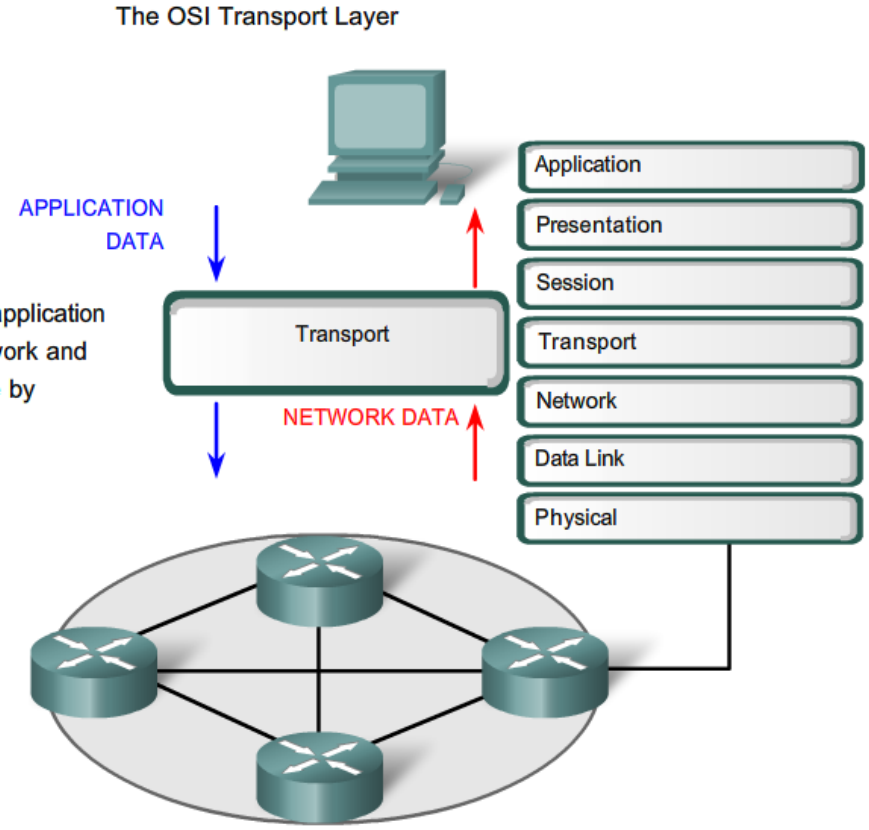
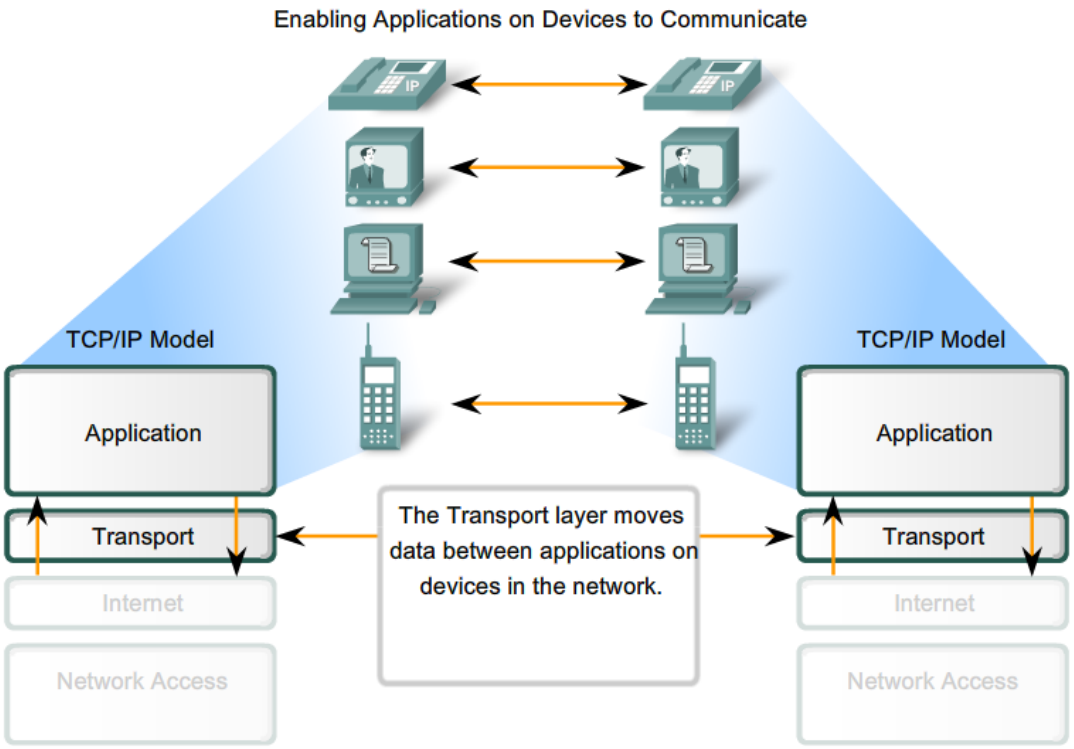
Transport Layer

Introduction

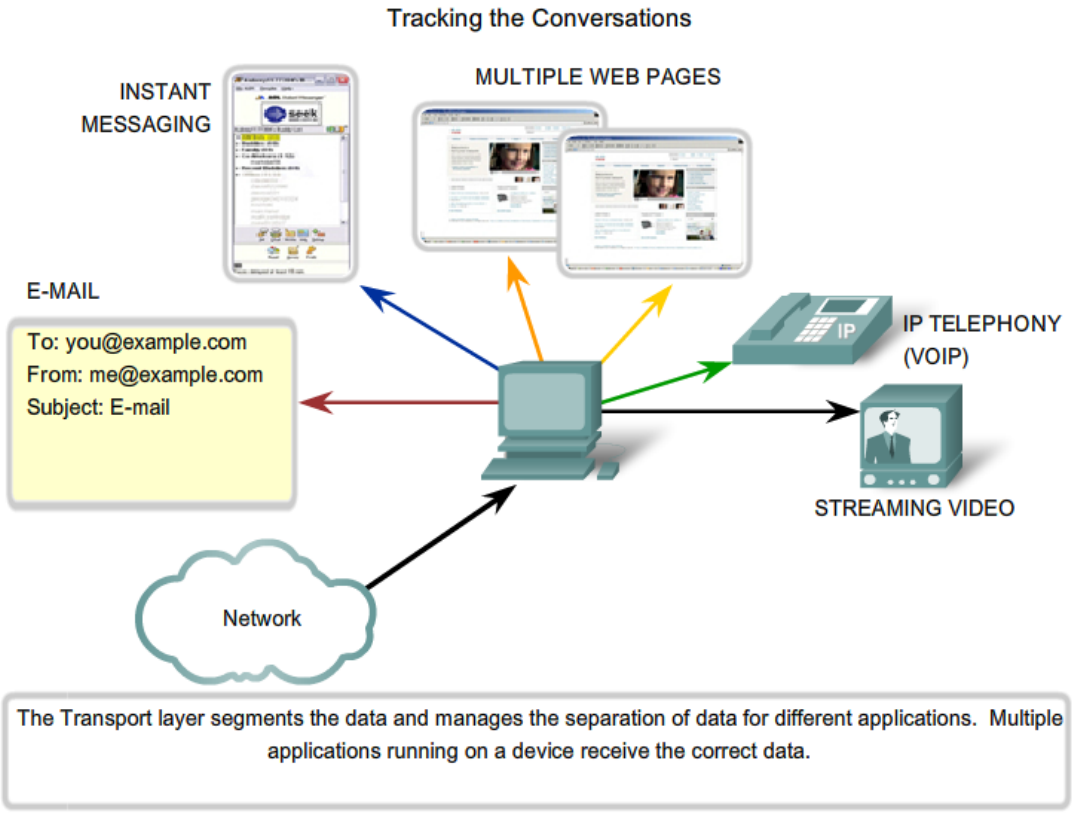
The Transport layer prepares application data for transport over the network and processes network data for use by applications.



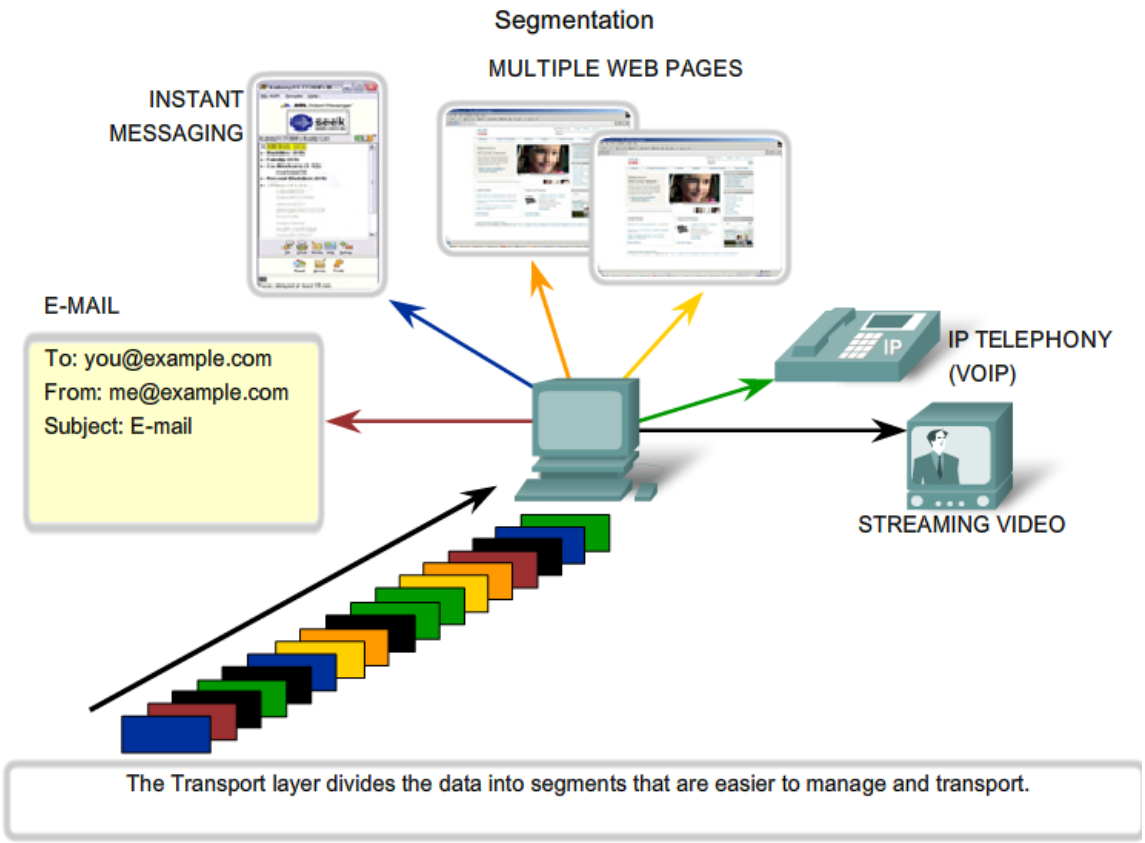
Purpose of the Transport Layer



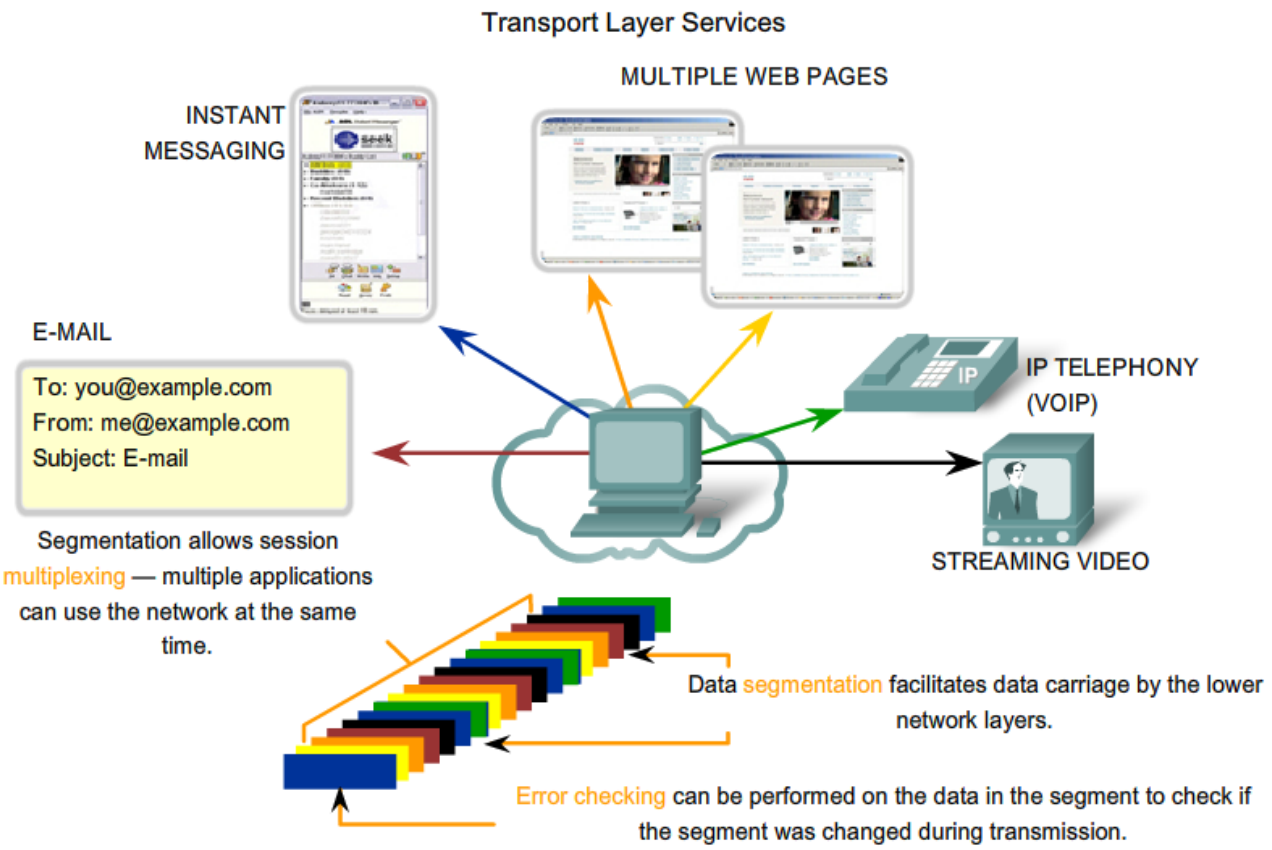
Purpose of the Transport Layer…



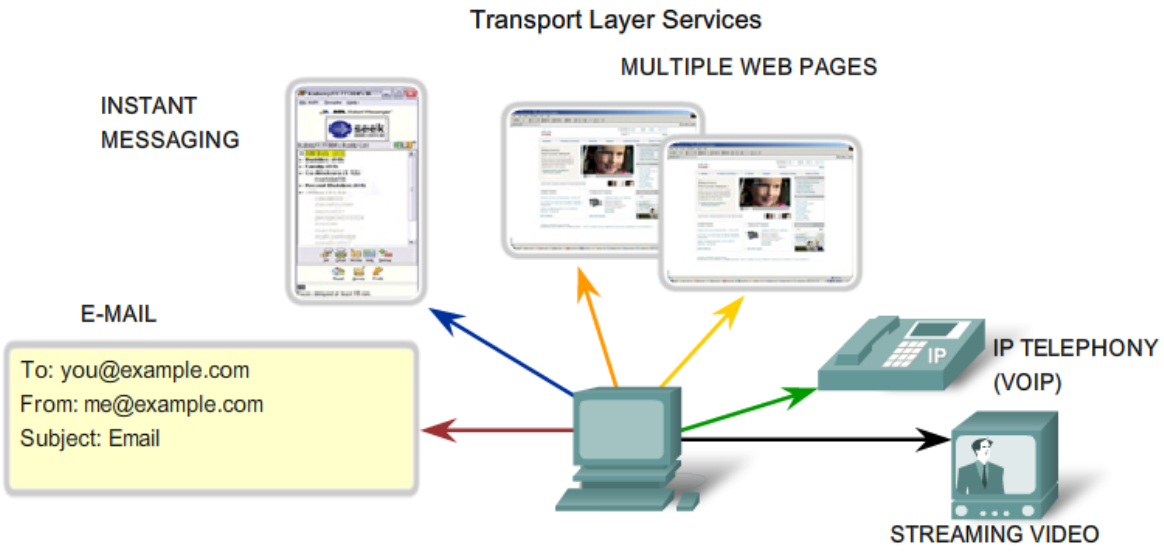
Purpose of the Transport Layer…



Controlling the Conversations



Controlling the Conversations…



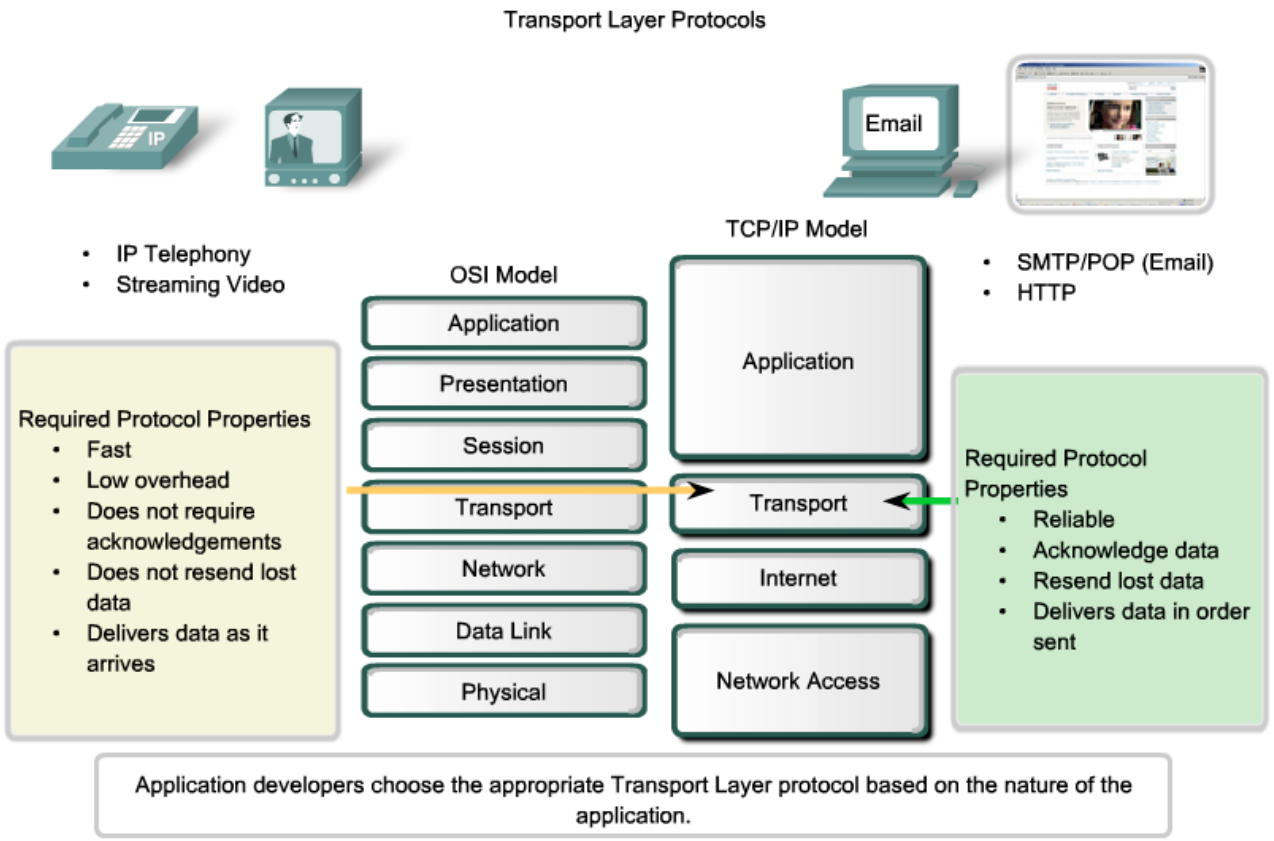
Establishing a Session ensures the application is ready to receive the data.

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

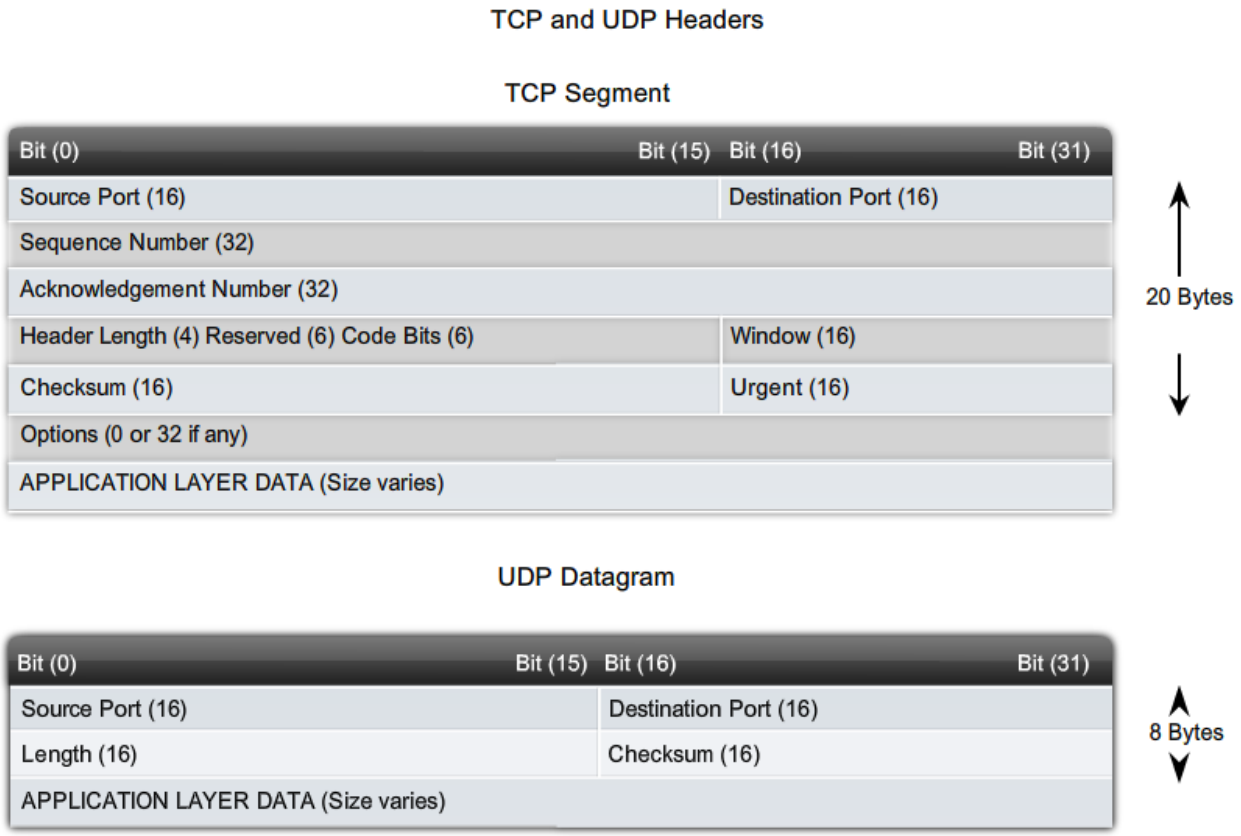
Same order delivery ensures that the segments are reassembled into the proper order.

Flow Control manages data delivery if there is congestion on the host.

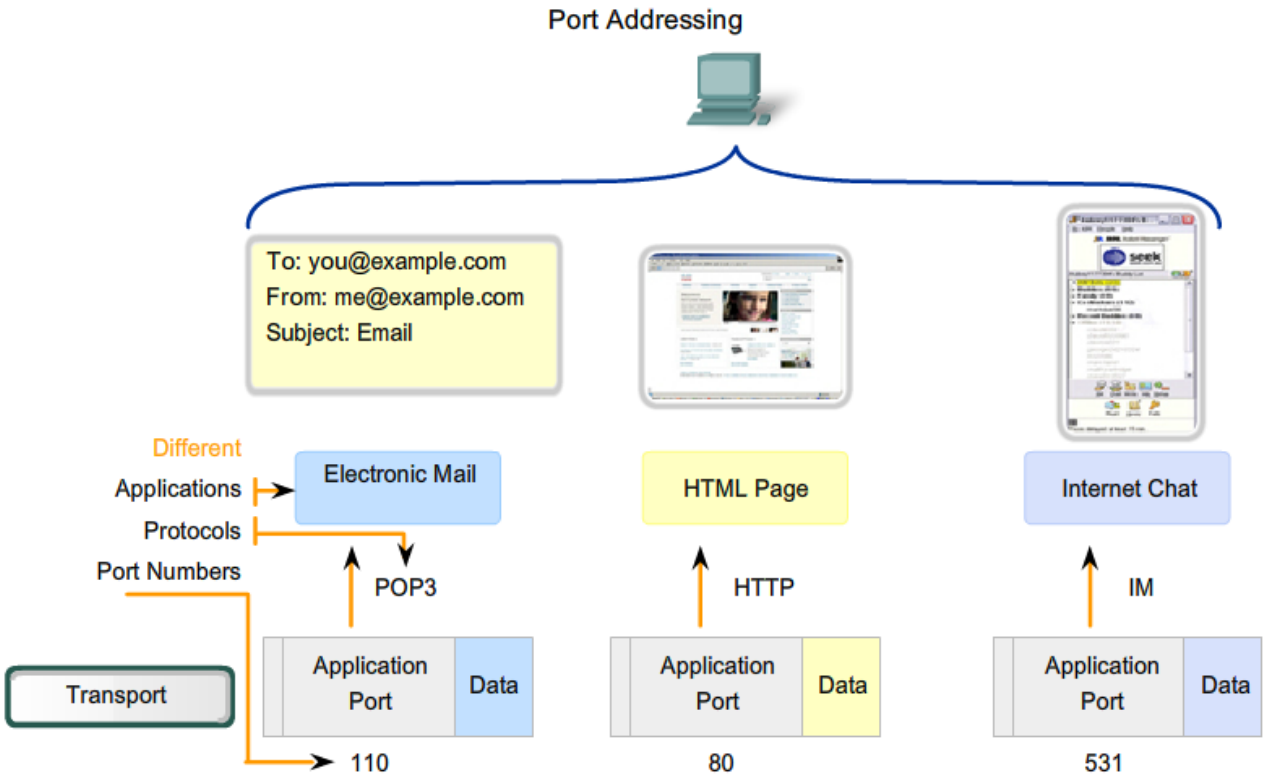
Support Reliable Communication



TCP and UDP

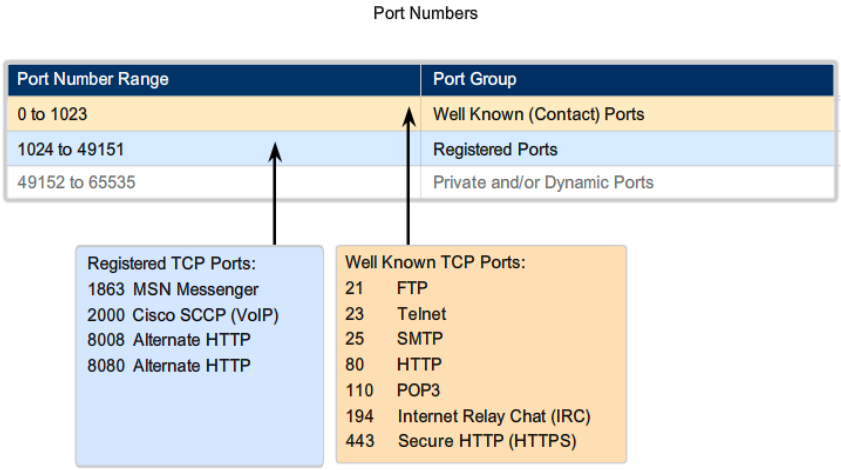

TCP and UDP Headers
TCP Segment
Bit (0)
Bit (15)
Bit (16)
Bit (31)
Source Port (16)
Destination Port (16)
Sequence Number (32)
Acknowledgement Number (32)
Header Length (4) Reserved (6) Code Bits (6)
Window (16)
Checksum (16)
Urgent (16)
Options (0 or 32 if any)
APPLICATION LAYER DATA (Size varies)
20 Bytes
UDP Datagram
Bit (0)
Bit (15)
Bit (16)
Bit (31)
Source Port (16)
Destination Port (16)
Length (16)
Checksum (16)
APPLICATION LAYER DATA (Size varies)
8 Bytes

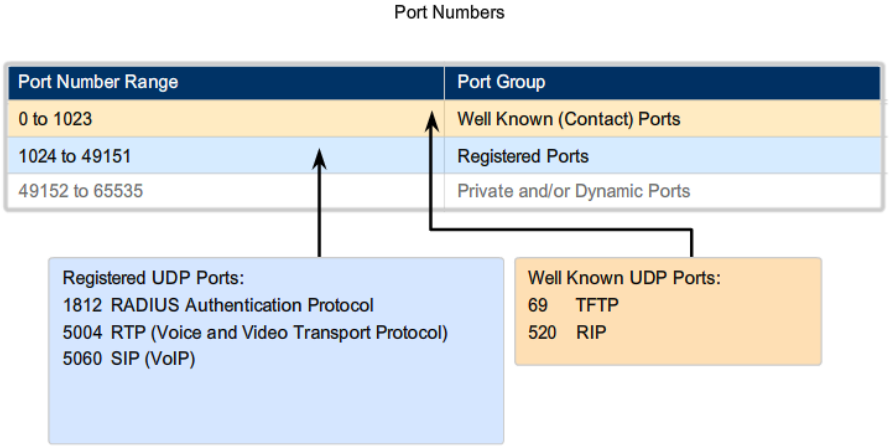
Port Addressing



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

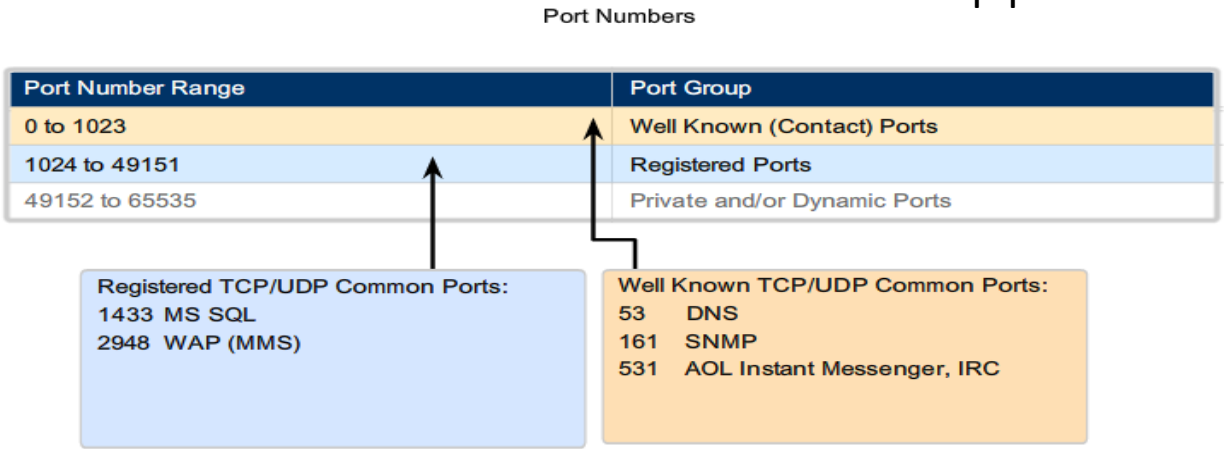
Port Addressing…





Tcp ports

Udp ports



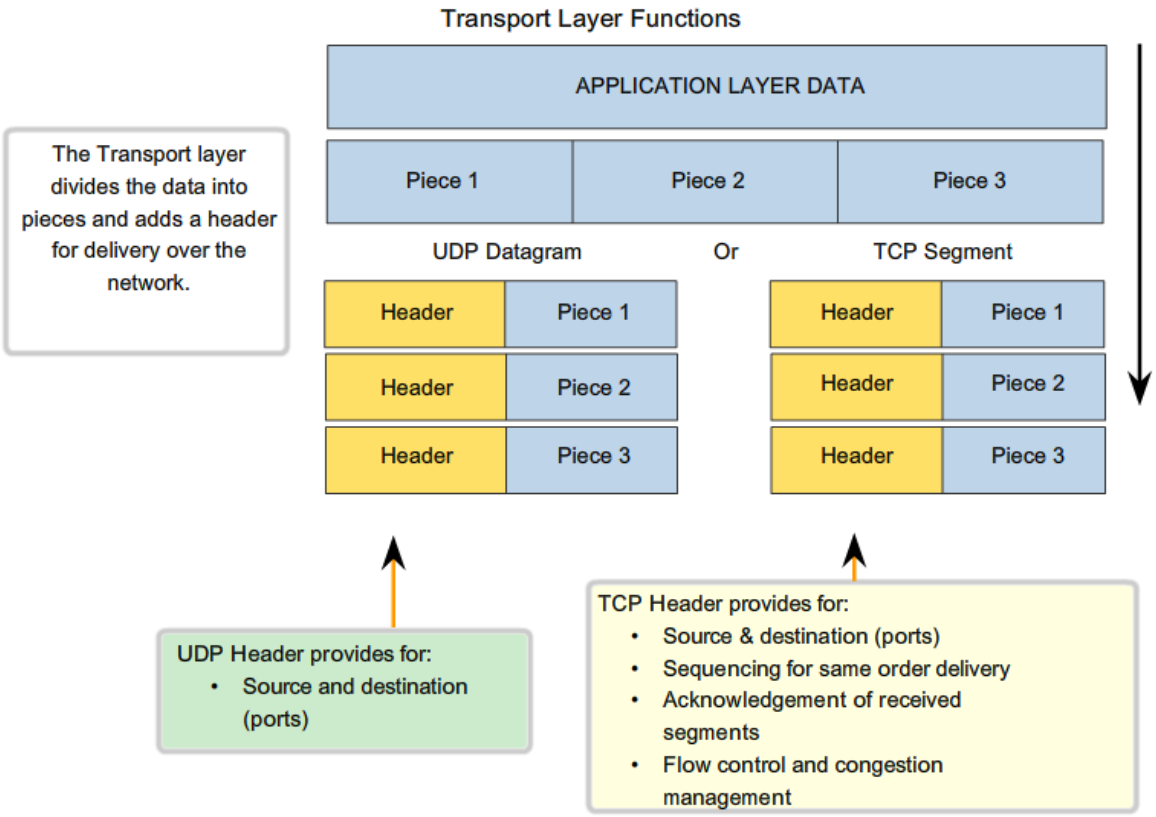
Port Addressing…

Netstat Output

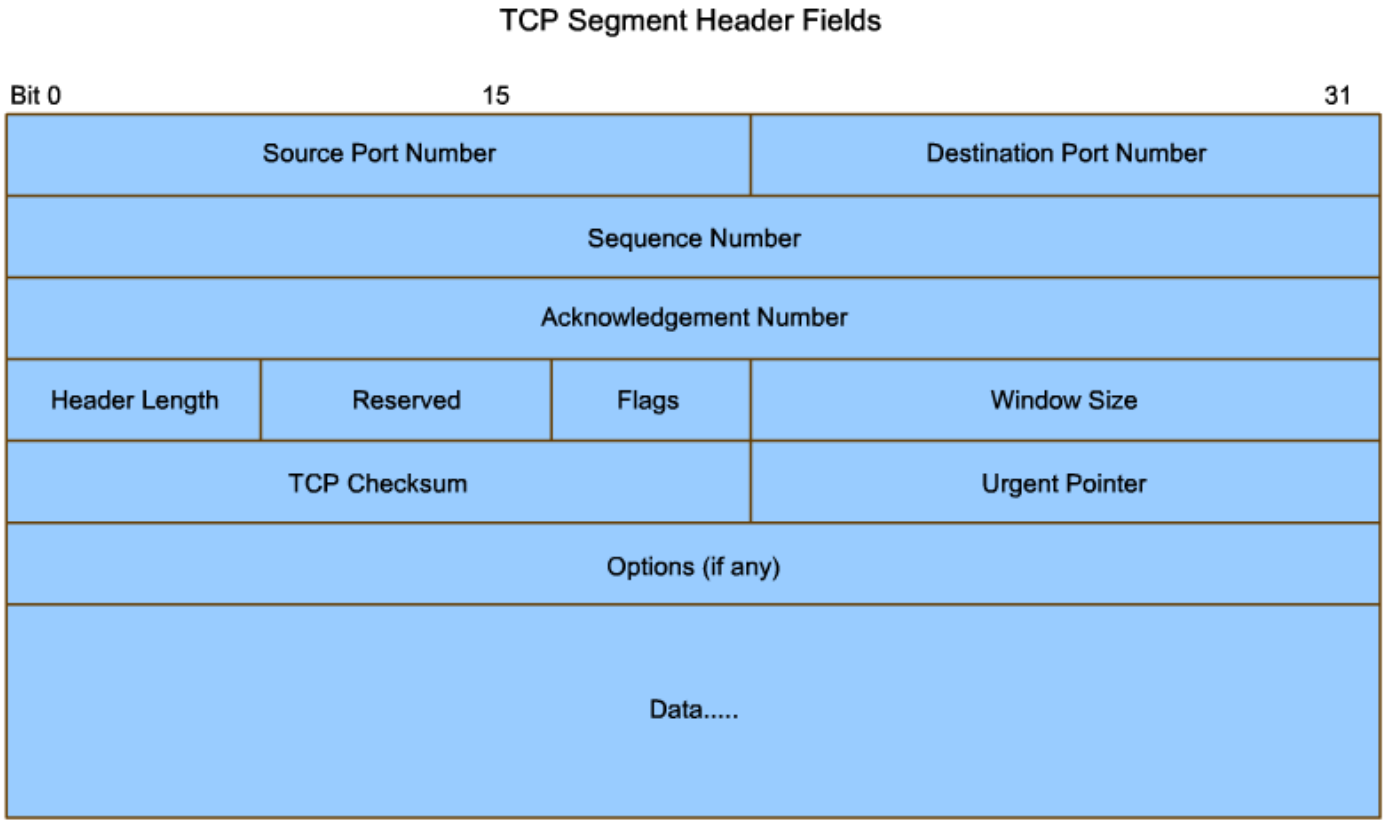
```
C:\>netstat

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:\>
```

Protocol used

Roles of the Transport Layer- Segmentation and Reassembly – divide and conquer

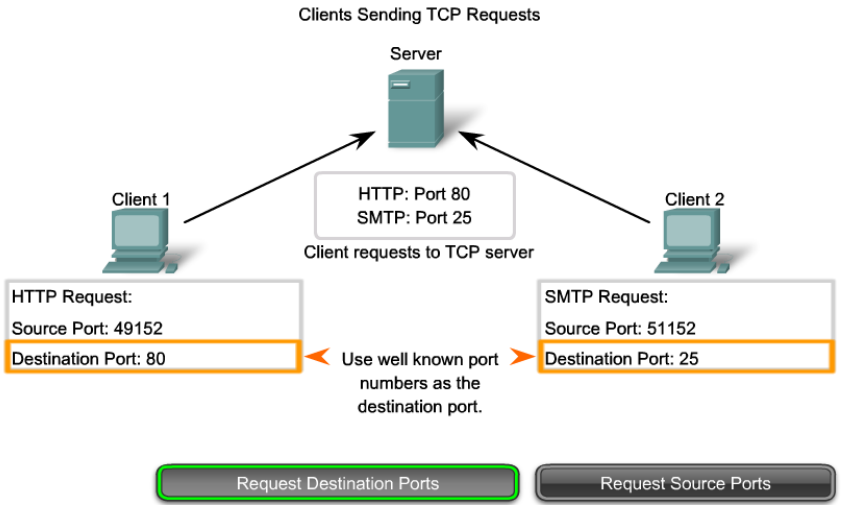


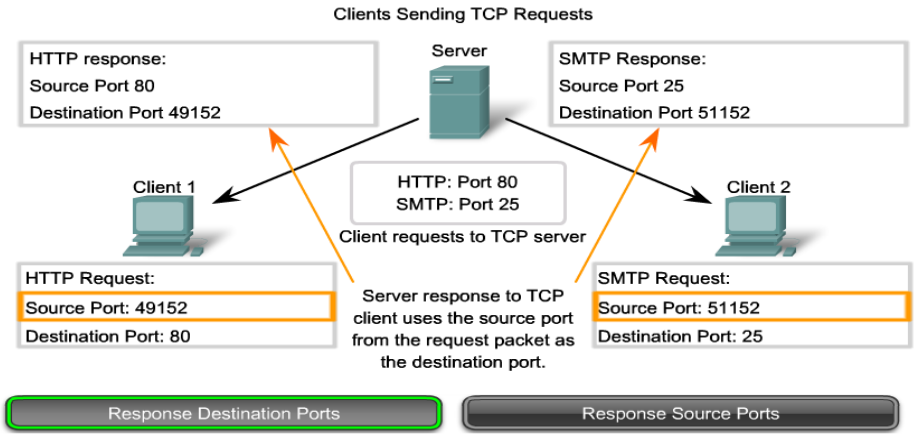
TCP Making Conversations Reliable



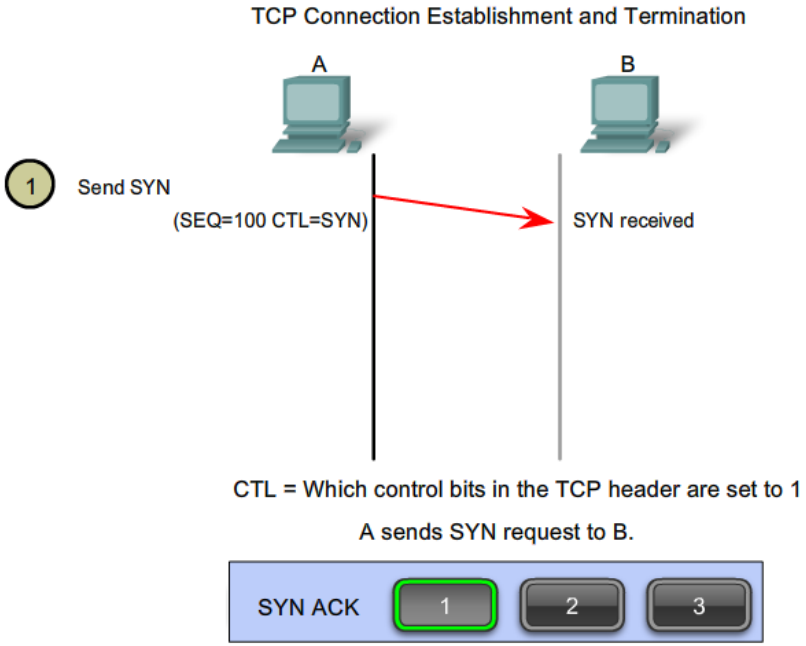
The fields of the TCP header enable TCP to provide connection-oriented, reliable data communications.

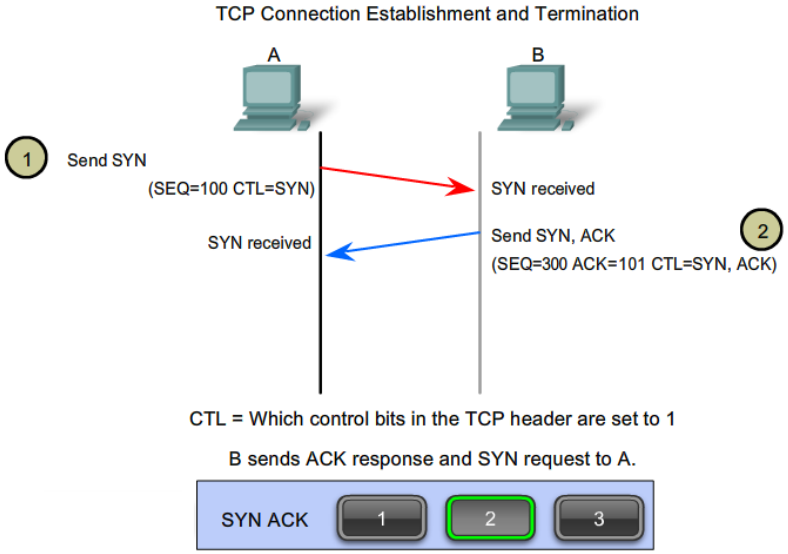
TCP server Processes

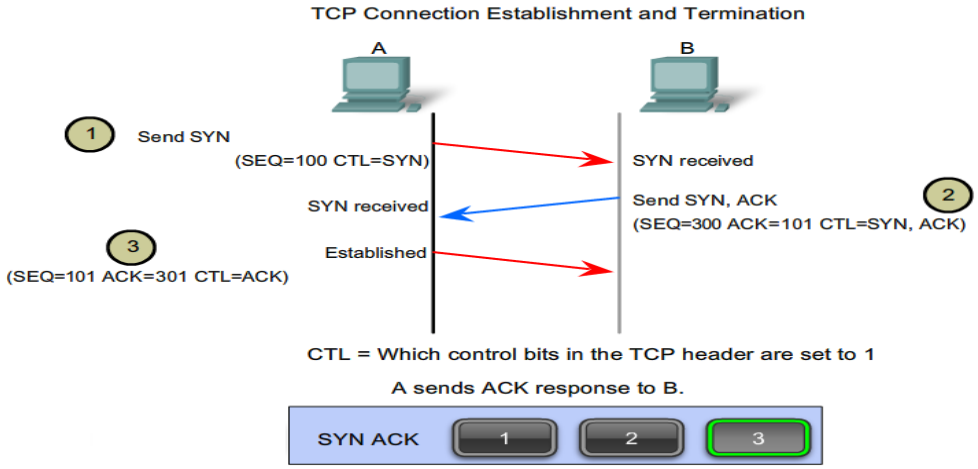




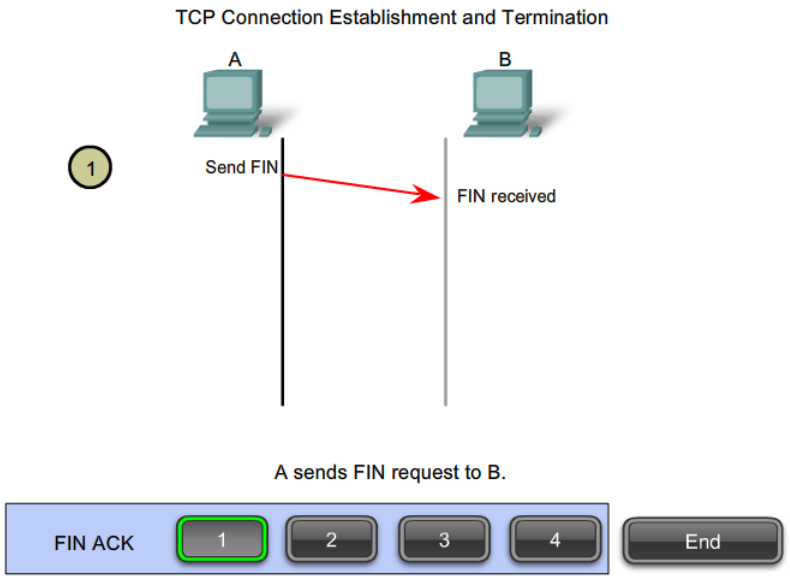
TCP connection Establishment and Termination

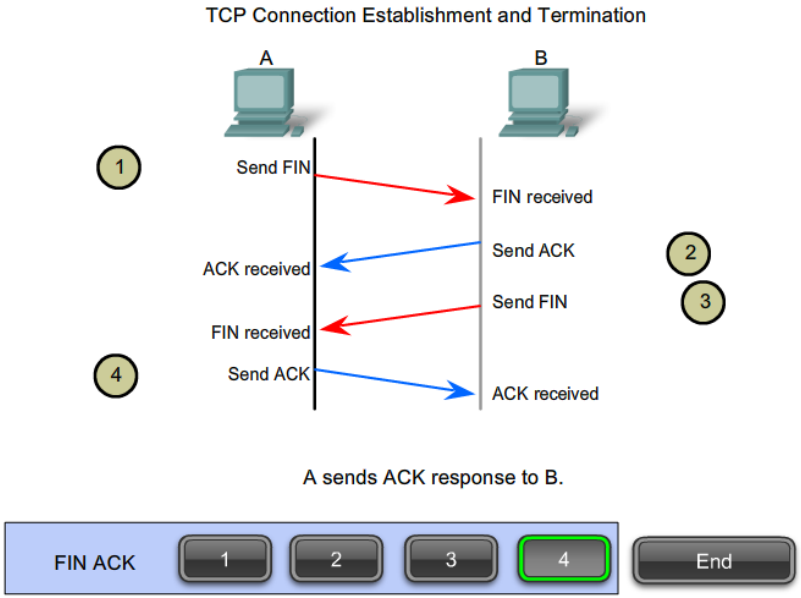






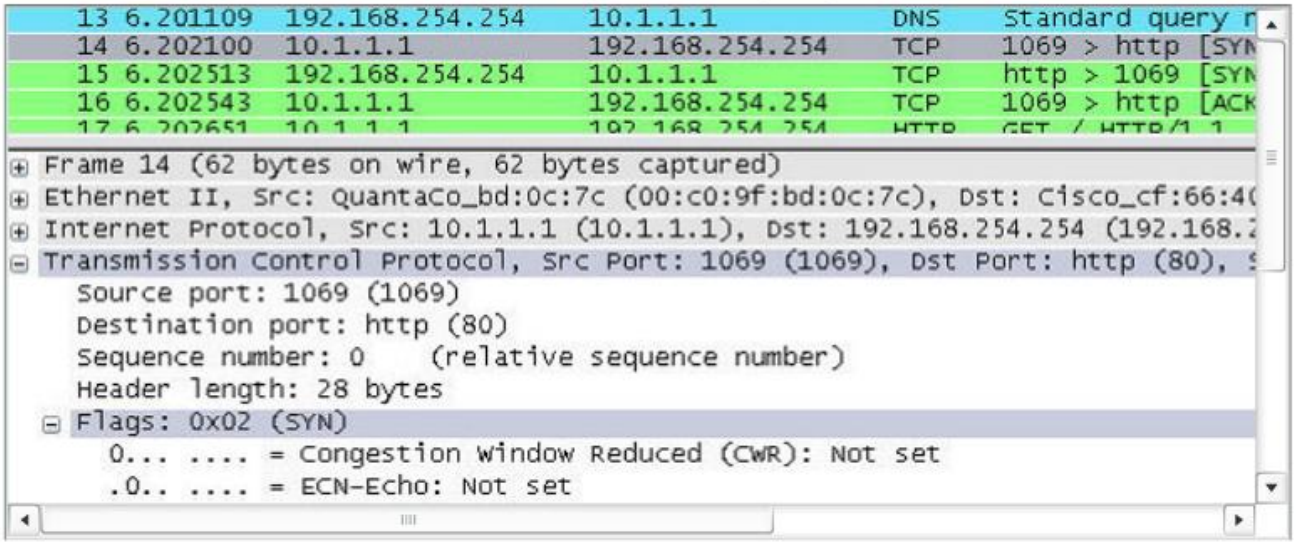
TCP connection Establishment and Termination….





TCP 3-way Handshake – step 1

TCP 3-way Handshake (SYN)



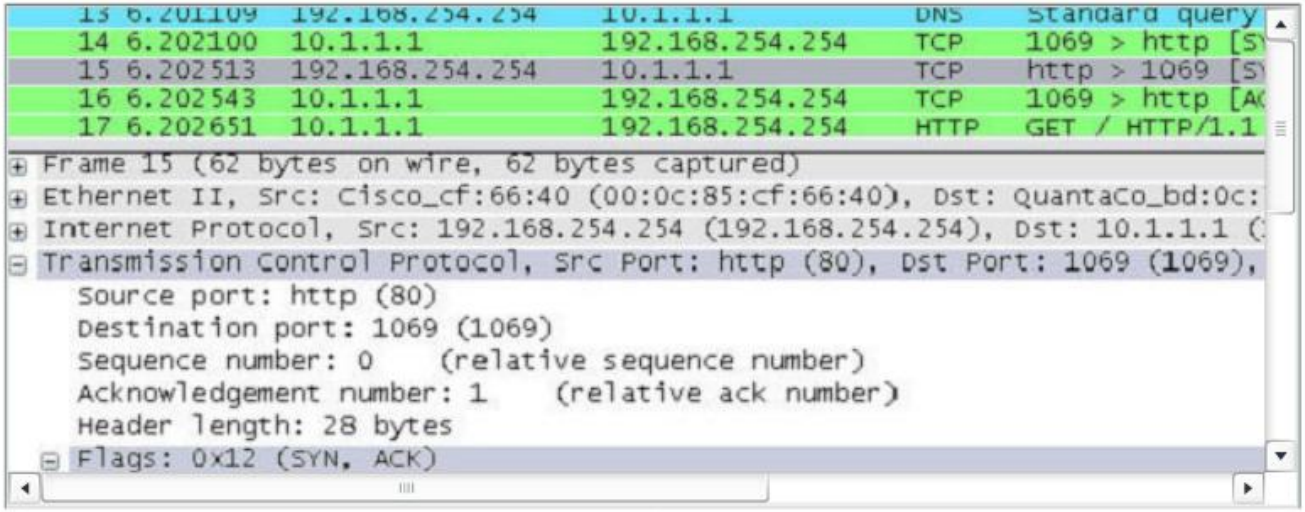
Protocol Analyzer shows initial client request for session in frame 14

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 1069
- Well known destination port is 80 (HTTP port) indicates web server (httpd)

TCP 3-way Handshake – step 2

TCP 3-way Handshake (SYN, ACK)

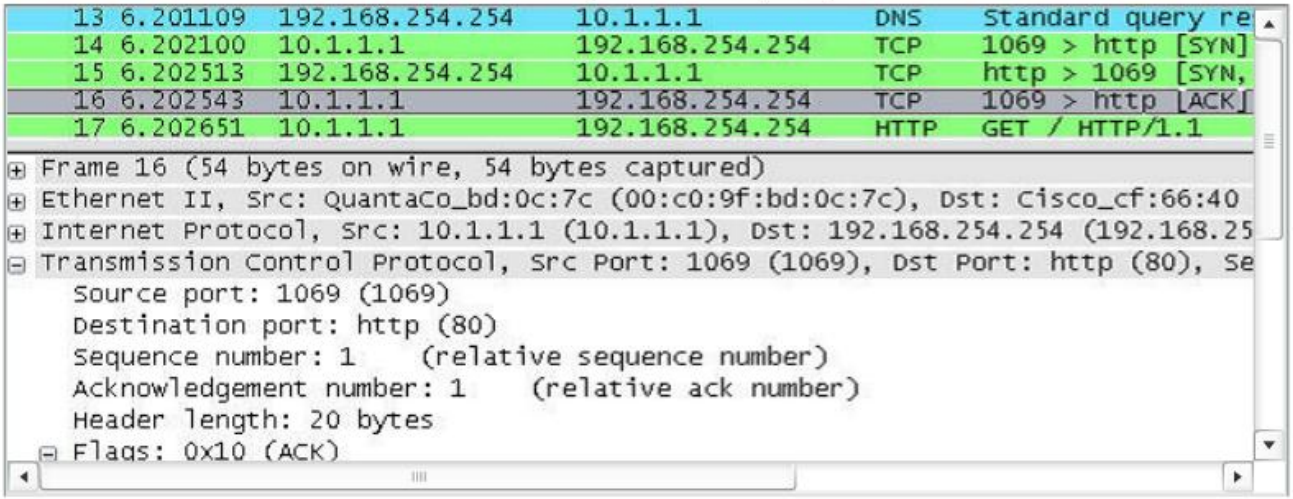


A protocol analyzer shows server response in frame 15

- 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 1069 to corresponding to the clients source port
- Source port number of 80 (HTTP) indicating the web server service (httpd)

TCP 3-way Handshake – step 3

TCP 3-way Handshake (ACK)



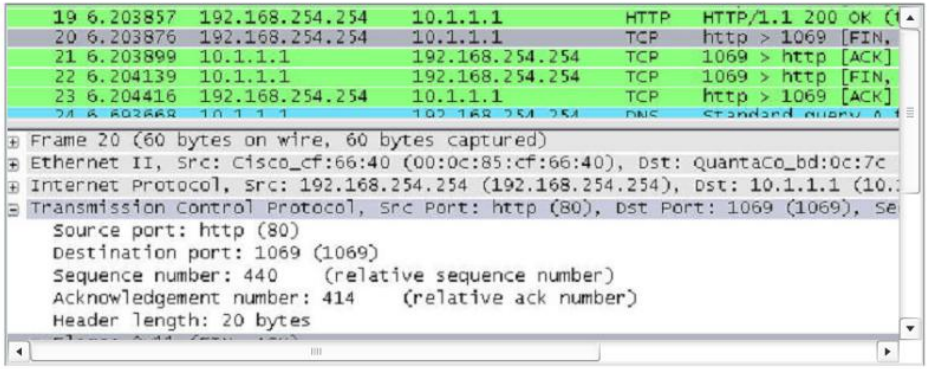
Protocol Analyzer shows client response to session in frame 16

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 1069 to corresponding
- Destination port number of 80 (HTTP) indicating the web server service (httpd)

TCP Session Termination

TCP Session Termination (FIN)



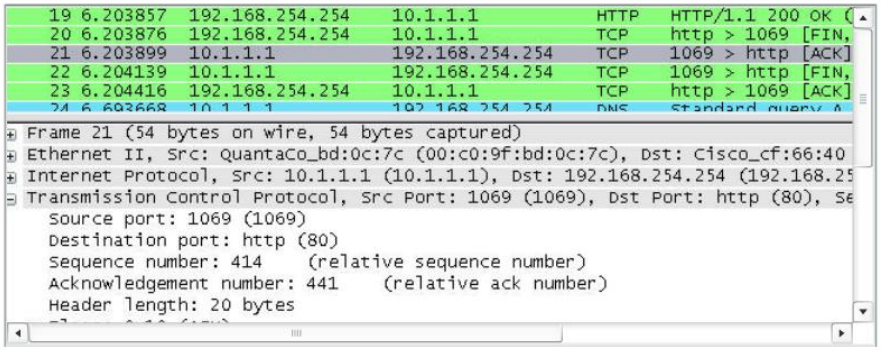
A protocol analyzer shows details of frame 20, TCP FIN request.

The destination and source ports
The header field contents and values

FIN

ACK

TCP Session Termination (ACK)



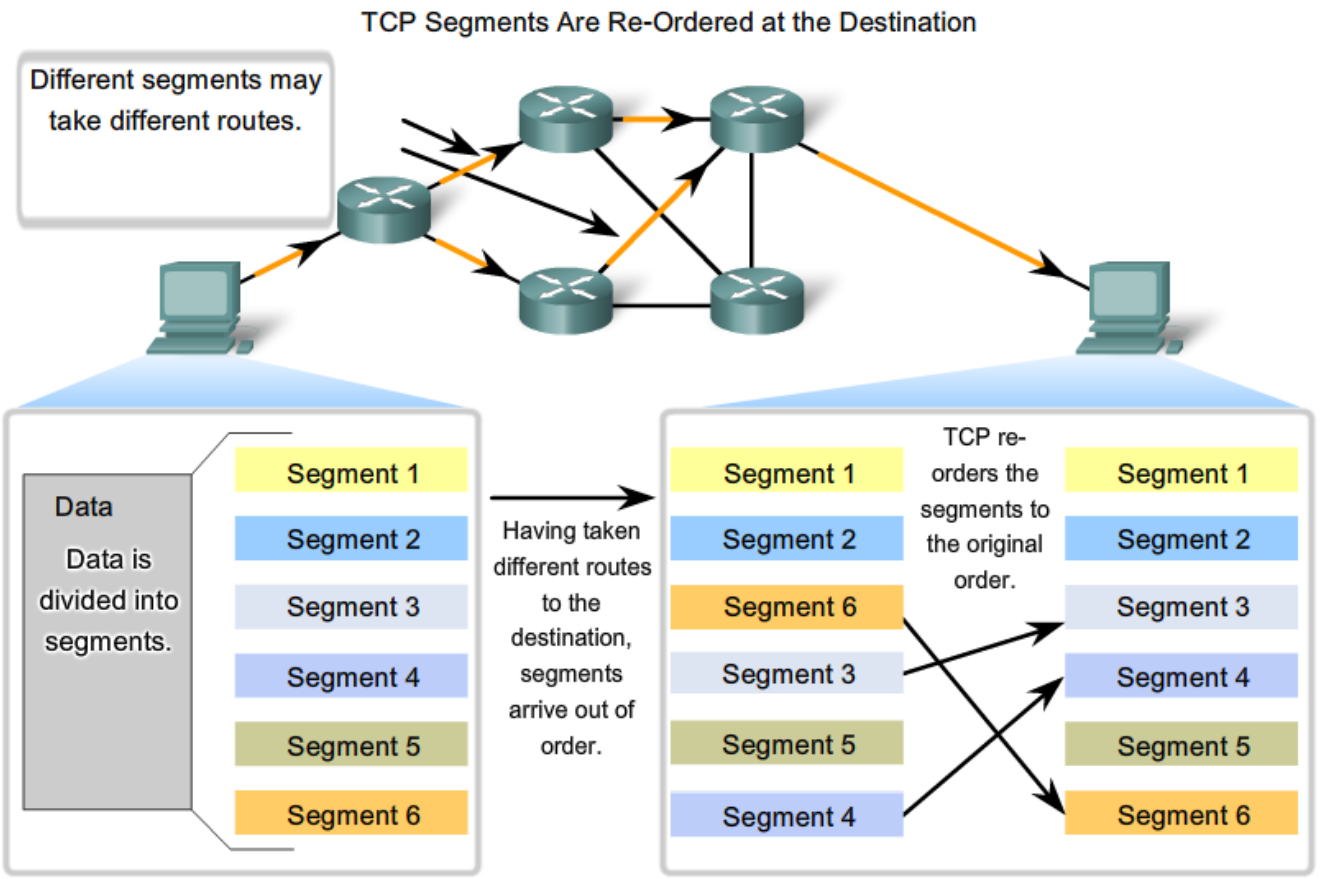
A protocol analyzer shows details of frame 21, TCP ACK response.

The destination and source ports
The header field contents and values

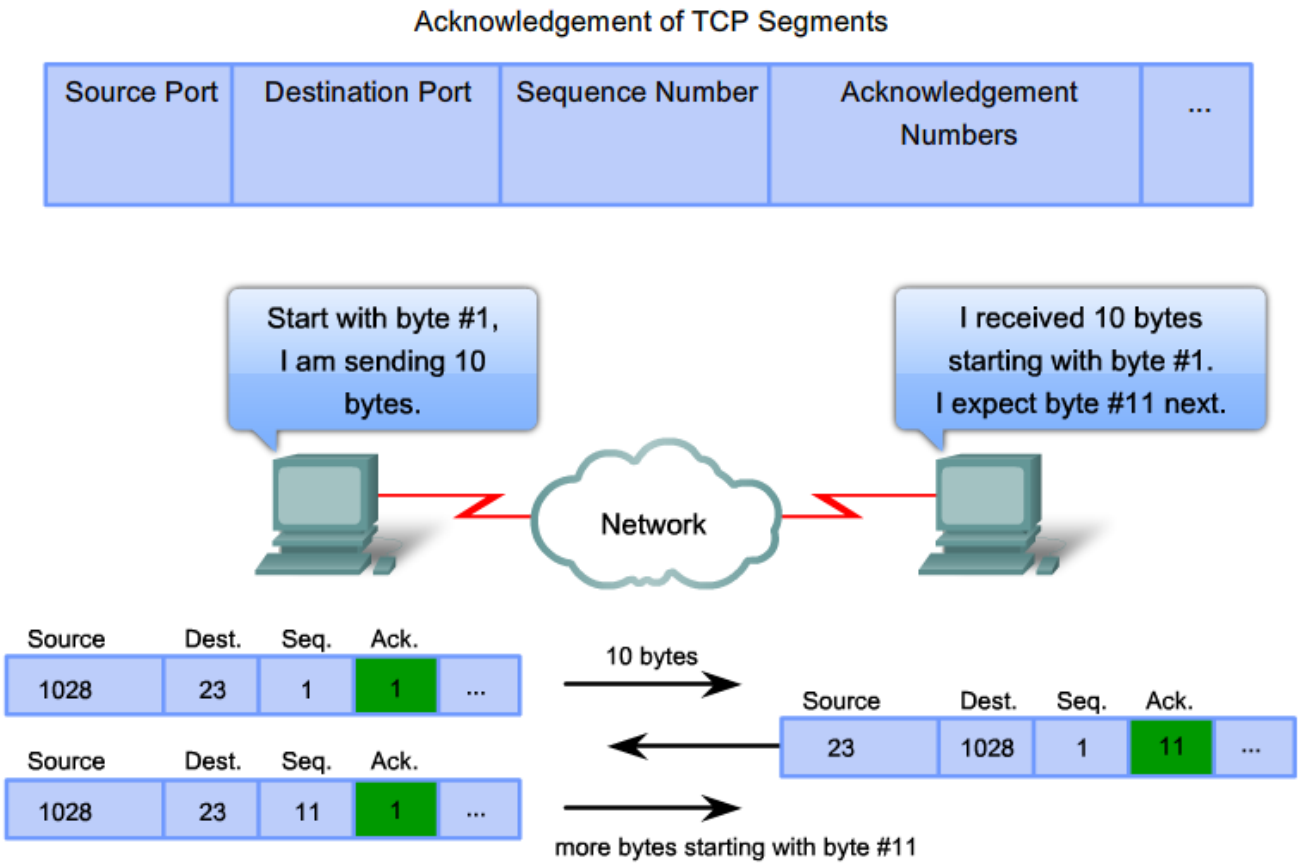
FIN

ACK

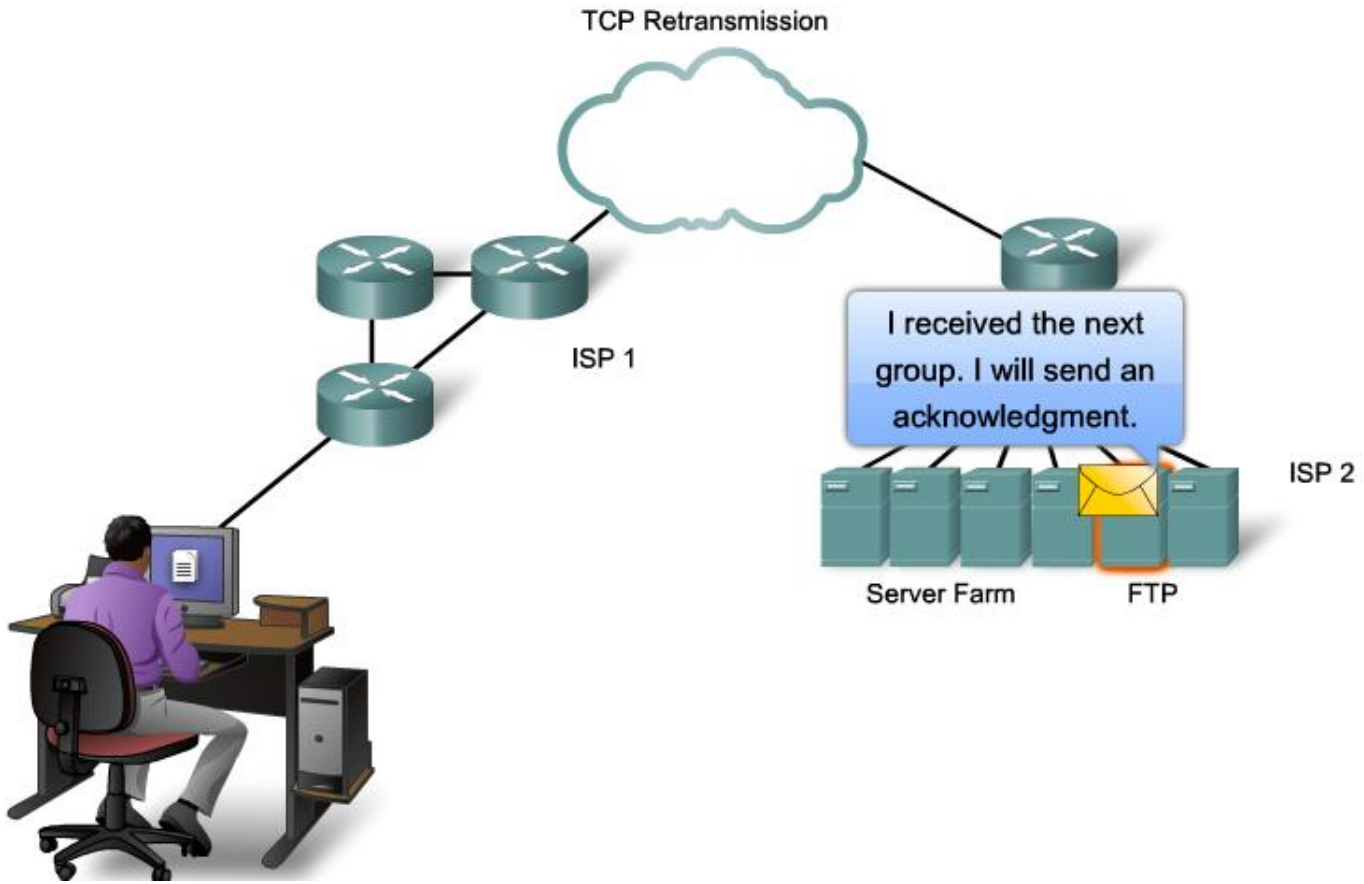
TCP Segments Reassembly



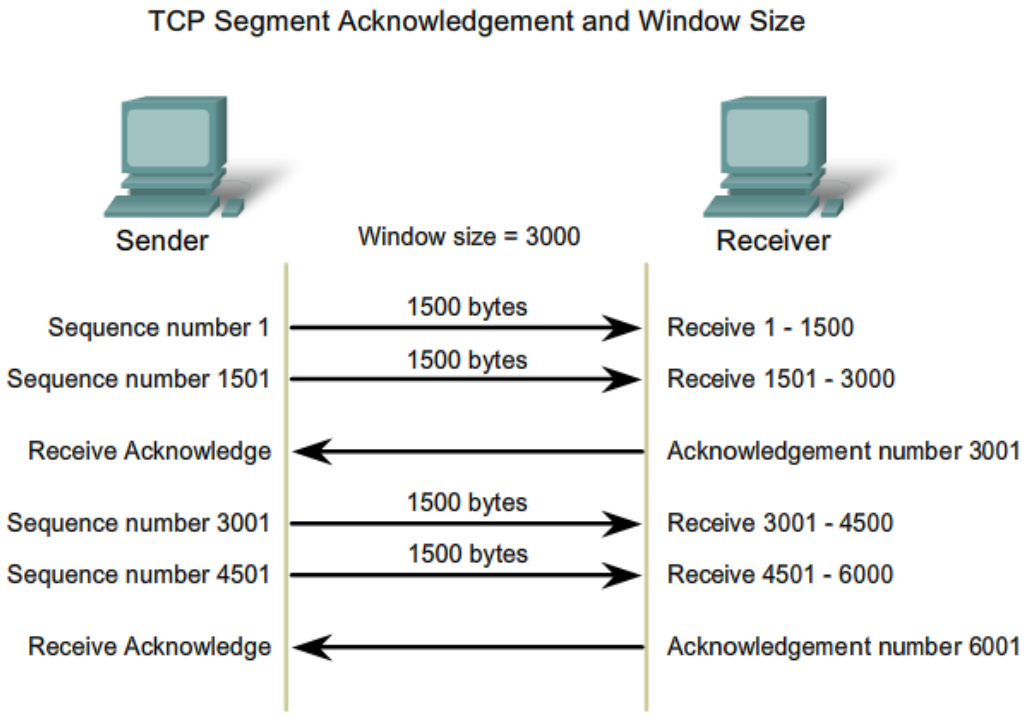
TCP Acknowledgment with Windowing



TCP Retransmission

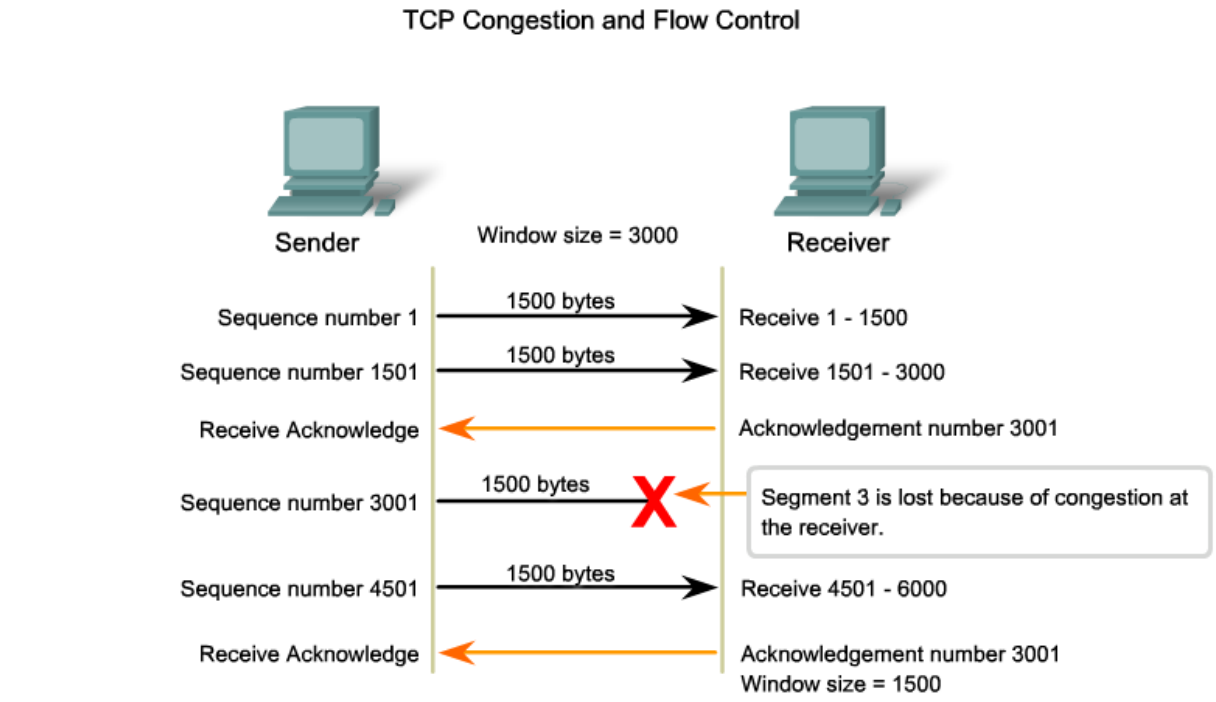


TCP Congestion Control- Minimize Segment Loss



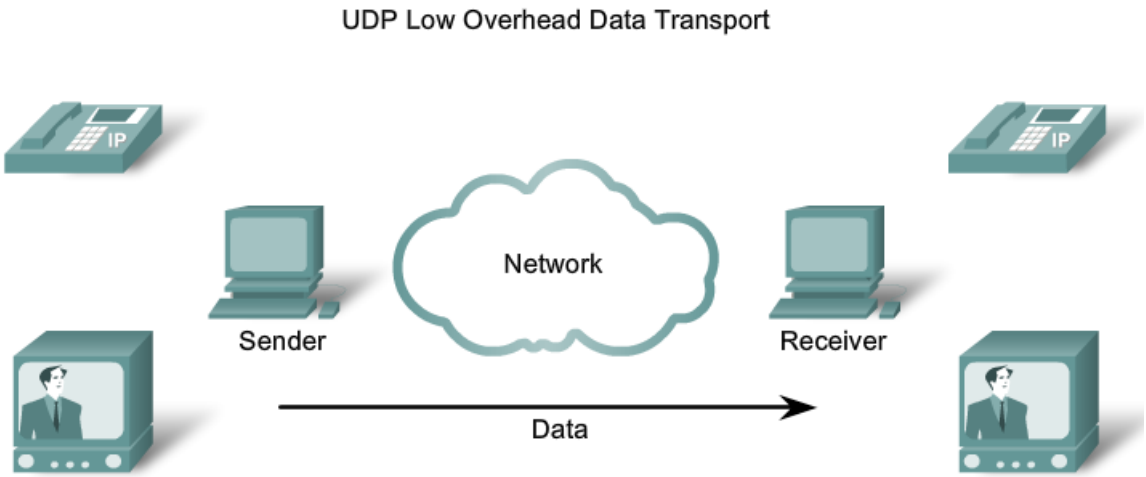
The window size determines the number of bytes sent before an acknowledgment is expected.
The acknowledgement number is the number of the next expected byte.

TCP Congestion Control- Minimize Segment Loss…



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

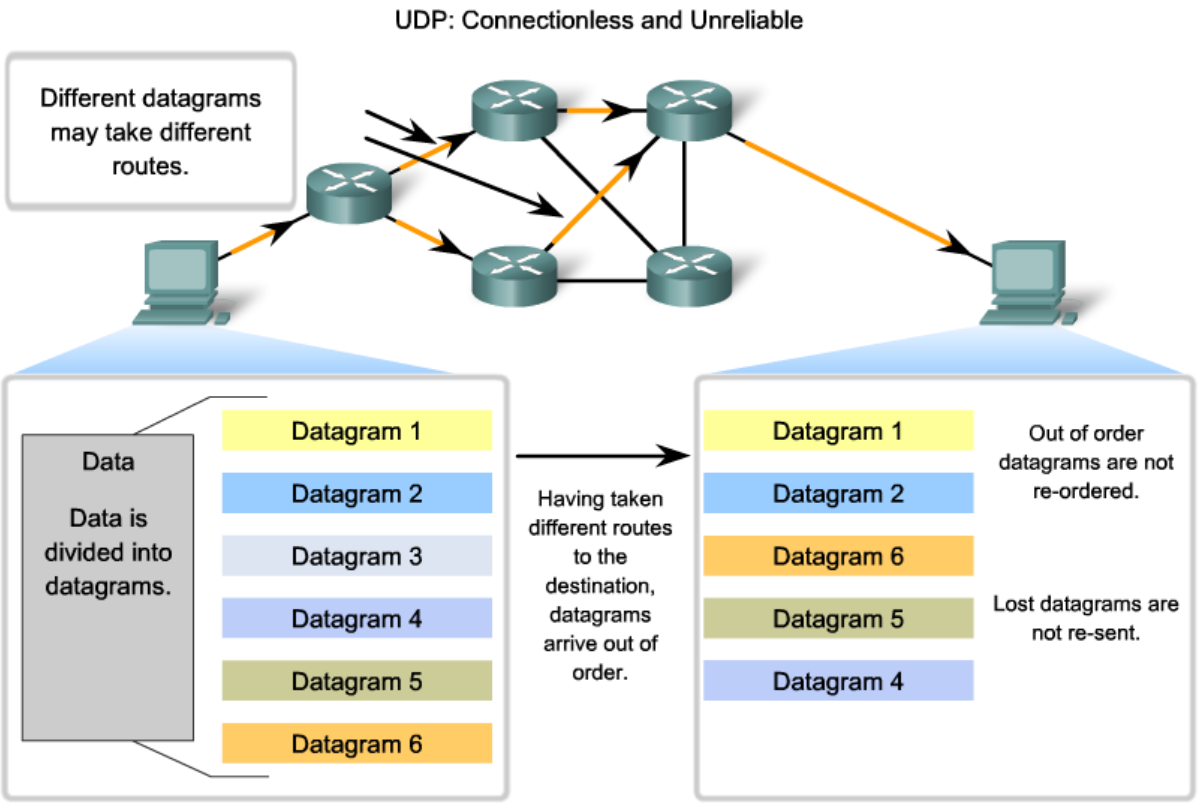
UDP



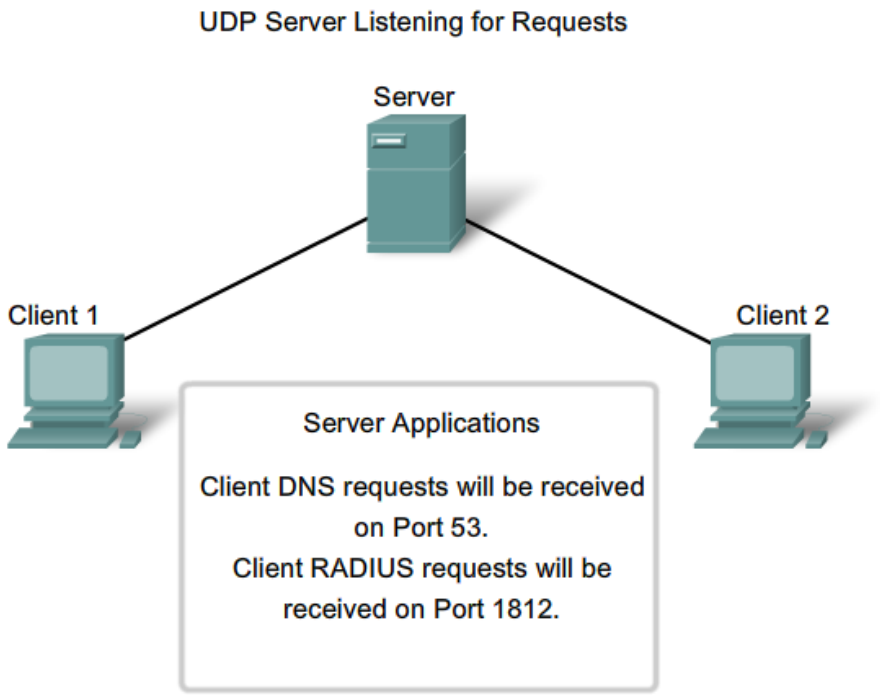
UDP does not establish a connection before sending data.

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

UDP Datagram Reassembly

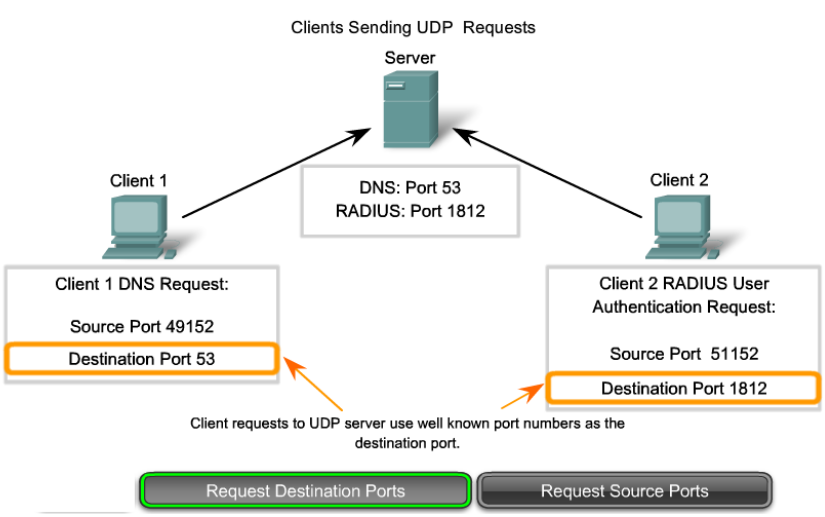


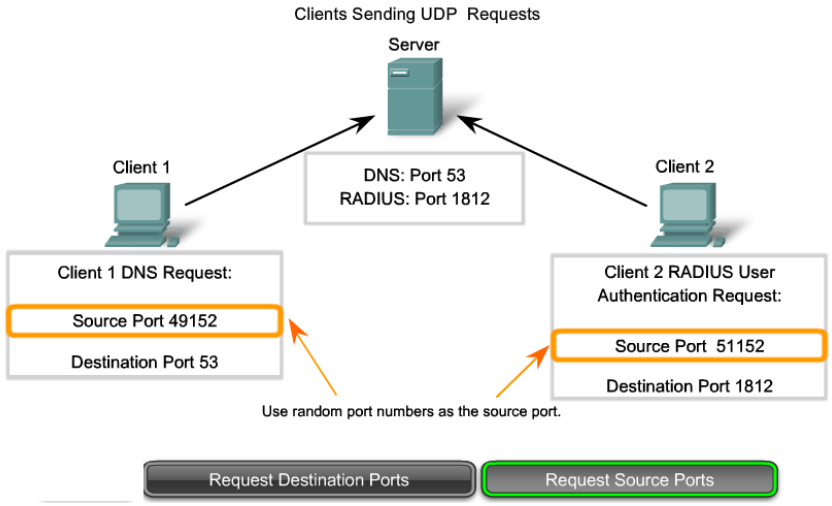
UDP server Processes and Requests



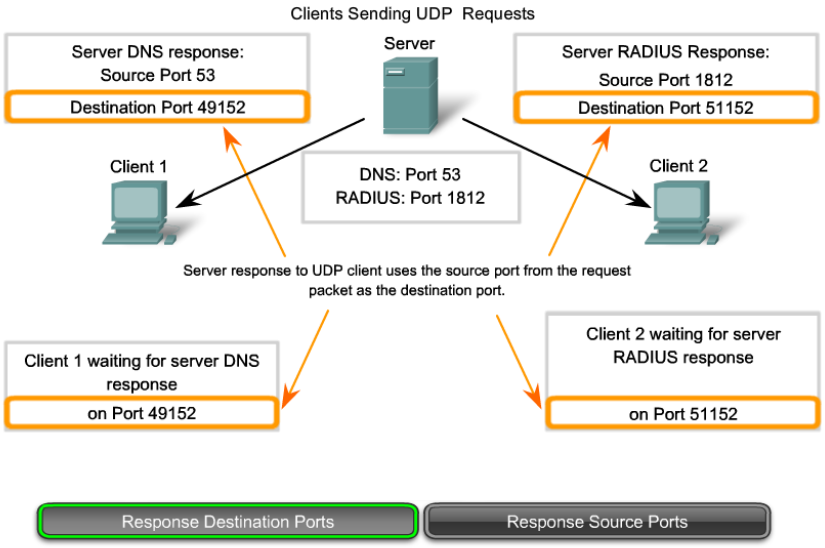
Client requests to servers have well known ports numbers as the destination port.

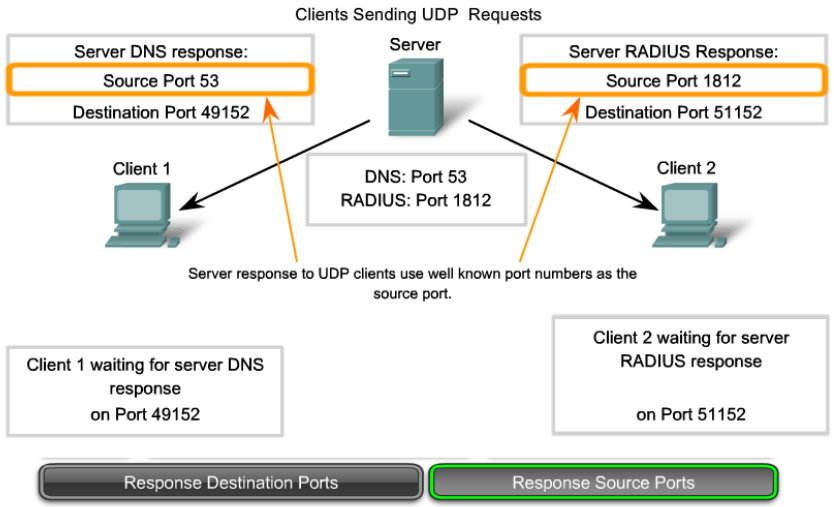
UDP Client Processes





UDP Client Processes…





Summary

- The Transport layer provides for data network needs by:
- Dividing data received from an application into segments
- Adding a header to identify and manage each segment
- Using the header information to reassemble the segments back into application data
- Passing the assembled data to the correct application

- UDP and TCP are common Transport layer protocols.

- UDP datagrams and TCP segments have headers prefixed to the data that include a source port number and destination port number. These port numbers enable data to be directed to the correct application running on the destination computer.

- TCP does not pass any data to the network until it knows that the destination is ready to receive it. TCP then manages the flow of the data and resends any data segments that are not acknowledged as being received at the destination. TCP uses mechanisms of handshaking, timers and acknowledgements, and dynamic windowing to achieve these reliable features. This reliability does, however, impose overhead on the network in terms of much larger segment headers and more network traffic between the source and destination managing the data transport.

- If the application data needs to be delivered across the network quickly, or if network bandwidth cannot support the overhead of control messages being exchanged between the source and the destination systems, UDP would be the developer's preferred Transport layer protocol. Because UDP does not track or acknowledge the receipt of datagrams at the destination - it just passes received datagrams to the Application layer as they arrive - and does not resend lost datagrams. However, this does not necessarily mean that the communication itself is unreliable; there may be mechanisms in the Application layer protocols and services that process lost or delayed datagrams if the application has these requirements.

- The choice of Transport layer protocol is made by the developer of the application to best meet the user requirements. The developer bears in mind, though, that the other layers all play a part in data network communications and will influence its performance.

The End