Chapter 9: Transport Layer

CCNA Routing and Switching

Introduction to Networks v6.0

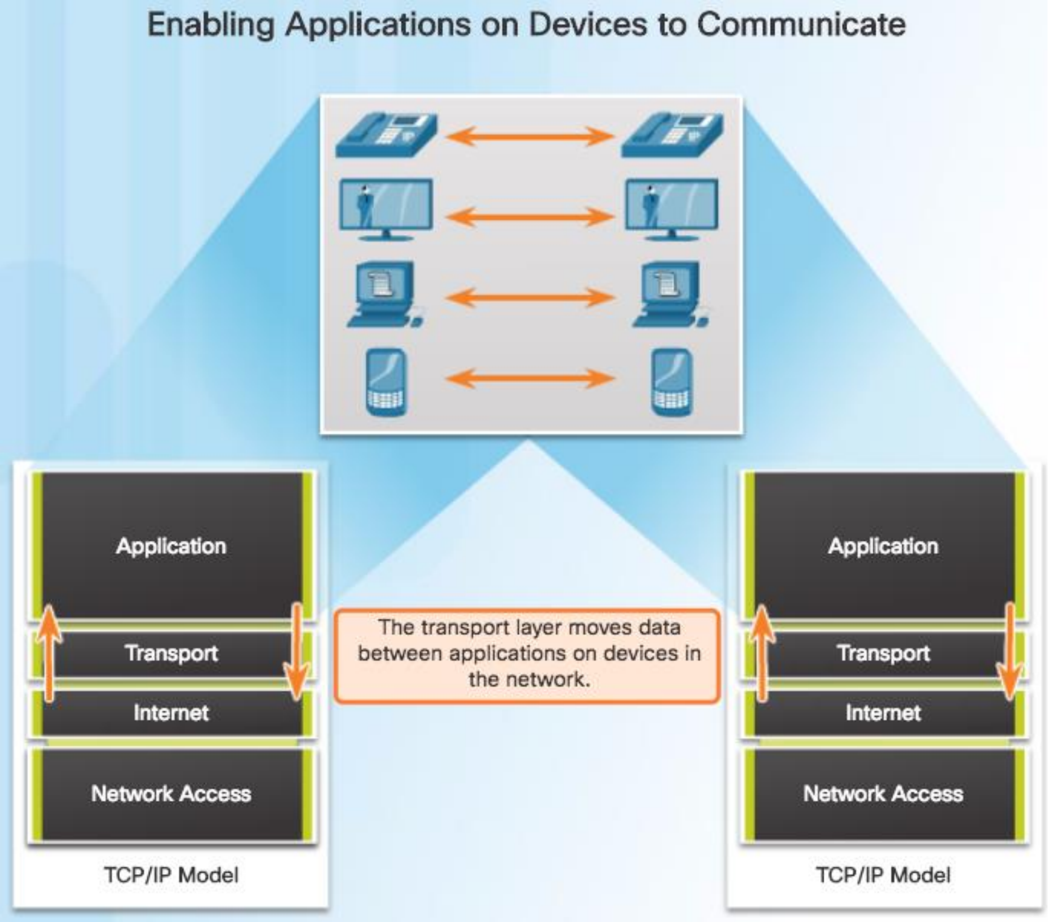
Chapter 9 - Sections & Objectives

- 9.1 Transport Layer Protocols
  - Explain how transport layer protocols and services support communications across data networks.
    - Explain the purpose of the transport layer in managing the transportation of data in end-to-end communication.
    - Explain characteristics of the TCP and UDP protocols, including port numbers and their uses.
- 9.2 TCP and UDP
  - Compare the operations of transport layer protocols in supporting end-to-end communication.
    - Explain how TCP session establishment and termination processes facilitate reliable communication.
    - Explain how TCP protocol data units are transmitted and acknowledged to guarantee delivery.
    - Describe 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.

9.1 Transport Layer Protocols

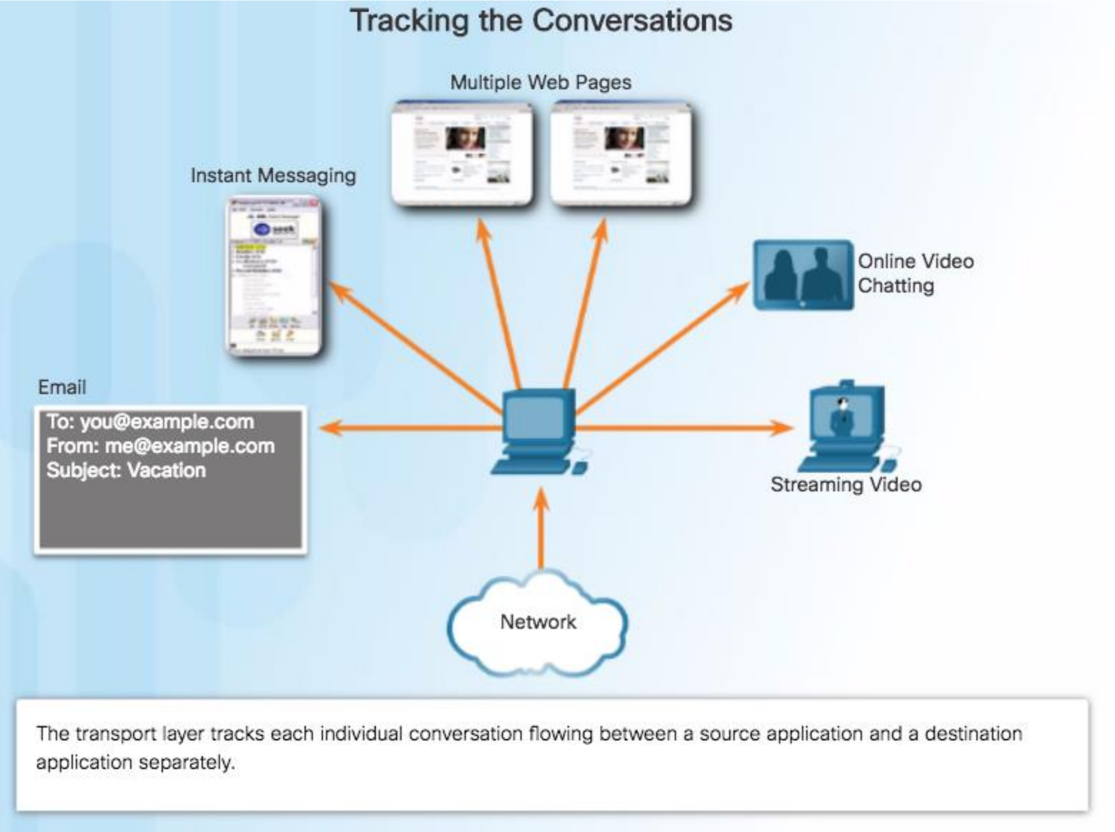
Role of the Transport Layer

- 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.



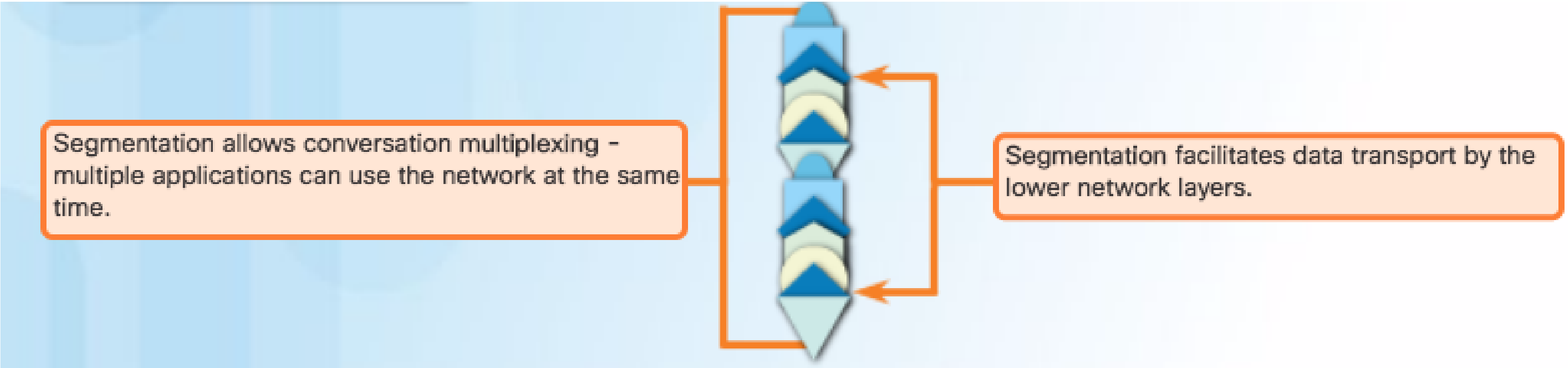
Transport Layer Responsibilities

- **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.



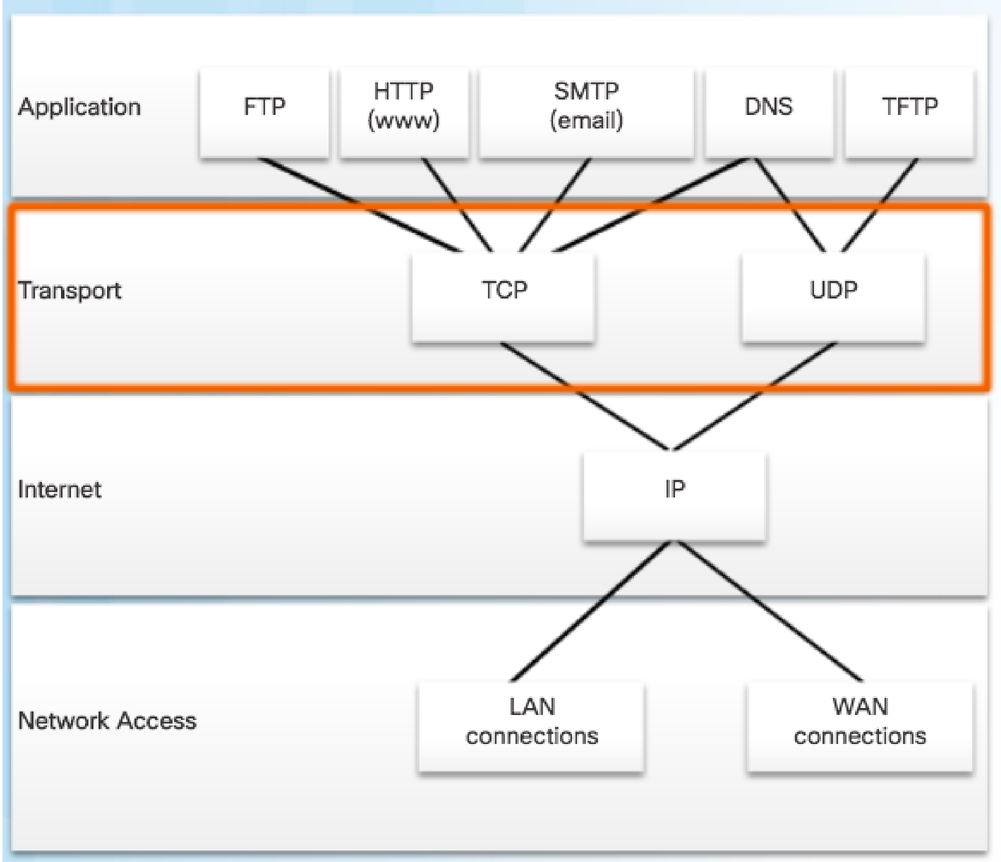
Conversation Multiplexing

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



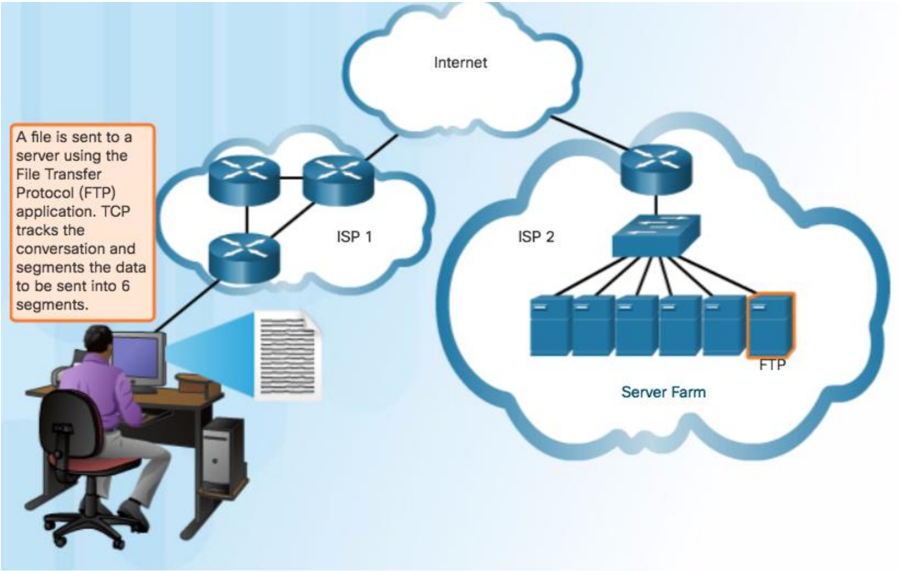
Transport Layer Reliability

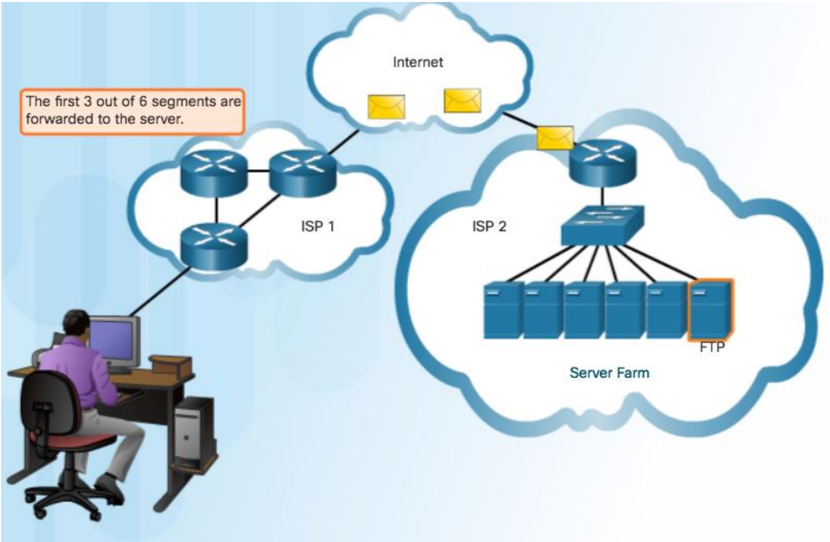
- TCP/IP provides two transport layer protocols:
  - Transmission Control Protocol (TCP)
    - Considered reliable which ensures that all of the data arrives at the destination.
    - Additional fields needed in header which increases size and delay.
  - User Datagram Protocol (UDP)
    - Does not provide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



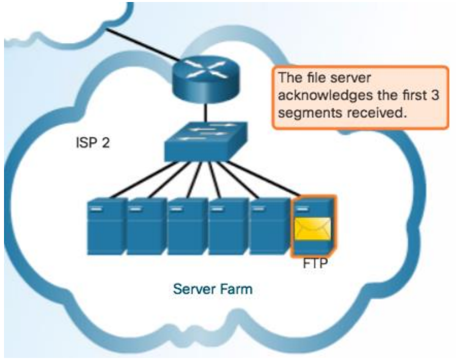
Transportation of Data
TCP

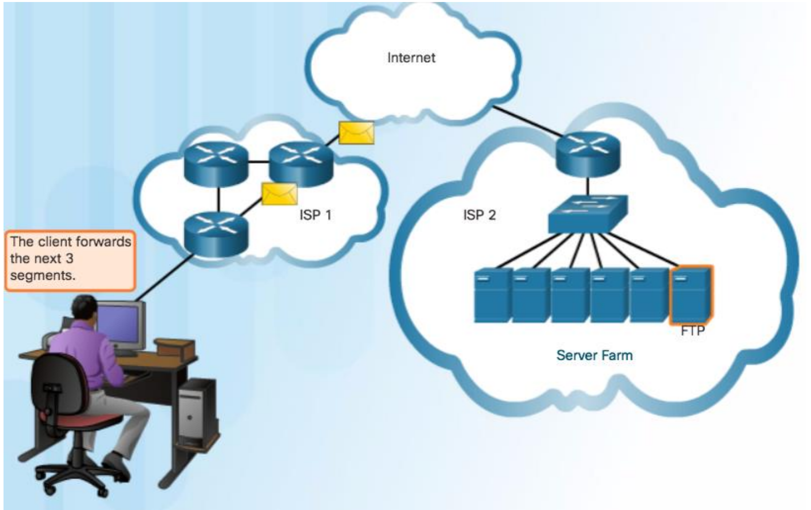
- TCP transport is similar to sending tracked packages. If a shipping order is broken up into several packages, a customer can check online to see the order of the delivery.

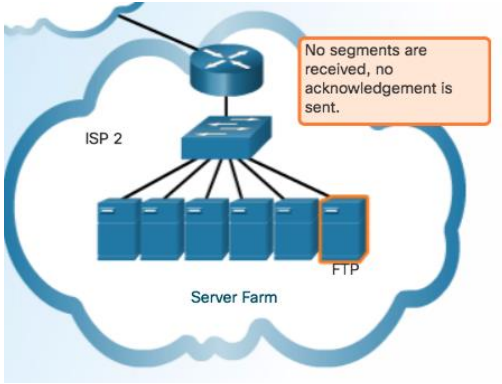




Transportation of Data
TCP (Cont.)

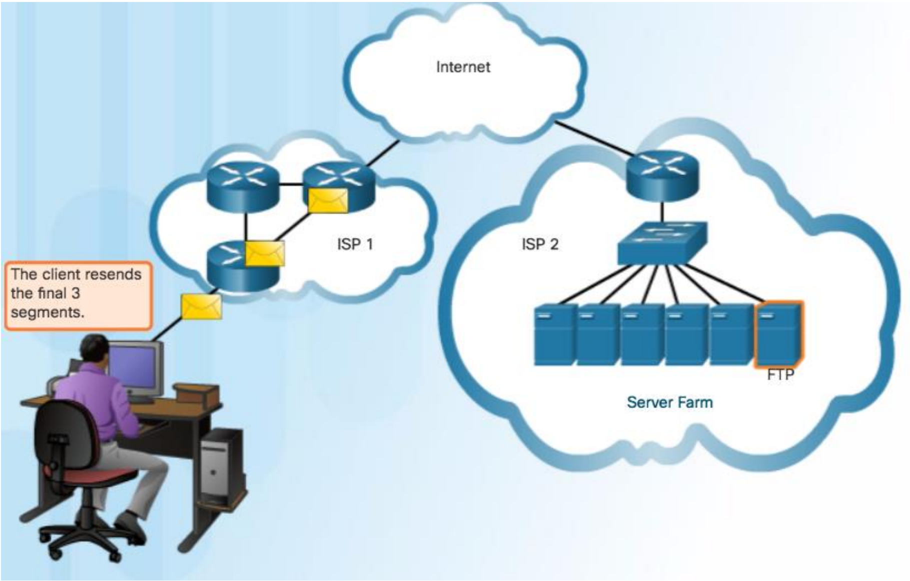






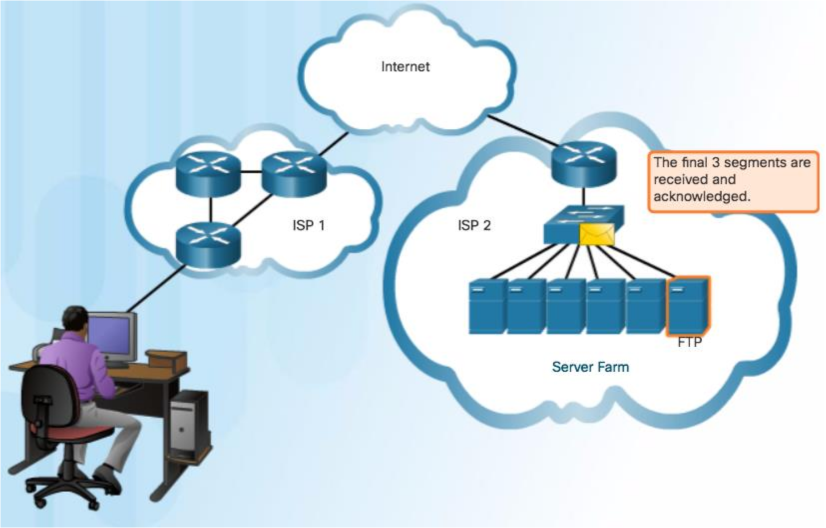
Transportation of Data

TCP (Cont.)

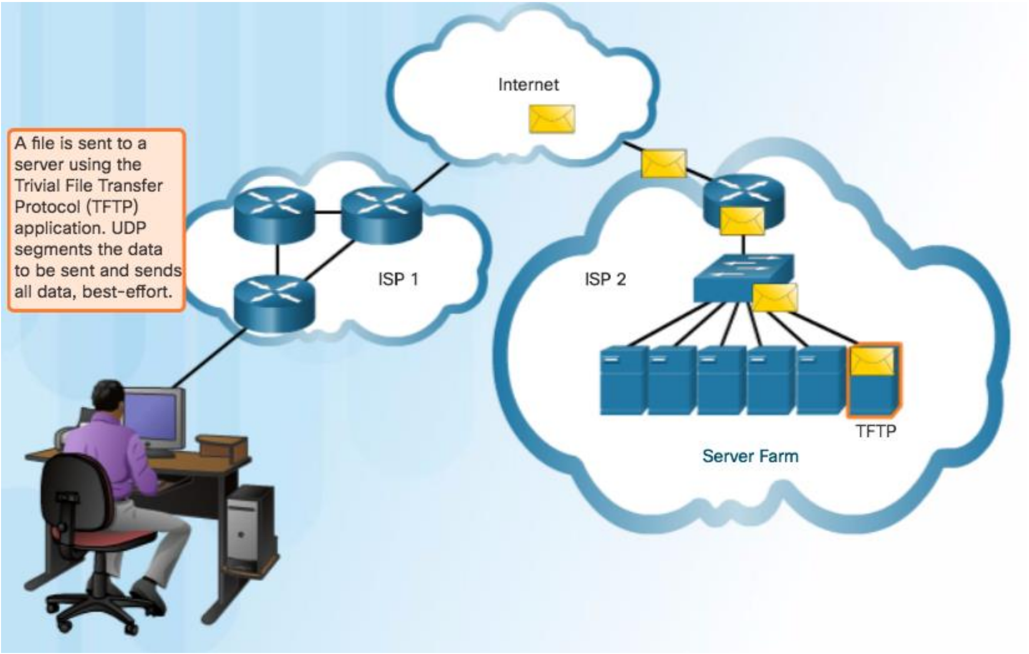


TCP Three Responsibilities:

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

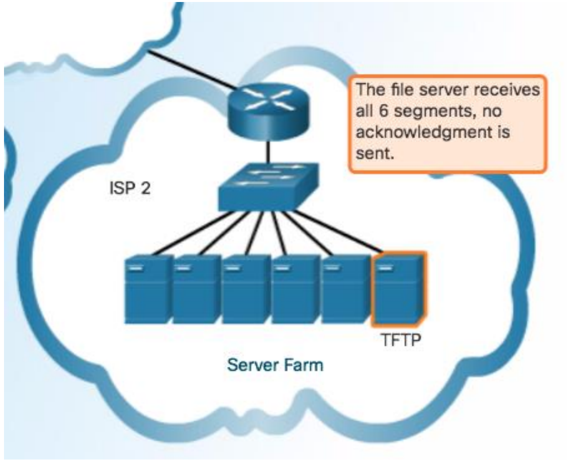


UDP



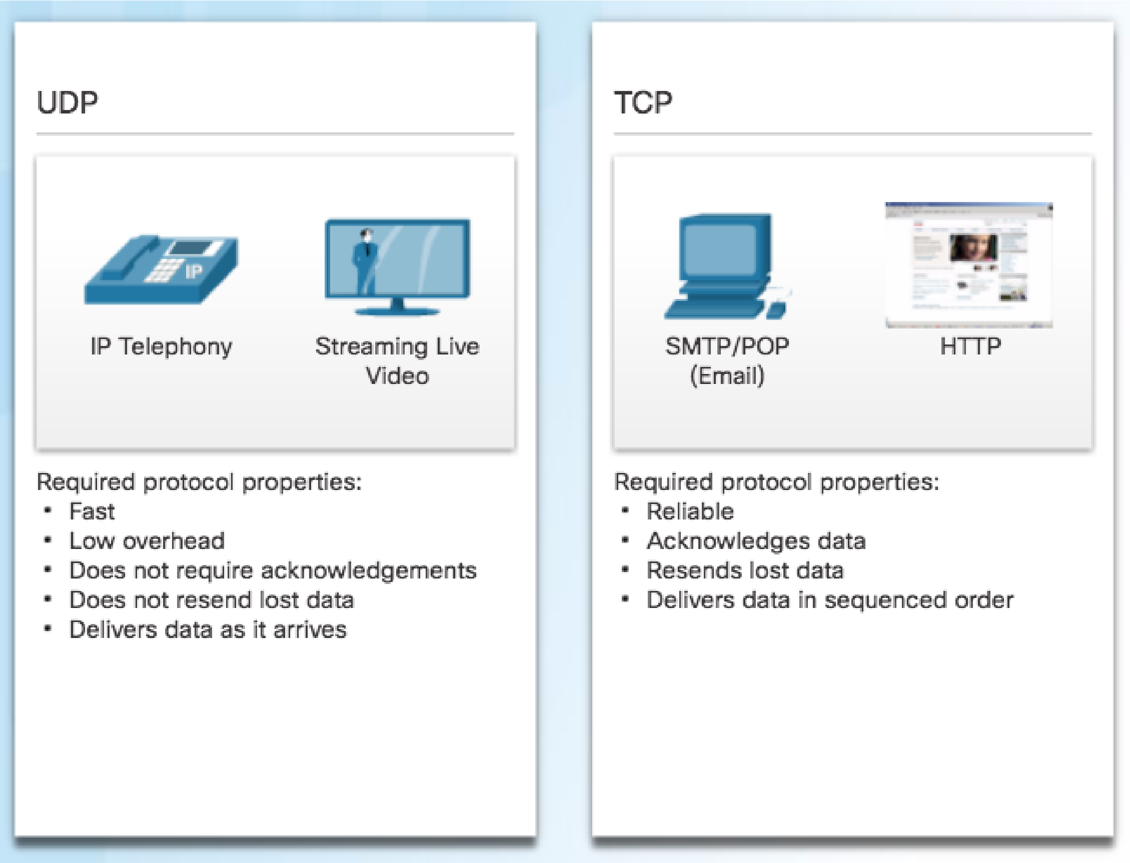
Use UDP for less overhead and to reduce possible delays.

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



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.



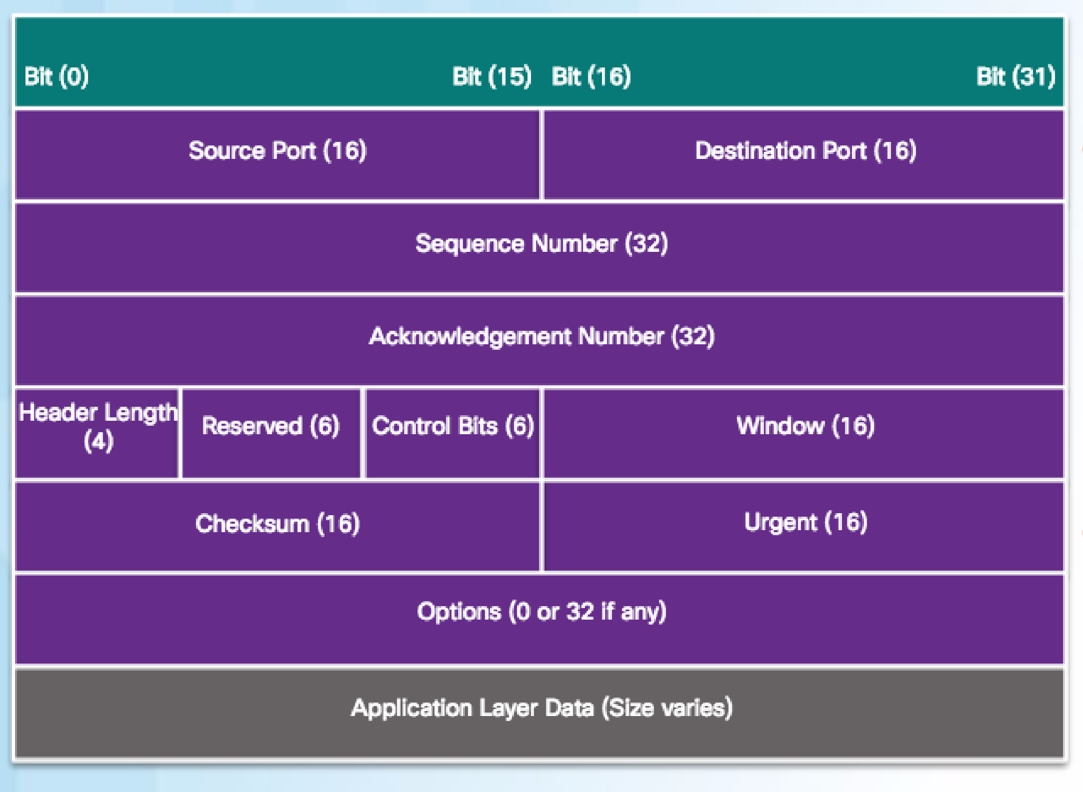
TCP Features

- 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

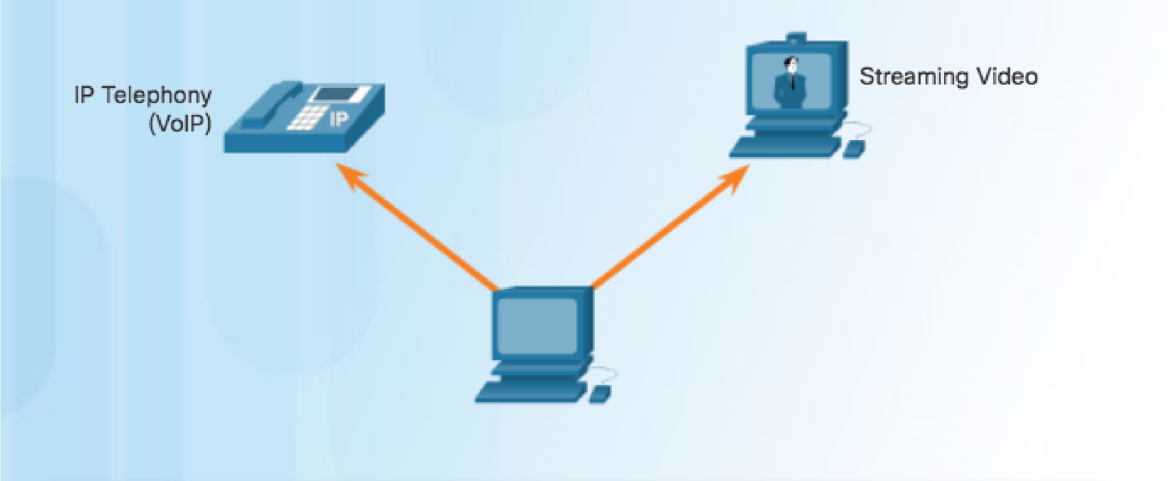
TCP Header

- 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



UDP Features



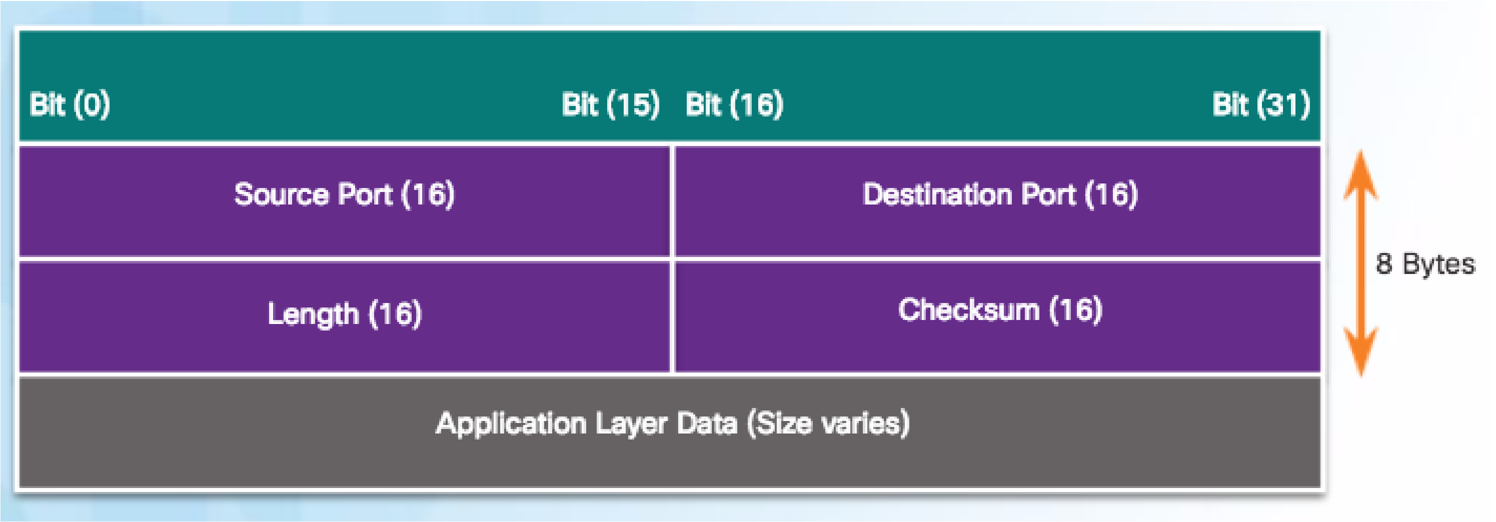
Features of UDP

- 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

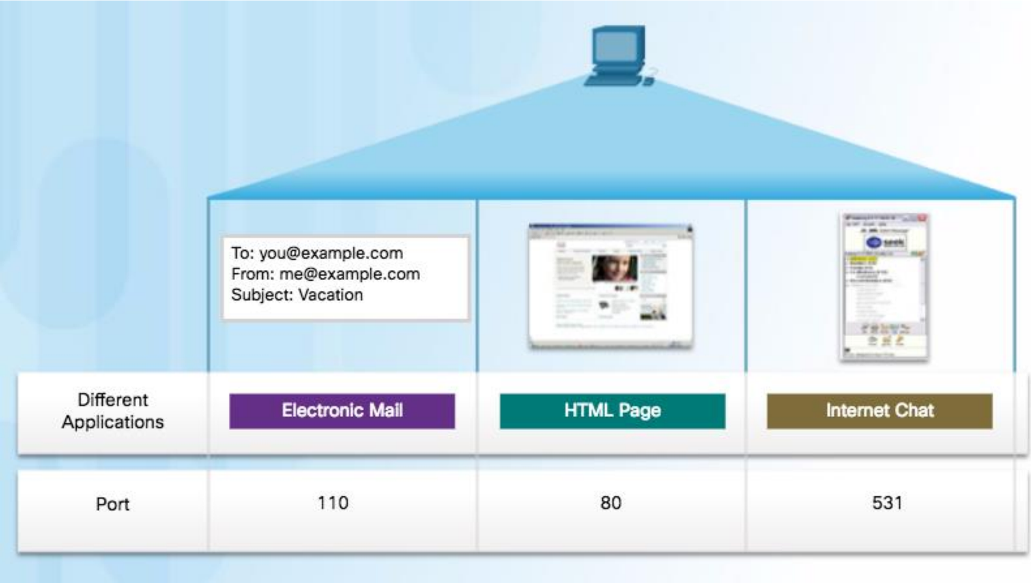
UDP Header

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



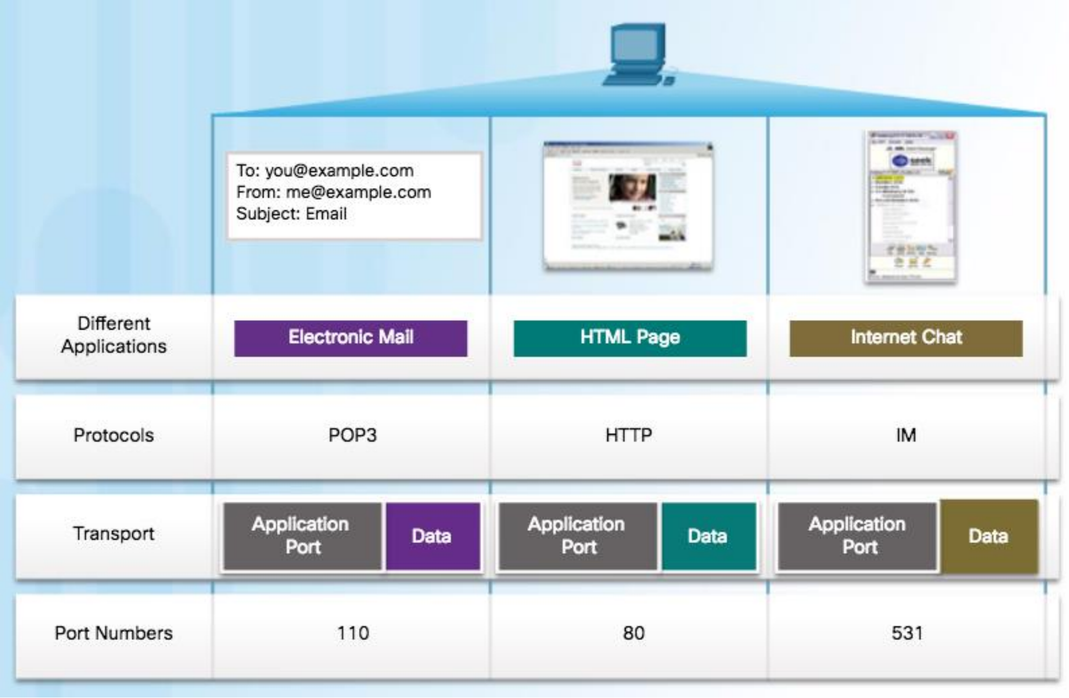
Multiple Separate Communications

- 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



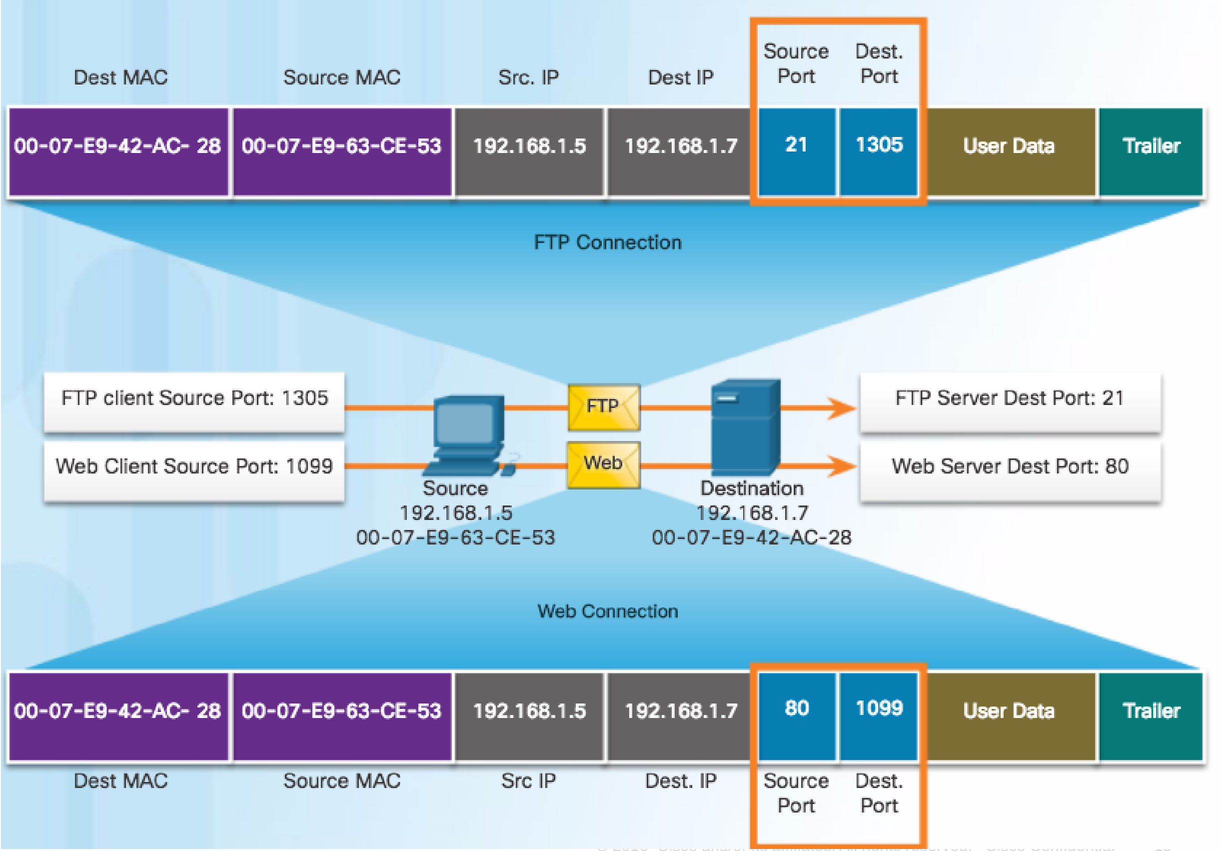
Port Numbers

- 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



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



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

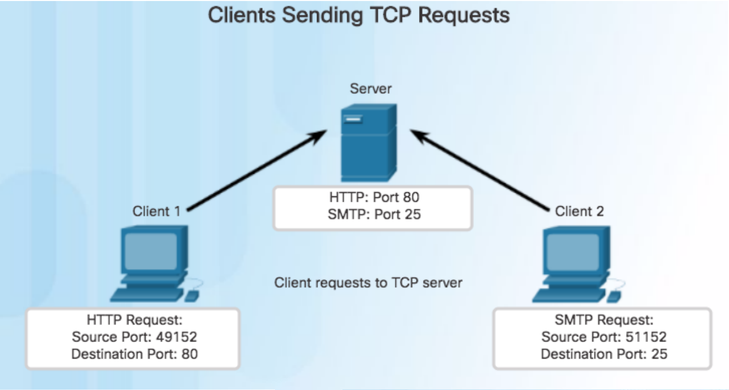
The netstat Command

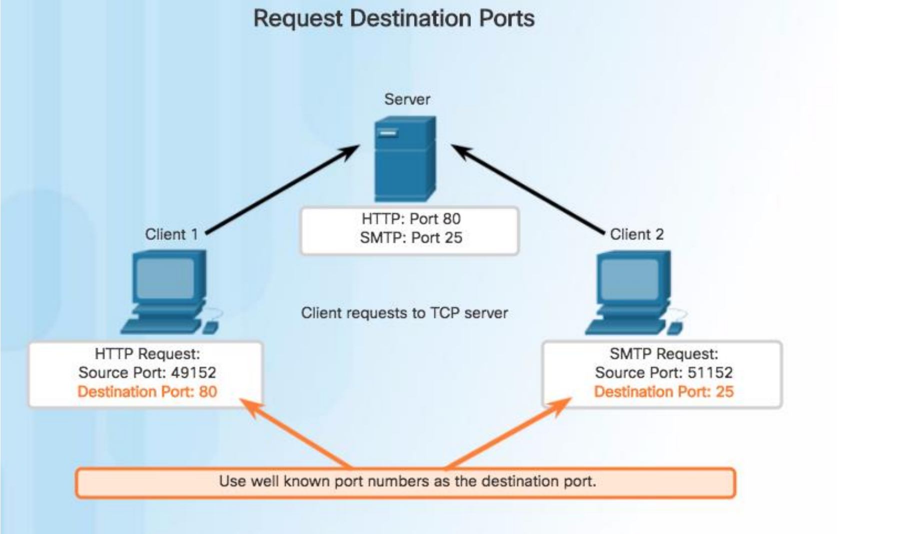
- Network utility that can be used to verify connections
- By default, will attempt to resolve IP addresses to domain names and port numbers to well-known applications
- -n option used to display IPs and ports in numerical form

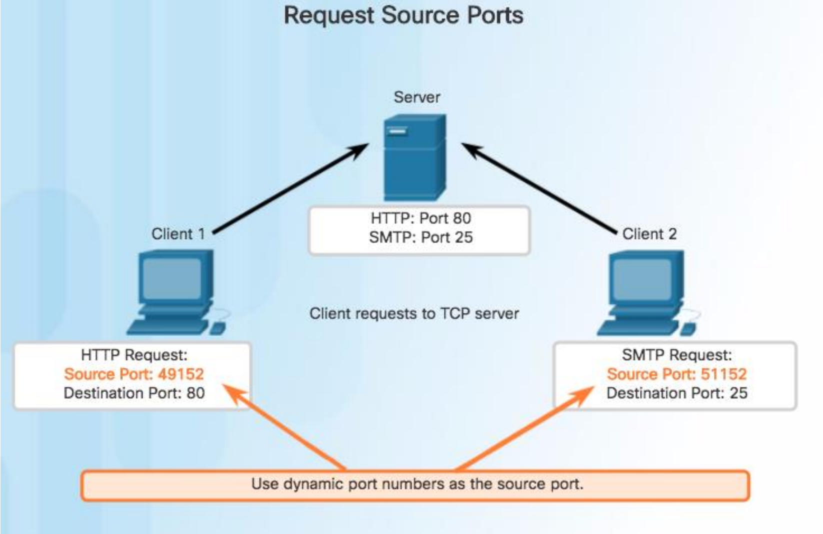
```
C:\> netstat

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:\>
```

9.2 TCP and UDP

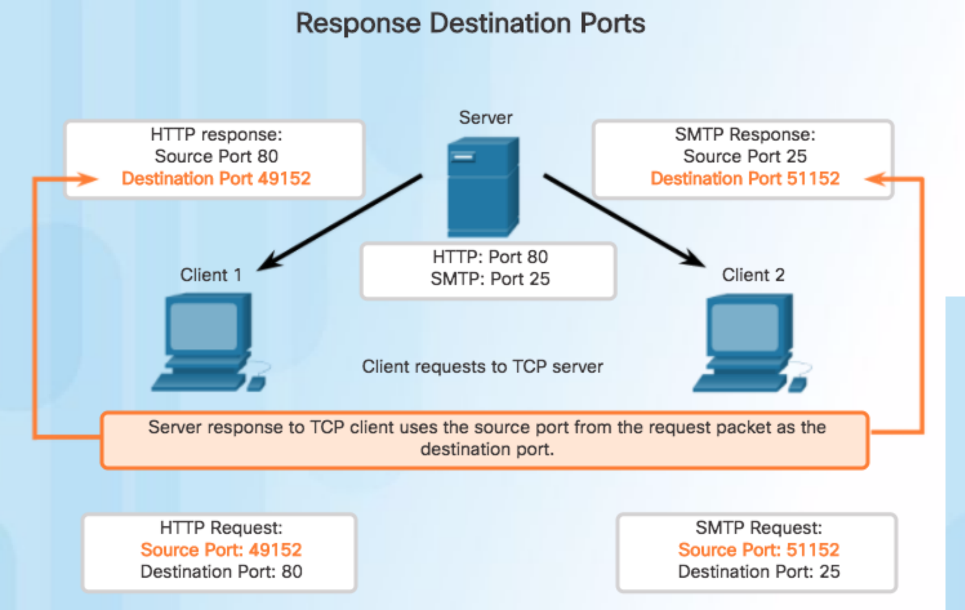
TCP Server Process

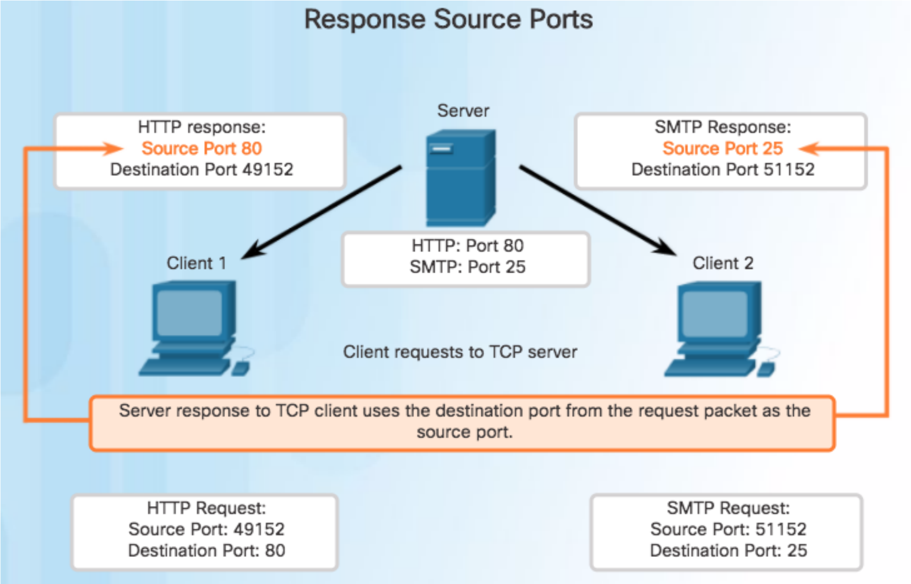




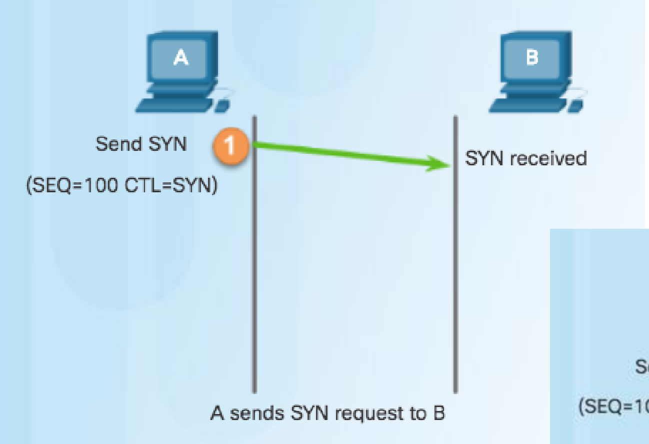


TCP Server Process (Cont.)

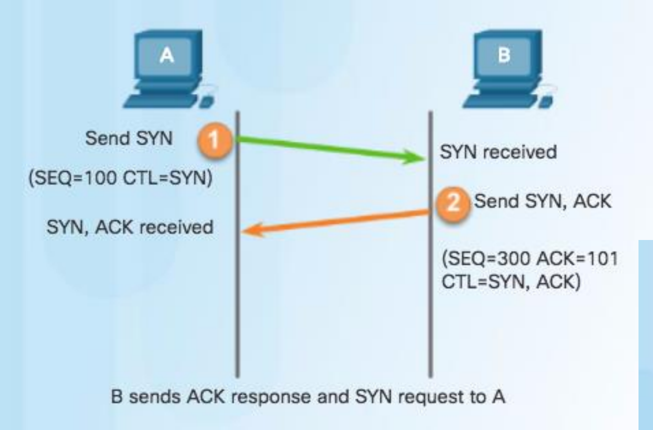




TCP Connection Establishment

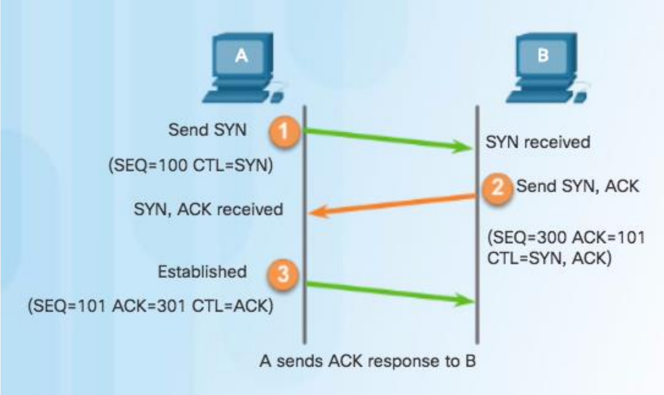


- Step 1 – Initiating client requests a session with server.

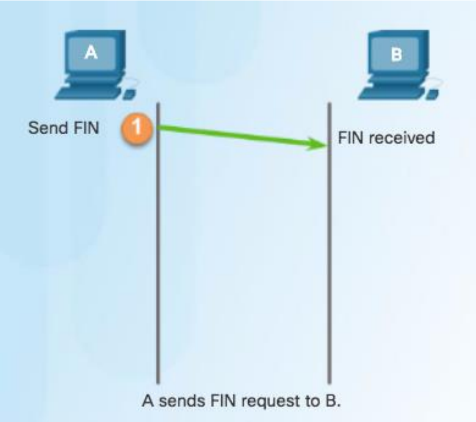


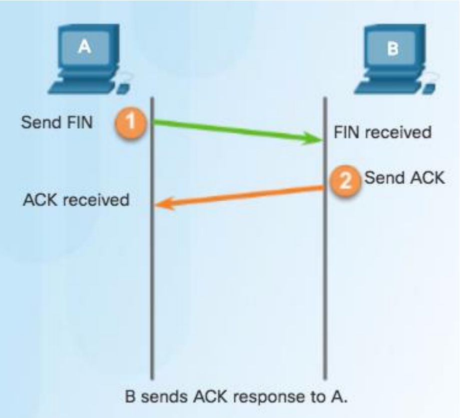
- Step 2 – Server acknowledges and requests a session with client.

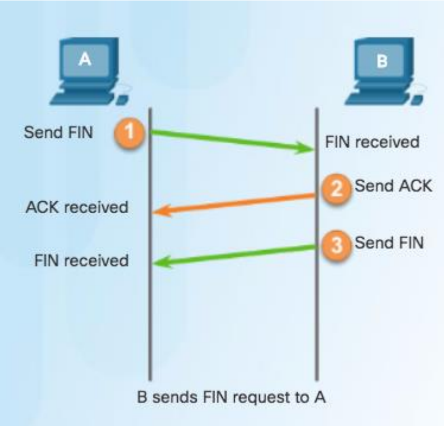
- Step 3 – Client acknowledges communication session with server.

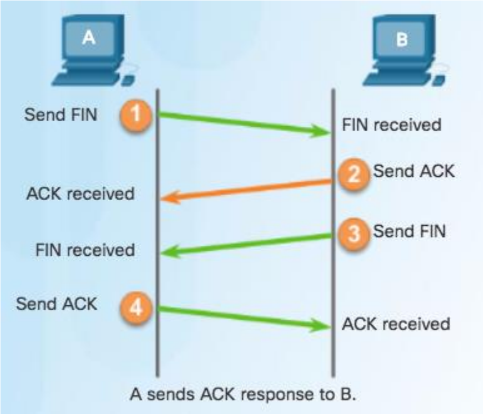


TCP Session Termination





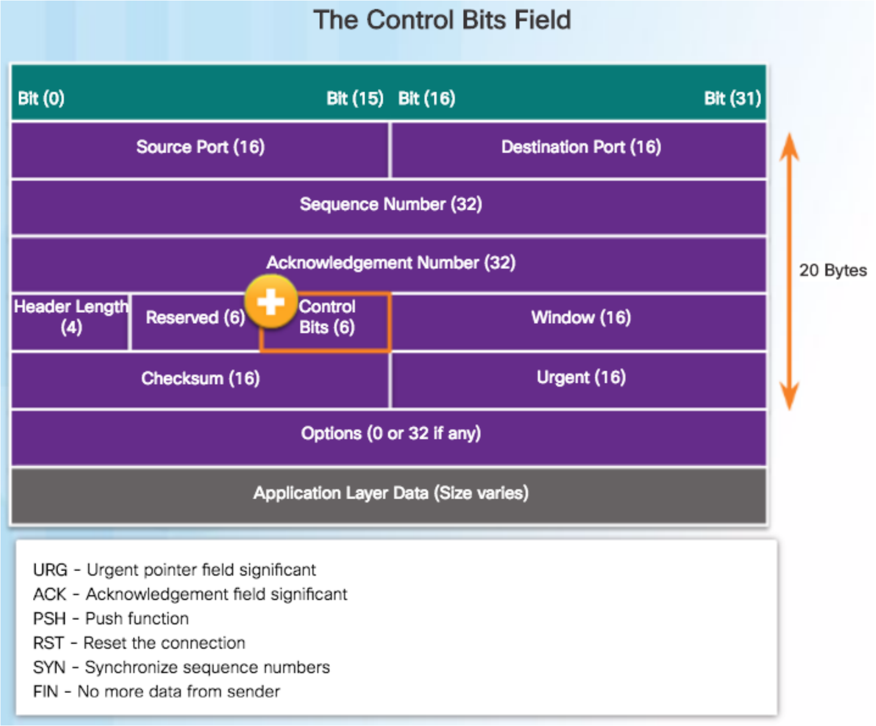




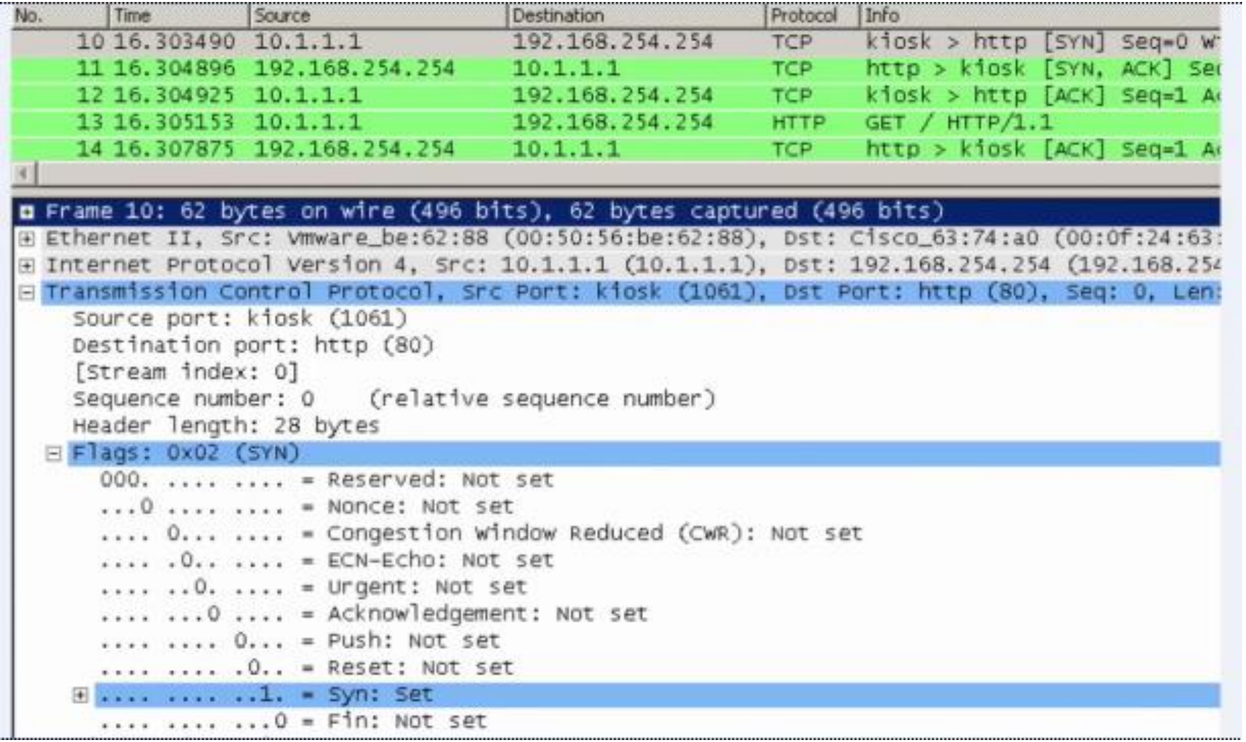
- To close a connection, the Finish (FIN) control flag must be set in the segment header.
- To end each one-way TCP session, a two-way handshake, consisting of a FIN segment and an Acknowledgment (ACK) segment, is used.
- To terminate a single conversation supported by TCP, four exchanges are needed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



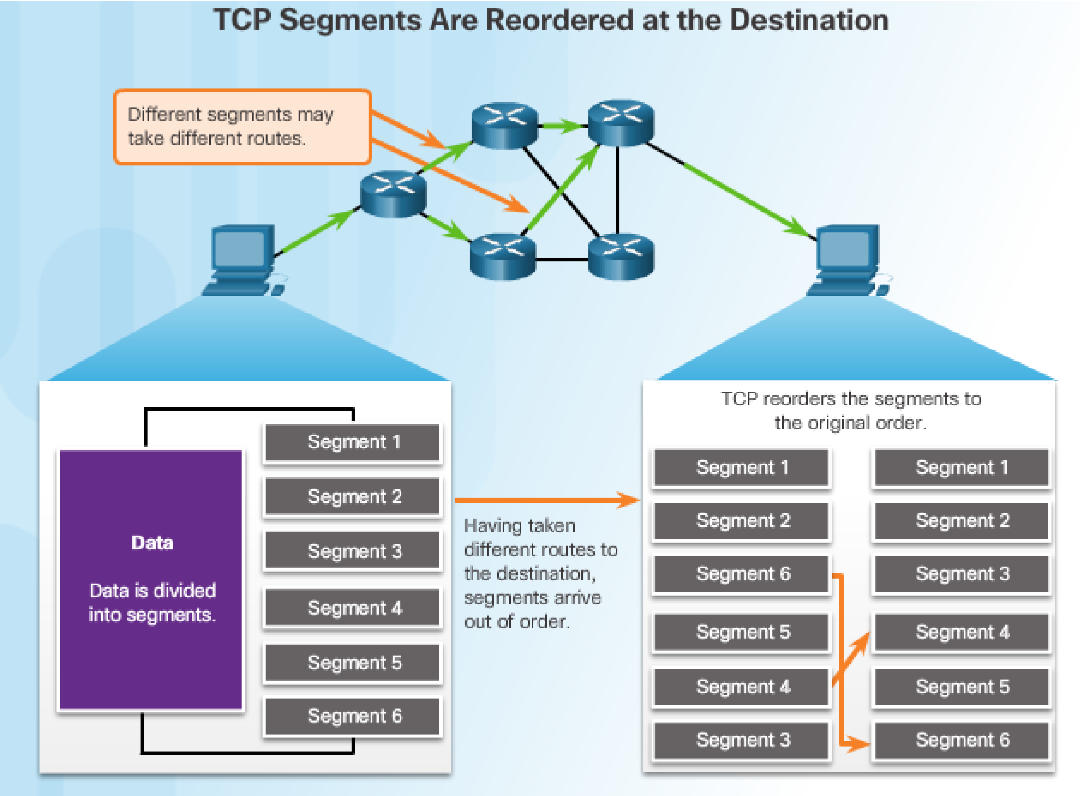
Video Demonstration - TCP 3-Way Handshake



SYN
SYN, ACK
ACK

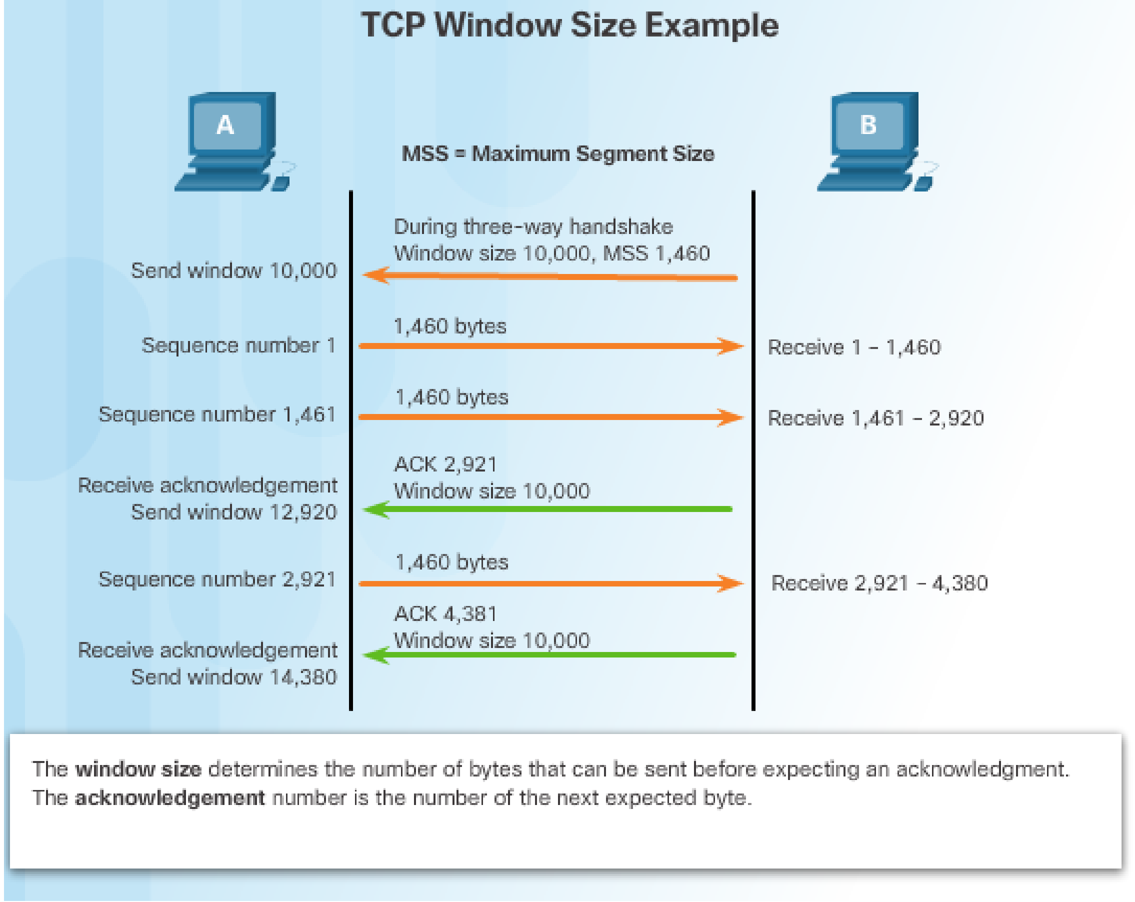
TCP Reliability – Ordered Delivery

- 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 Flow Control – Window Size and Acknowledgments

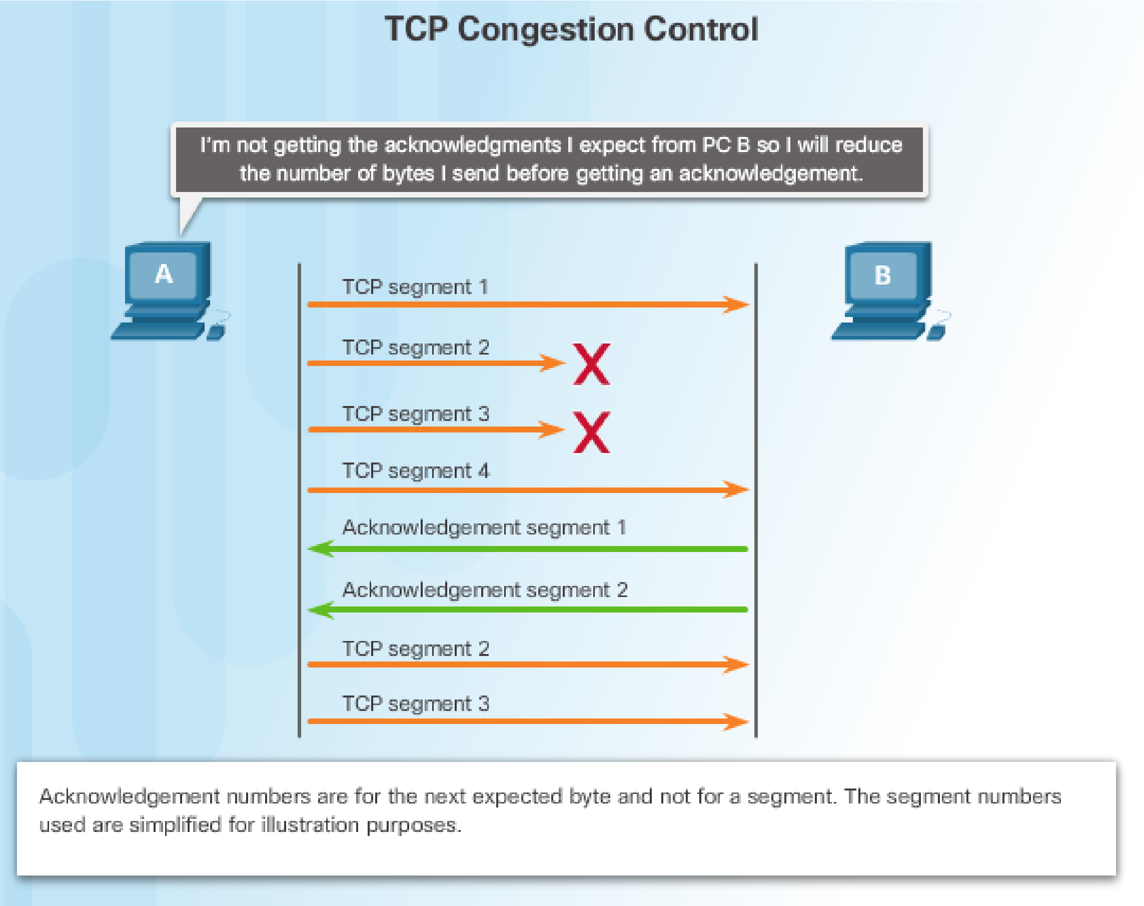
- In the figure, the source is transmitting 1,460 bytes of data within each segment.
- Window size agreed on during 3-way handshake.
- Typically, PC B will not wait for 10,000 bytes before sending an acknowledgment.
- PC A can adjust its send window as it receives acknowledgments from PC B.



Reliability and Flow Control

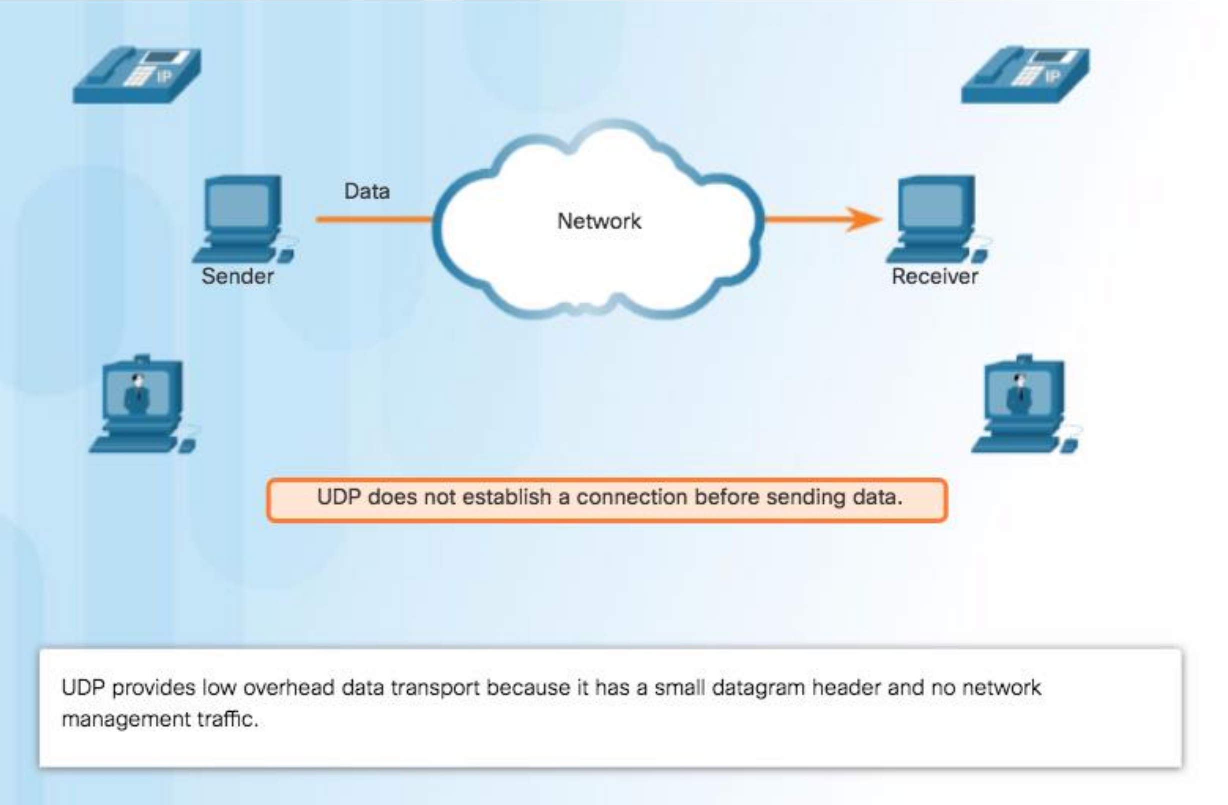
TCP Flow Control – Congestion Avoidance

- 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



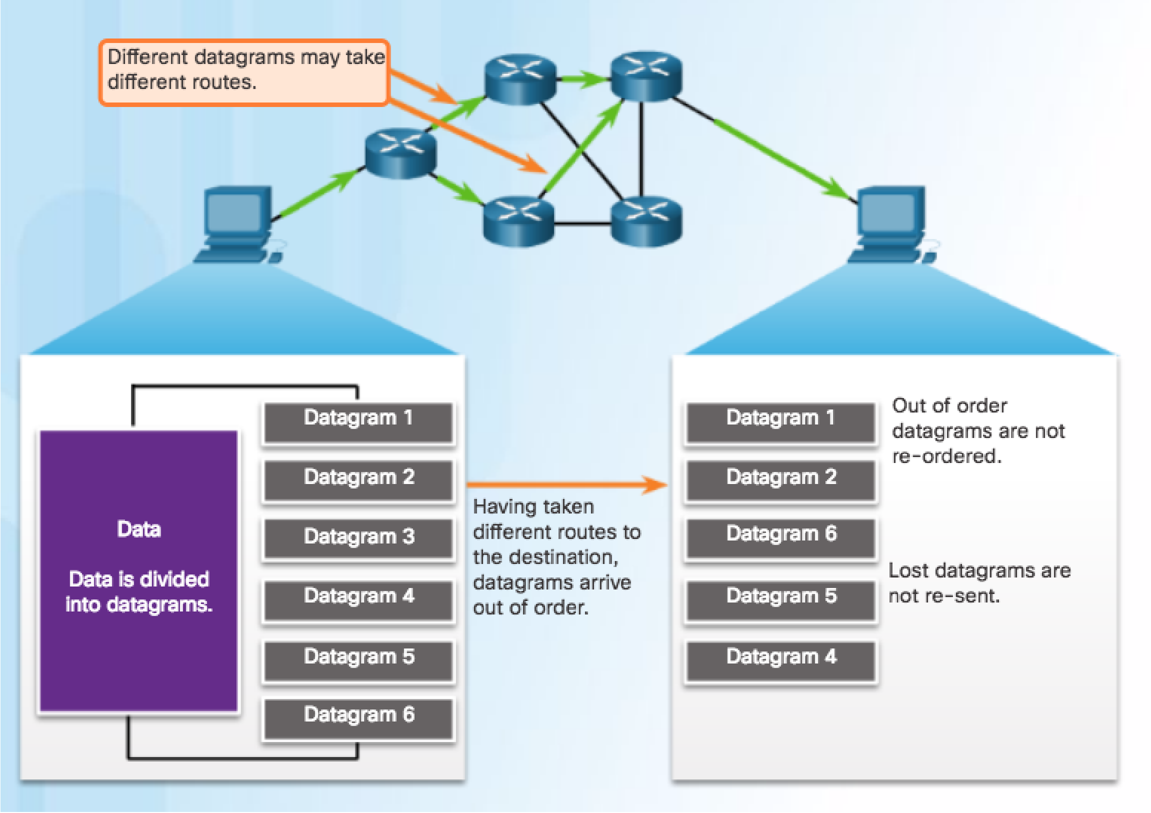
UDP Low Overhead versus Reliability

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



UDP Datagram Reassembly

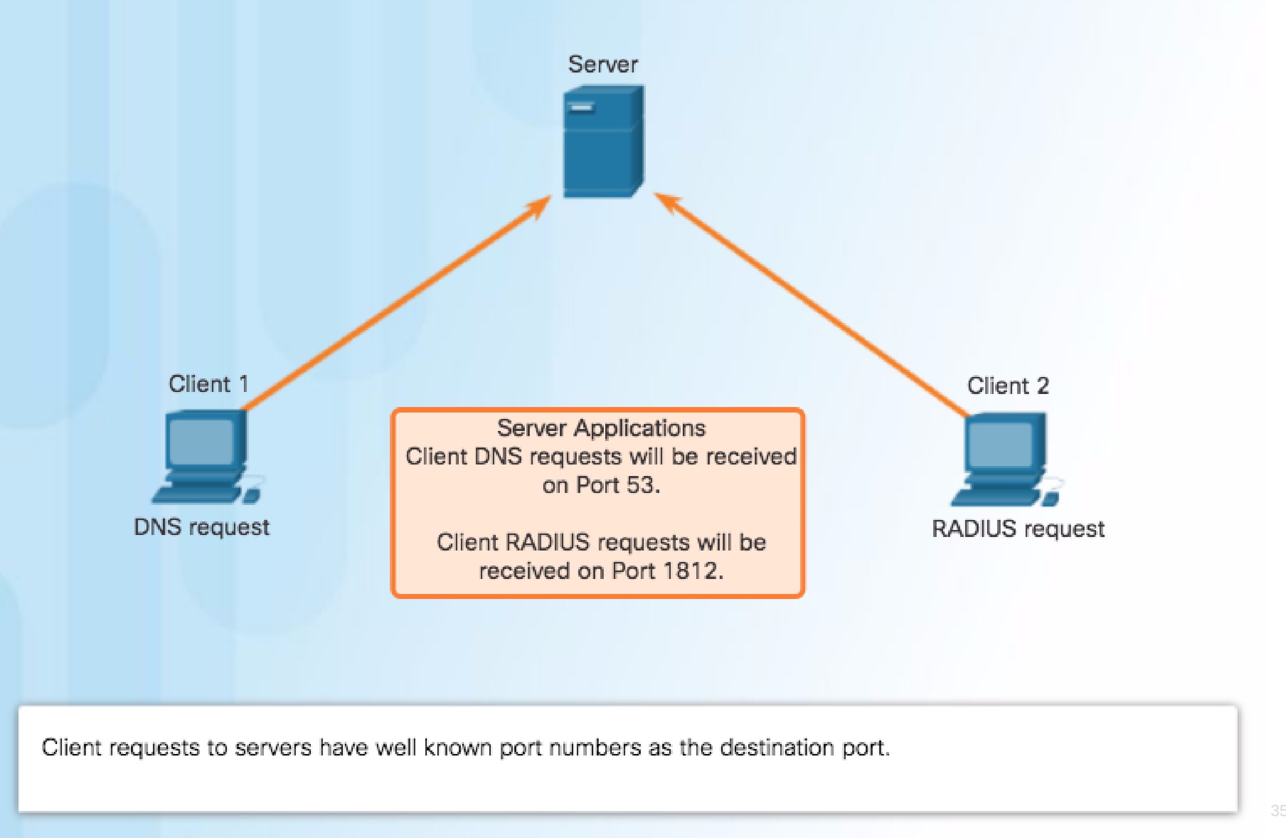
- UDP reassembles data in order received and forwards to application
- Application must identify the proper sequence



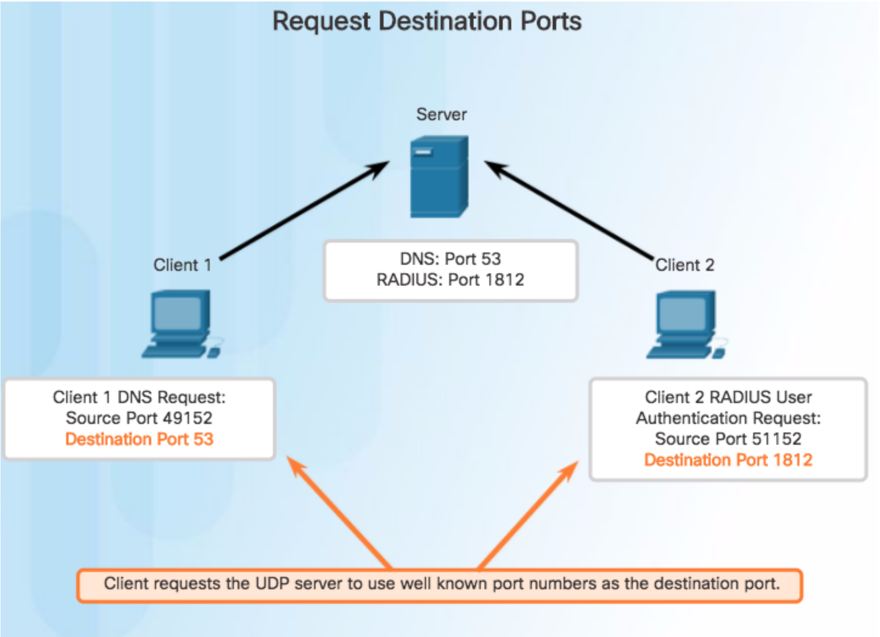
UDP: Connectionless and Unreliable

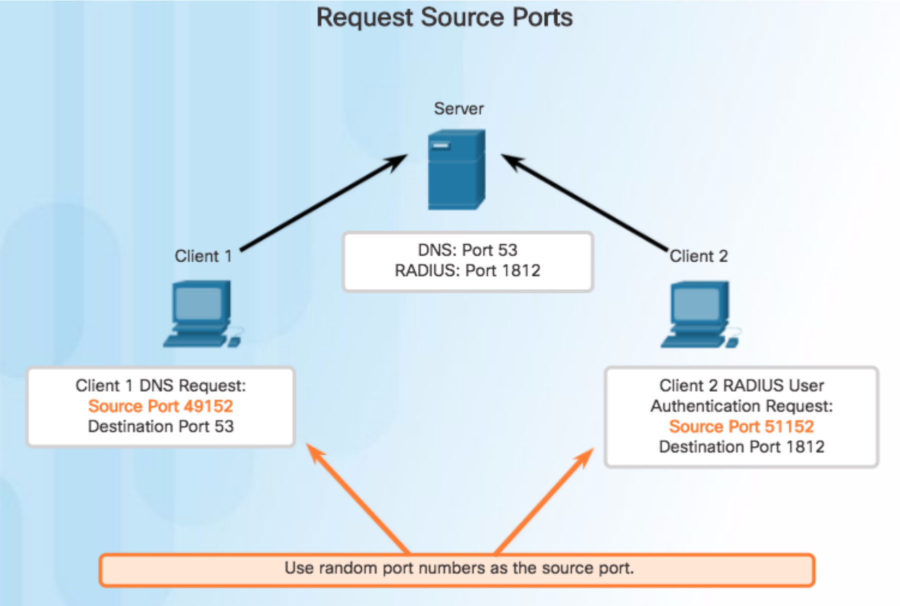
UDP Server Processes and Requests

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



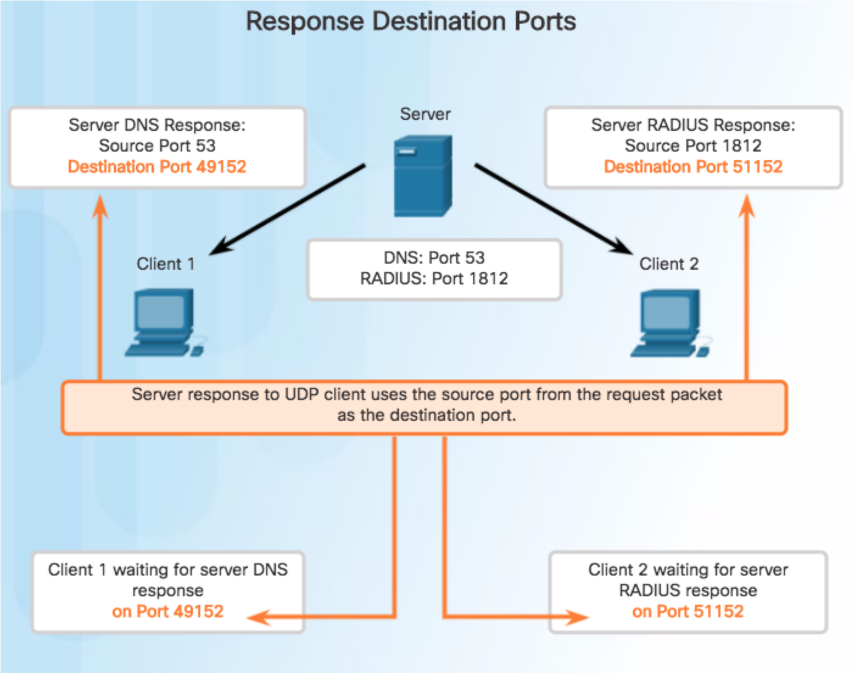
UDP Client Processes

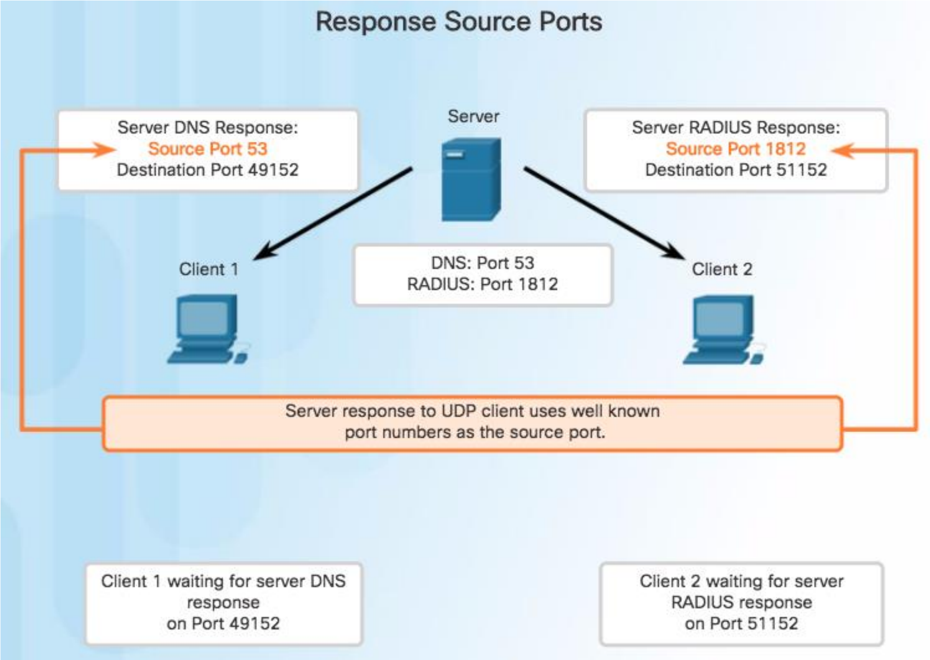




Clients Sending UDP Requests

UDP Client Processes (Cont.)

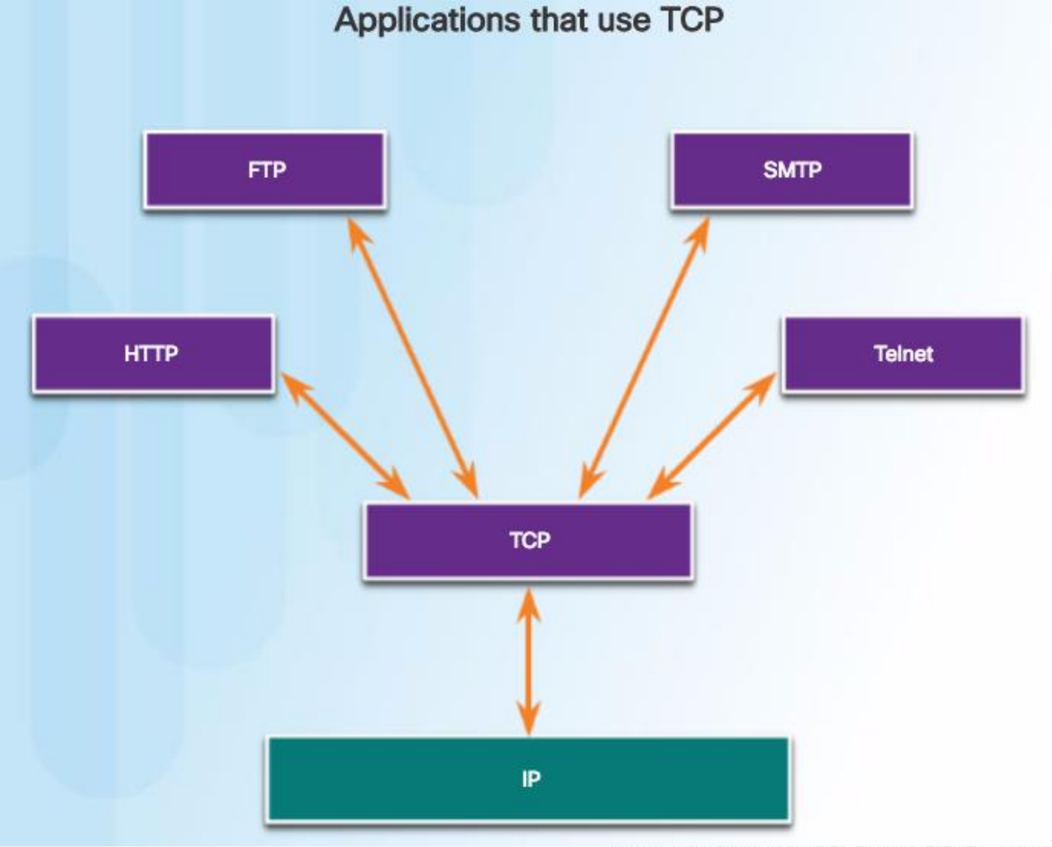




Clients Sending UDP Requests

Applications that use TCP

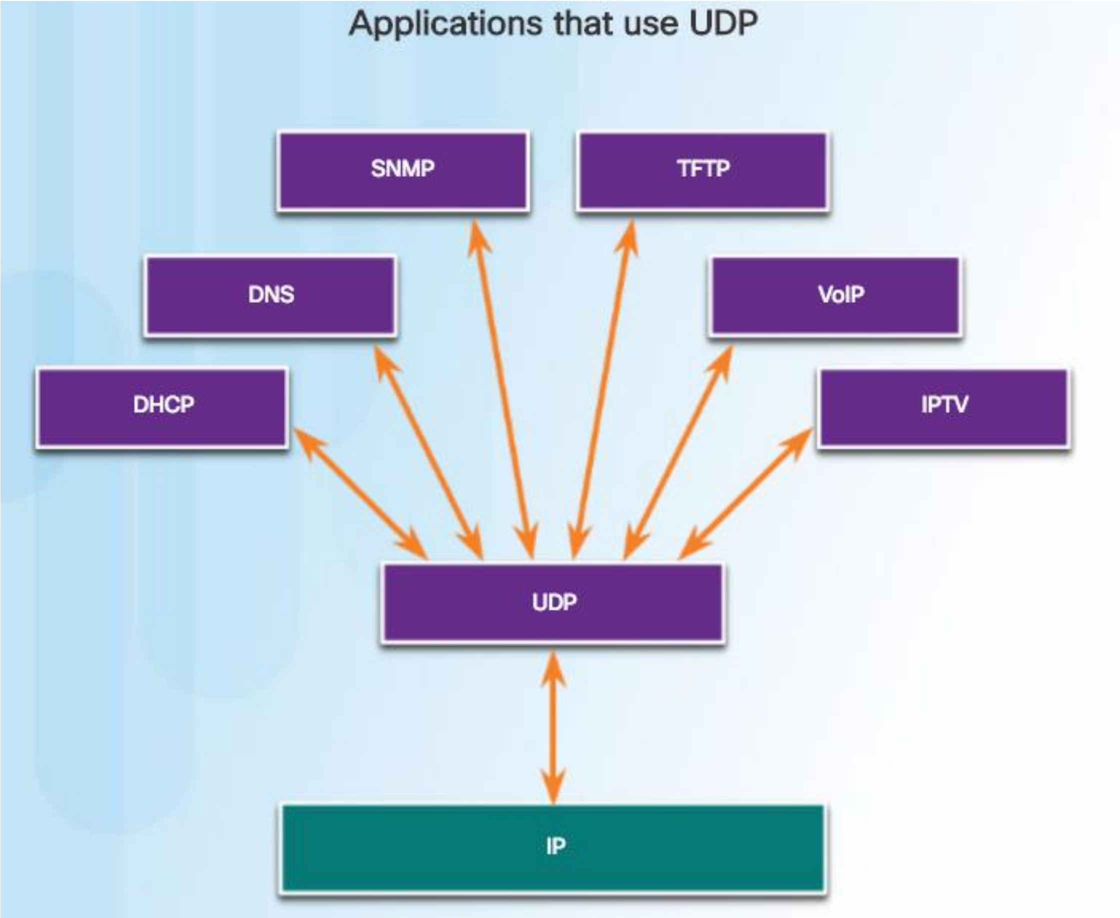
TCP frees applications from having to manage reliability



Applications that use UDP

Three types of applications best suited for UDP:

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



9.3 Chapter Summary

Conclusion

Chapter 9: Transport Layer

- Explain how transport layer protocols and services support communications across data networks.
- Compare the operations of transport layer protocols in supporting end-to-end communication.

CISCO