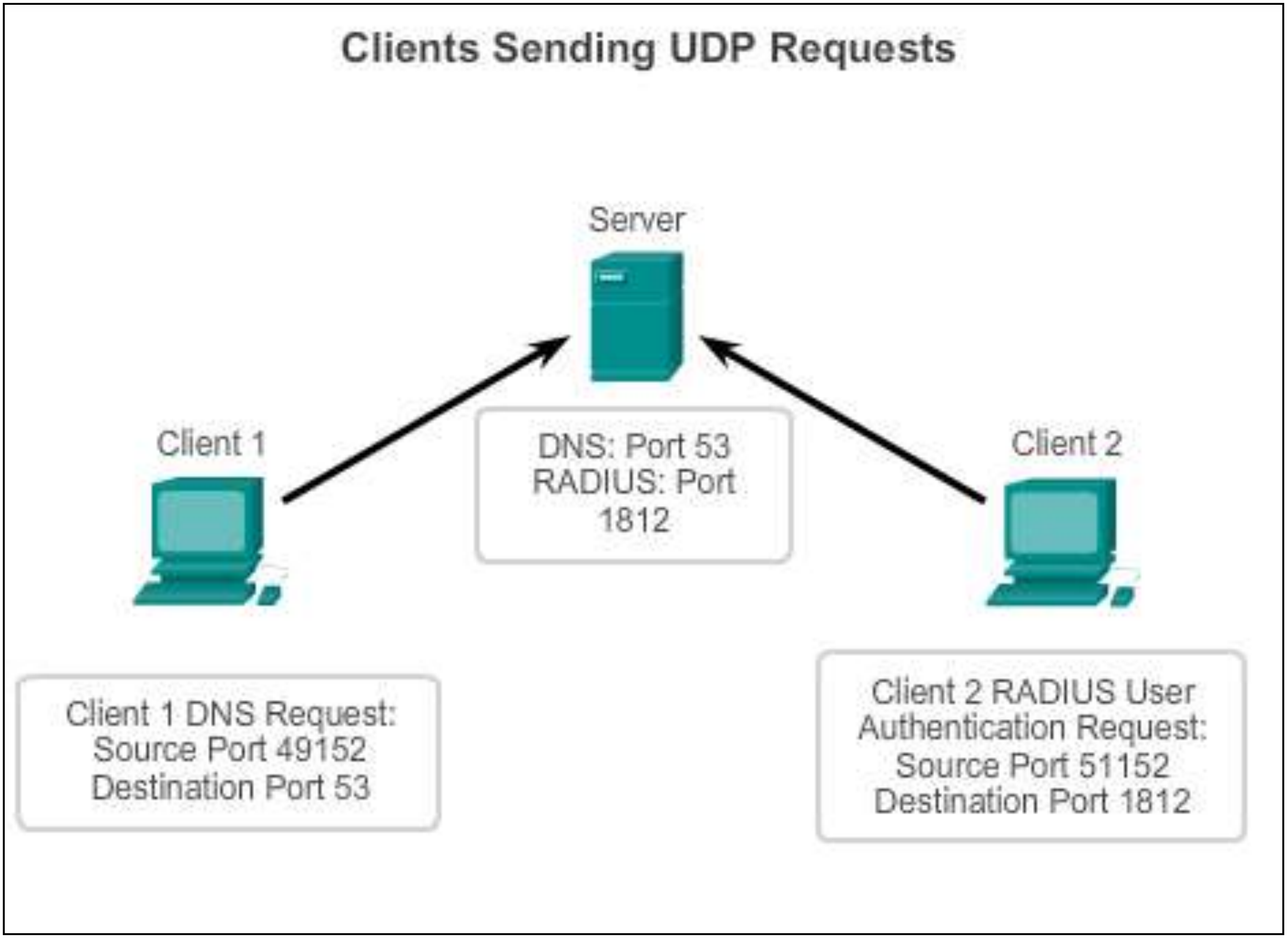
Datagram Reassembly


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.

UDP Communication

UDP Server and Client Processes

- UDP-based server applications are assigned well-known or registered port numbers.
- UDP client process randomly selects port number from range of dynamic port numbers as the source port.



Clients Sending UDP Requests

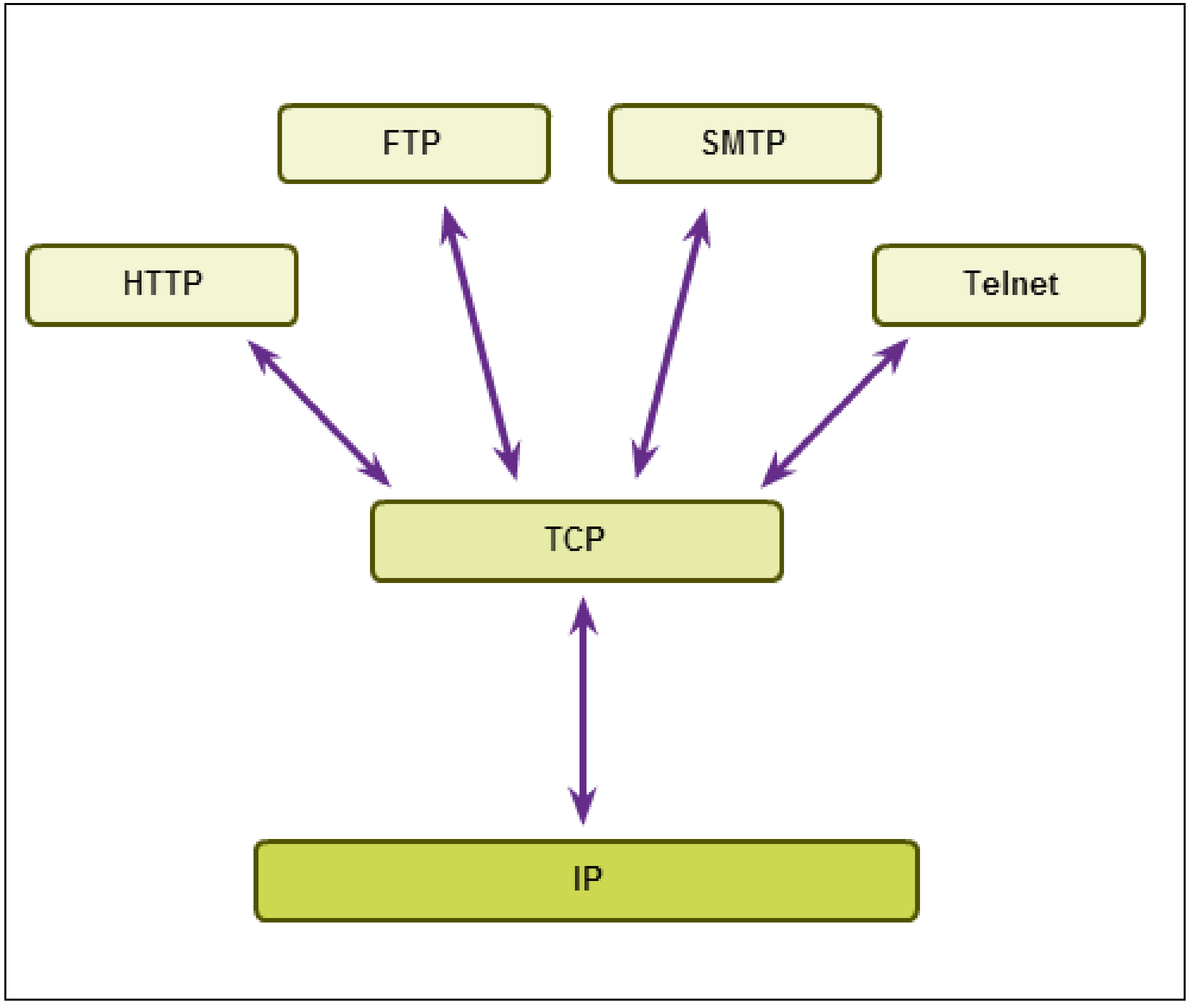
DNS: Port 53
RADIUS: Port 1812

Client 1 DNS Request:
Source Port 49152
Destination Port 53

Client 2 RADIUS User Authentication Request:
Source Port 51152
Destination Port 1812

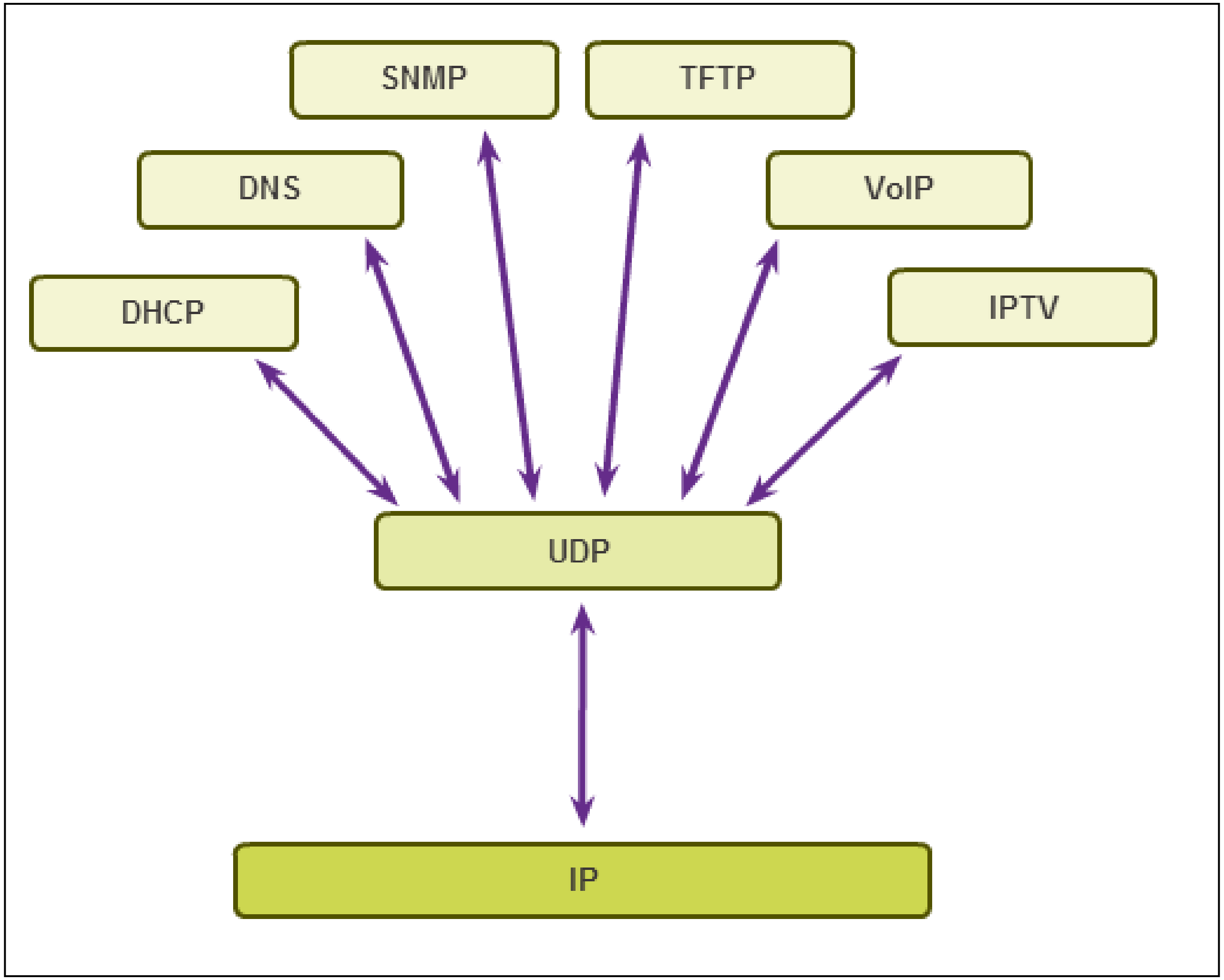
TCP or UDP

Applications that use TCP



TCP or UDP

Applications That Use UDP


SNMP
TFTP
DNS
VoIP
DHCP
IPTV
UDP
IP