TY B. Tech.
(Computer Engineering)
2020 Pattern

Prerequisites :

Digital Electronics

Course Objectives :

1. To study the fundamentals of networking.
2. To understand functionalities of Physical layer.
3. To understand the functionalities of Data Link Layer and Network Layer.
4. To learn Integrity checks and Authentication algorithms.
5. To learn various types of Cryptographic algorithm.

Course Outcomes :

After completion of the course, student will be able to

1. Explore network design issues.

2. Recognize the functions of OSI layers & TCP/IP protocol stack.

3. Describe and Demonstrate the functionality of Data Link Layer and Network Layer.

4. Describe the functionality of Transport and Application Layer.

5. Examine the protocols for integrity and authentication.

6. Make use of various Cryptographic algorithm.

UNIT 4- **Transport Layer and Application Layer**

- Transport Layer: Transport Layer Protocols, Role of transport layer, Responsibilities of Transport layer, Transport layer reliability.

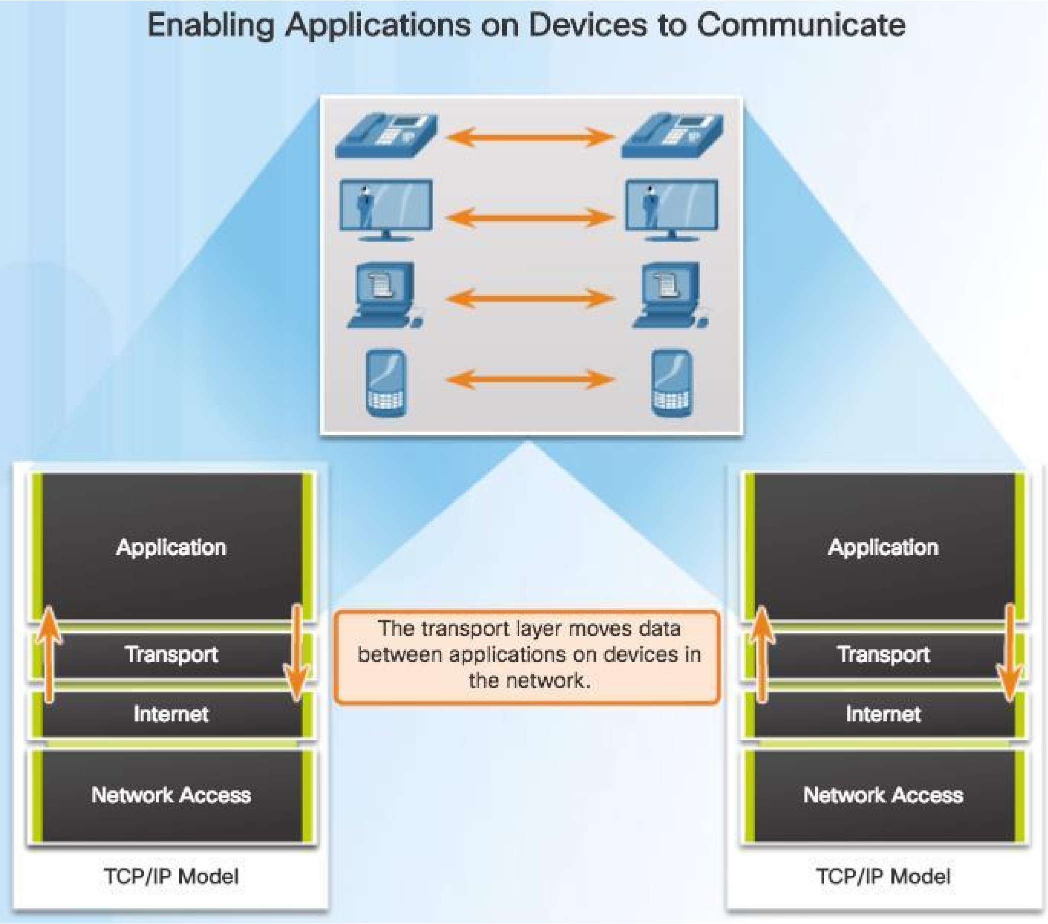
- TCP and UDP: TCP communication Process, Reliability and flow control, UDP Communication, applications of TCP and UDP.

- Application Layer: Application Layer Protocols, Application layer protocols interaction with end-user applications, Presentation and Session layers. Well-Known Application Protocols and Services.

Transport Layer Protocols

Role of the Transport Layer

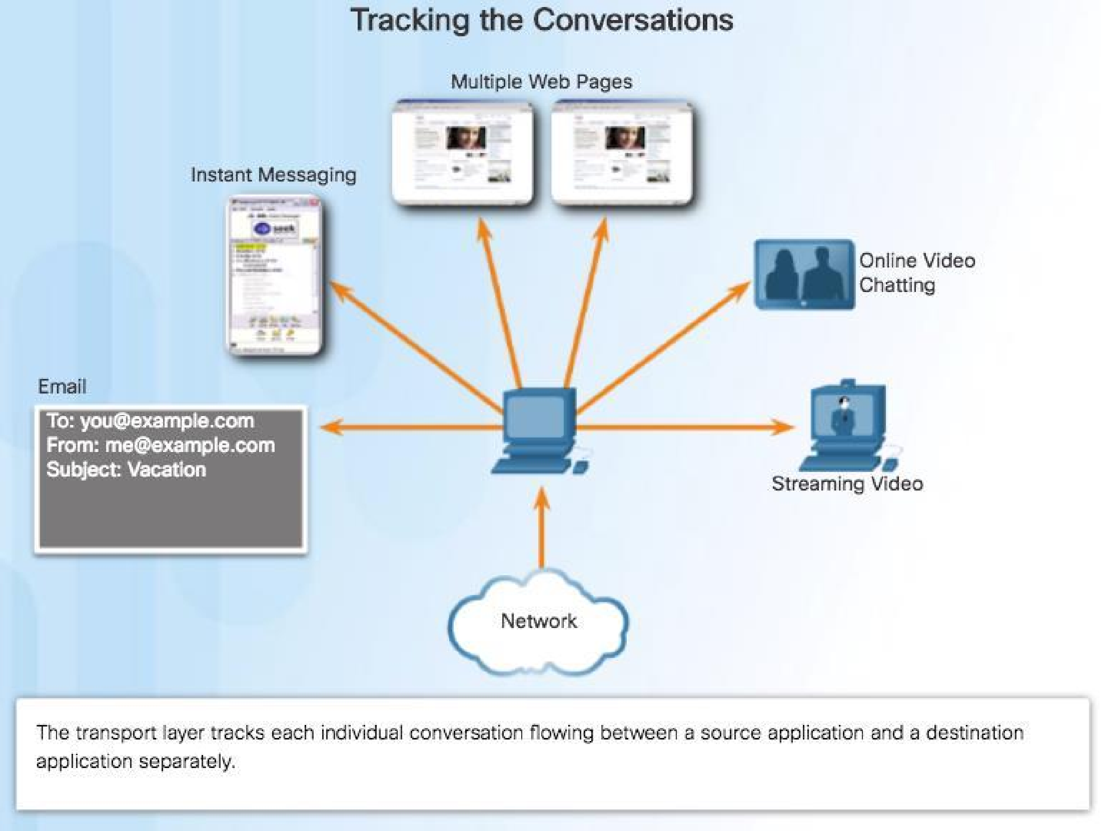
- Responsible for establishing a temporary communication session between two applications and delivering data between them.
- Link between the application layer and the lower layers that are responsible for network transmission.



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

Transport Layer Responsibilities

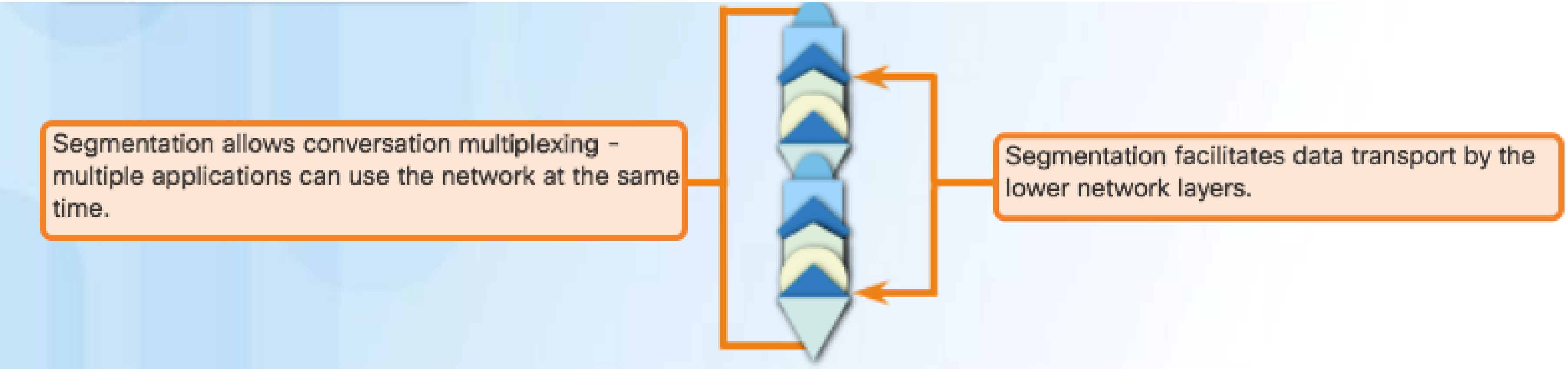
- **Tracking the Conversation** - Tracks each individual conversation flowing between a source and a destination application.
- **Segmentation** - Divides the data into segments that are easier to manage and transport. Header used for reassembly is used for tracking.
- **Identifying the Application** - Ensures that even with multiple applications running on a device, all applications receive the correct data via port numbers.



The transport layer tracks each individual conversation flowing between a source application and a destination application separately.

Conversation Multiplexing

- Segmenting the data into smaller chunks enables many different communications to be multiplexed on the same network.



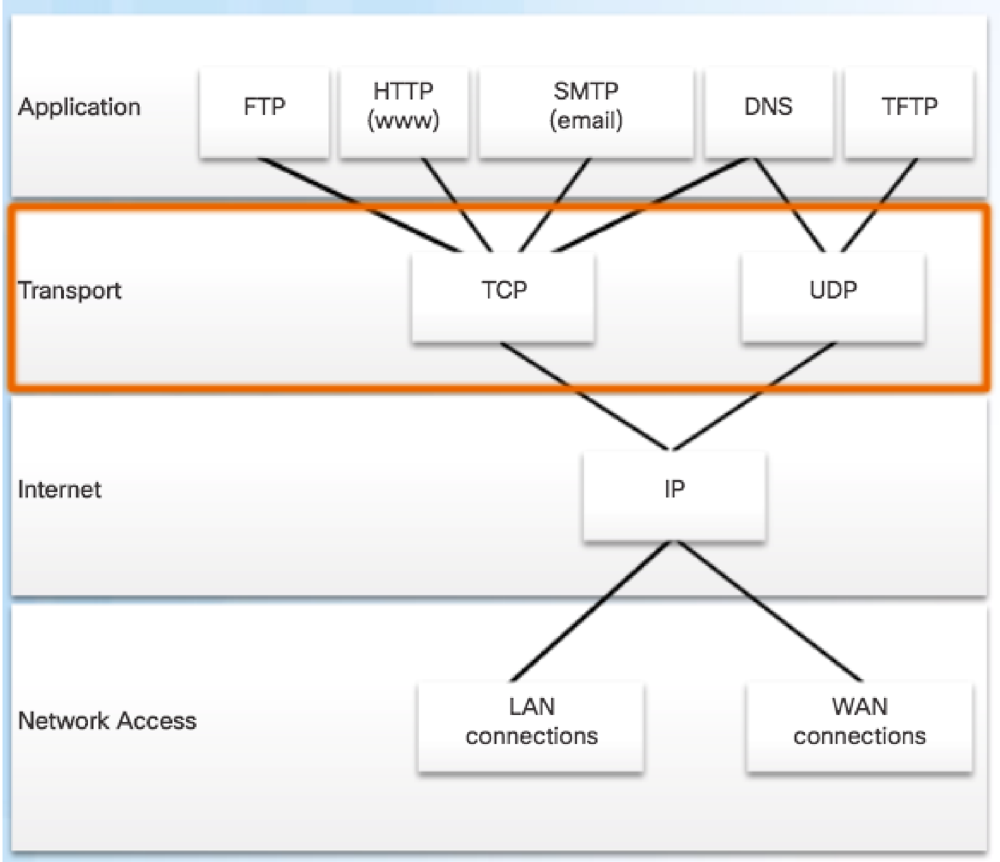
Segmentation allows conversation multiplexing - multiple applications can use the network at the same time.

Segmentation facilitates data transport by the lower network layers.

Transportation of Data

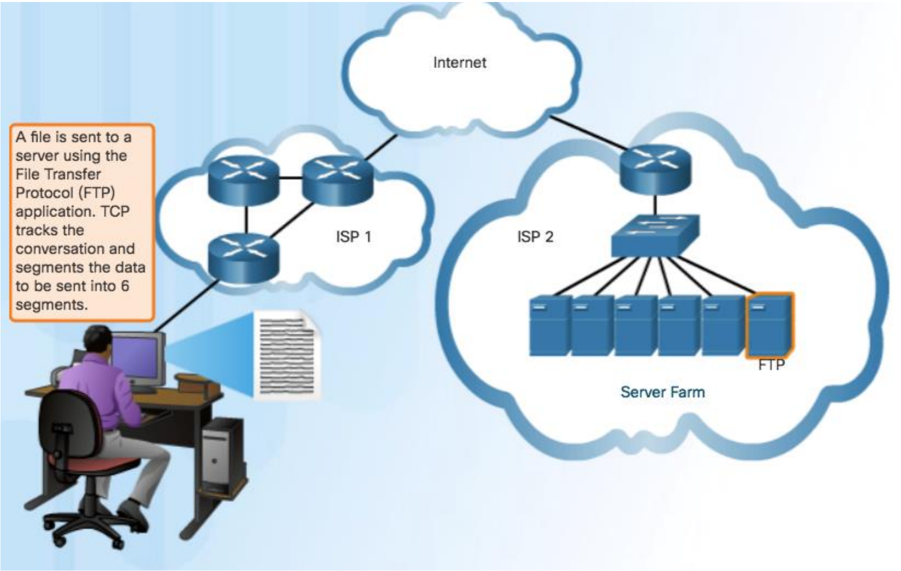
Transport Layer Reliability

- TCP/IP provides two transport layer protocols:
  - Transmission Control Protocol (TCP)
    - Considered reliable which ensures that all of the data arrives at the destination.
    - Additional fields needed in header which increases size and delay.
  - User Datagram Protocol (UDP)
    - Does not provide for reliability.
    - Fewer fields and is faster than TCP.



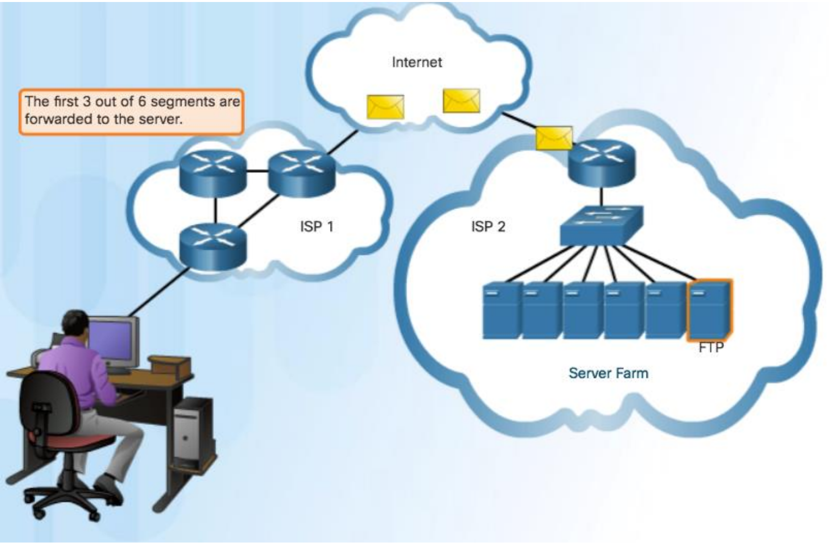
Transportation of Data

TCP



A file is sent to a server using the File Transfer Protocol (FTP) application. TCP tracks the conversation and segments the data to be sent into 6 segments.

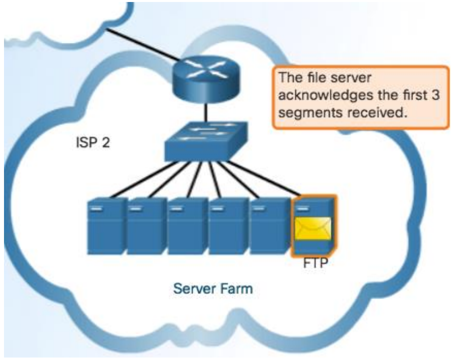
- TCP transport is similar to sending tracked packages. If a shipping order is broken up into several packages, a customer can check online to see the order of the delivery.

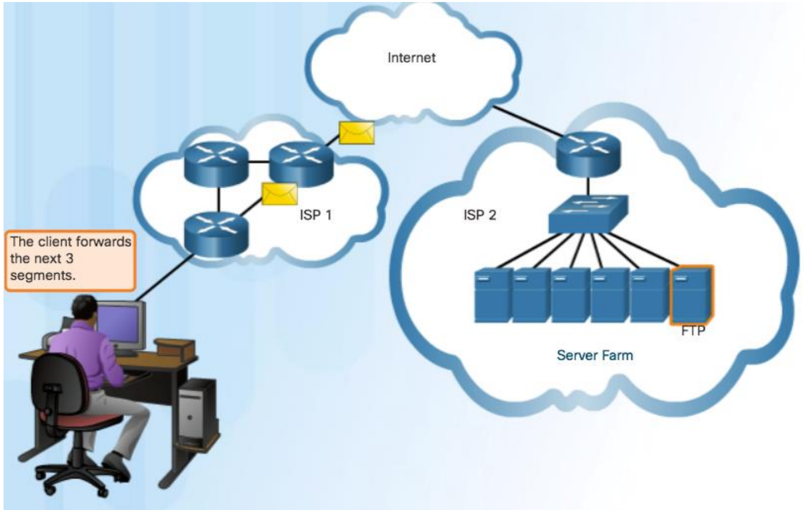


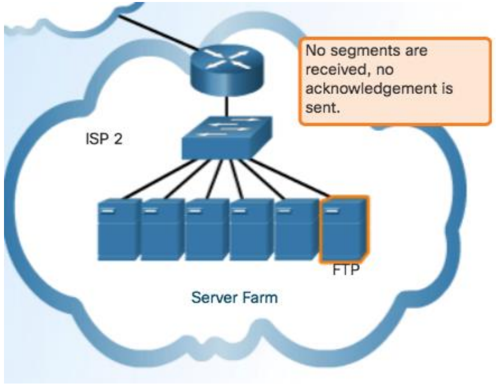
The first 3 out of 6 segments are forwarded to the server.

Transportation of Data

TCP (Cont.)

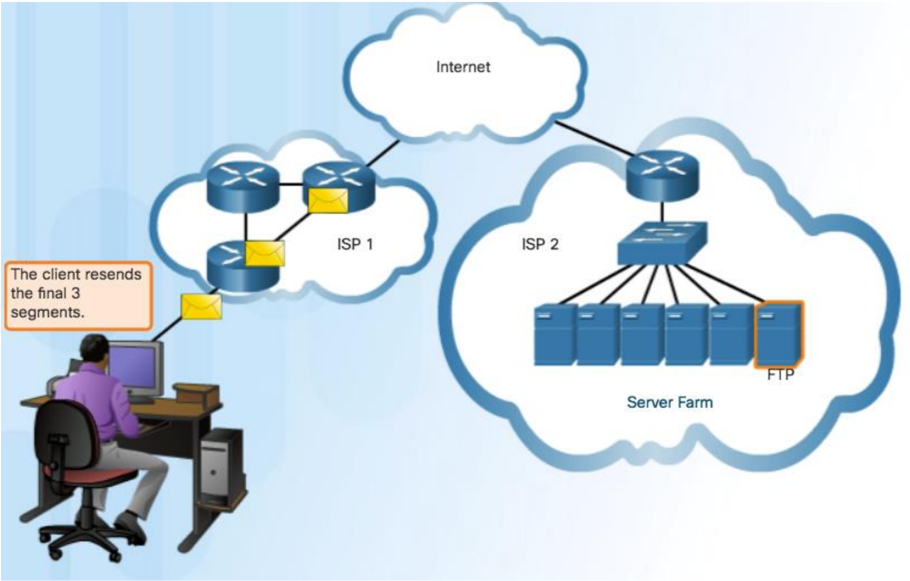

The file server acknowledges the first 3 segments received.


The client forwards the next 3 segments.


No segments are received, no acknowledgement is sent.

Transportation of Data

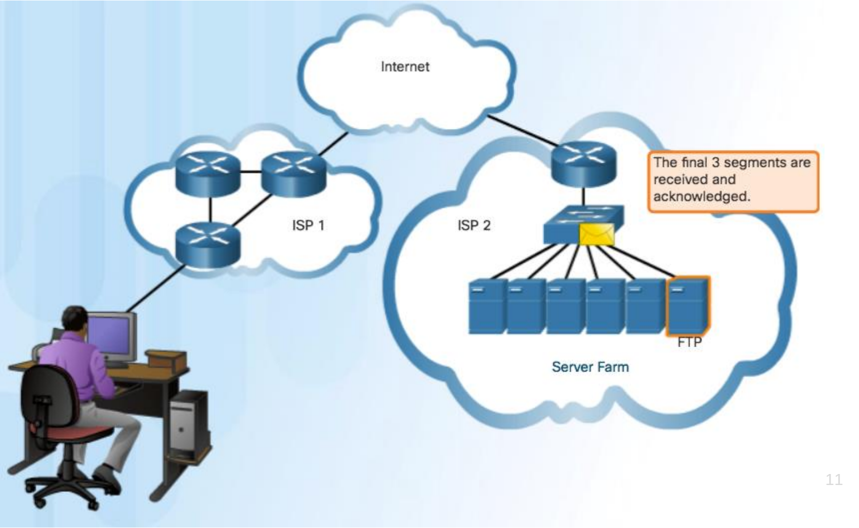
TCP (Cont.)



The client resends the final 3 segments.

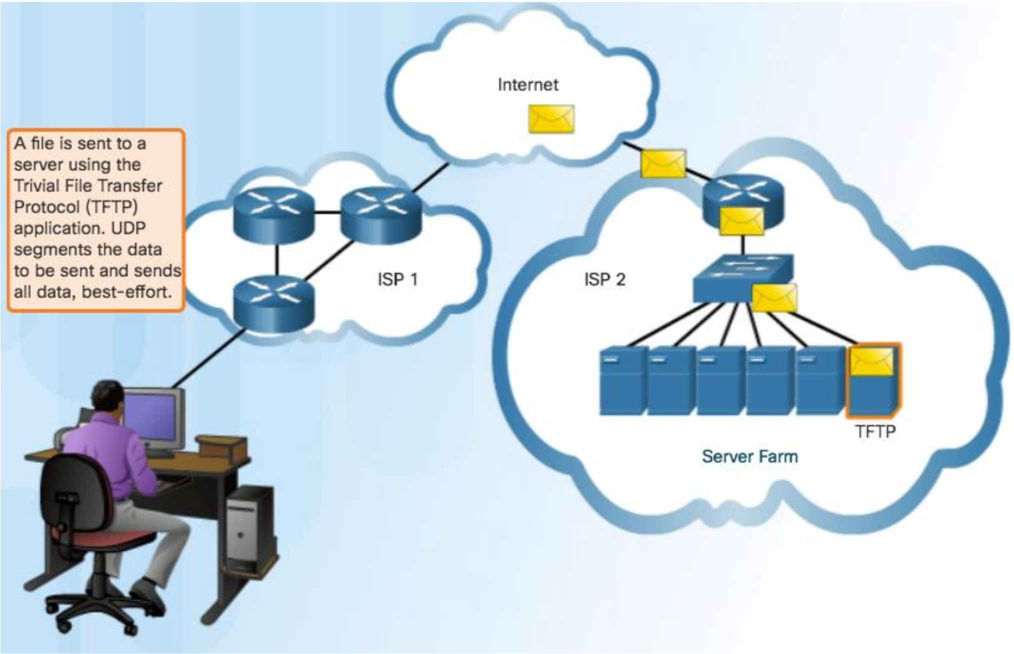
TCP Three Responsibilities:

- Numbering and tracking data segments
- Acknowledging received data
- Retransmitting any unacknowledged data after a certain period of time



The final 3 segments are received and acknowledged.

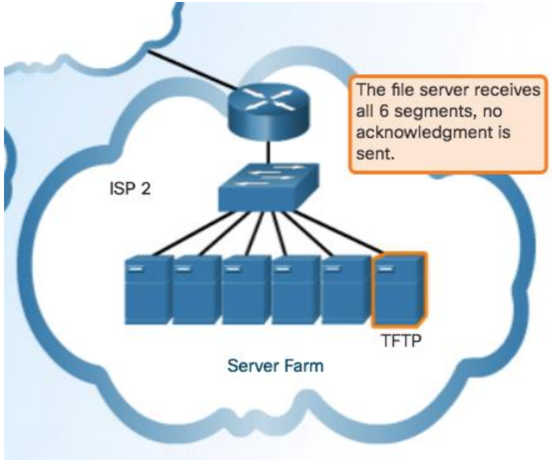
UDP



A file is sent to a server using the Trivial File Transfer Protocol (TFTP) application. UDP segments the data to be sent and sends all data, best-effort.

Use UDP for less overhead and to reduce possible delays.

- Best-effort delivery (unreliable)
- No acknowledgment
- Similar to a non-registered letter

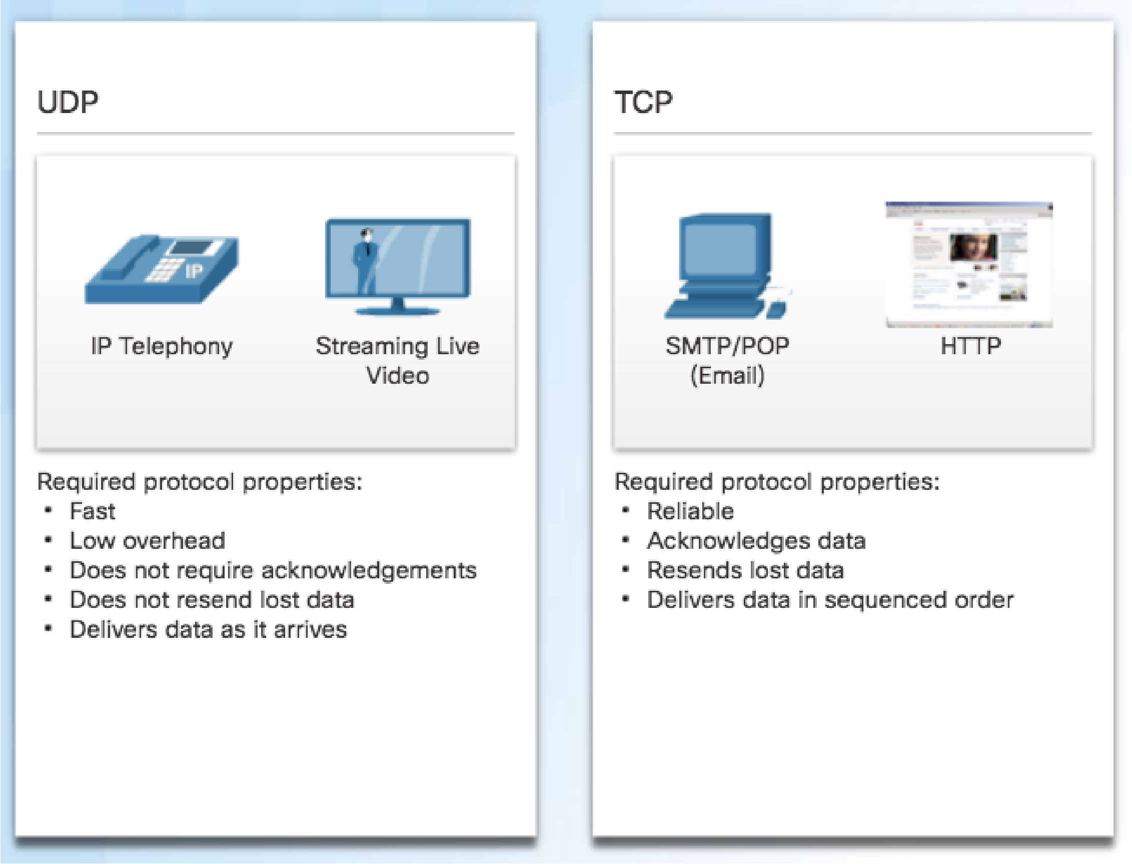


The file server receives all 6 segments, no acknowledgment is sent.

Transportation of Data

The Right Transport Layer Protocol for the Right Application

- TCP - databases, web browsers, and email clients require that all data that is sent arrives at the destination in its original condition.
- UDP - if one or two segments of a live video stream fail to arrive, if disruption in the stream, may not be noticeable to the user.



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
- Acknowledges data
- Resends lost data
- Delivers data in sequenced order

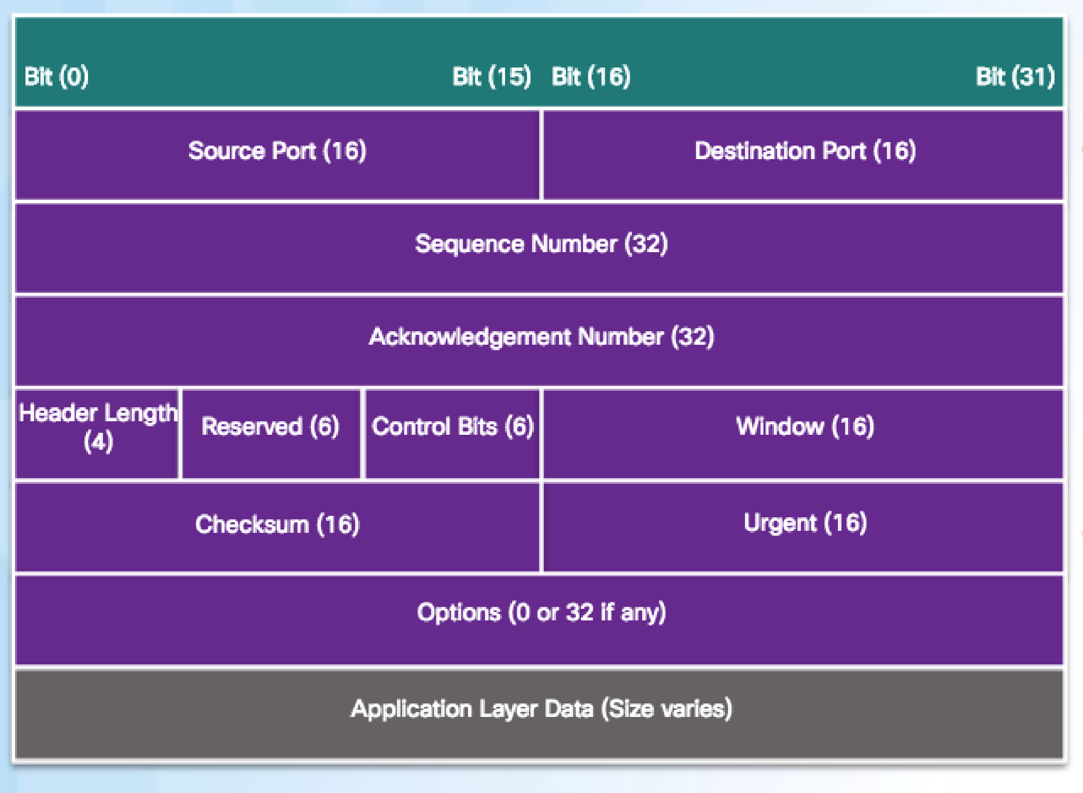
TCP Features

- Establishing a Session
  - Connection-oriented protocol
  - Ensures the application is ready to receive the data
  - Negotiate the amount of traffic that can be forwarded at a given time
- Reliable Delivery
  - Ensuring that each segment that the source sends arrives at the destination
- Same-Order Delivery
  - Numbering & Sequencing the segments guarantees reassembly into the proper order
- Flow Control
  - Regulate the amount of data the source transmits

TCP Header

- Source and Destination Port used to identify application
- Sequence number used for data reassembly
- Acknowledgement number indicates data has been received and ready for next byte from source
- Header length – length of TCP segment header
- Control bits – purpose and function of TCP segment
- Window size – number of bytes that can be accepted at one time
- Checksum – Used for error checking of segment header and data

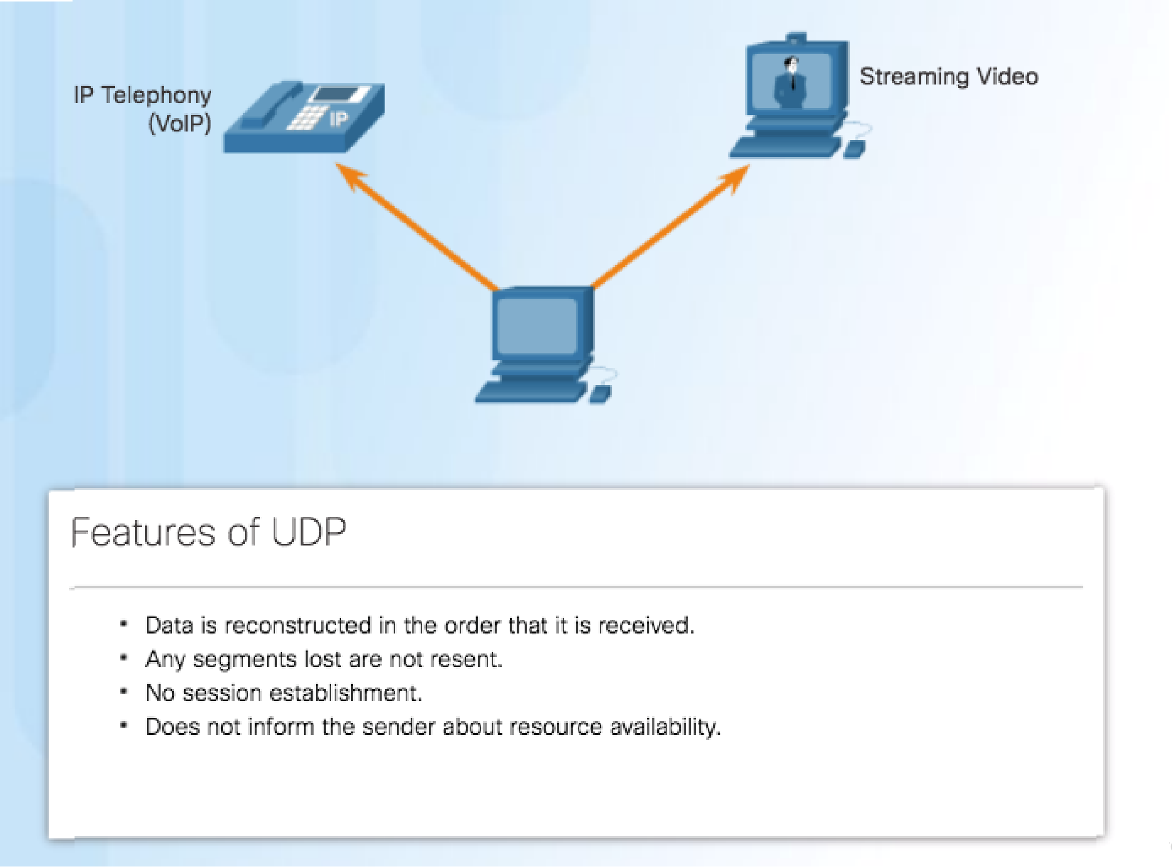
20 Bytes Total



<table>
<tr><th colspan="3">Bit (0) Bit (15)</th><th>Bit (16) Bit (31)</th></tr>
<tr><td colspan="3">Source Port (16)</td><td>Destination Port (16)</td></tr>
<tr><td colspan="4">Sequence Number (32)</td></tr>
<tr><td colspan="4">Acknowledgement Number (32)</td></tr>
<tr><td>Header Length (4)</td><td>Reserved (6)</td><td>Control Bits (6)</td><td>Window (16)</td></tr>
<tr><td colspan="3">Checksum (16)</td><td>Urgent (16)</td></tr>
<tr><td colspan="4">Options (0 or 32 if any)</td></tr>
<tr><td colspan="4">Application Layer Data (Size varies)</td></tr>
</table>

TCP and UDP Overview

UDP Features



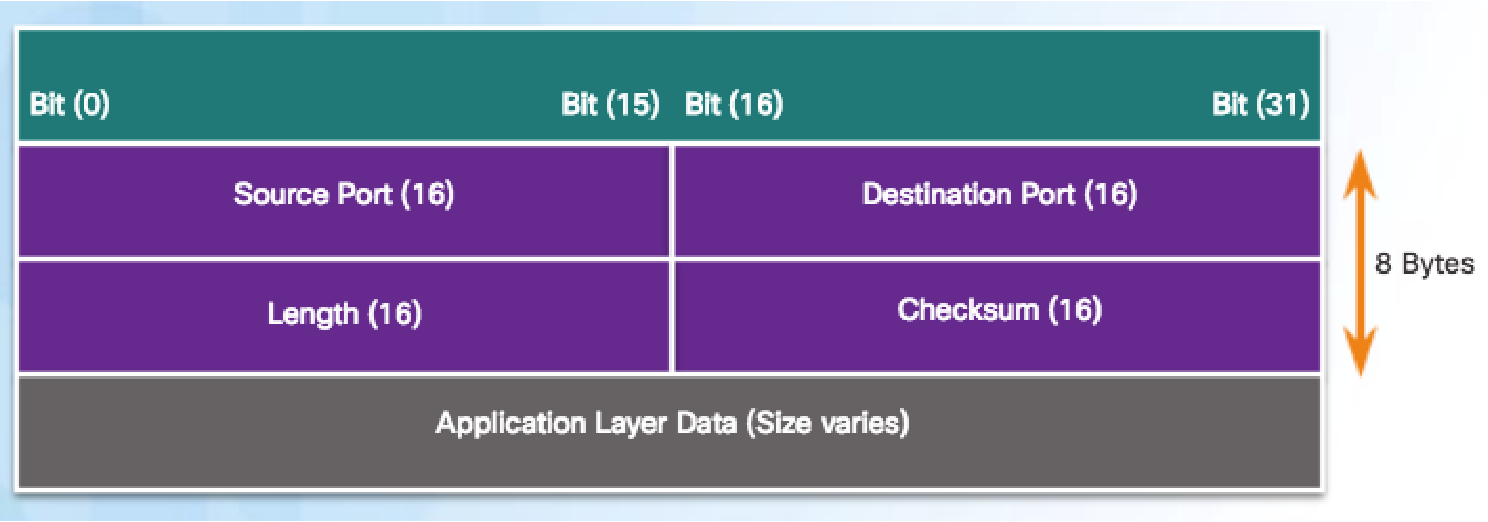
Features of UDP

- Data is reconstructed in the order that it is received.
- Any segments lost are not resent.
- No session establishment.
- Does not inform the sender about resource availability.

TCP and UDP Overview

UDP Header

- UDP is a stateless protocol – no tracking
- Reliability handled by application

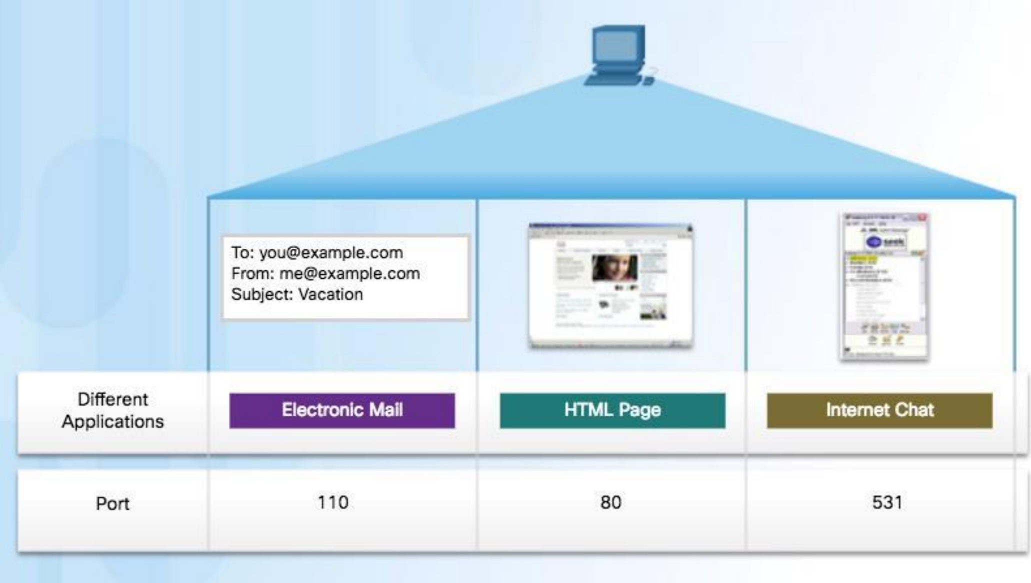


Bit (0) Bit (15)	Bit (16) Bit (31)
Source Port (16)	Destination Port (16)
Length (16)	Checksum (16)
Application Layer Data (Size varies)	

TCP and UDP Overview

Multiple Separate Communications

- Users expect to simultaneously receive and send email, view websites and make a VoIP phone call
- TCP and UDP manage multiple conversations by using unique identifiers called port numbers

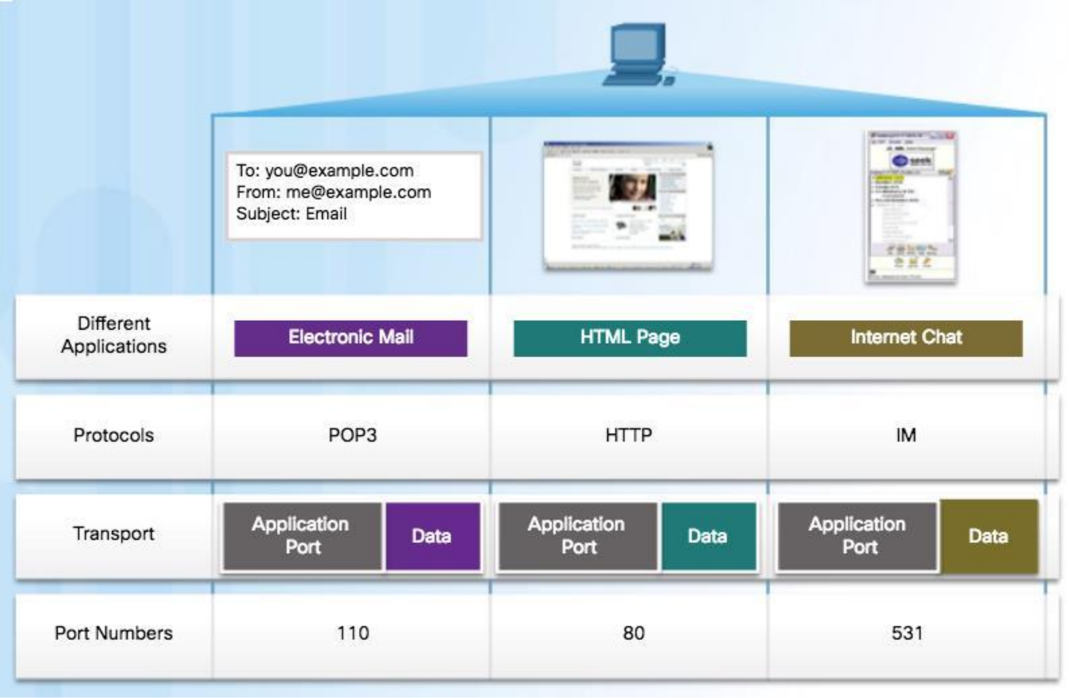


To: you@example.com
From: me@example.com
Subject: Vacation

Different Applications	Electronic Mail	HTML Page	Internet Chat
Port	110	80	531

Port Numbers

- Source Port
  - Originating application port that is dynamically generated by sending device
  - Example: Each separate HTTP conversation is tracked based on the source ports.
- Destination Port
  - Tell the destination what service is being requested
  - Example: Port 80 web services are being requested

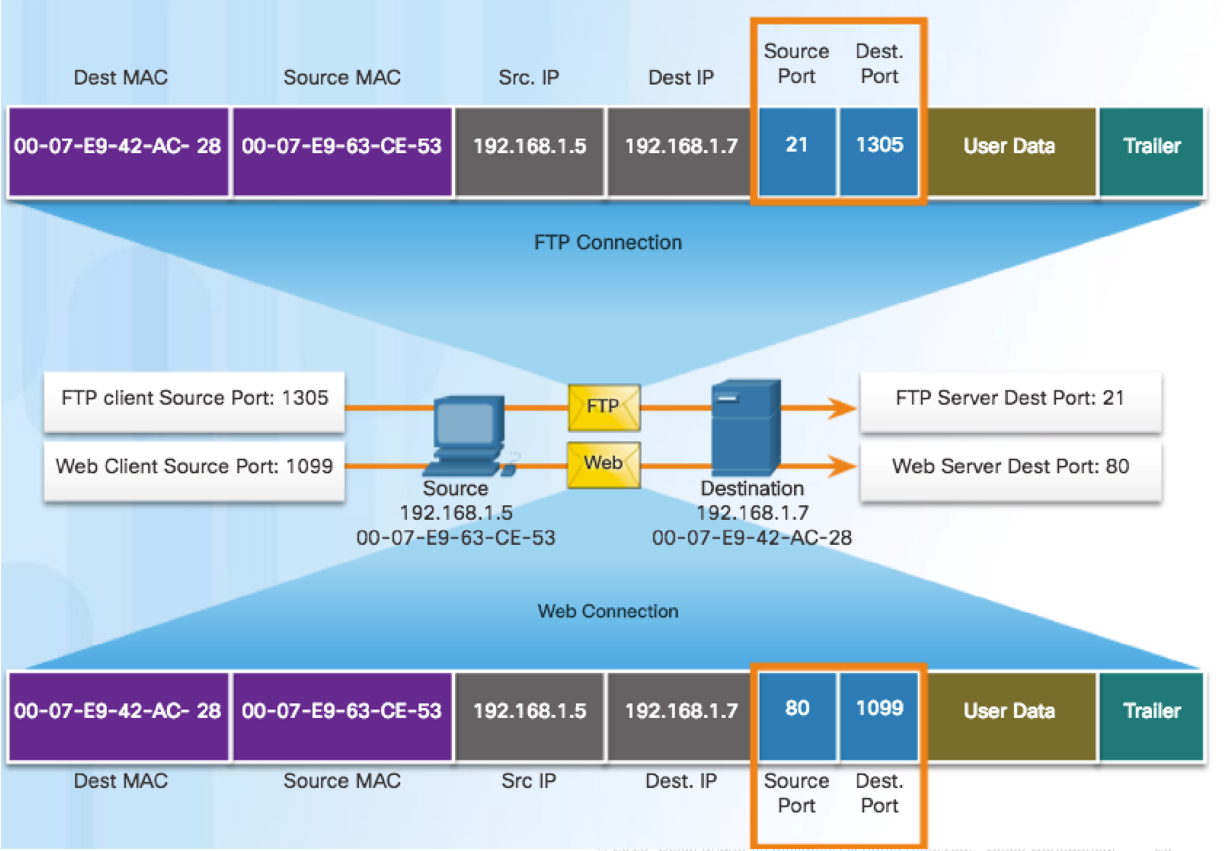


To: you@example.com
From: me@example.com
Subject: Email

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

Socket Pairs

- Source and destination port placed in segment
- Segments encapsulated in IP packet
- IP and port number = socket
- Example: 192.168.1.7:80
- Sockets enable multiple processes to be distinguished
- Source port acts as a return address

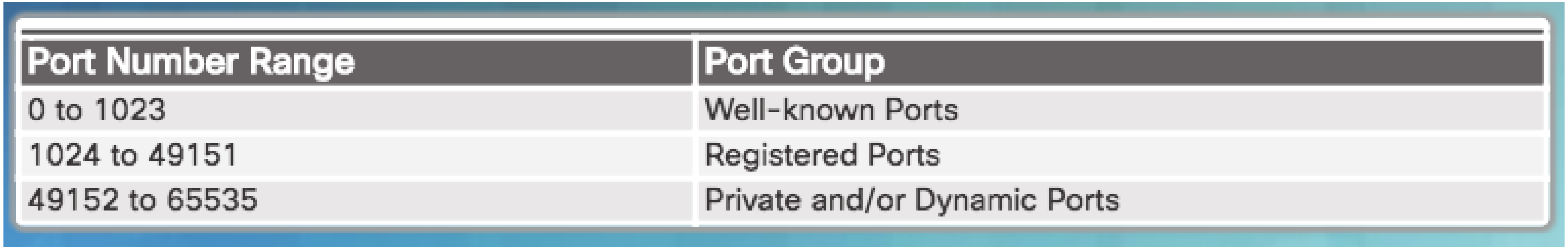


Dest MAC	Source MAC	Src. IP	Dest IP	Source Port	Dest. Port		
00-07-E9-42-AC- 28	00-07-E9-63-CE-53	192.168.1.5	192.168.1.7	21	1305	User Data	Trailer

Dest MAC	Source MAC	Src IP	Dest. IP	Source Port	Dest. 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

TCP and UDP Overview

Port Number Groups

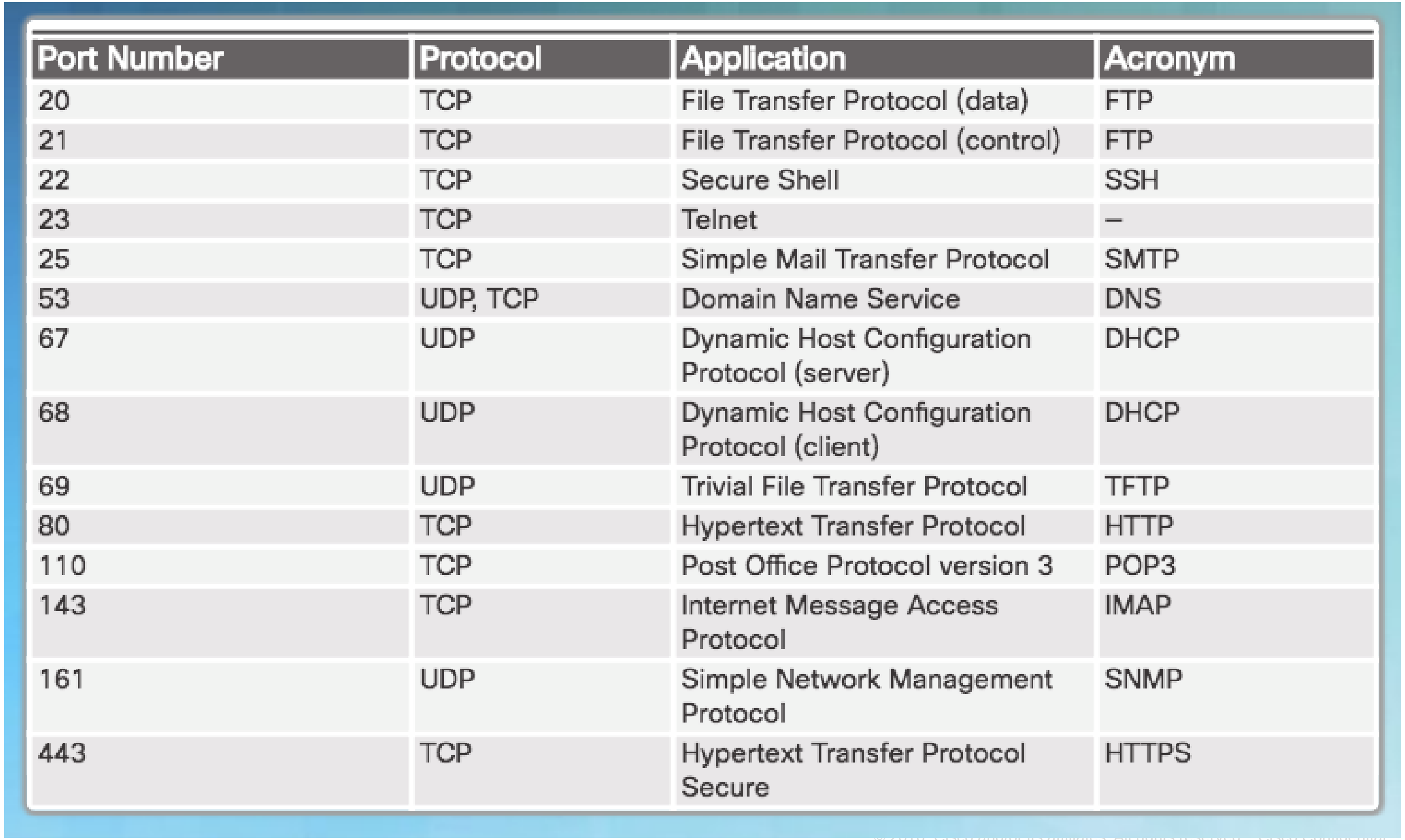


Port Number Range	Port Group
0 to 1023	Well-known Ports
1024 to 49151	Registered Ports
49152 to 65535	Private and/or Dynamic Ports

- Well-known Ports (Numbers 0 to 1023) - These numbers are reserved for services and applications.
- Registered Ports (Numbers 1024 to 49151) - These port numbers are assigned by IANA to a requesting entity to use with specific processes or applications.
- Dynamic or Private Ports (Numbers 49152 to 65535) - Usually assigned dynamically by the client's OS and used to identify the client application during communication.

Port Number Groups (Cont.)

Well Known Port Numbers

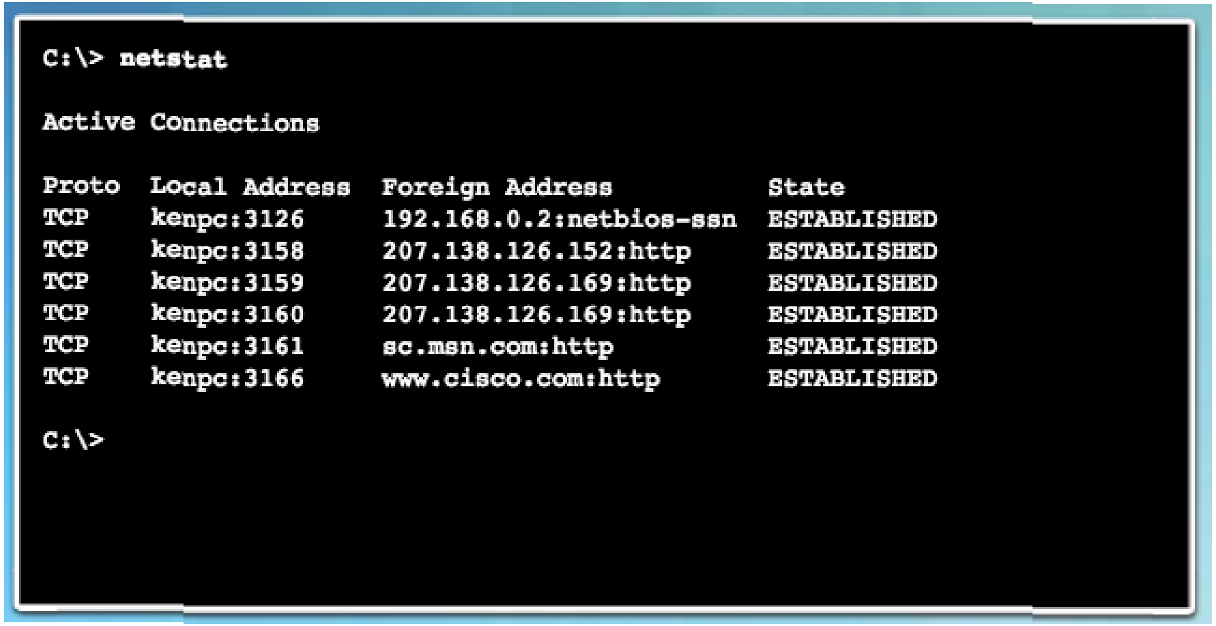


Port Number	Protocol	Application	Acronym
20	TCP	File Transfer Protocol (data)	FTP
21	TCP	File Transfer Protocol (control)	FTP
22	TCP	Secure Shell	SSH
23	TCP	Telnet	–
25	TCP	Simple Mail Transfer Protocol	SMTP
53	UDP, TCP	Domain Name Service	DNS
67	UDP	Dynamic Host Configuration Protocol (server)	DHCP
68	UDP	Dynamic Host Configuration Protocol (client)	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

TCP and UDP Overview

The netstat Command

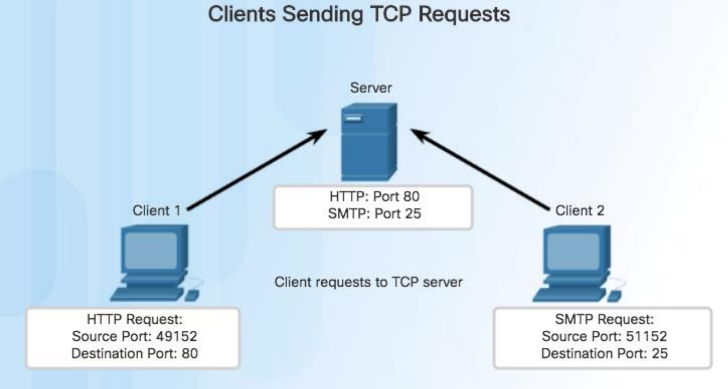
- Network utility that can be used to verify connections
- By default, will attempt to resolve IP addresses to domain names and port numbers to well-known applications
- -n option used to display IPs and ports in numerical form

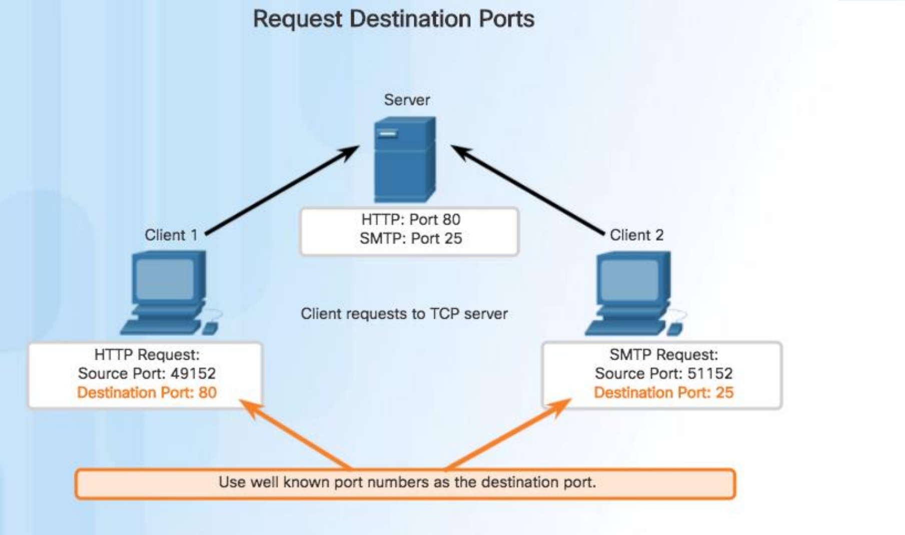


```
C:\> netstat

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:\>
```

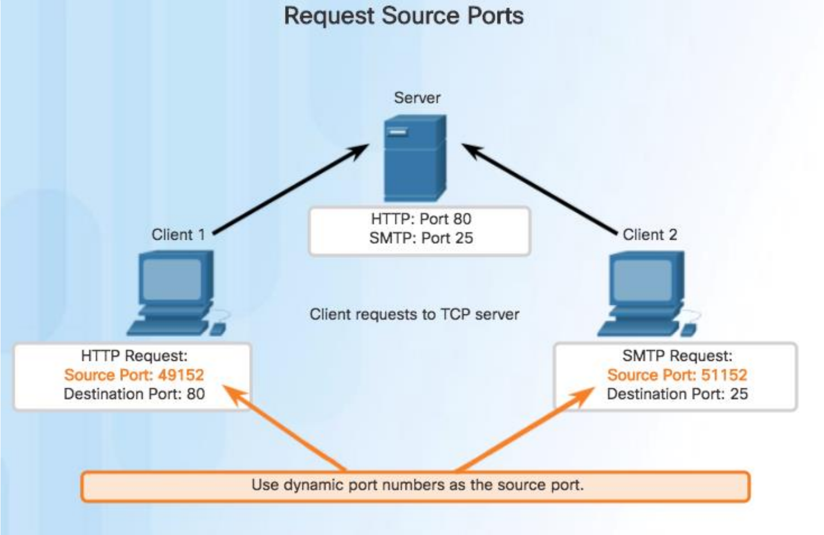
TCP and UDP

TCP Server Process


Clients Sending TCP Requests


Request Destination Ports

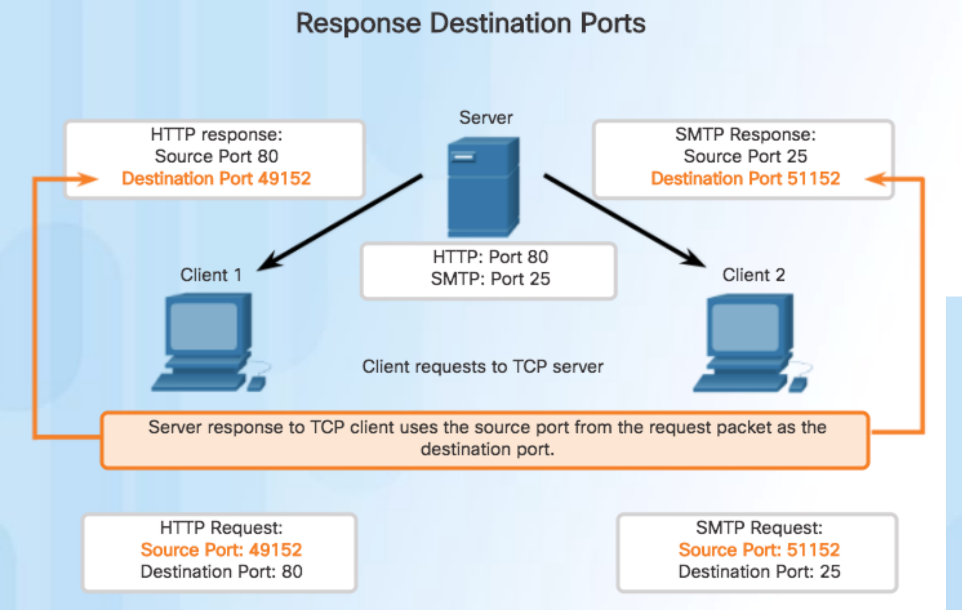
Use well known port numbers as the destination port.


Request Source Ports

Use dynamic port numbers as the source port.

TCP Communication Process

TCP Server Process (Cont.)

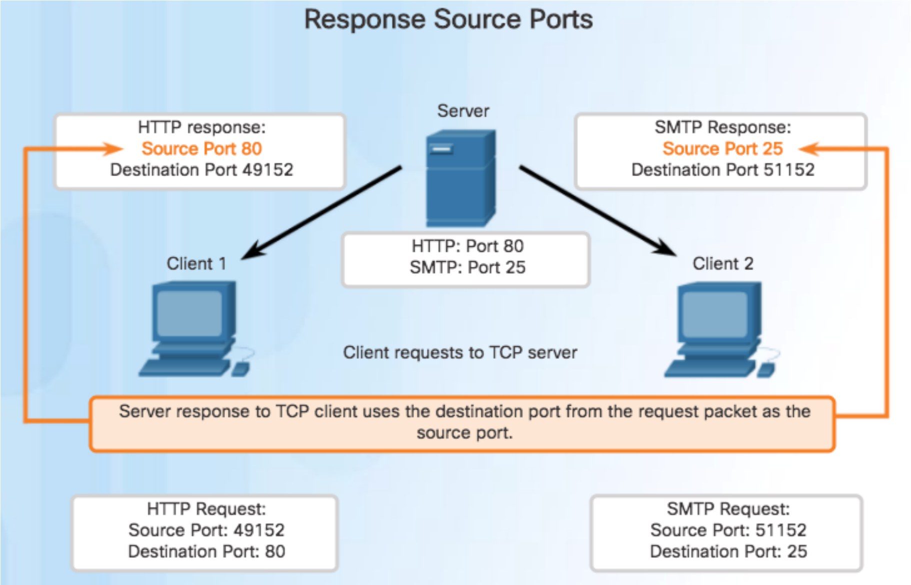


Response Destination Ports

Server response to TCP client uses the source port from the request packet as the destination port.

HTTP Request:
Source Port: 49152
Destination Port: 80

SMTP Request:
Source Port: 51152
Destination Port: 25



Response Source Ports

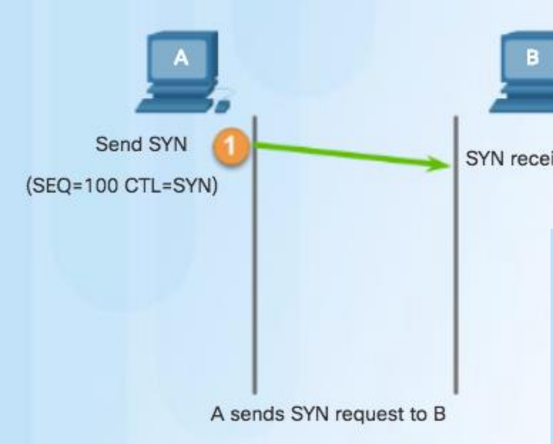
Server response to TCP client uses the destination port from the request packet as the source port.

HTTP Request:
Source Port: 49152
Destination Port: 80

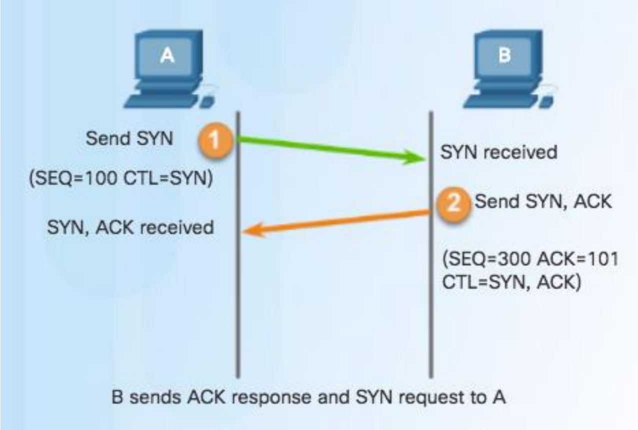
SMTP Request:
Source Port: 51152
Destination Port: 25

TCP Communication Process

TCP Connection Establishment

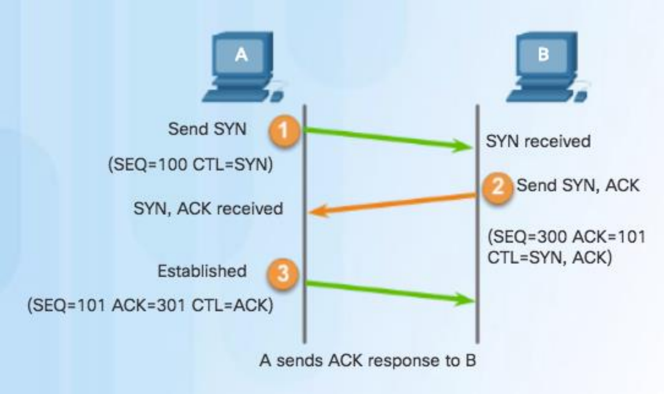


- Step 1 – Initiating client requests a session with server.



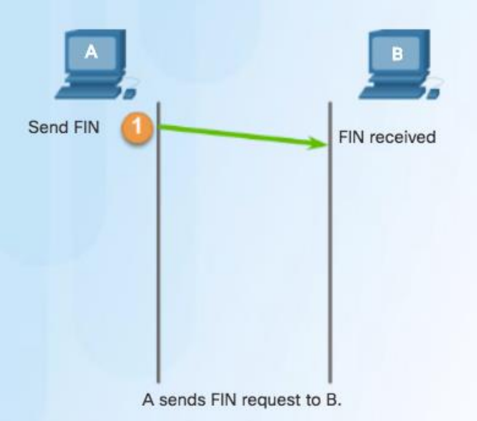
- Step 2 – Server acknowledges and requests a session with client.

- Step 3 – Client acknowledges communication session with server.

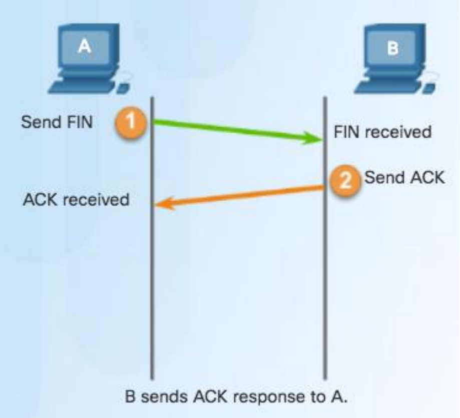


TCP Communication Process

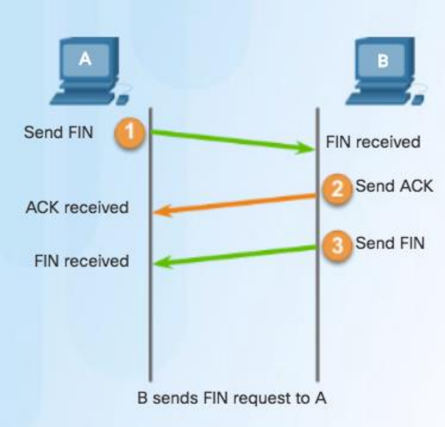
TCP Session Termination



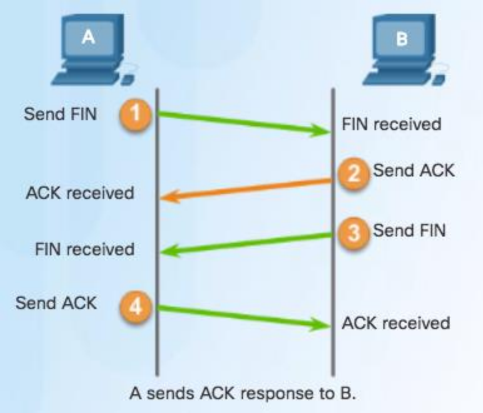
A sends FIN request to B.



B sends ACK response to A.



B sends FIN request to A

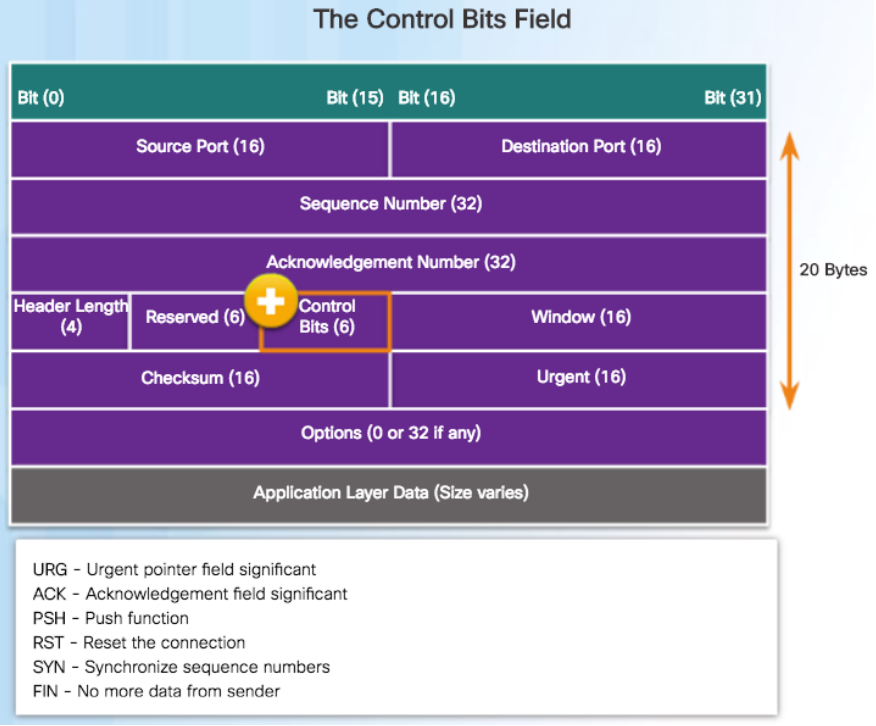


A sends ACK response to B.

- To close a connection, the Finish (FIN) control flag must be set in the segment header.
- To end each one-way TCP session, a two-way handshake, consisting of a FIN segment and an Acknowledgment (ACK) segment, is used.
- To terminate a single conversation supported by TCP, four exchanges are needed to end both sessions.

TCP Three-way Handshake Analysis

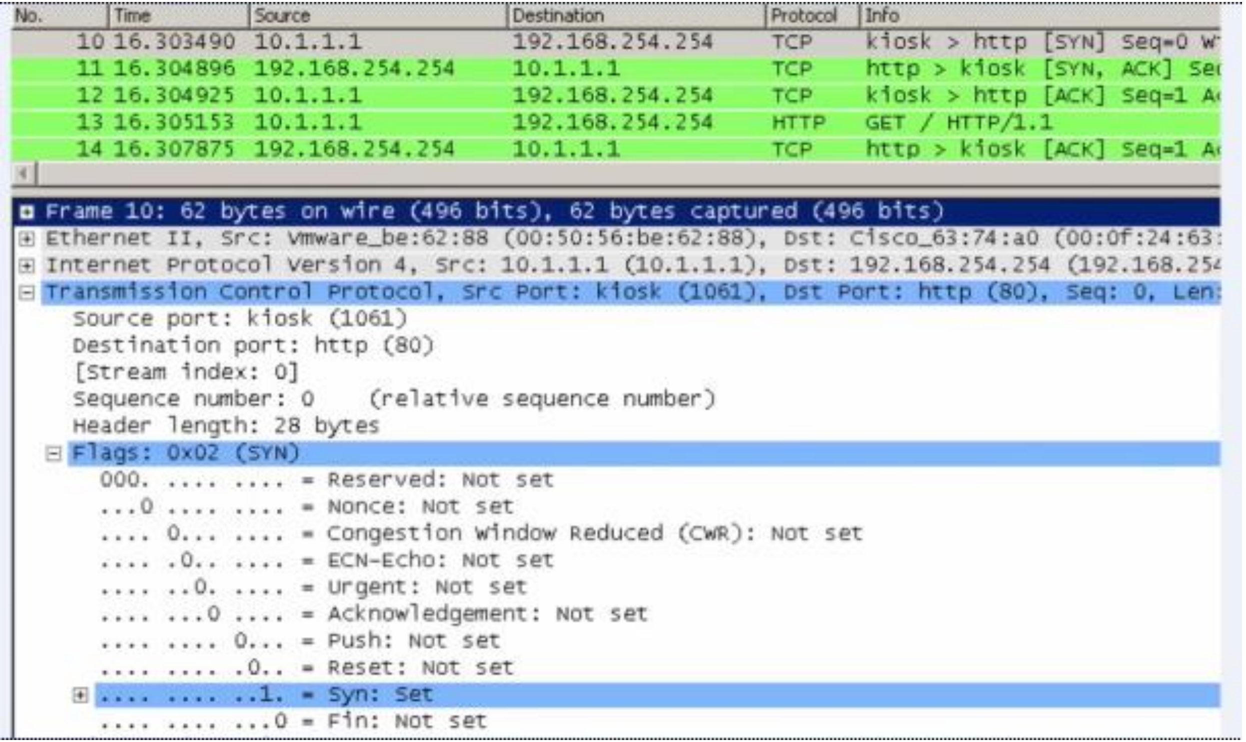
- The three-way handshake:
  - Establishes that the destination device is present on the network.
  - Verifies that the destination device has an active service and is accepting requests on the destination port number that the initiating client intends to use.
  - Informs the destination device that the source client intends to establish a communication session on that port number.
- The six bits in the Control Bits field of the TCP segment header are also known as flags.
  - RST flag is used to reset a connection when an error or timeout occurs



The Control Bits Field

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

Video Demonstration - TCP 3-Way Handshake

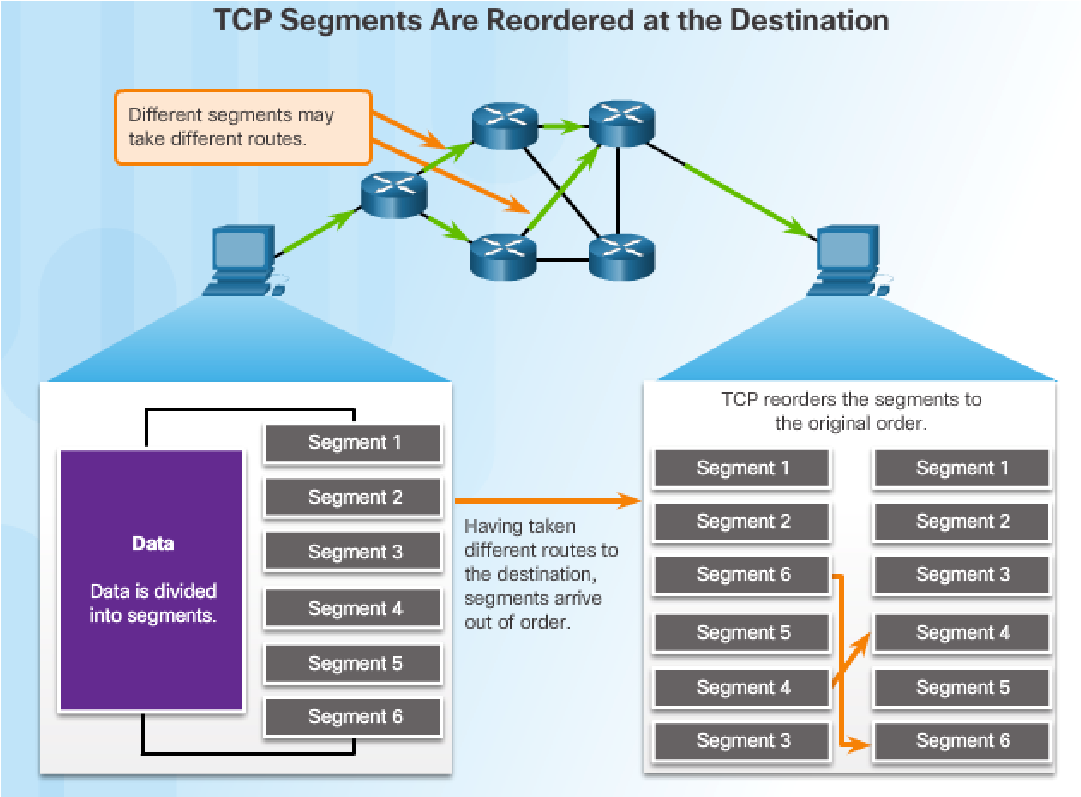


```
No.   Time          Source            Destination       Protocol  Info
   10 16.303490     10.1.1.1          192.168.254.254   TCP       kiosk > http [SYN] Seq=0 W
   11 16.304896     192.168.254.254   10.1.1.1          TCP       http > kiosk [SYN, ACK] Se
   12 16.304925     10.1.1.1          192.168.254.254   TCP       kiosk > http [ACK] Seq=1 A
   13 16.305153     10.1.1.1          192.168.254.254   HTTP      GET / HTTP/1.1
   14 16.307875     192.168.254.254   10.1.1.1          TCP       http > kiosk [ACK] Seq=1 A

⊞ Frame 10: 62 bytes on wire (496 bits), 62 bytes captured (496 bits)
⊞ Ethernet II, Src: Vmware_be:62:88 (00:50:56:be:62:88), Dst: Cisco_63:74:a0 (00:0f:24:63:
⊞ Internet Protocol Version 4, Src: 10.1.1.1 (10.1.1.1), Dst: 192.168.254.254 (192.168.25
⊟ Transmission Control Protocol, Src Port: kiosk (1061), Dst Port: http (80), Seq: 0, Len:
    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
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...0 .... = Acknowledgement: Not set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
    ⊞ .... .... ..1. = Syn: Set
      .... .... ...0 = Fin: Not set
```

SYN
SYN, ACK
ACK

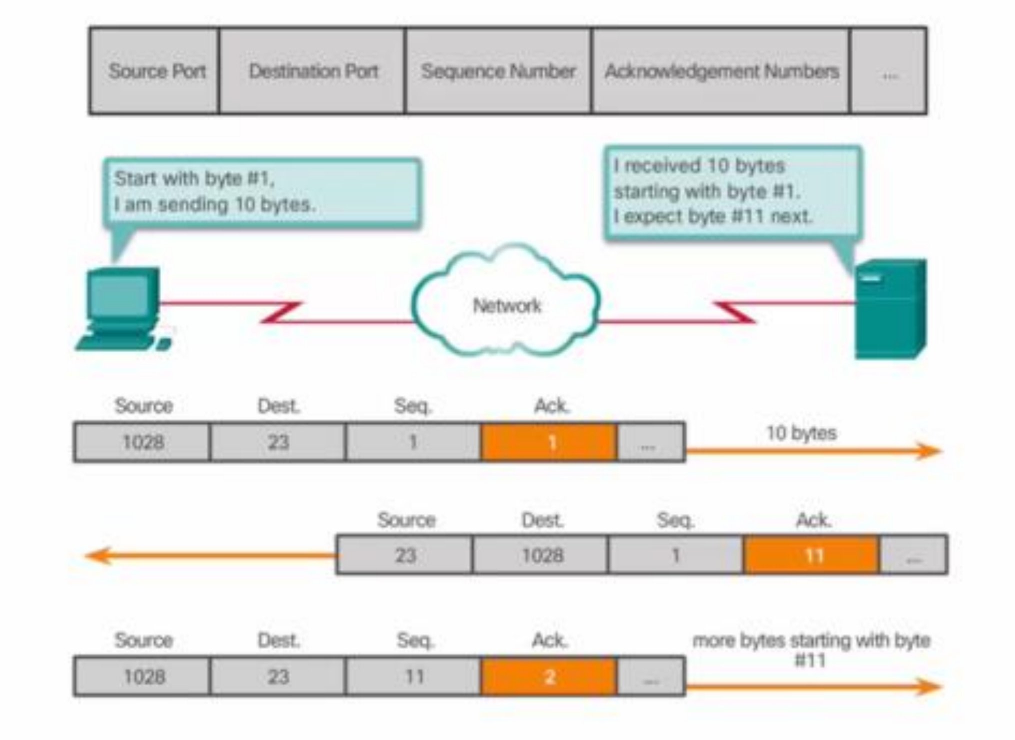
TCP Reliability – Ordered Delivery

- Sequence numbers are assigned in the header of each packet.
- Represents the first data byte of the TCP segment.
- During session setup, an initial sequence number (ISN) is set - represents the starting value of the bytes.
- As data is transmitted during the session, the sequence number is incremented by the number of bytes that have been transmitted.
- Missing segments can then be identified.


TCP Segments Are Reordered at the Destination
Different segments may take different routes.
Data
Data is divided into segments.
Segment 1
Segment 2
Segment 3
Segment 4
Segment 5
Segment 6
Having taken different routes to the destination, segments arrive out of order.
TCP reorders the segments to the original order.
Segment 1
Segment 2
Segment 6
Segment 5
Segment 4
Segment 3
Segment 1
Segment 2
Segment 3
Segment 4
Segment 5
Segment 6

Reliability and Flow Control

Video Demonstration - TCP Reliability – Sequence Numbers and Acknowledgments



Source Port	Destination Port	Sequence Number	Acknowledgement Numbers	...

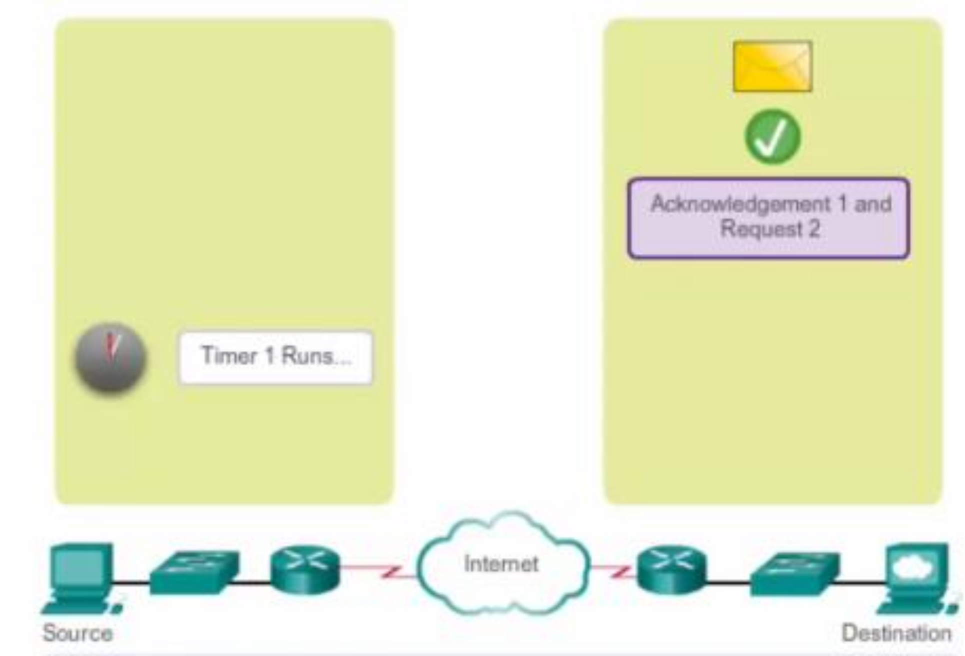
Source	Dest.	Seq.	Ack.	
1028	23	1	1	...

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

Source	Dest.	Seq.	Ack.	
1028	23	11	2	...

Reliability and Flow Control

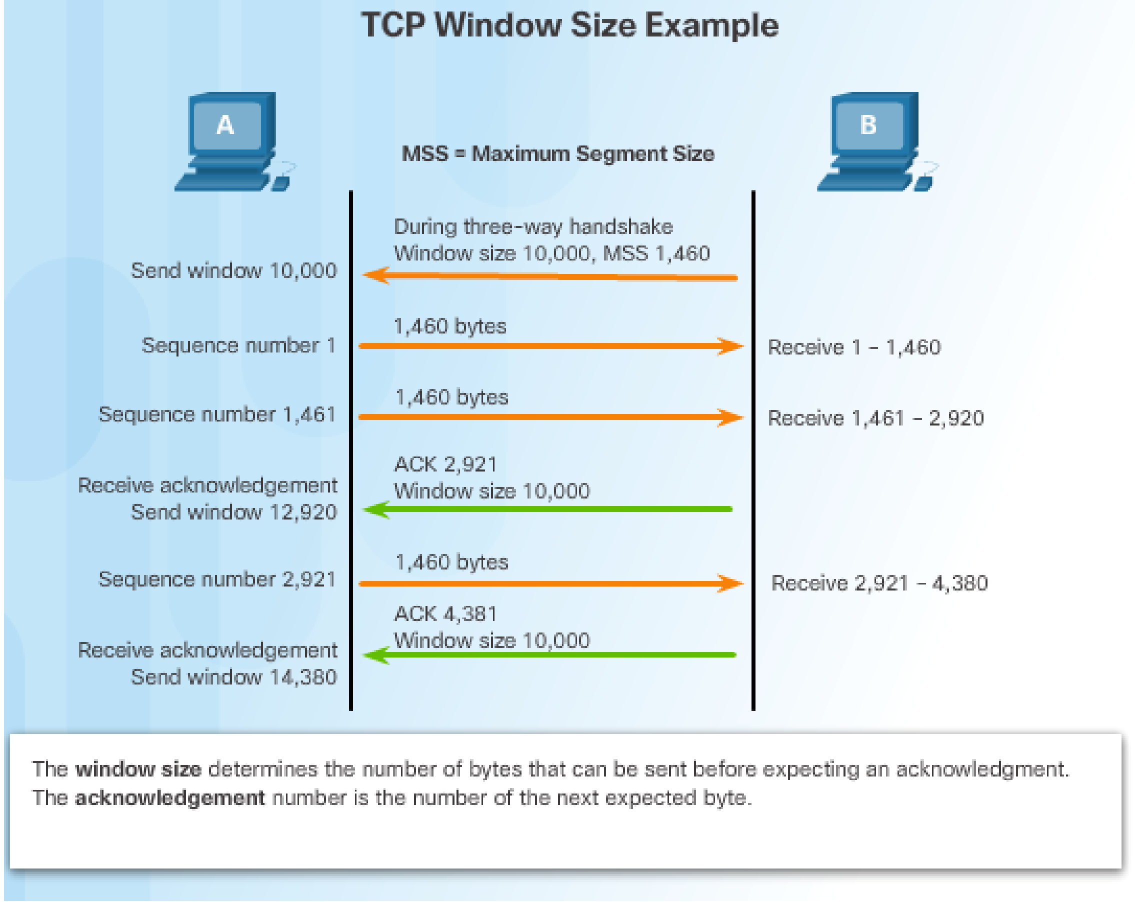
Video Demonstration – Data Loss and Retransmission



TCP Reliability-Data Loss and Retransmission
Demonstration
TCP Reliability - Data Loss and Retransmission

TCP Flow Control – Window Size and Acknowledgments

- In the figure, the source is transmitting 1,460 bytes of data within each segment.
- Window size agreed on during 3-way handshake.
- Typically, PC B will not wait for 10,000 bytes before sending an acknowledgment.
- PC A can adjust its send window as it receives acknowledgments from PC B.



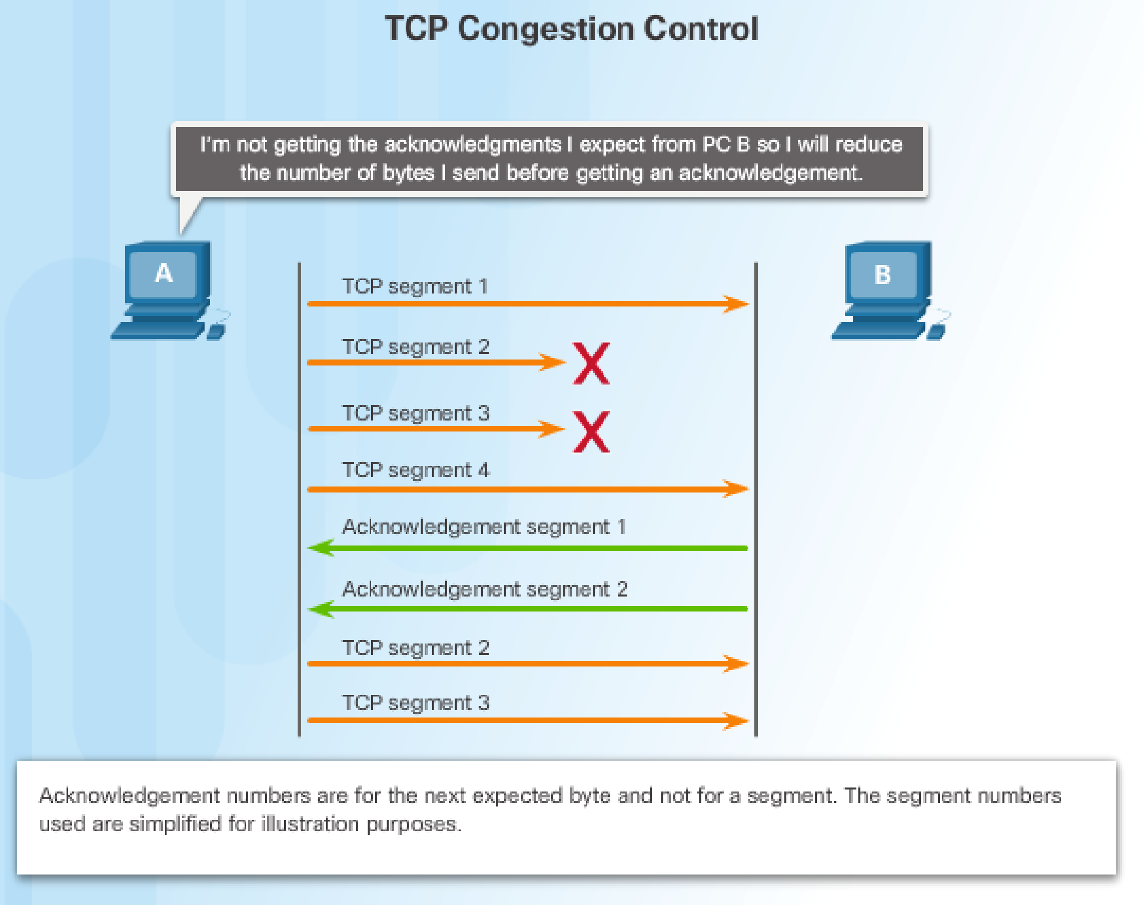
TCP Window Size Example

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.

Reliability and Flow Control

TCP Flow Control – Congestion Avoidance

- Congestion causes retransmission of lost TCP segments
- Retransmission of segments can make the congestion worse
- To avoid and control congestion, TCP employs several congestion handling mechanisms, timers, and algorithms
- Example: Reduce the number of bytes it sends before receiving an acknowledgment



TCP Congestion Control

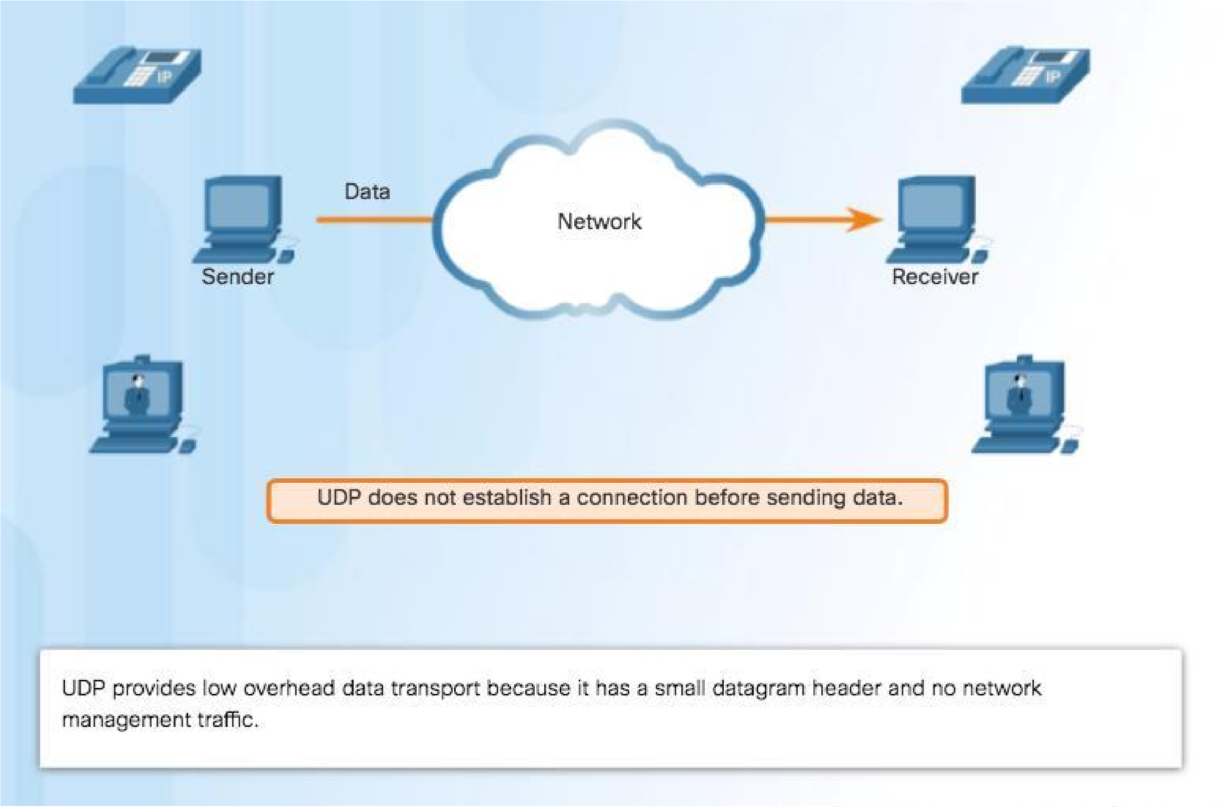
I'm not getting the acknowledgments I expect from PC B so I will reduce the number of bytes I send before getting an acknowledgement.

Acknowledgement numbers are for the next expected byte and not for a segment. The segment numbers used are simplified for illustration purposes.

UDP Communication

UDP Low Overhead versus Reliability

- UDP not connection-oriented
- No retransmission, sequencing, and flow control
- Functions not provided by the transport layer implemented elsewhere

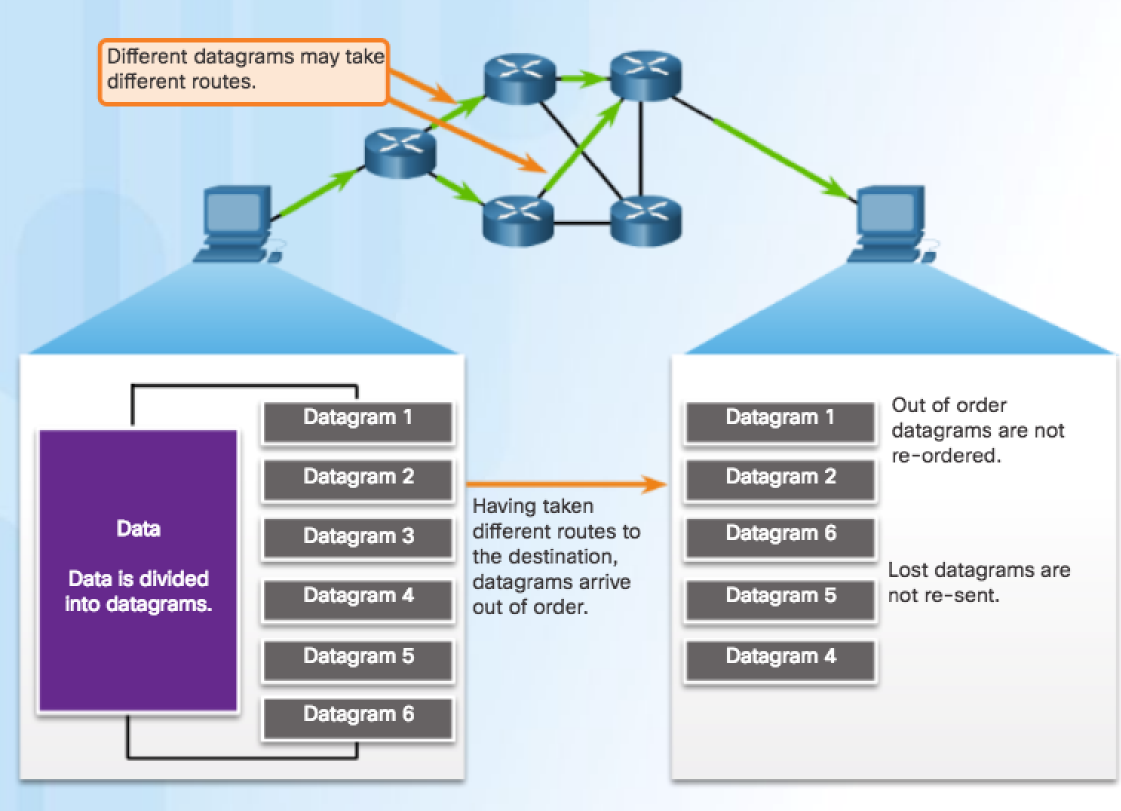


UDP does not establish a connection before sending data.

UDP provides low overhead data transport because it has a small datagram header and no network management traffic.

UDP Datagram Reassembly

- UDP reassembles data in order received and forwards to application
- Application must identify the proper sequence



Different datagrams may take different routes.

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

Out of order datagrams are not re-ordered.

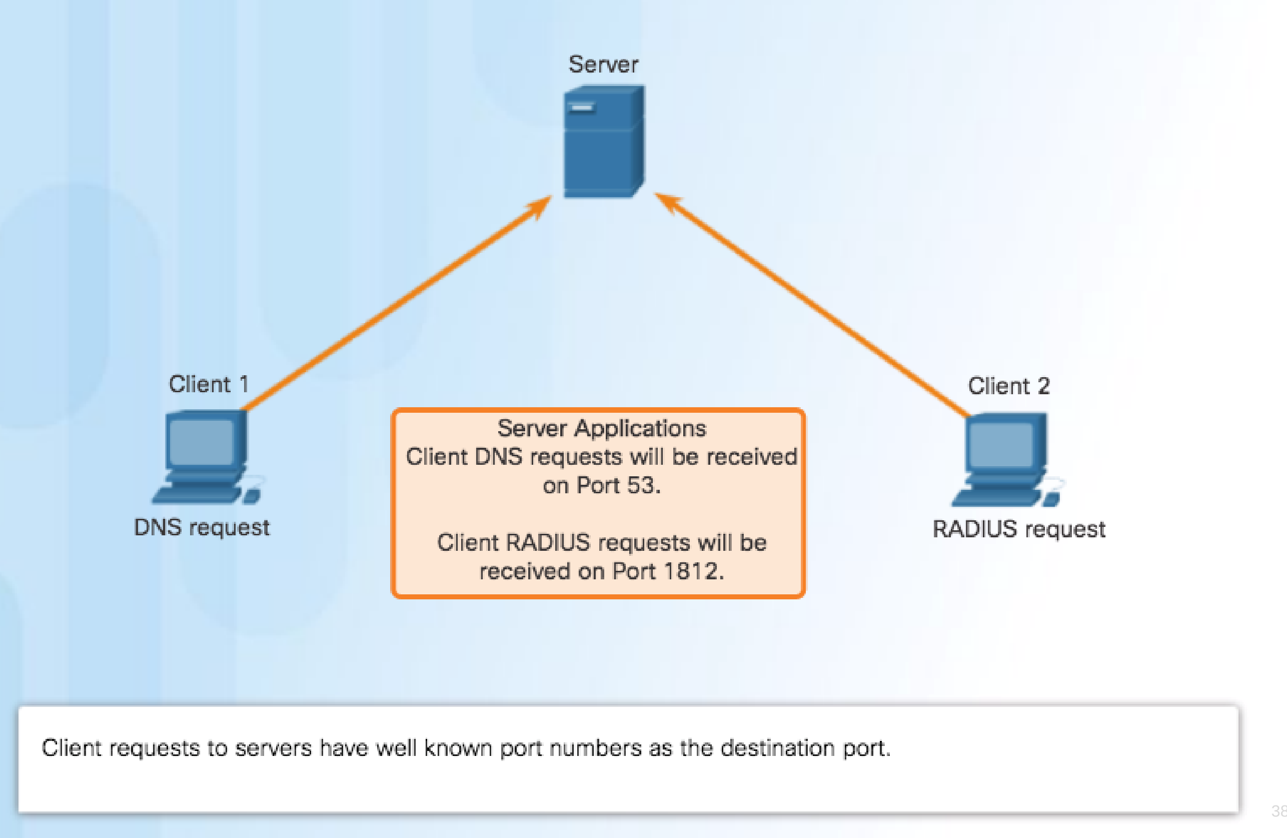
Lost datagrams are not re-sent.

UDP: Connectionless and Unreliable

UDP Communication

UDP Server Processes and Requests

Note: The Remote Authentication Dial-in User Service (RADIUS) server shown in the figure provides authentication, authorization, and accounting services to manage user access.



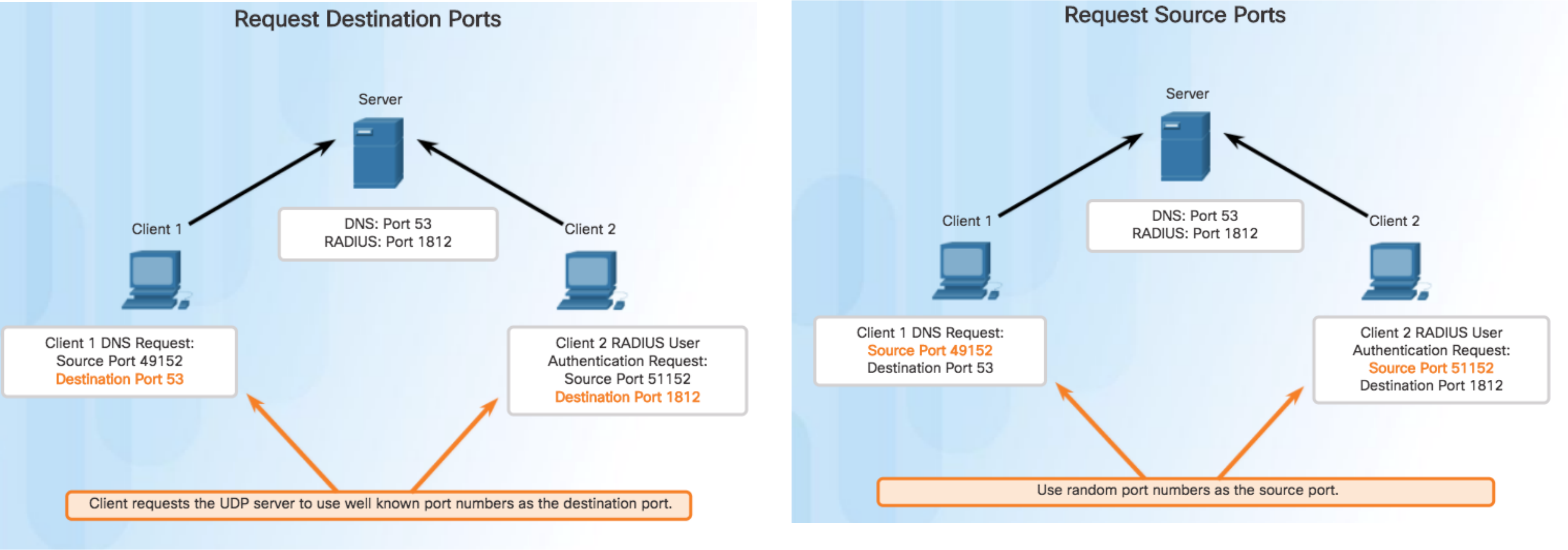
Server Applications
Client DNS requests will be received on Port 53.

Client RADIUS requests will be received on Port 1812.

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

UDP Communication

UDP Client Processes



Request Destination Ports

Server

DNS: Port 53
RADIUS: Port 1812

Client 1

Client 2

Client 1 DNS Request:
Source Port 49152
Destination Port 53

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

Client requests the UDP server to use well known port numbers as the destination port.

Request Source Ports

Server

DNS: Port 53
RADIUS: Port 1812

Client 1

Client 2

Client 1 DNS Request:
Source Port 49152
Destination Port 53

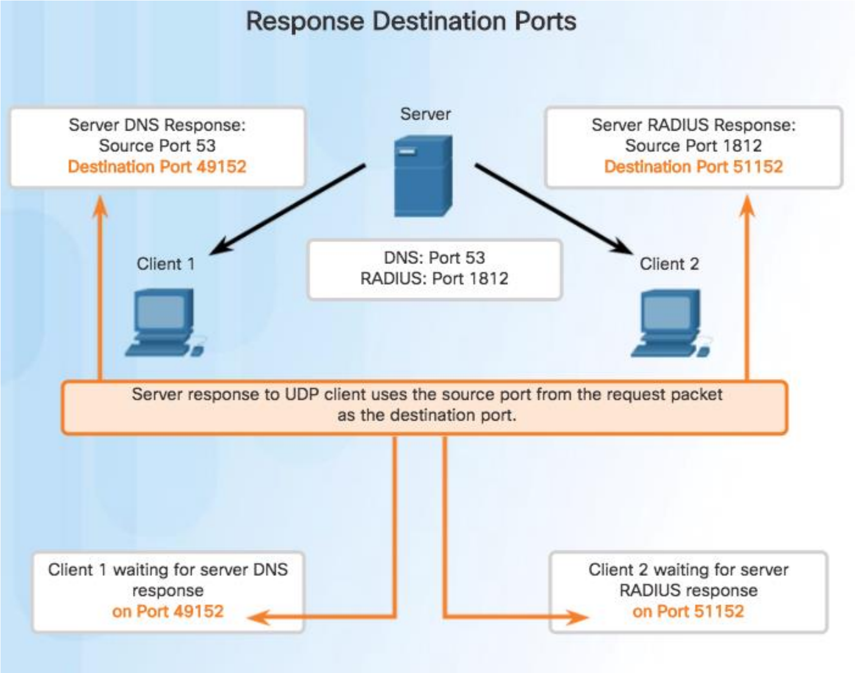
Client 2 RADIUS User Authentication Request:
Source Port 51152
Destination Port 1812

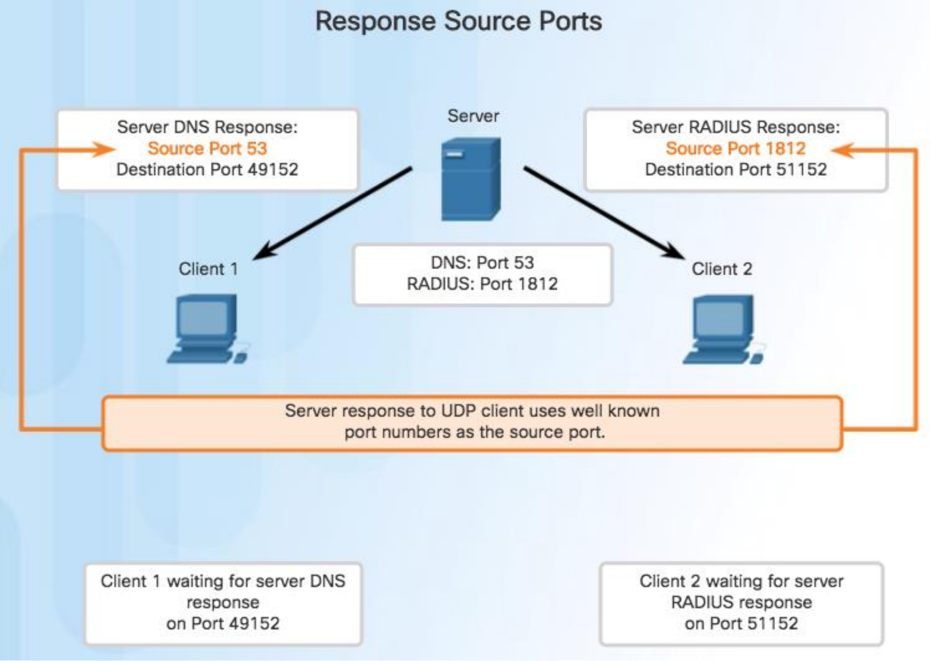
Use random port numbers as the source port.

Clients Sending UDP Requests

UDP Communication

UDP Client Processes (Cont.)

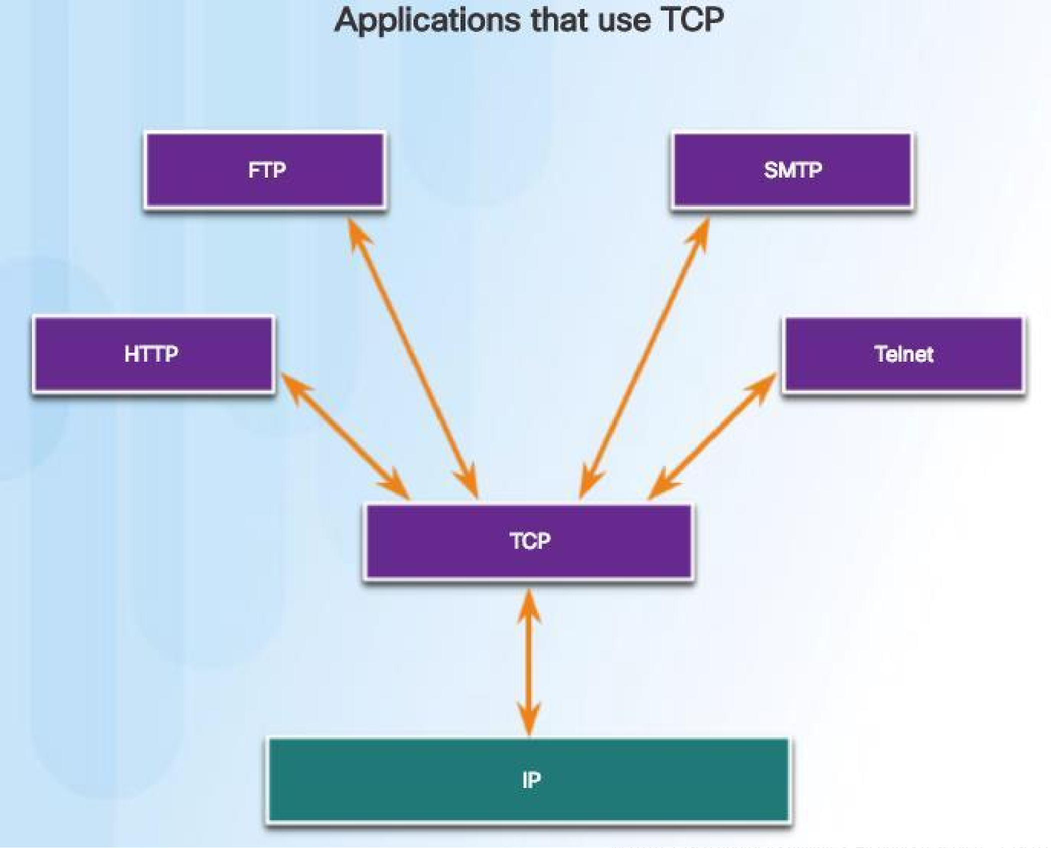

Response Destination Ports
Server DNS Response:
Source Port 53
Destination Port 49152
Server
Server RADIUS Response:
Source Port 1812
Destination Port 51152
Client 1
DNS: Port 53
RADIUS: Port 1812
Client 2
Server response to UDP client uses the source port from the request packet as the destination port.
Client 1 waiting for server DNS response on Port 49152
Client 2 waiting for server RADIUS response on Port 51152


Response Source Ports
Server DNS Response:
Source Port 53
Destination Port 49152
Server
Server RADIUS Response:
Source Port 1812
Destination Port 51152
Client 1
DNS: Port 53
RADIUS: Port 1812
Client 2
Server response to UDP client uses 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

Clients Sending UDP Requests

Applications that use TCP

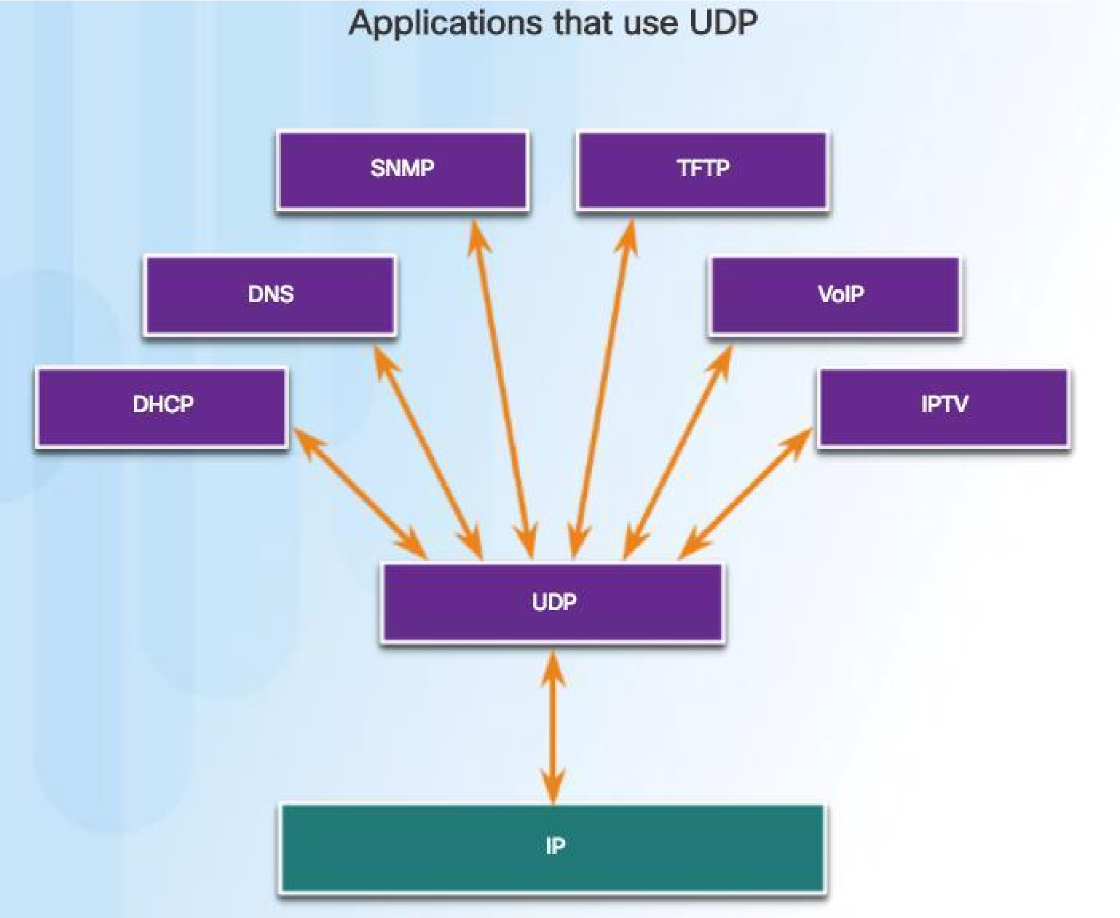
TCP frees applications from having to manage reliability



Applications that use UDP

Three types of applications best suited for UDP:

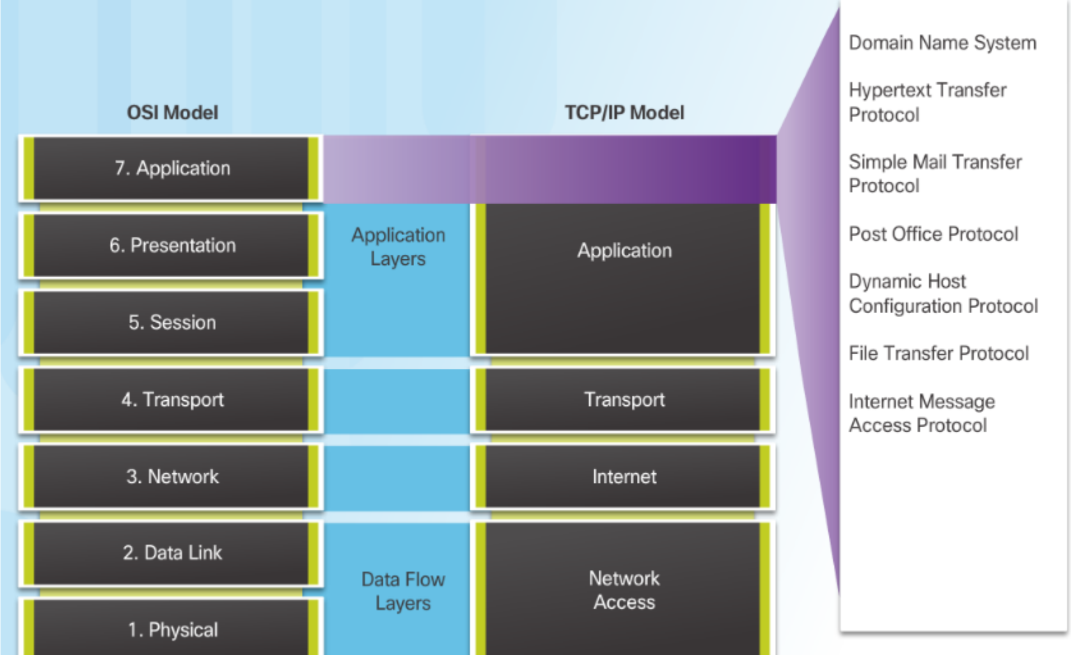
- Live video and multimedia
- Simple request and reply
- Handle reliability themselves



Application Layer Protocols

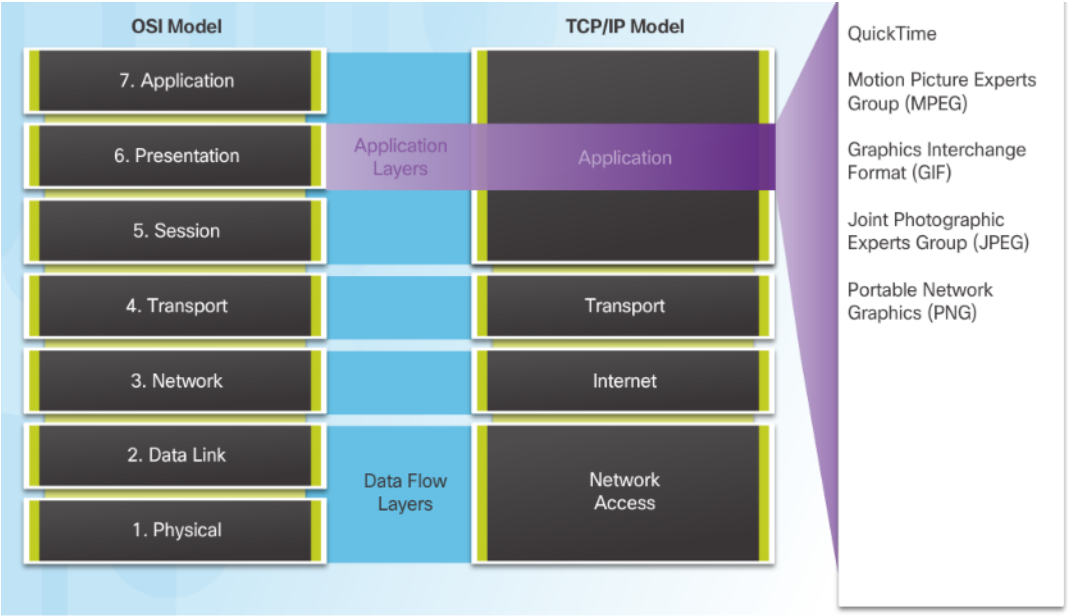
Application, Presentation, and Session

Application Layer



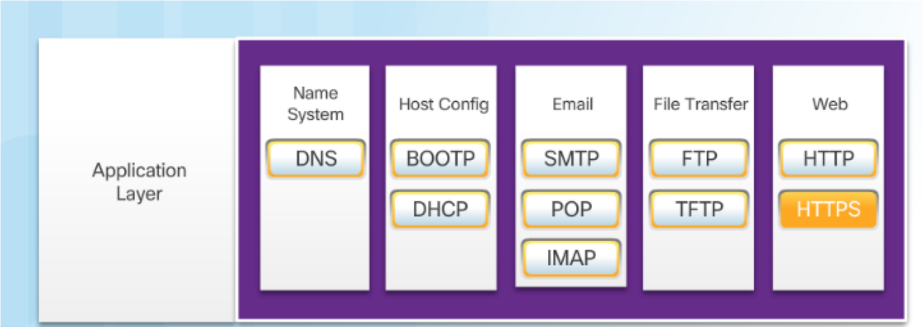
- Application Layer:
  - Closest to the end user.
  - Used to exchange data between programs running on the source and destination hosts.

Presentation and Session Layer



- Presentation Layer function:
  - Formatting data at the source device into a compatible form for the receiving device.
  - Compressing data.
  - Encrypting data.
- Session Layer Function
  - Create and maintain dialogs between source and destination applications.

TCP/IP Application Layer Protocols

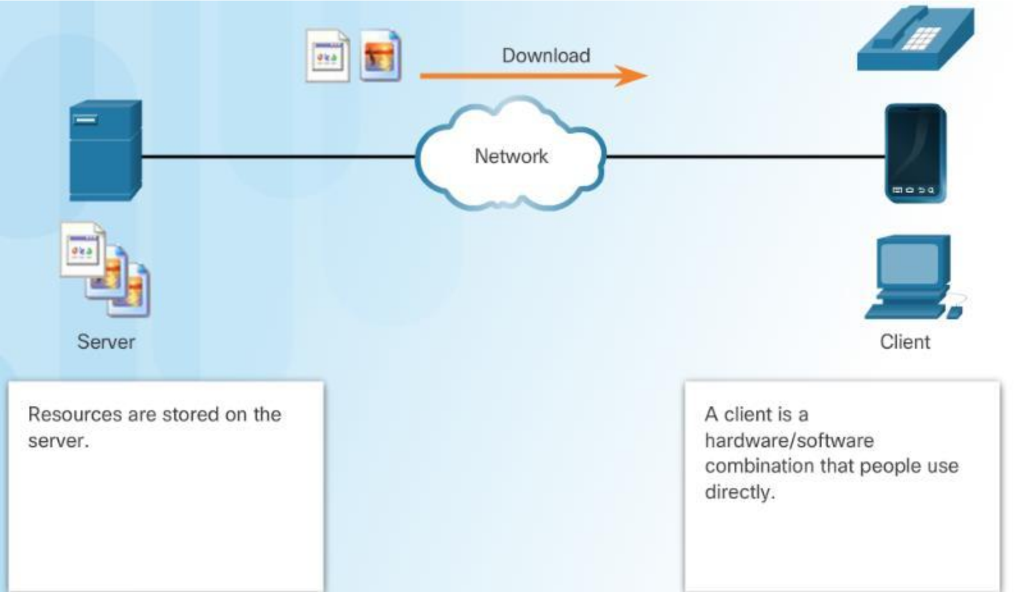


- Domain Name Server (DNS) TCP,UDP 53 - Translates domain names, such as cisco.com, into IP addresses.
- (BOOTP) – Bootstrap Protocol - BOOTP is being superseded by DHCP.
- Dynamic Host Configuration Protocol (DHCP) UDP client 68, server 67 – Dynamically assigns IP addresses to client stations at start-up.
- Simple Mail Transport Protocol (SMTP) TCP 25 - Enables clients to send email to a mail server.
- Post Office Protocol (POP) TCP 110 - Enables clients to retrieve email from a mail server.
- Internet Message Access Protocol (IMAP) TCP 143 - Enables clients to retrieve email from a mail server, maintains email on server.
- File Transfer Protocol (FTP) TCP 20 and 21 - Reliable, connection-oriented, and acknowledged file delivery protocol.
- Trivial File Transfer Protocol (TFTP) UDP 69 – simple connectionless file transfer protocol.
- Hypertext Transfer Protocol (HTTP) TCP 80, 8080 - Set of rules for exchanging text, graphic images, etc. on the World Wide Web.
- Hypertext Transfer Protocol Secure (HTTPS) TCP, UDP 443 – Uses encryption and authentication to secure communication.

How Application Protocols Interact with End-User Applications

Client-Server Model

- Client and server processes are considered to be in the application layer.
- Application layer protocols describe the format of the requests and responses between clients and servers.
- Example of a client-server network is using an ISP’s email service to send, receive and store email.



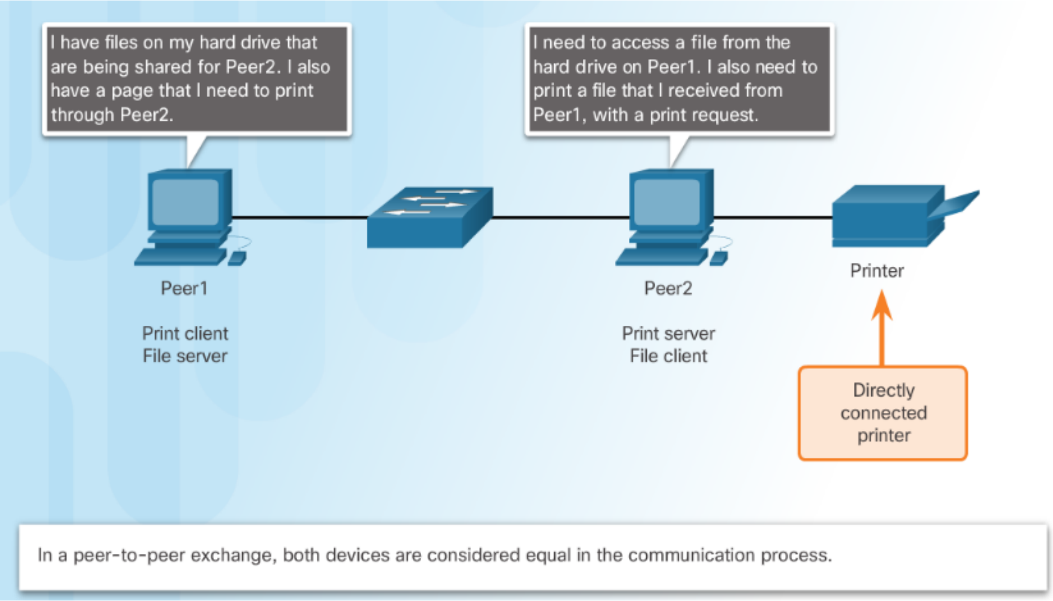
Resources are stored on the server.

A client is a hardware/software combination that people use directly.

How Application Protocols Interact with End-User Applications

Peer-to-Peer Networks

- Data is accessed from a peer device without the use of a dedicated server.
- Each device (known as a peer) can function as both a server and a client.

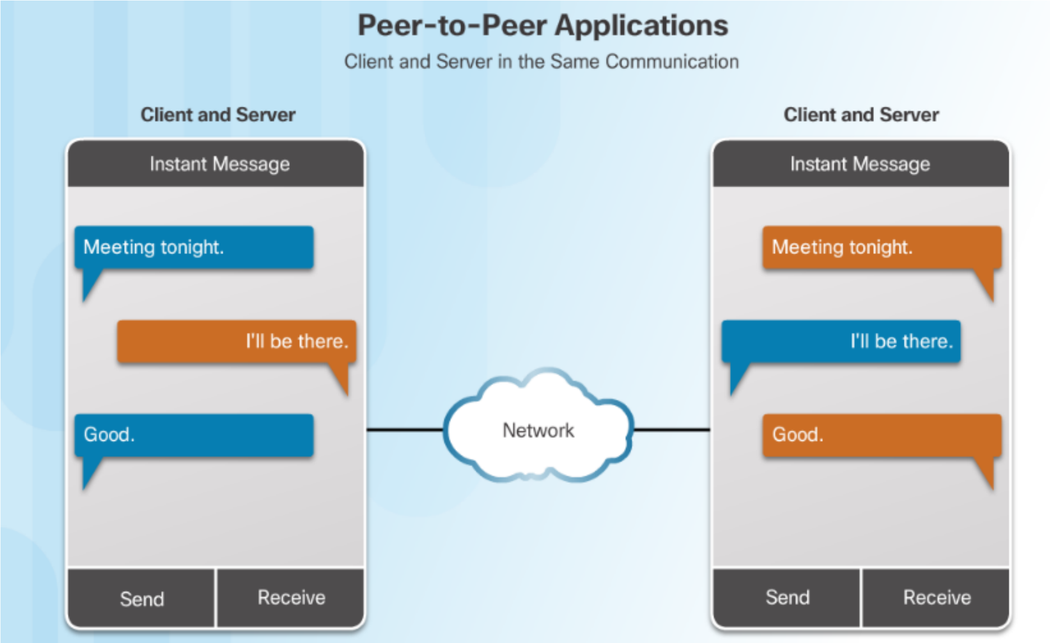

I have files on my hard drive that are being shared for Peer2. I also have a page that I need to print through Peer2.
I need to access a file from the hard drive on Peer1. I also need to print a file that I received from Peer1, with a print request.
Peer1
Print client
File server
Peer2
Print server
File client
Printer
Directly connected printer

In a peer-to-peer exchange, both devices are considered equal in the communication process.

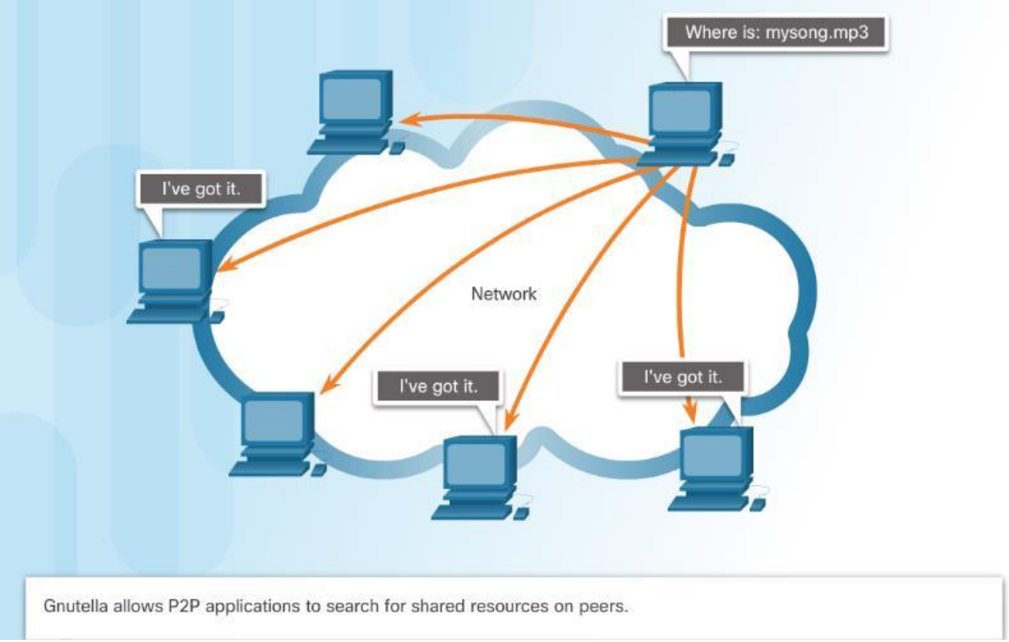
How Application Protocols Interact with End-User Applications

Peer-to-Peer Applications

- A P2P application allows a device to act as both a client and a server within the same communication.
- P2P applications require that each end device provide a user interface and run a background service.



Common P2P Applications



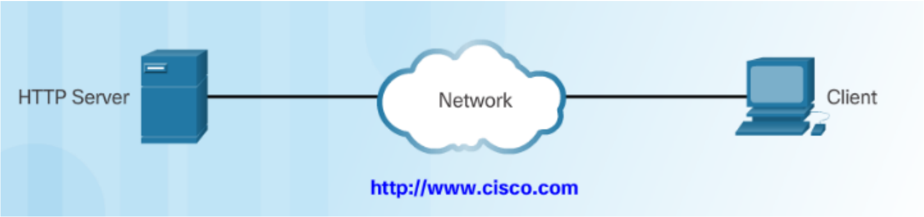
Gnutella allows P2P applications to search for shared resources on peers.

- Common P2P networks include:
  - G2
  - Bitcoin
  - BitTorrent
  - eDonkey
- Some P2P applications are based on the Gnutella protocol, where each user shares whole files with other users.
- Many P2P applications allow users to share pieces of many files with each other at the same time –this is BitTorrent technology.

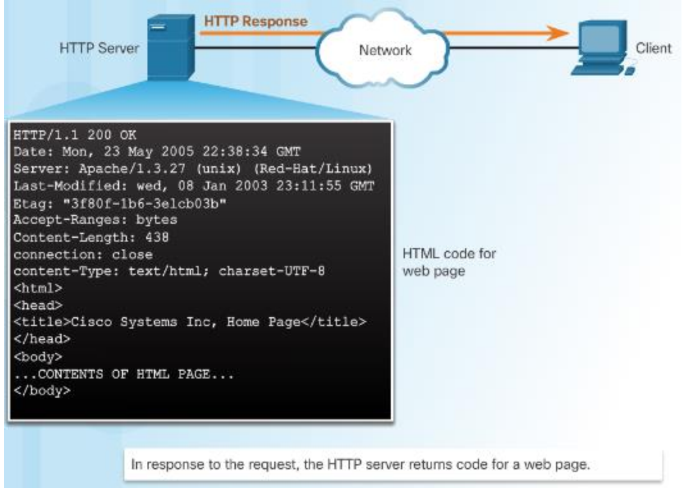
Well-Known Application Layer Protocols and Services

Web and Email Protocols

Hypertext Transfer Protocol and Hypertext Markup Language

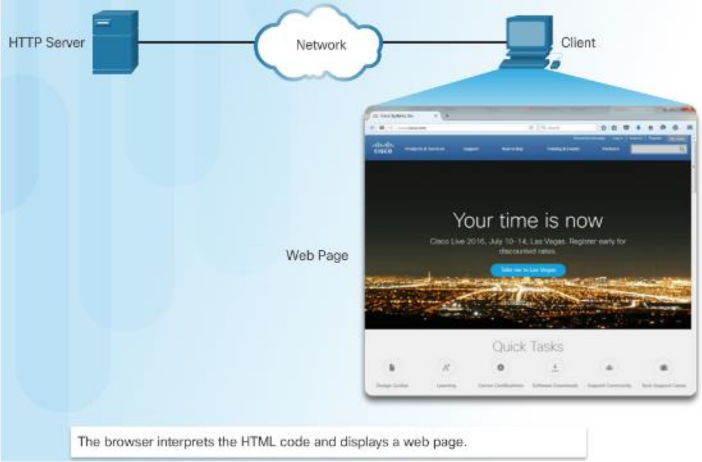


- When a web address or uniform resource locator (URL) is typed into a web browser, the web browser establishes a connection to the web service running on the server, using the HTTP protocol.



```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Server: Apache/1.3.27 (unix) (Red-Hat/Linux)
Last-Modified: wed, 08 Jan 2003 23:11:55 GMT
Etag: "3f80f-1b6-3elcb03b"
Accept-Ranges: bytes
Content-Length: 438
connection: close
content-Type: text/html; charset-UTF-8
<html>
<head>
<title>Cisco Systems Inc, Home Page</title>
</head>
<body>
...CONTENTS OF HTML PAGE...
</body>
```

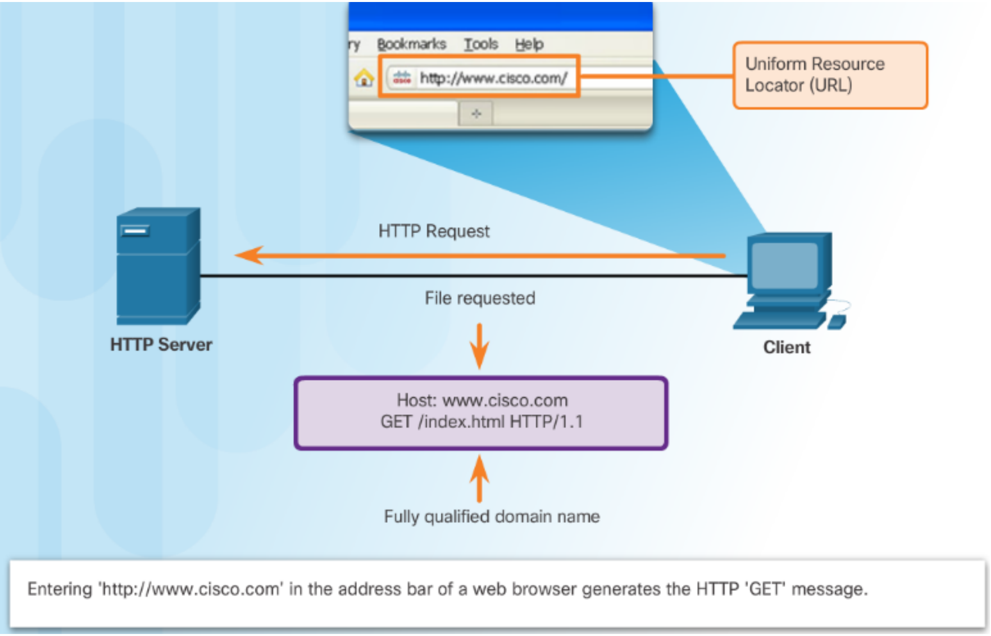
In response to the request, the HTTP server returns code for a web page.



The browser interprets the HTML code and displays a web page.

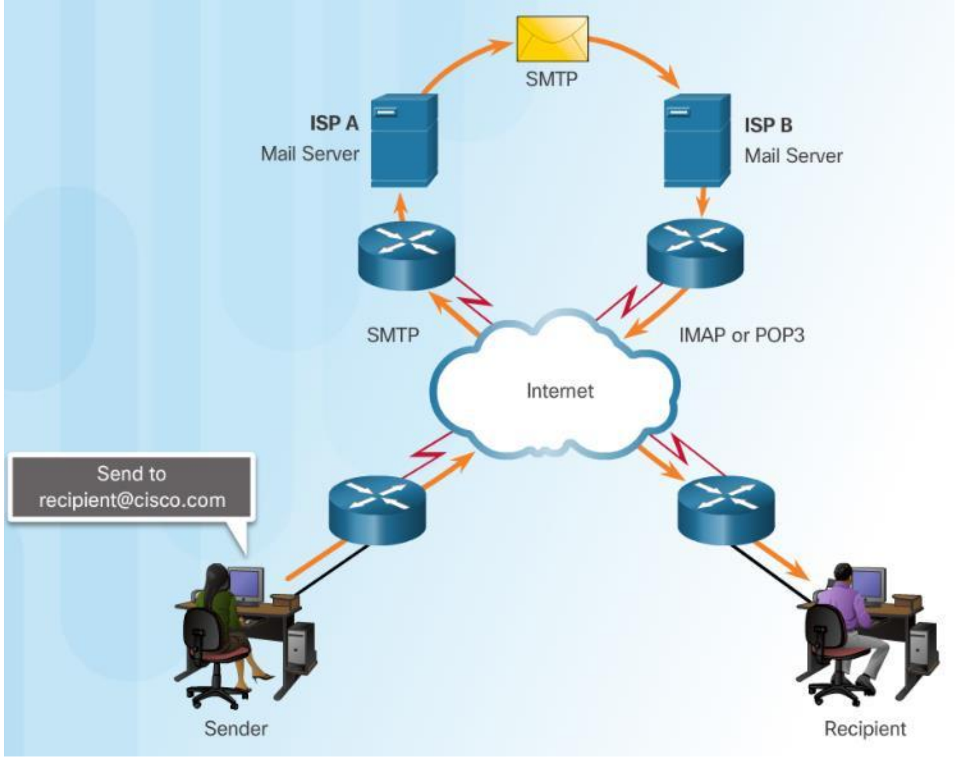
HTTP and HTTPS

- HTTP is a request/response protocol.
- Three common HTTP message types are:
  - GET - A client request for data.
  - POST - Uploads data files to the web server.
  - PUT - Uploads resources or content to the web server.
- HTTP Secure (HTTPS) protocol uses encryption and authentication to secure data.



Entering 'http://www.cisco.com' in the address bar of a web browser generates the HTTP 'GET' message.

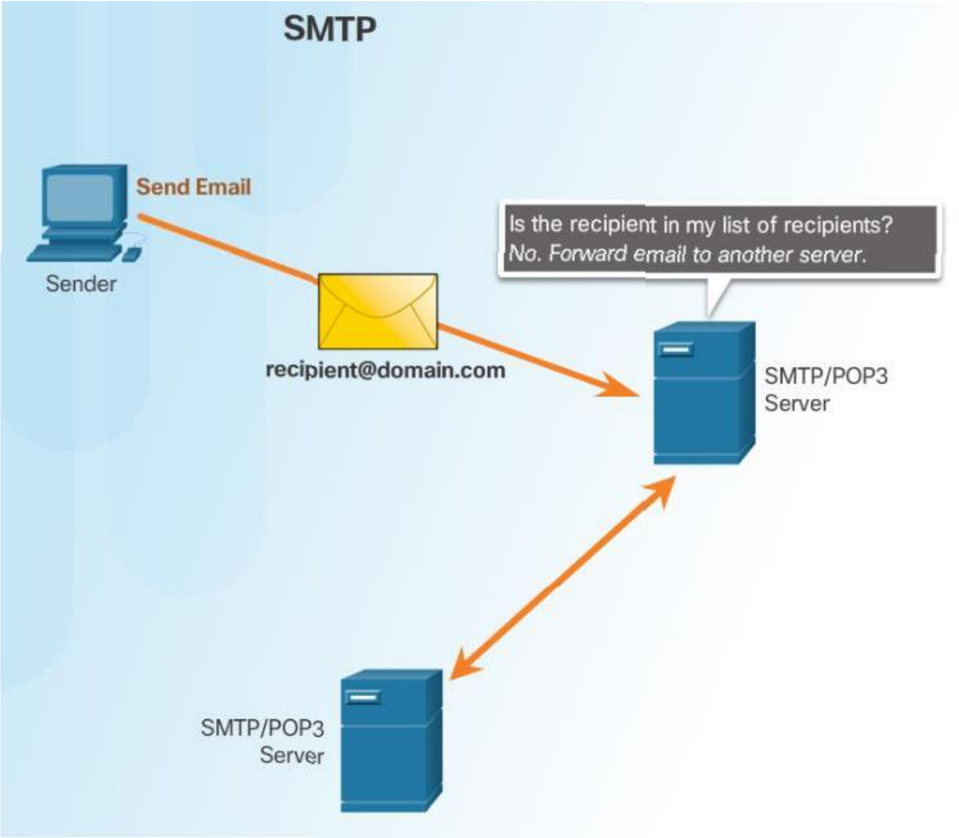
Email Protocols



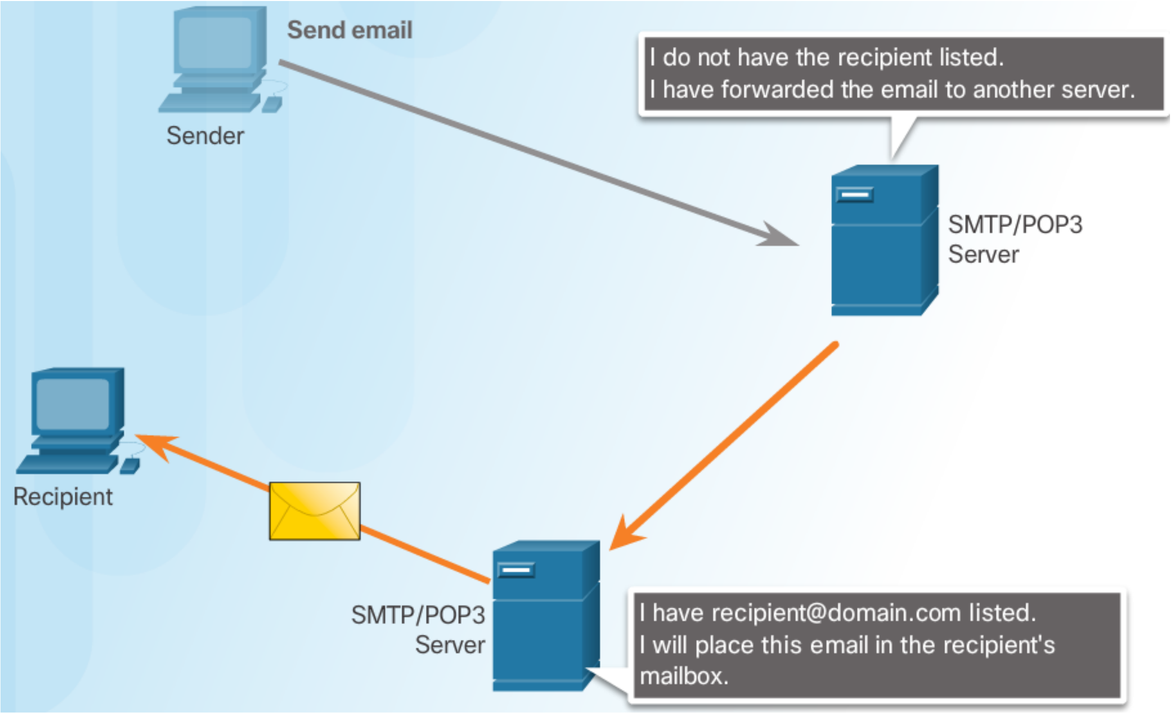
- Email clients communicate with mail servers to send and receive email.
- Mail servers communicate with other mail servers to transport messages from one domain to another.
- Three protocols for email:
  - Simple Mail Transfer Protocol (SMTP) to send email.
  - Post Office Protocol (POP) to retrieve email.
  - Internet Message Access Protocol (IMAP) to retrieve email.

SMTP Operation

- SMTP is used to send email


SMTP
Send Email
Sender
recipient@domain.com
Is the recipient in my list of recipients?
No. Forward email to another server.
SMTP/POP3 Server
SMTP/POP3 Server

POP Operation

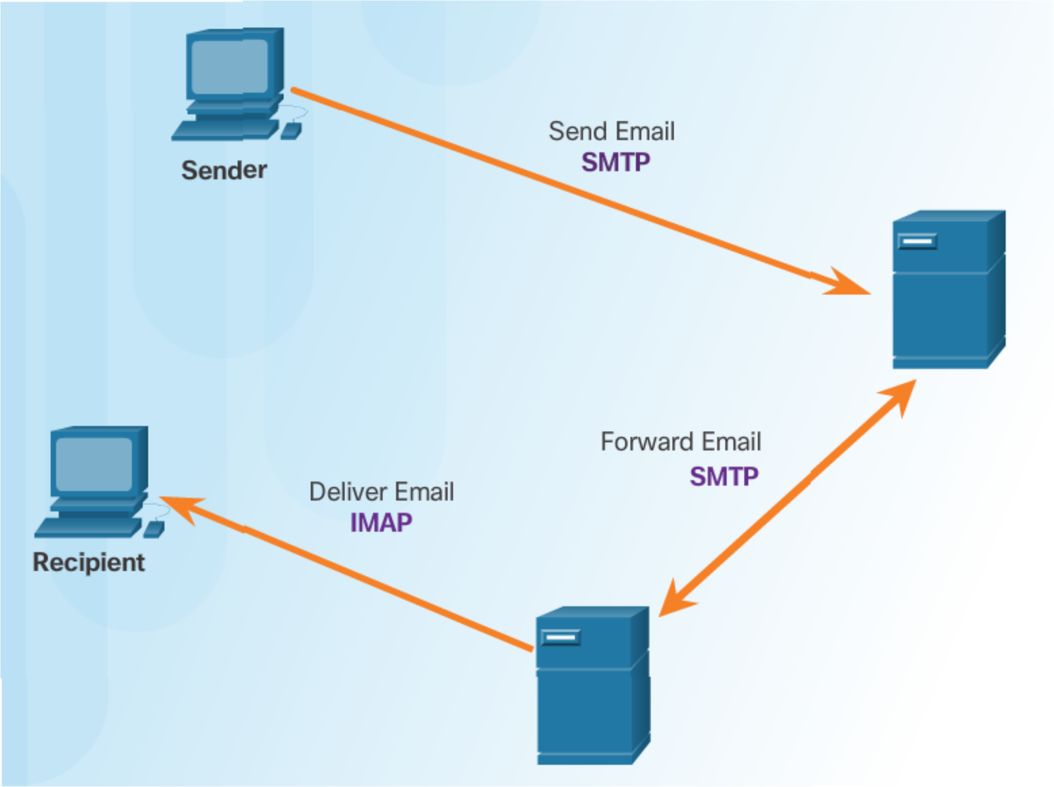


I do not have the recipient listed.
I have forwarded the email to another server.

I have recipient@domain.com listed.
I will place this email in the recipient's mailbox.

- POP is used to retrieve email from a mail server.
- Email is downloaded from the server to the client and then deleted on the server.

IMAP Operation

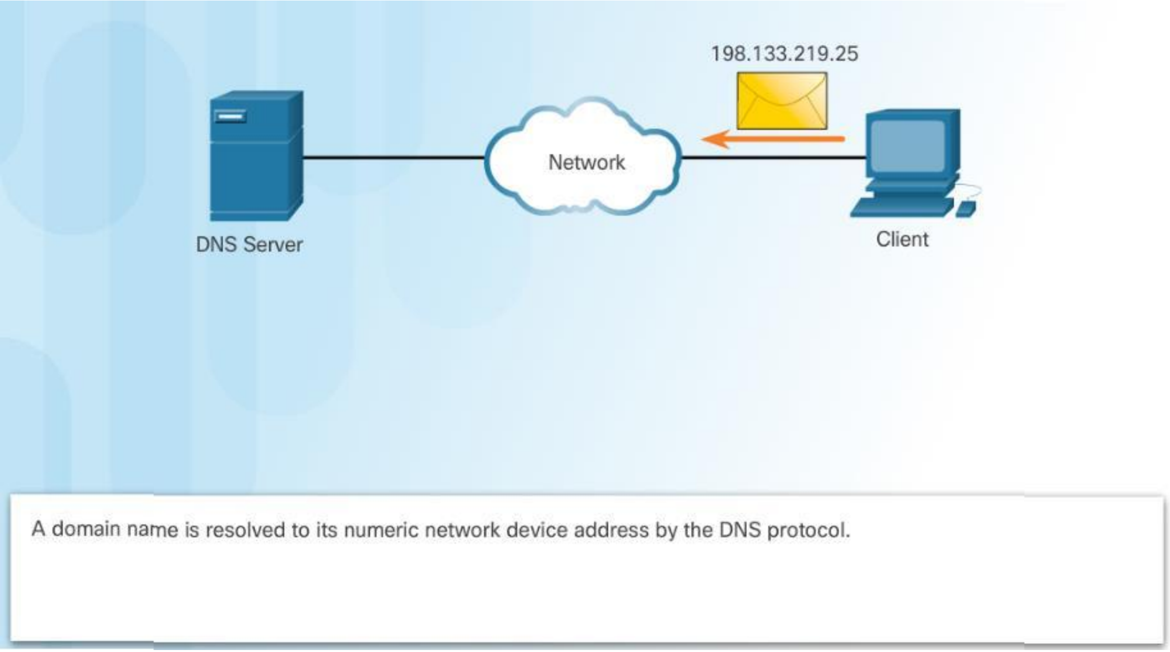


- IMAP is used to retrieve mail from a mail server.
- Copies of messages are downloaded from the server to the client and the original messages are stored on the server.

IP Addressing Services

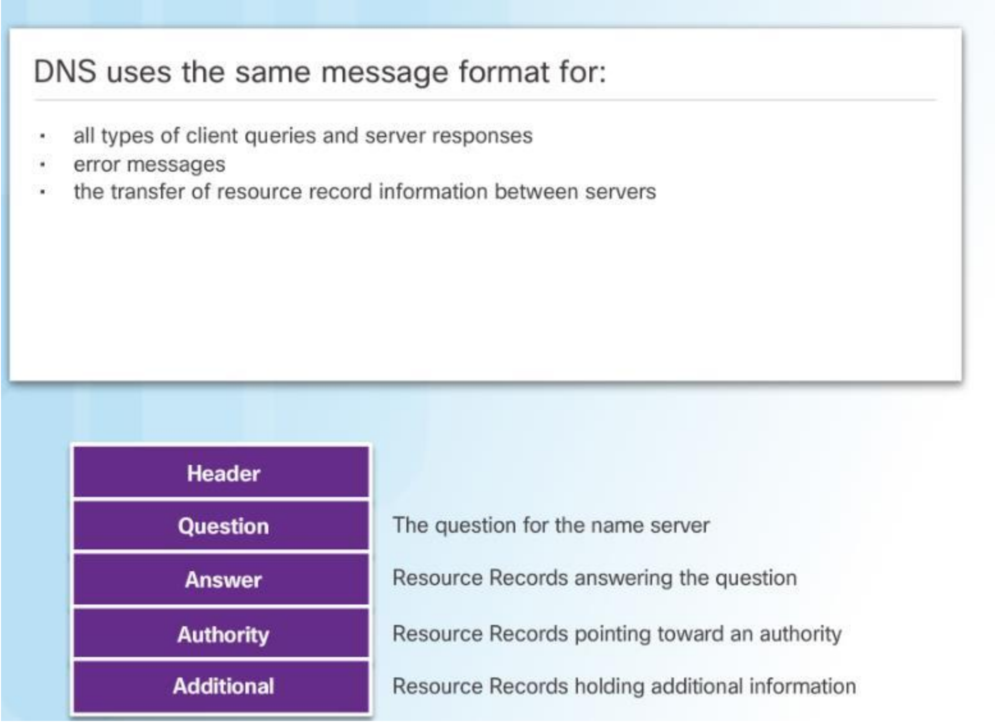
Domain Name Service

- Domain names convert the numeric address into a simple, recognizable name.
- The DNS protocol defines an automated service that matches resource names with the required numeric network address.



A domain name is resolved to its numeric network device address by the DNS protocol.

DNS Message Format



DNS uses the same message format for:

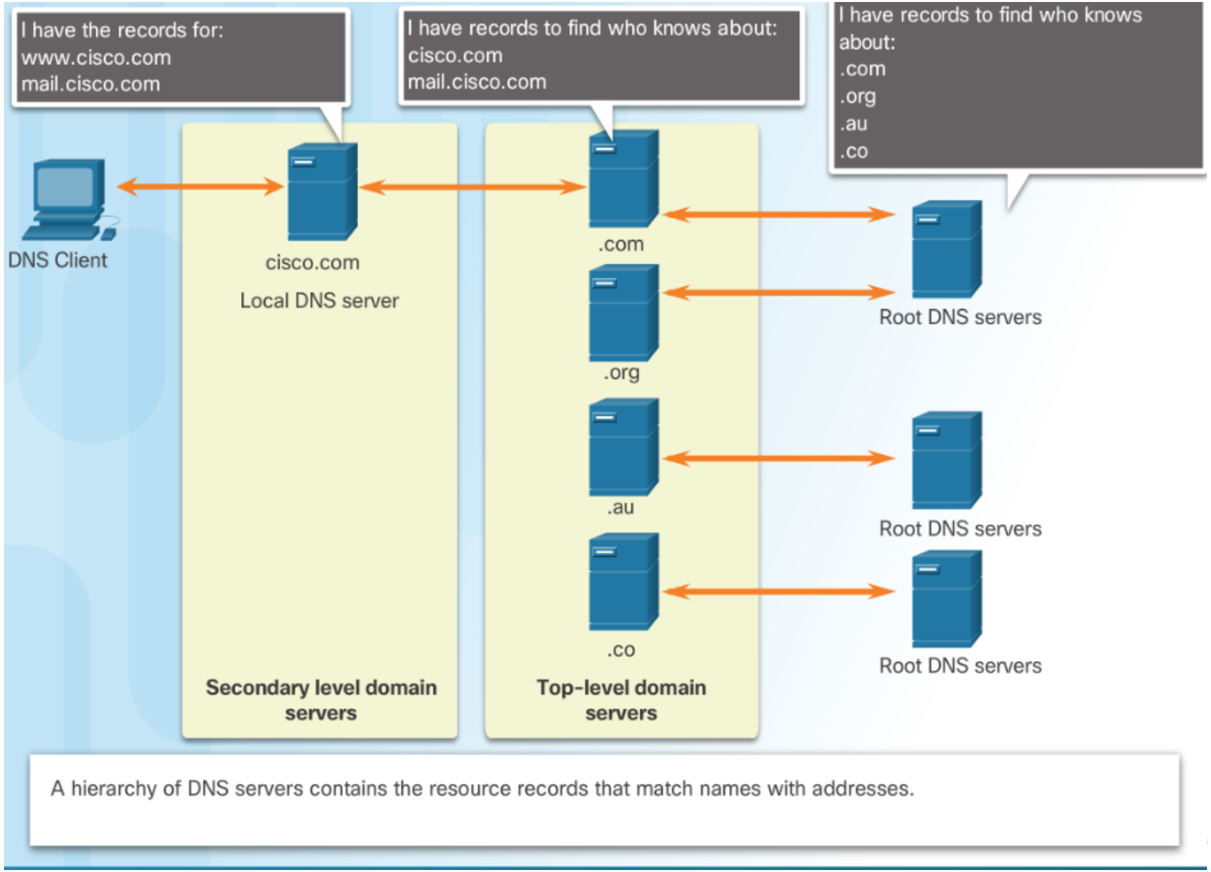
- all types of client queries and server responses
- error messages
- the transfer of resource record information between servers

Header	
Question	The question for the name server
Answer	Resource Records answering the question
Authority	Resource Records pointing toward an authority
Additional	Resource Records holding additional information

- When a client makes a query, the server's DNS process first looks at its own records to resolve the name.
- If unable to resolve, it contacts other servers to resolve the name.
- The server temporarily stores the numbered address in the event that the same name is requested again.
- The **ipconfig /displaydns** command displays all of the cached DNS entries on a Windows PC.

IP Addressing Services

DNS Hierarchy



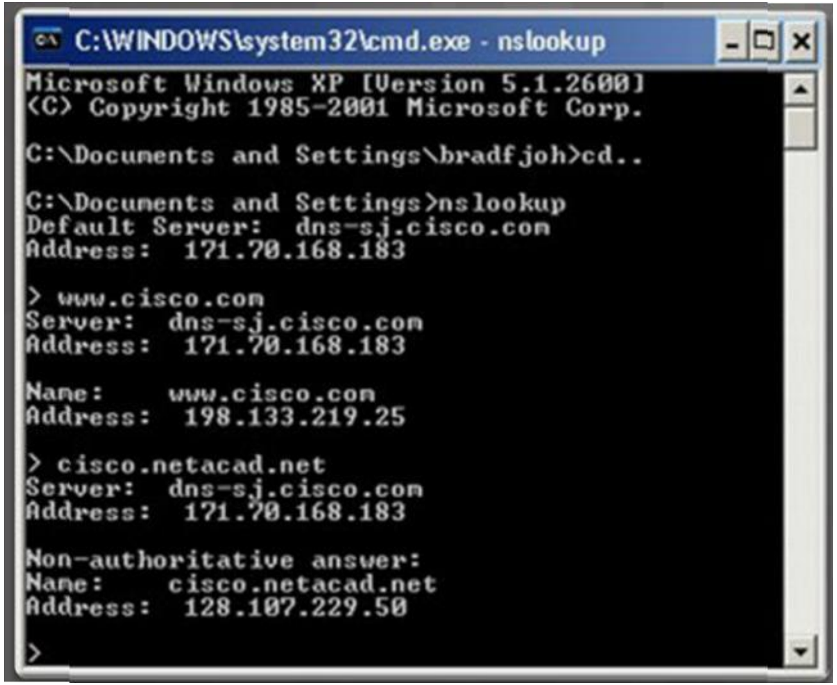
I have the records for:
www.cisco.com
mail.cisco.com

I have records to find who knows about:
cisco.com
mail.cisco.com

I have records to find who knows about:
.com
.org
.au
.co

A hierarchy of DNS servers contains the resource records that match names with addresses.

The nslookup Command



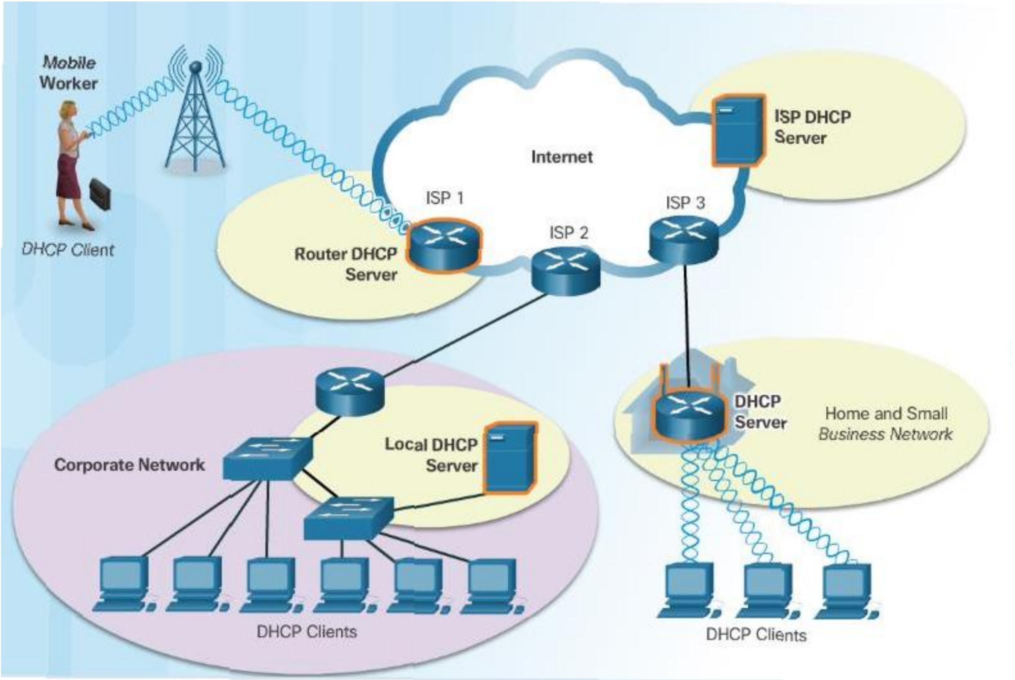
```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\bradfjoh>cd..

C:\Documents and Settings>nslookup
Default Server:  dns-sj.cisco.com
Address:  171.70.168.183

> www.cisco.com
Server:  dns-sj.cisco.com
Address:  171.70.168.183

Name:    www.cisco.com
Address:  198.133.219.25

> cisco.netacad.net
Server:  dns-sj.cisco.com
Address:  171.70.168.183

Non-authoritative answer:
Name:    cisco.netacad.net
Address:  128.107.229.50

>
```

- **Nslookup** - a utility that allows a user to manually query the name servers to resolve a given host.
  - Can also be used to troubleshoot name resolution issues and to verify the current status of the name servers.

IP Addressing Services

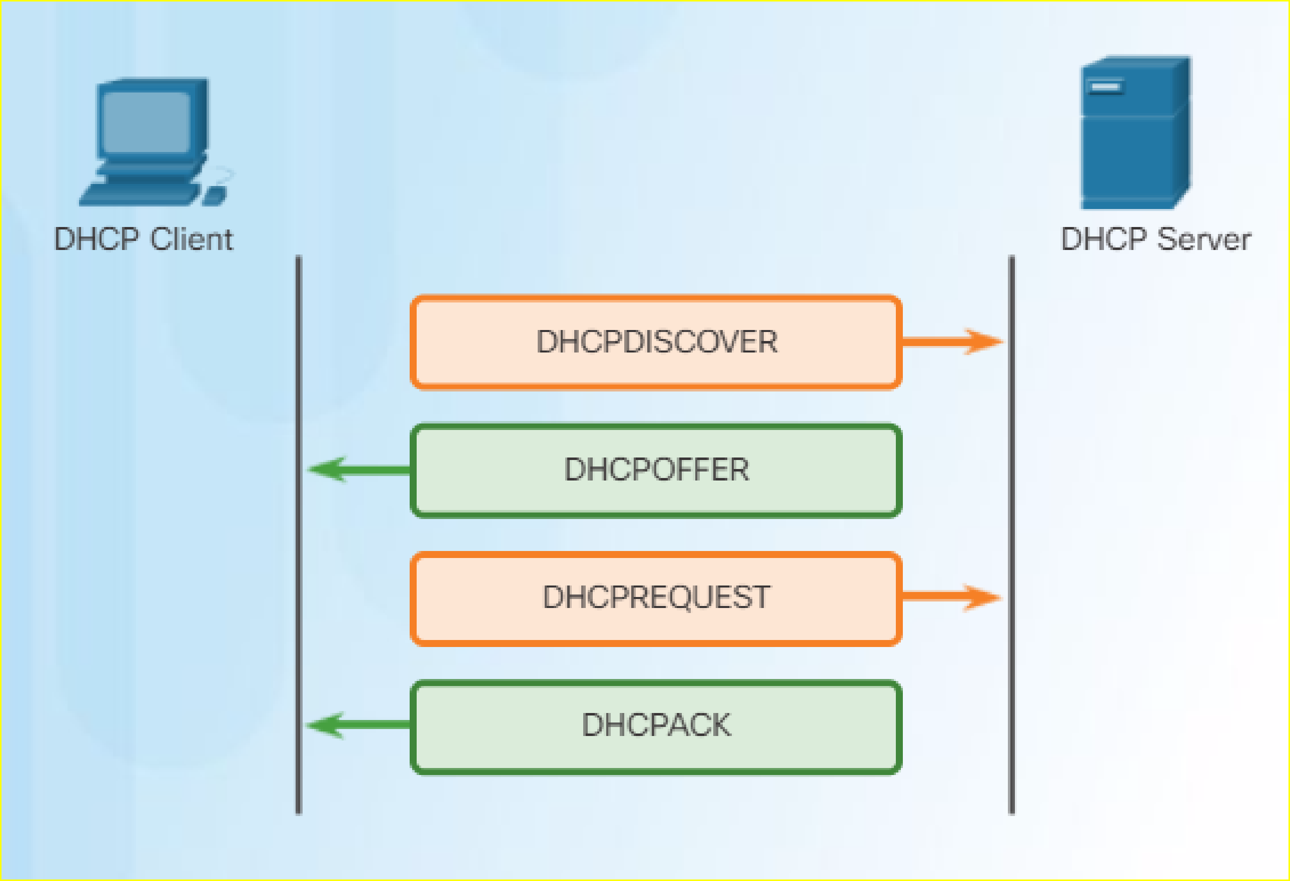
Dynamic Host Configuration Protocol

- The Dynamic Host Configuration Protocol (DHCP) for IPv4 automates the assignment of IPv4 addresses, subnet masks, gateways, and other parameters.
- DHCP-distributed addresses are leased for a set period of time, then returned to pool for reuse.
- DHCP is usually employed for end user devices. Static addressing is used for network devices, such as gateways, switches, servers, and printers.
- DHCPv6 (DHCP for IPv6) provides similar services for IPv6 clients.



IP Addressing Services

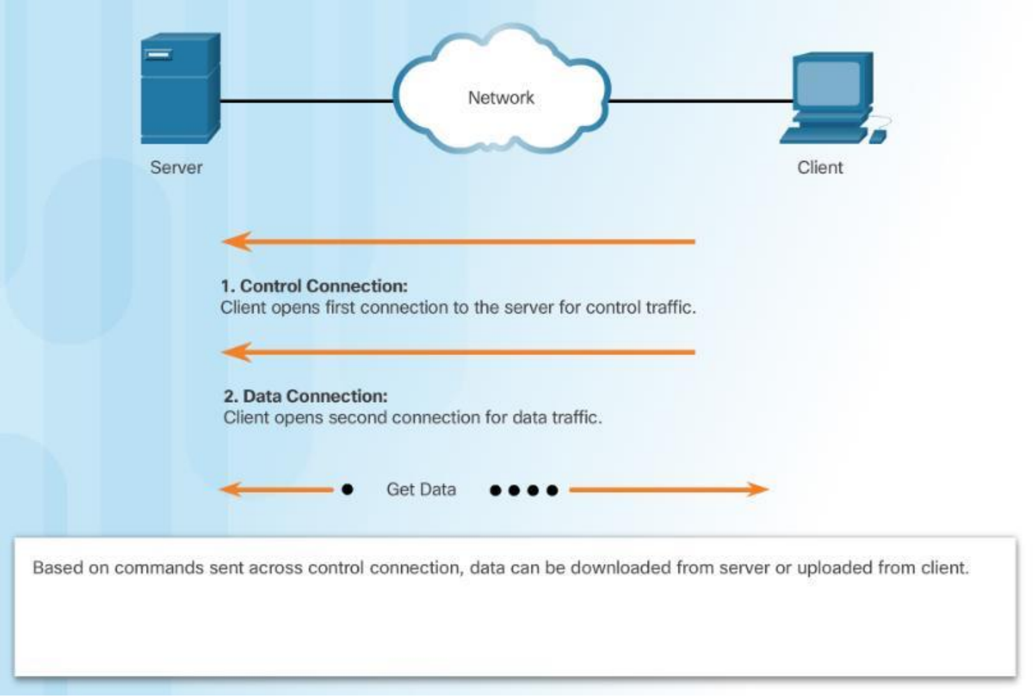
DHCP Operation



File Sharing Services

File Transfer Protocol

- FTP requires two connections between the client and the server, one for commands and replies, the other for the actual file transfer:
  - The client establishes the first connection to the server for control traffic using TCP port 21.
  - The client establishes the second connection to the server for the actual data transfer using TCP port 20.



1. Control Connection:
Client opens first connection to the server for control traffic.

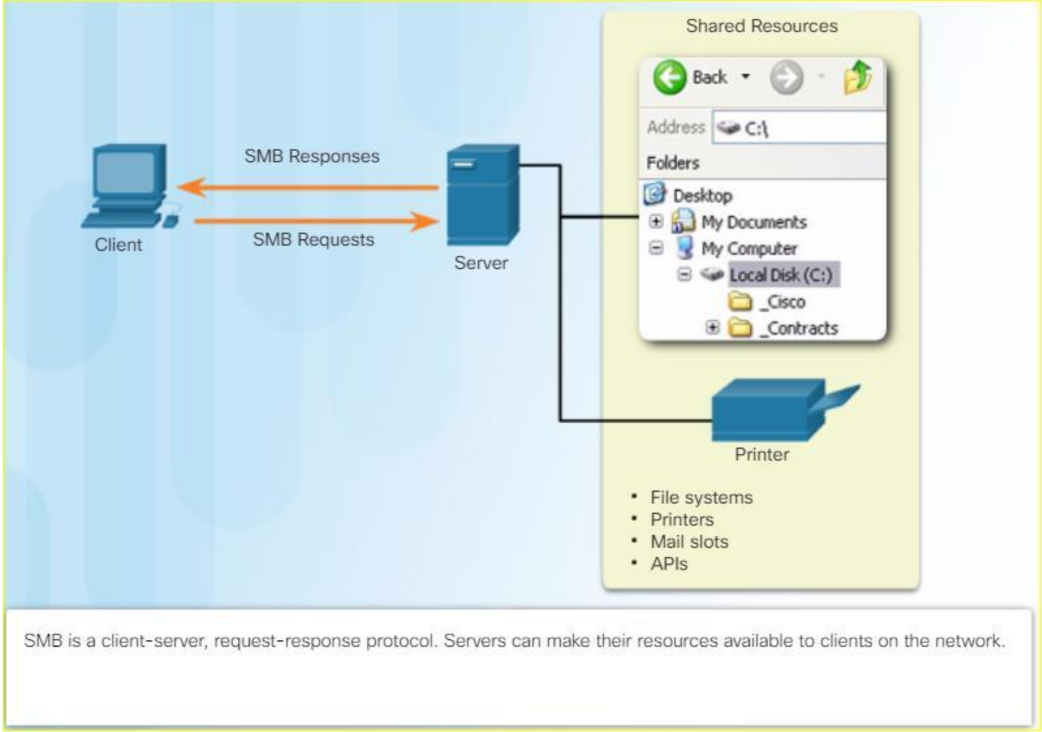
2. Data Connection:
Client opens second connection for data traffic.

Based on commands sent across control connection, data can be downloaded from server or uploaded from client.

File Sharing Services

Server Message Block

- The Server Message Block (SMB) is a client/server file sharing protocol:
  - SMB file-sharing and print services have become the mainstay of Microsoft networking.
  - Clients establish a long-term connection to servers and can access the resources on the server as if the resource is local to the client host.



SMB is a client-server, request-response protocol. Servers can make their resources available to clients on the network.