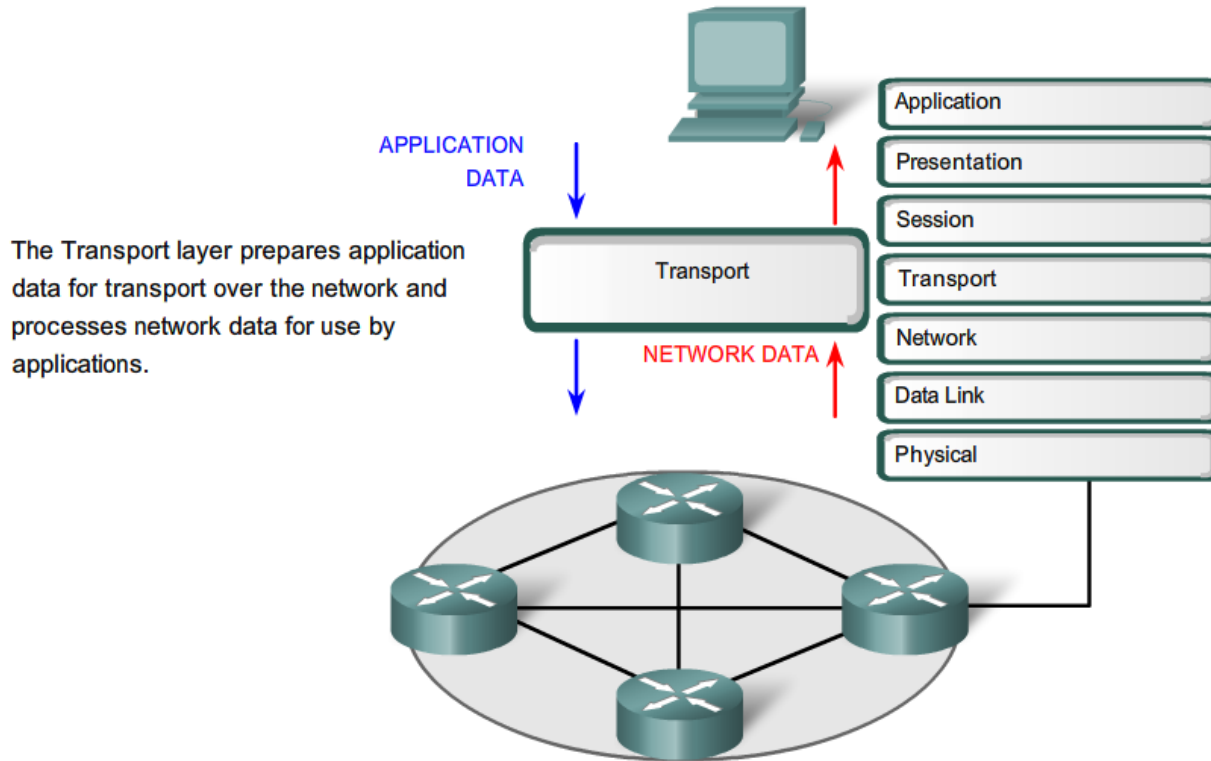


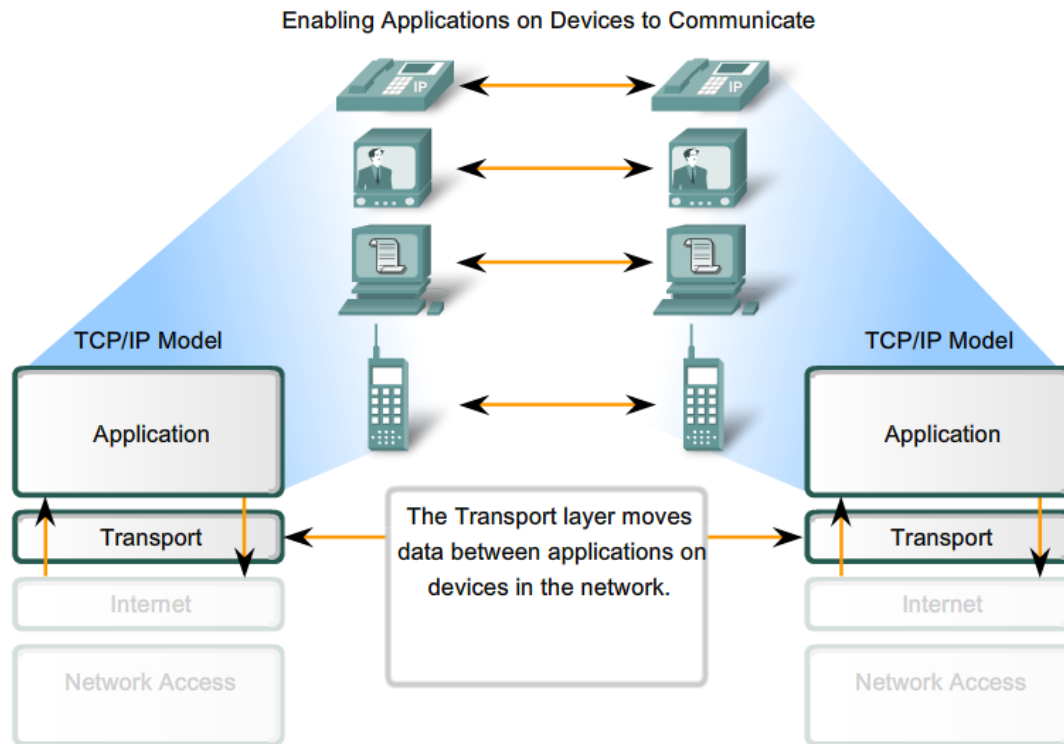
Transport Layer

Introduction

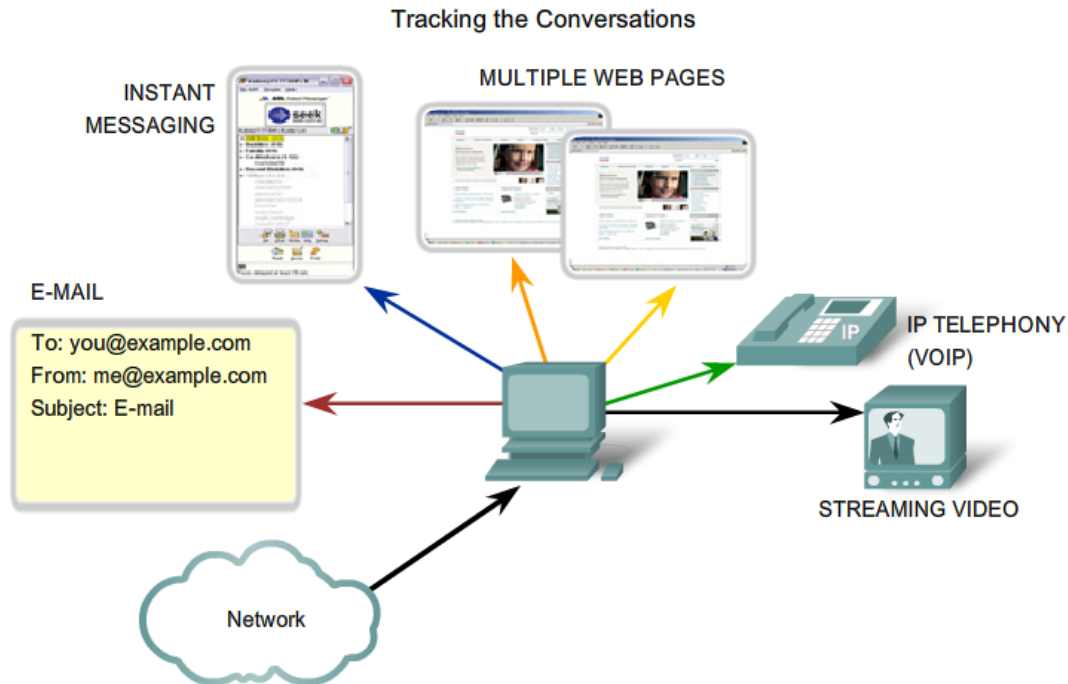
The OSI Transport Layer



Purpose of the Transport Layer

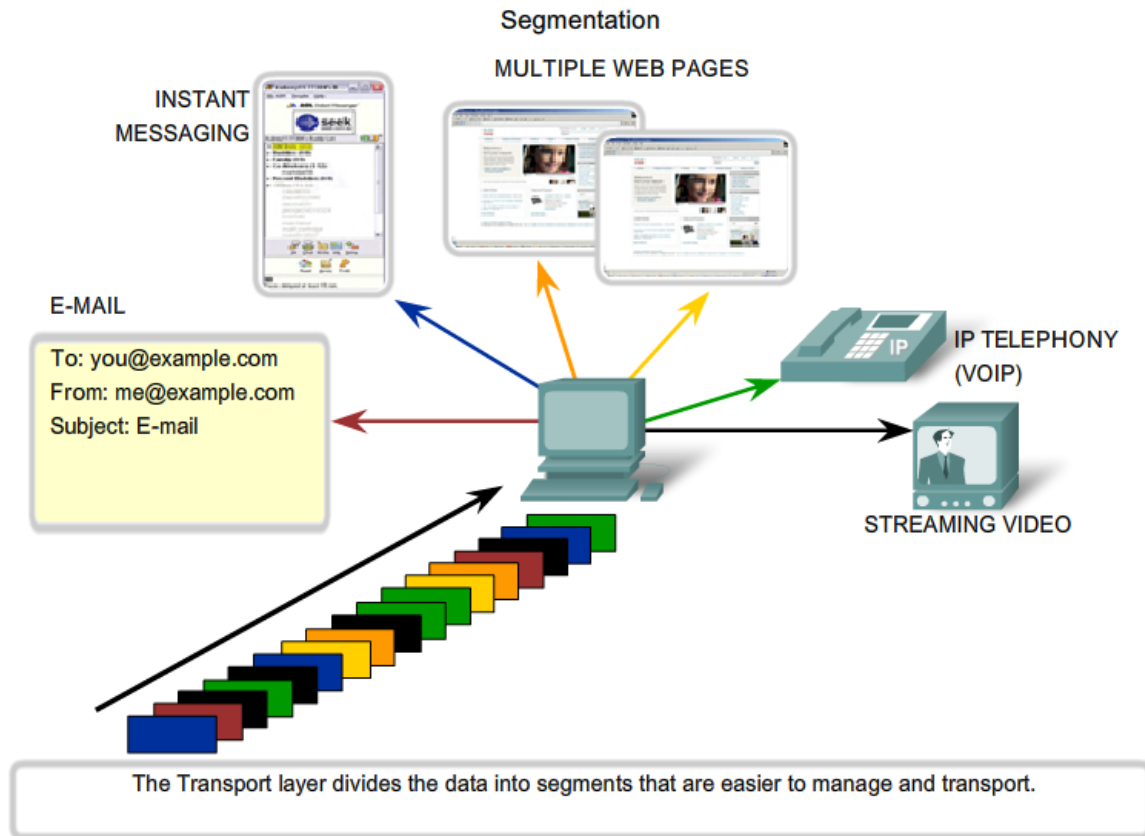


Purpose of the Transport Layer...

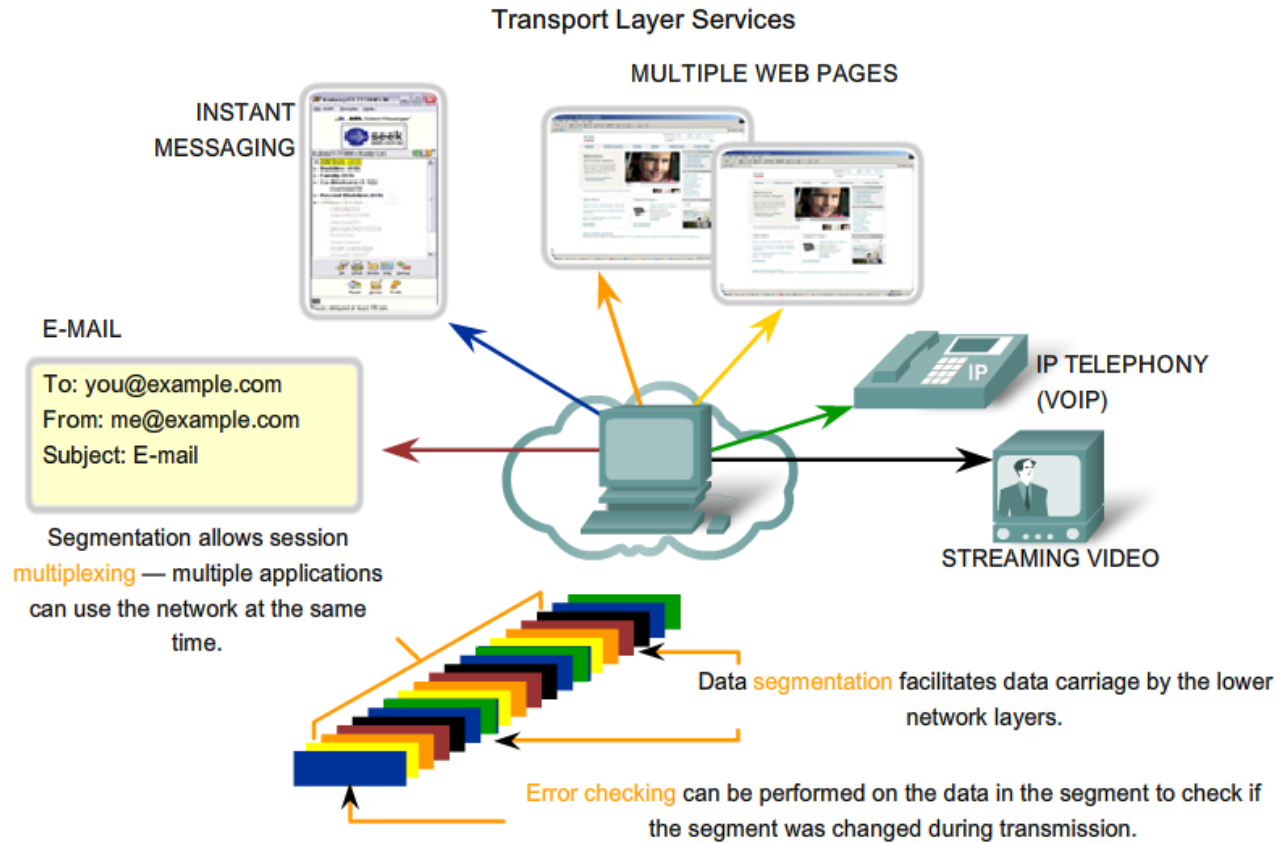


The Transport layer segments the data and manages the separation of data for different applications. Multiple applications running on a device receive the correct data.

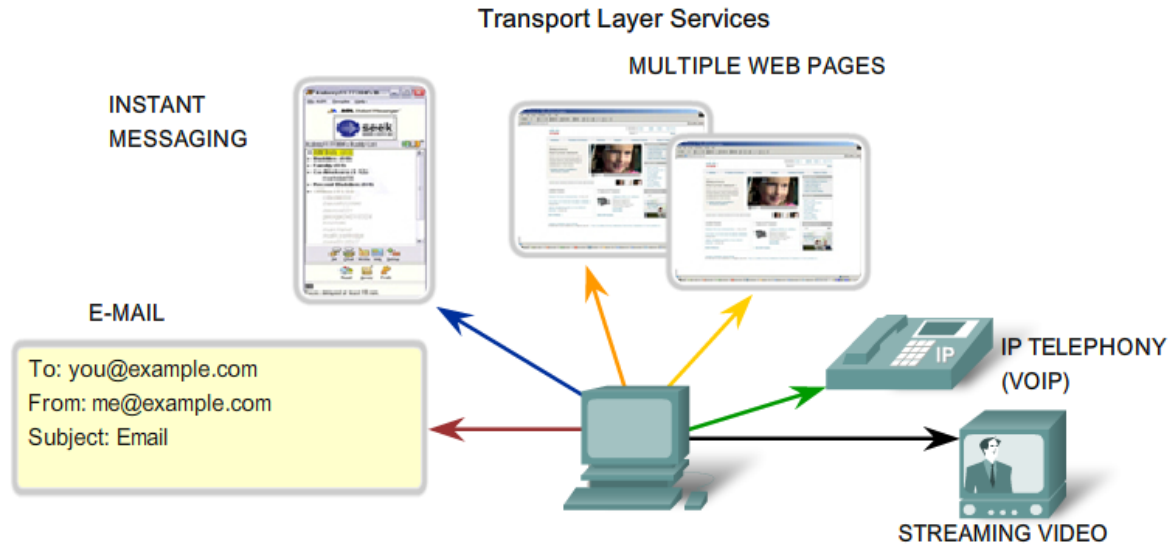
Purpose of the Transport Layer...



Controlling the Conversations



Controlling the Conversations...



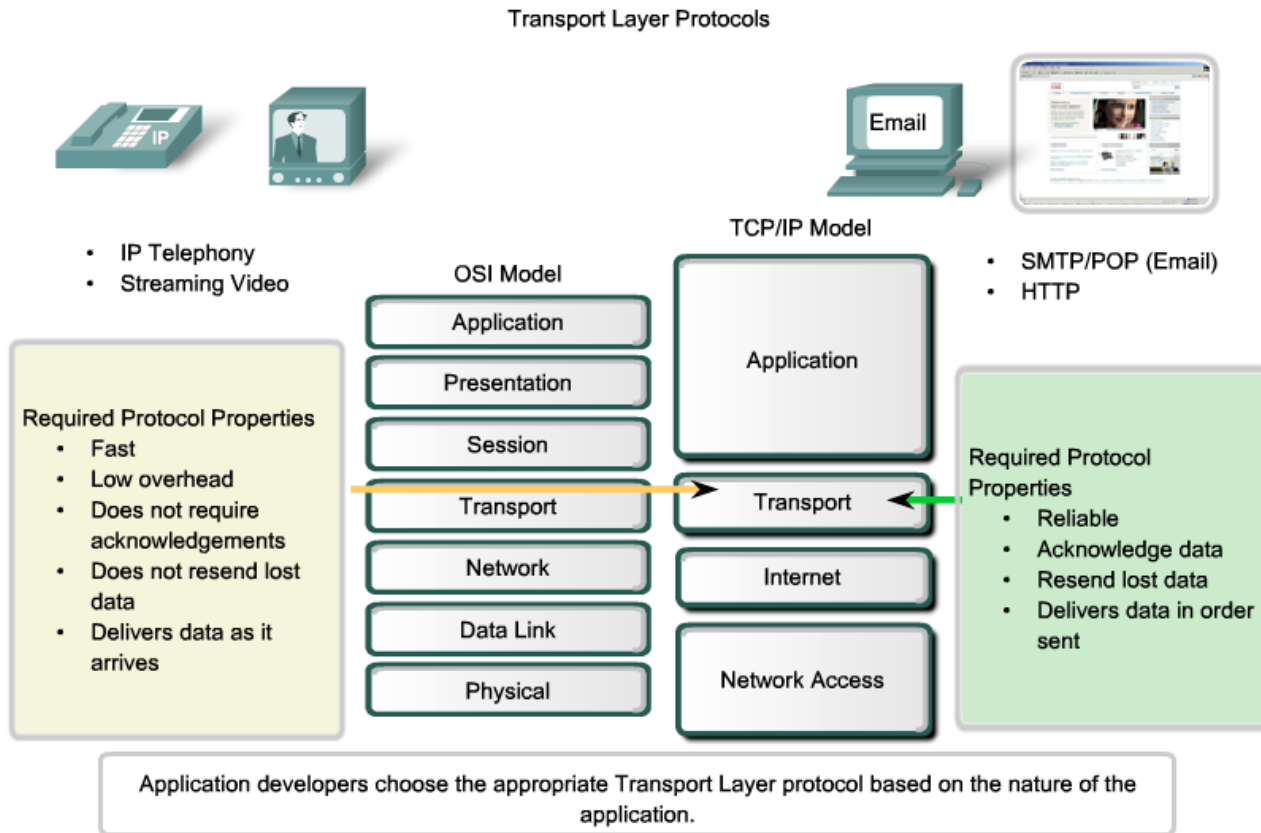
Establishing a Session ensures the application is ready to receive the data.

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.

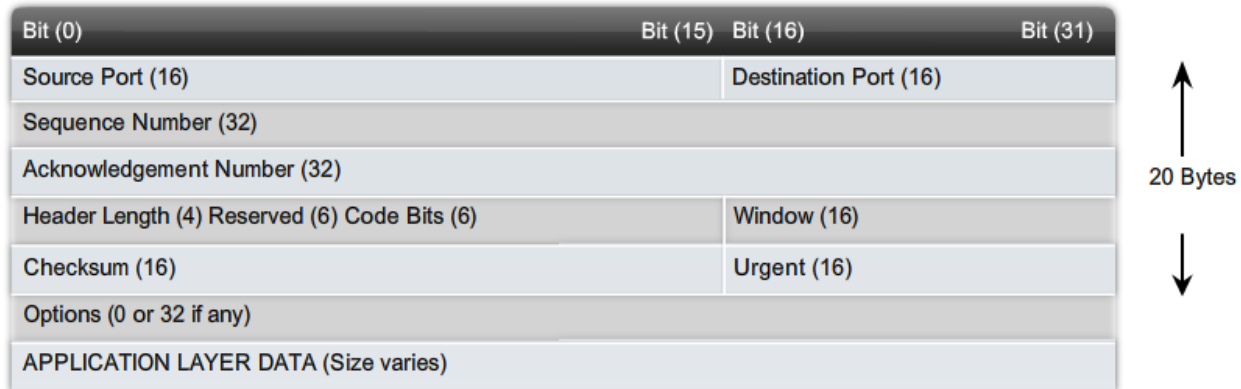
Support Reliable Communication



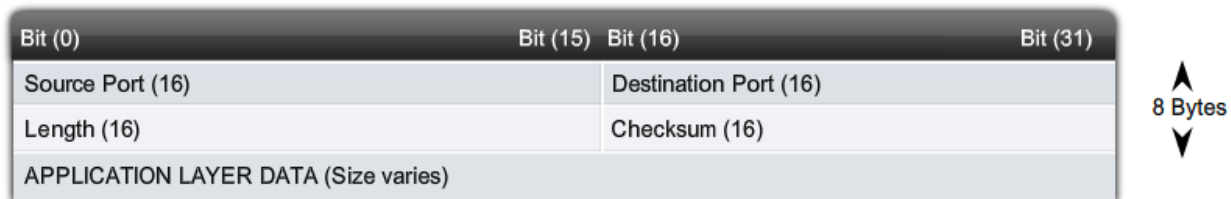
TCP and UDP

TCP and UDP Headers

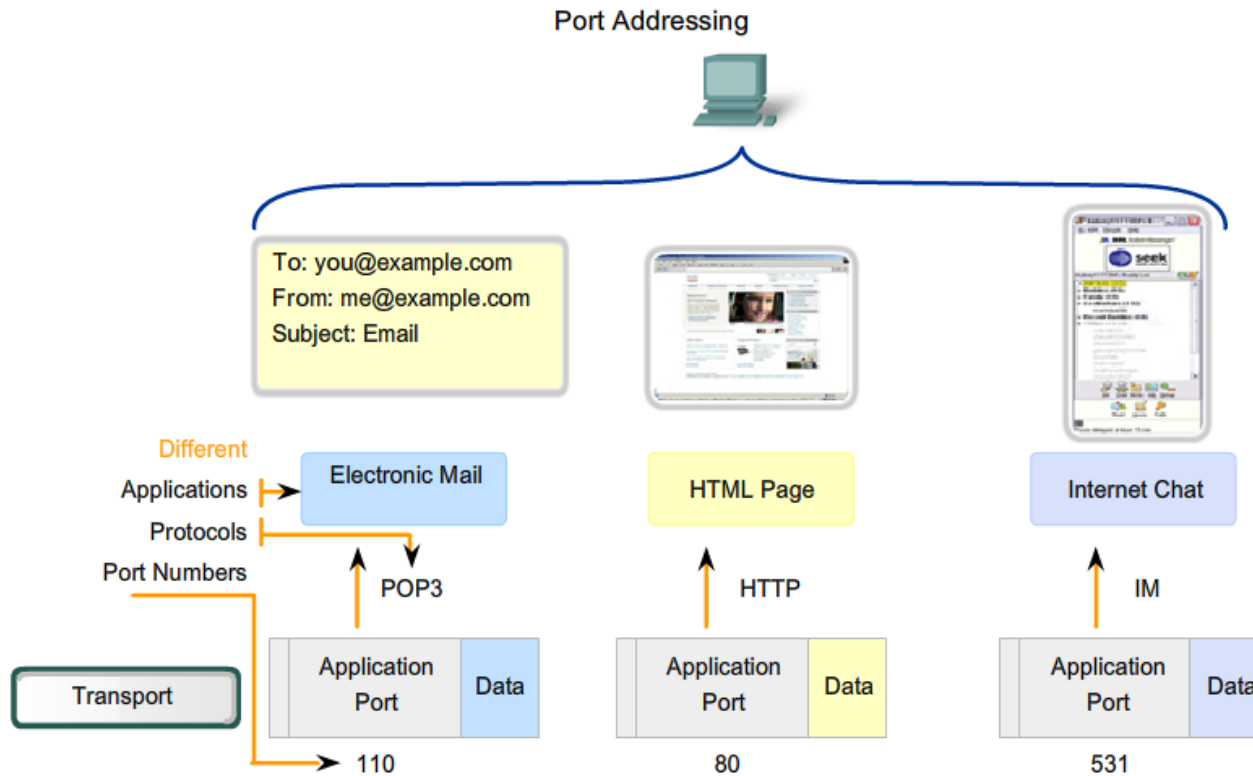
TCP Segment



UDP Datagram



Port Addressing



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

Port Addressing...

Port Numbers

Port Number Range	Port Group
0 to 1023	Well Known (Contact) Ports
1024 to 49151	Registered Ports
49152 to 65535	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)

Port Numbers

Port Number Range	Port Group
0 to 1023	Well Known (Contact) Ports
1024 to 49151	Registered Ports
49152 to 65535	Private and/or Dynamic Ports

Registered UDP Ports:
 1812 RADIUS Authentication Protocol
 5004 RTP (Voice and Video Transport Protocol)
 5060 SIP (VoIP)

Well Known UDP Ports:
 69 TFTP
 520 RIP

Tcp ports

Udp ports

Port Numbers

Port Number Range	Port Group
0 to 1023	Well Known (Contact) Ports
1024 to 49151	Registered Ports
49152 to 65535	Private and/or Dynamic Ports

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

Port Addressing...

netstat Output

```
C:\>netstat
```

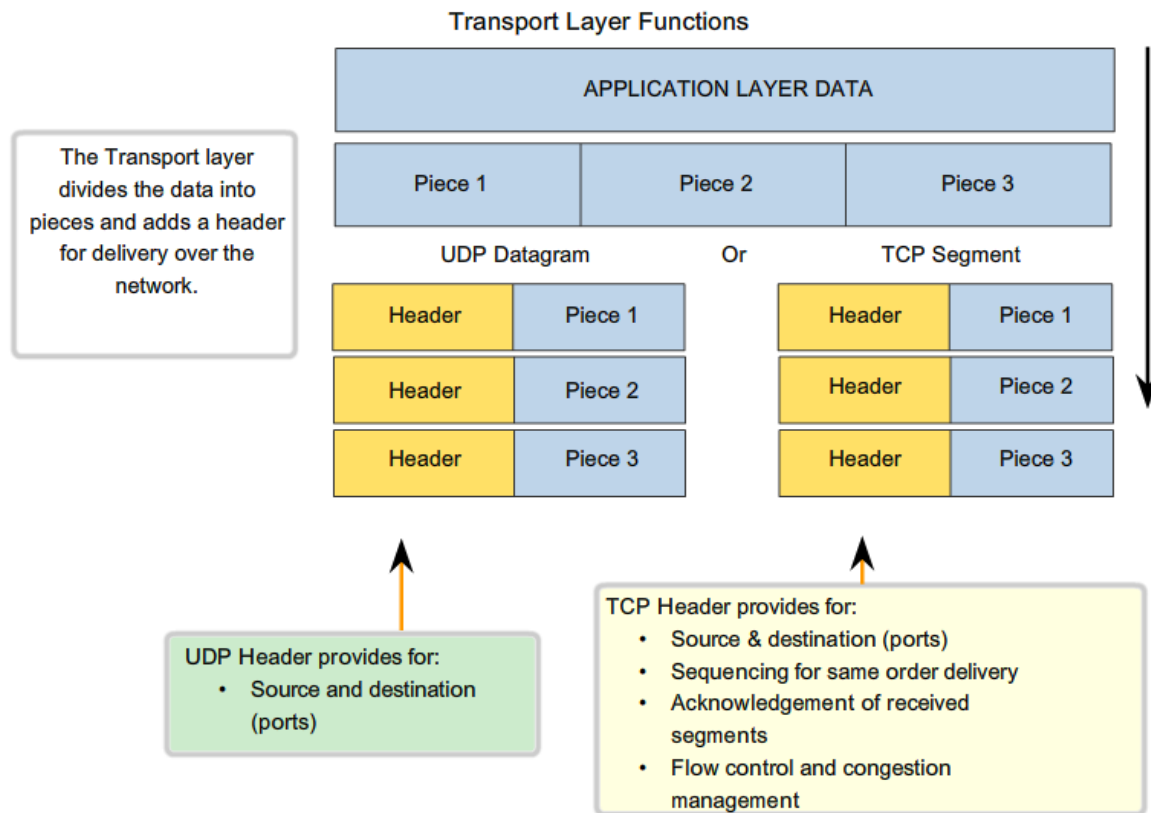
Active Connections

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

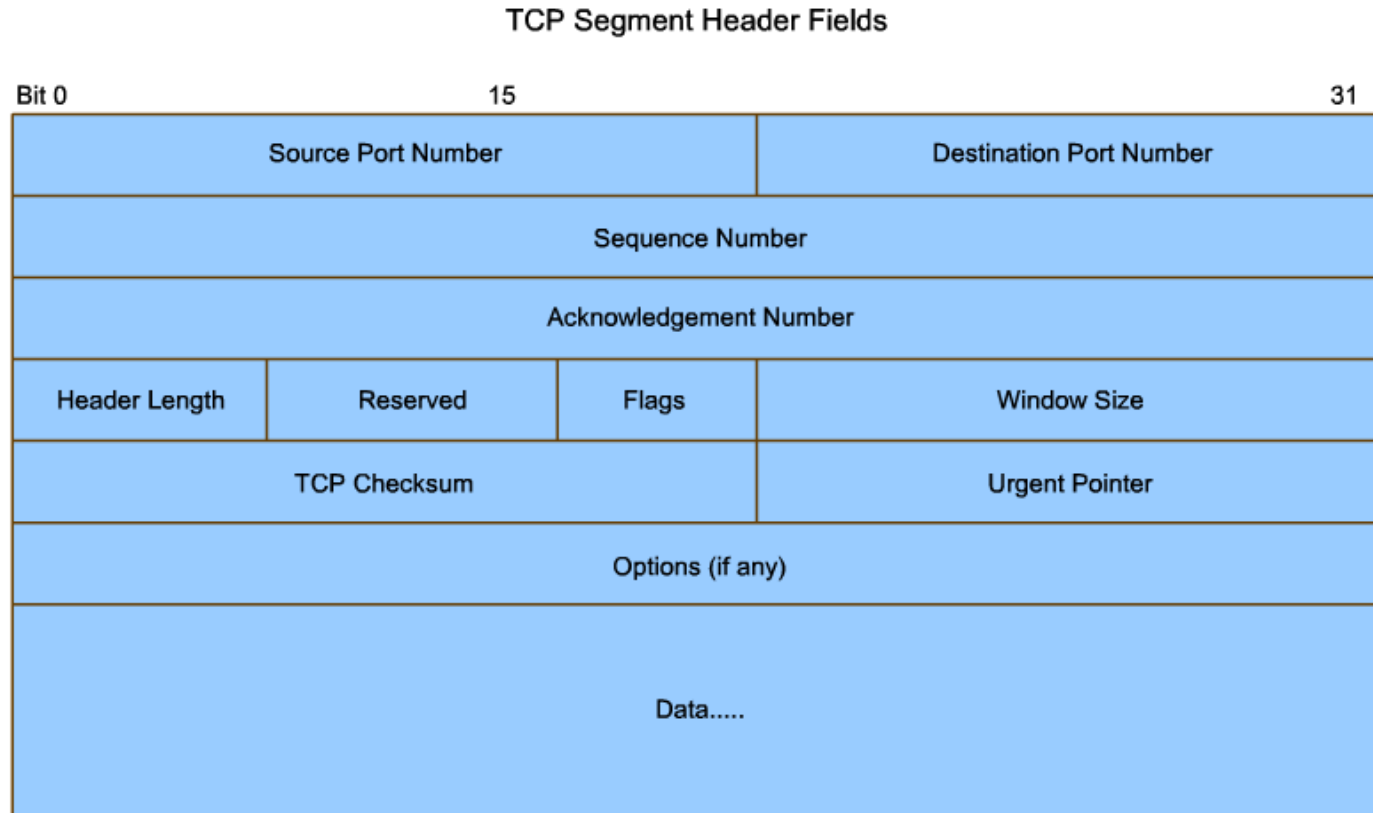
```
C:\>
```

Protocol used

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

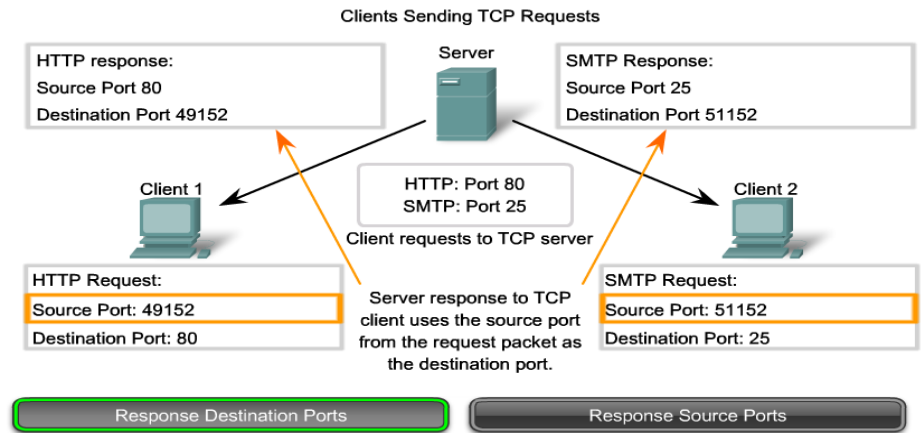
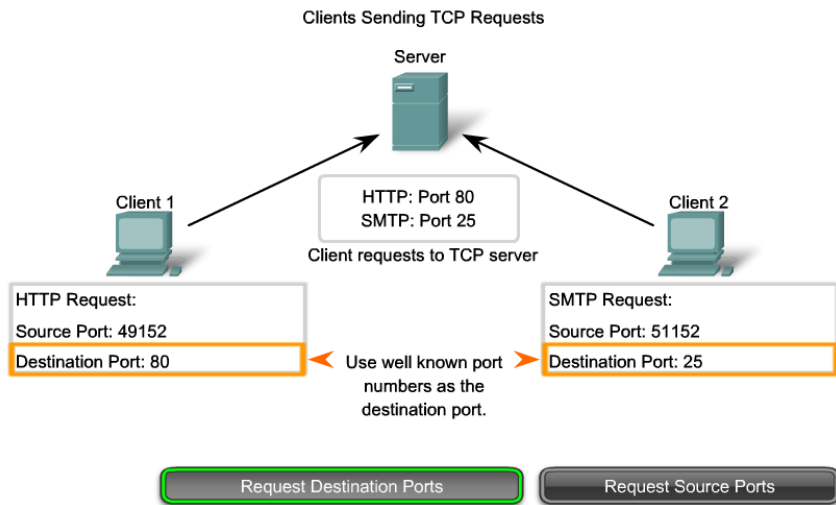


TCP Making Conversations Reliable



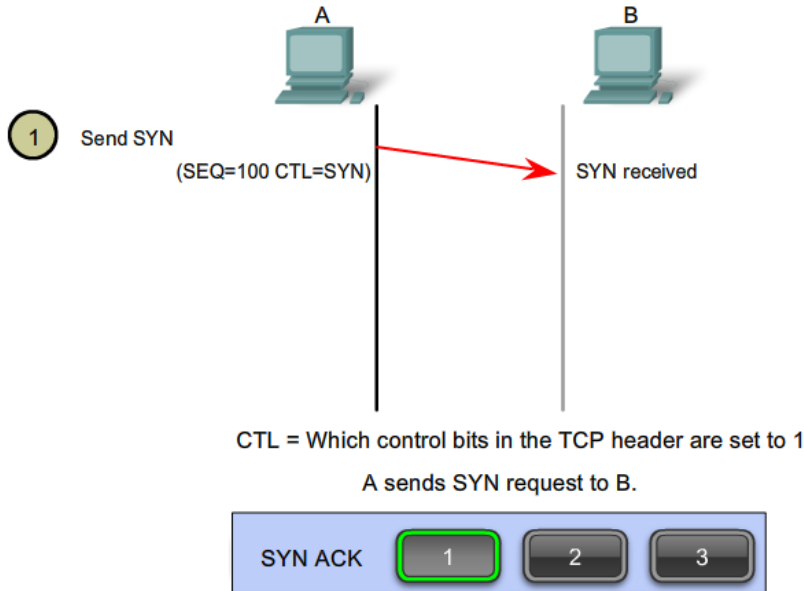
The fields of the TCP header enable TCP to provide connection-oriented, reliable data communications.

TCP server Processes

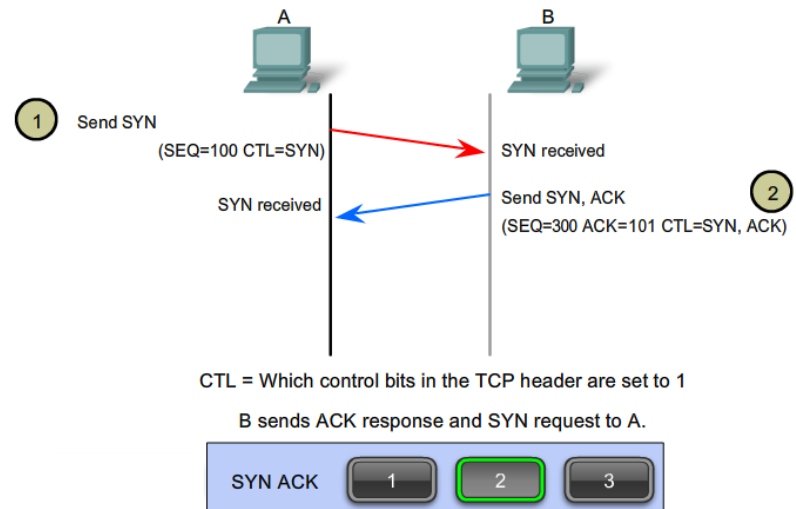


TCP connection Establishment and Termination

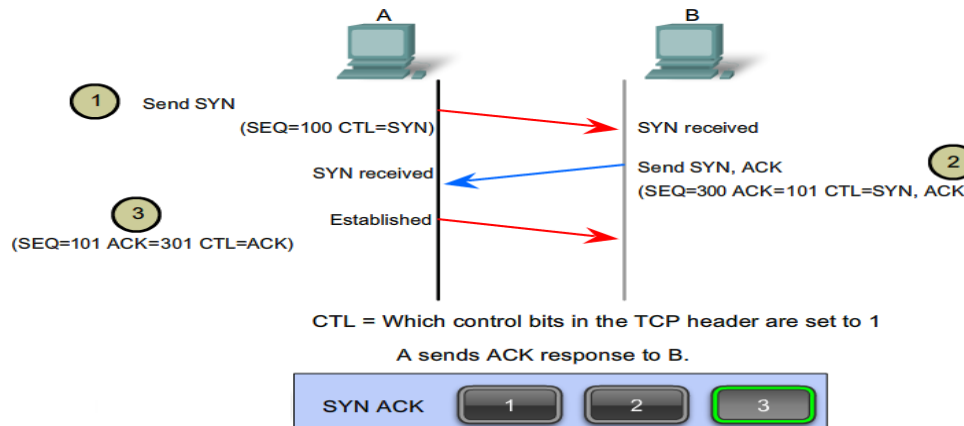
TCP Connection Establishment and Termination



TCP Connection Establishment and Termination

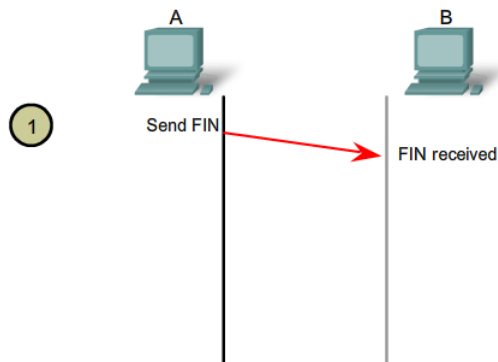


TCP Connection Establishment and Termination



TCP connection Establishment and Termination....

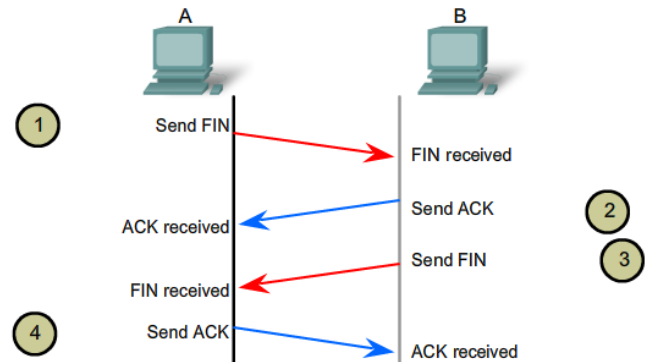
TCP Connection Establishment and Termination



A sends FIN request to B.

FIN ACK 1 2 3 4

TCP Connection Establishment and Termination



A sends ACK response to B.

FIN ACK 1 2 3 4

TCP 3-way Handshake – step 1

TCP 3-way Handshake (SYN)

13	6.201109	192.168.254.254	10.1.1.1	DNS	Standard query r
14	6.202100	10.1.1.1	192.168.254.254	TCP	1069 > http [SYN
15	6.202513	192.168.254.254	10.1.1.1	TCP	http > 1069 [SYN
16	6.202543	10.1.1.1	192.168.254.254	TCP	1069 > http [ACK
17	6.202651	10.1.1.1	192.168.254.254	HTTP	GET / HTTP/1.1

⊕ Frame 14 (62 bytes on wire, 62 bytes captured)

⊕ Ethernet II, Src: quantaCo_bd:0c:7c (00:c0:9f:bd:0c:7c), Dst: Cisco_cf:66:40

⊕ Internet Protocol, Src: 10.1.1.1 (10.1.1.1), Dst: 192.168.254.254 (192.168.2

⊖ Transmission Control Protocol, Src Port: 1069 (1069), Dst Port: http (80), s

- Source port: 1069 (1069)
- Destination port: http (80)
- Sequence number: 0 (relative sequence number)
- Header length: 28 bytes
- ⊖ Flags: 0x02 (SYN)
 - 0... = Congestion window Reduced (CWR): Not set
 - .0.. = ECN-Echo: Not set

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)

Time	Source	Destination	Protocol	Length	Info
6.201109	192.168.254.254	10.1.1.1	DNS	Standard query	
6.202100	10.1.1.1	192.168.254.254	TCP	1069 > http [S]	
6.202513	192.168.254.254	10.1.1.1	TCP	http > 1069 [S]	
6.202543	10.1.1.1	192.168.254.254	TCP	1069 > http [A]	
6.202651	10.1.1.1	192.168.254.254	HTTP	GET / HTTP/1.1	

Frame 15 (62 bytes on wire, 62 bytes captured)

Ethernet II, Src: Cisco_cf:66:40 (00:0c:85:cf:66:40), Dst: quantaCo_bd:0c:...

Internet Protocol, Src: 192.168.254.254 (192.168.254.254), Dst: 10.1.1.1 (10.1.1.1)

Transmission Control Protocol, Src Port: http (80), Dst Port: 1069 (1069),

- Source port: http (80)
- Destination port: 1069 (1069)
- Sequence number: 0 (relative sequence number)
- Acknowledgement number: 1 (relative ack number)
- Header length: 28 bytes
- Flags: 0x12 (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)

13	6.201109	192.168.254.254	10.1.1.1	DNS	Standard query re
14	6.202100	10.1.1.1	192.168.254.254	TCP	1069 > http [SYN]
15	6.202513	192.168.254.254	10.1.1.1	TCP	http > 1069 [SYN,
16	6.202543	10.1.1.1	192.168.254.254	TCP	1069 > http [ACK]
17	6.202651	10.1.1.1	192.168.254.254	HTTP	GET / HTTP/1.1

⊕ Frame 16 (54 bytes on wire, 54 bytes captured)

⊕ Ethernet II, Src: Quantaco_bd:0c:7c (00:c0:9f:bd:0c:7c), Dst: Cisco_cf:66:40

⊕ Internet Protocol, Src: 10.1.1.1 (10.1.1.1), Dst: 192.168.254.254 (192.168.254.254)

⊖ Transmission Control Protocol, Src Port: 1069 (1069), Dst Port: http (80), Seq: 1069, Win: 0, Len: 0

Source port: 1069 (1069)

Destination port: http (80)

Sequence number: 1 (relative sequence number)

Acknowledgement number: 1 (relative ack number)

Header length: 20 bytes

⊖ Flags: 0x10 (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)

The image shows a Wireshark packet capture window. The packet list pane shows five packets. Packet 20 is selected and highlighted in green. The packet details pane shows the structure of the selected packet: Ethernet II, Internet Protocol, and Transmission Control Protocol. The TCP section is expanded to show: Source port: http (80), Destination port: 1069 (1069), Sequence number: 440 (relative sequence number), Acknowledgement number: 414 (relative ack number), and Header length: 20 bytes.

No.	Time	Source	Destination	Protocol	Length	Info
19	6.203857	192.168.254.254	10.1.1.1	HTTP	200	OK
20	6.203876	192.168.254.254	10.1.1.1	TCP		http > 1069 [FIN, Seq=440]
21	6.203899	10.1.1.1	192.168.254.254	TCP		1069 > http [ACK, Seq=414]
22	6.204139	10.1.1.1	192.168.254.254	TCP		1069 > http [FIN, Seq=414]
23	6.204416	192.168.254.254	10.1.1.1	TCP		http > 1069 [ACK, Seq=415]

Frame 20 (60 bytes on wire, 60 bytes captured)
Ethernet II, Src: Cisco_cf:66:40 (00:0c:85:cf:66:40), Dst: quantaCo_bd:0c:7c
Internet Protocol, Src: 192.168.254.254 (192.168.254.254), Dst: 10.1.1.1 (10.1.1.1)
Transmission Control Protocol, Src Port: http (80), Dst Port: 1069 (1069), Seq: 440 (relative sequence number)
Source port: http (80)
Destination port: 1069 (1069)
Sequence number: 440 (relative sequence number)
Acknowledgement number: 414 (relative ack number)
Header length: 20 bytes

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)

The image shows a Wireshark packet capture window. The packet list pane shows five packets. Packet 21 is selected and highlighted in green. The packet details pane shows the structure of the selected packet: Ethernet II, Internet Protocol, and Transmission Control Protocol. The TCP section is expanded to show: Source port: 1069 (1069), Destination port: http (80), Sequence number: 414 (relative sequence number), Acknowledgement number: 441 (relative ack number), and Header length: 20 bytes.

No.	Time	Source	Destination	Protocol	Length	Info
19	6.203857	192.168.254.254	10.1.1.1	HTTP	200	OK
20	6.203876	192.168.254.254	10.1.1.1	TCP		http > 1069 [FIN, Seq=440]
21	6.203899	10.1.1.1	192.168.254.254	TCP		1069 > http [ACK, Seq=414]
22	6.204139	10.1.1.1	192.168.254.254	TCP		1069 > http [FIN, Seq=414]
23	6.204416	192.168.254.254	10.1.1.1	TCP		http > 1069 [ACK, Seq=415]

Frame 21 (54 bytes on wire, 54 bytes captured)
Ethernet II, Src: quantaCo_bd:0c:7c (00:c0:9f:bd:0c:7c), Dst: Cisco_cf:66:40
Internet Protocol, Src: 10.1.1.1 (10.1.1.1), Dst: 192.168.254.254 (192.168.254.254)
Transmission Control Protocol, Src Port: 1069 (1069), Dst Port: http (80), Seq: 414 (relative sequence number)
Source port: 1069 (1069)
Destination port: http (80)
Sequence number: 414 (relative sequence number)
Acknowledgement number: 441 (relative ack number)
Header length: 20 bytes

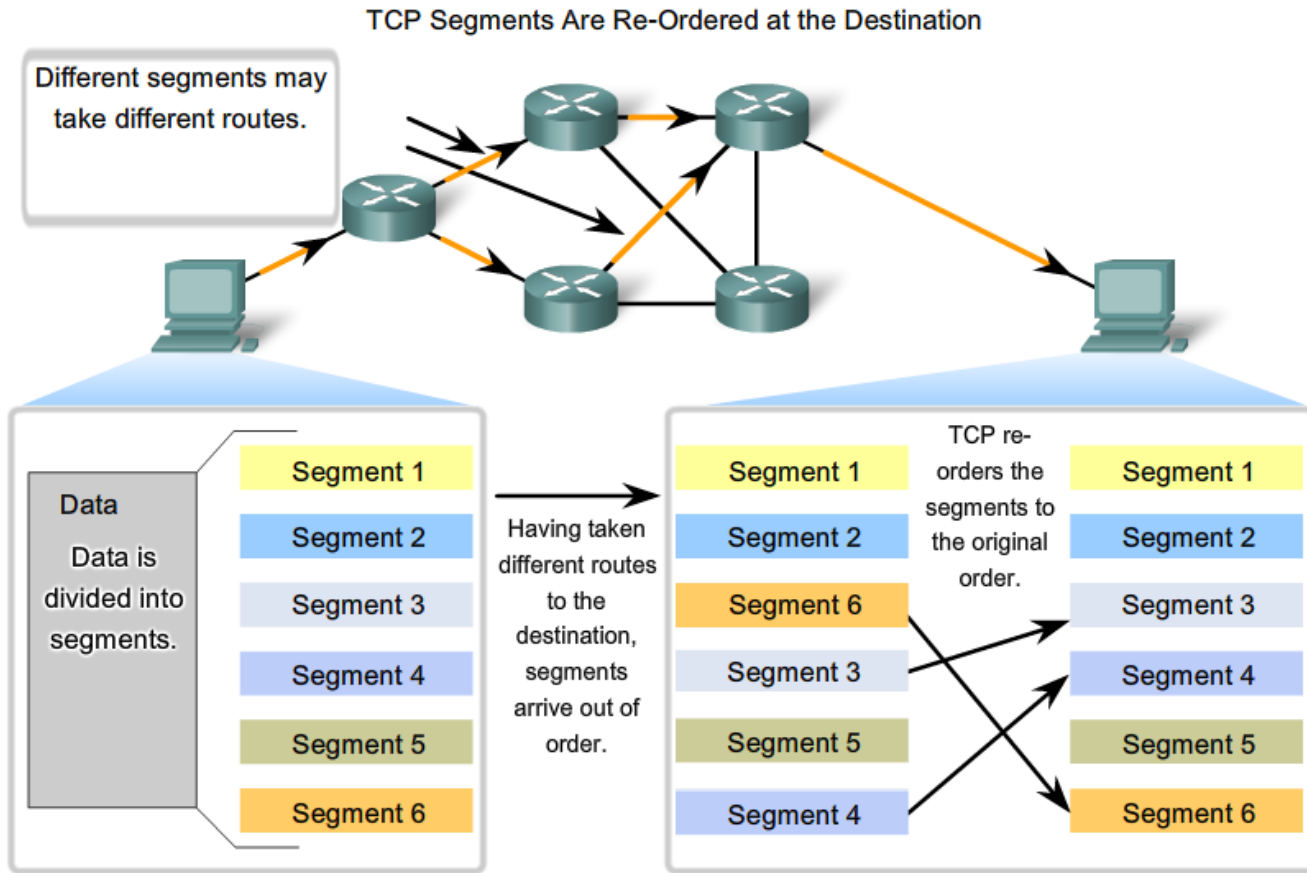
A protocol analyzer shows details of frame 21, TCP ACK response.

The destination and source ports
The header field contents and values

FIN

ACK

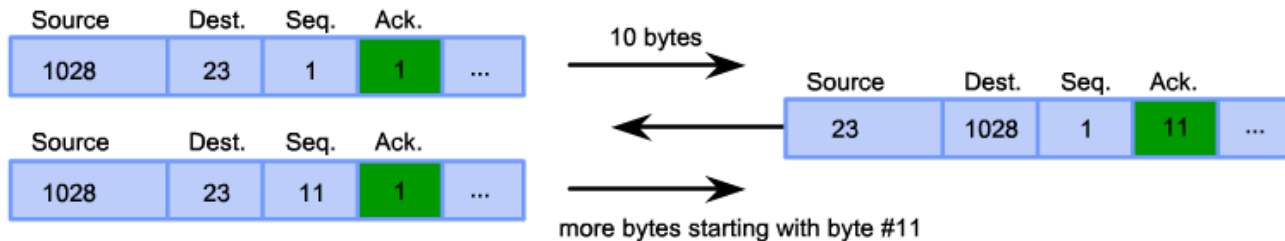
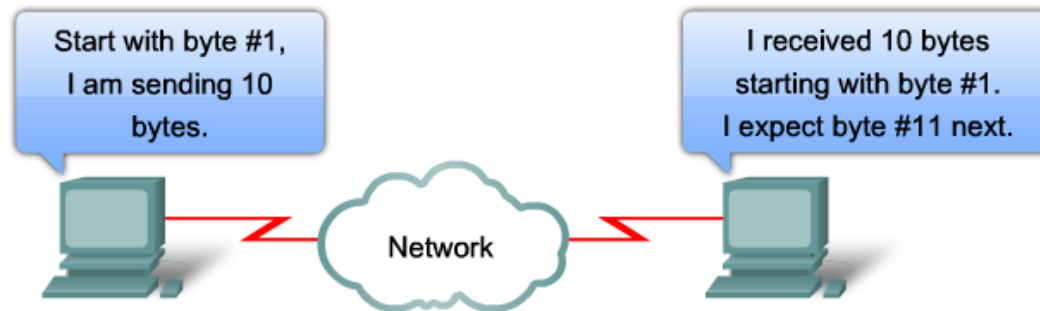
TCP Segments Reassembly



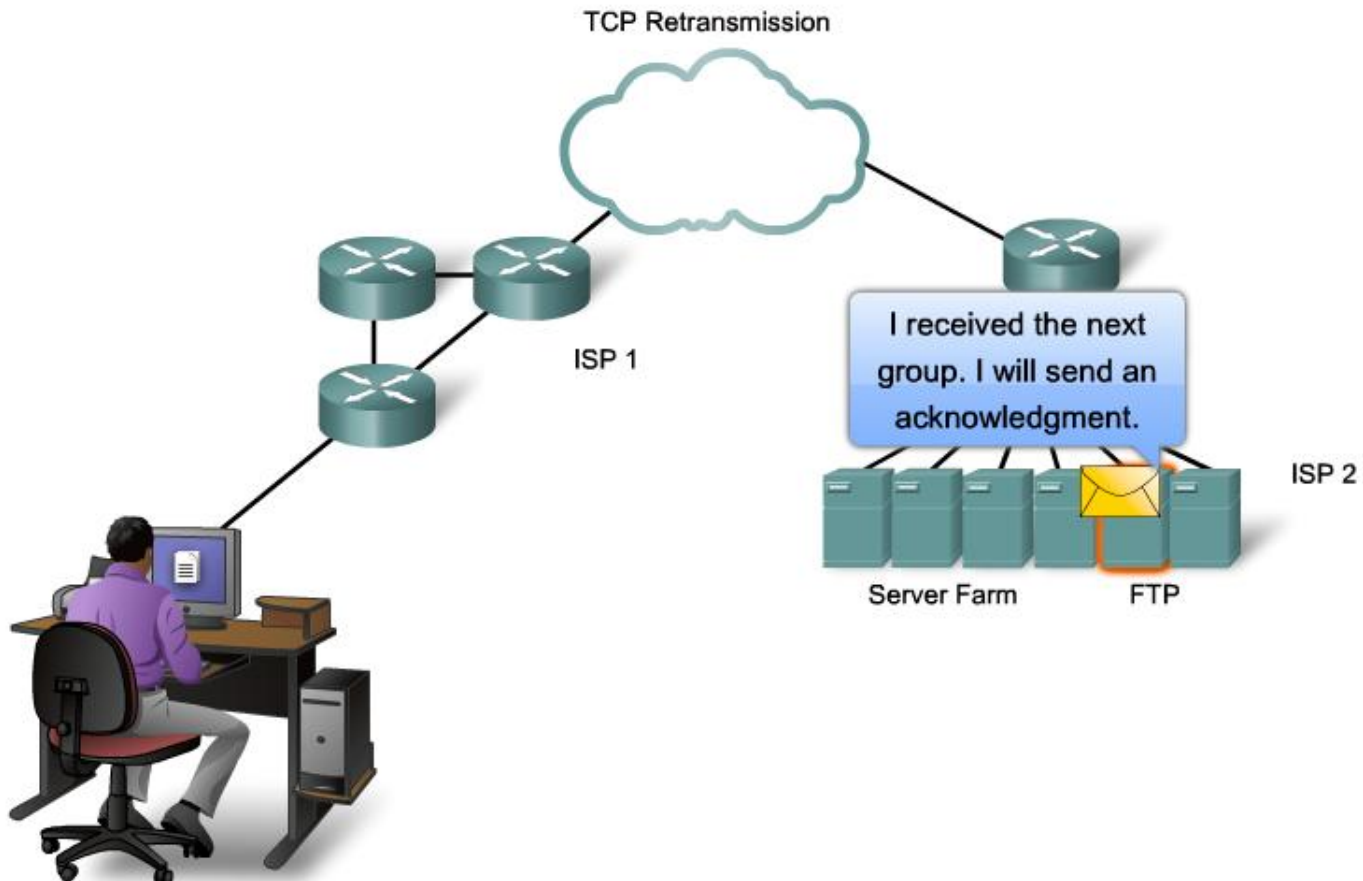
TCP Acknowledgment with Windowing

Acknowledgement of TCP Segments

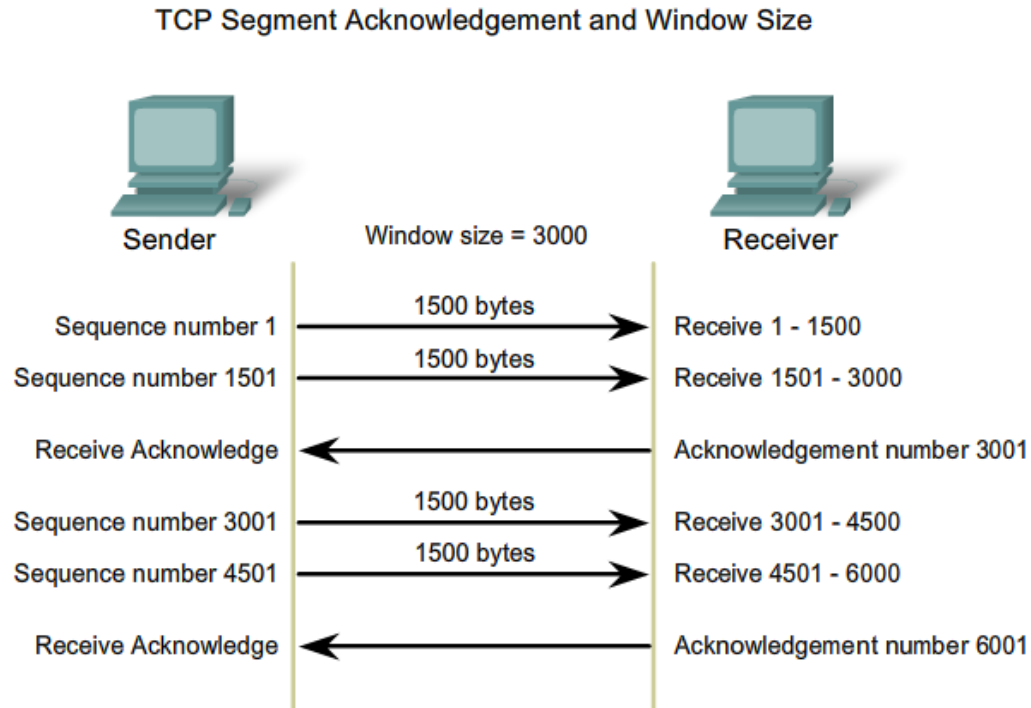
Source Port	Destination Port	Sequence Number	Acknowledgement Numbers	...



TCP Retransmission



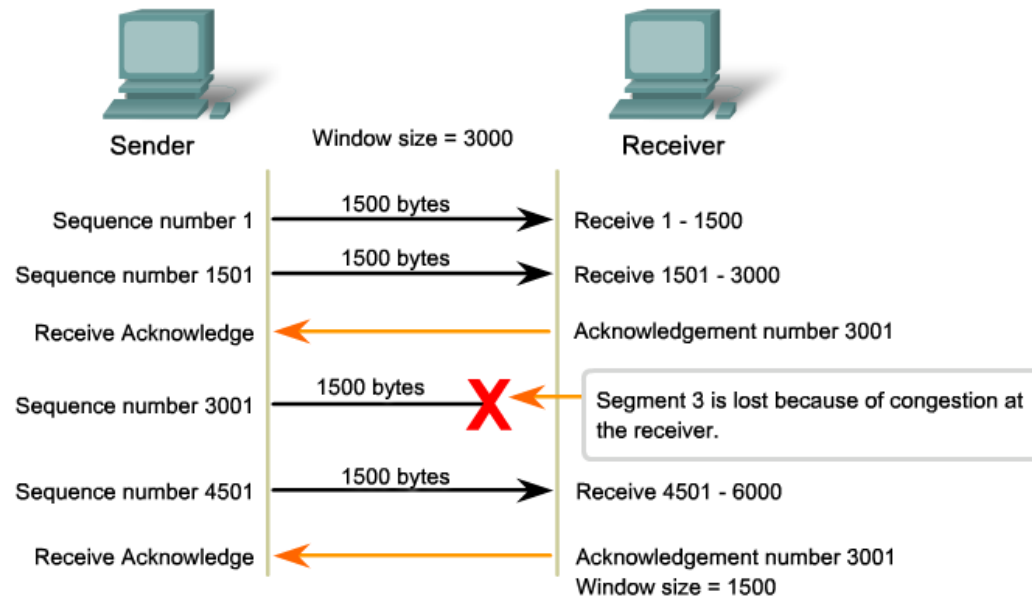
TCP Congestion Control- Minimize Segment Loss



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...

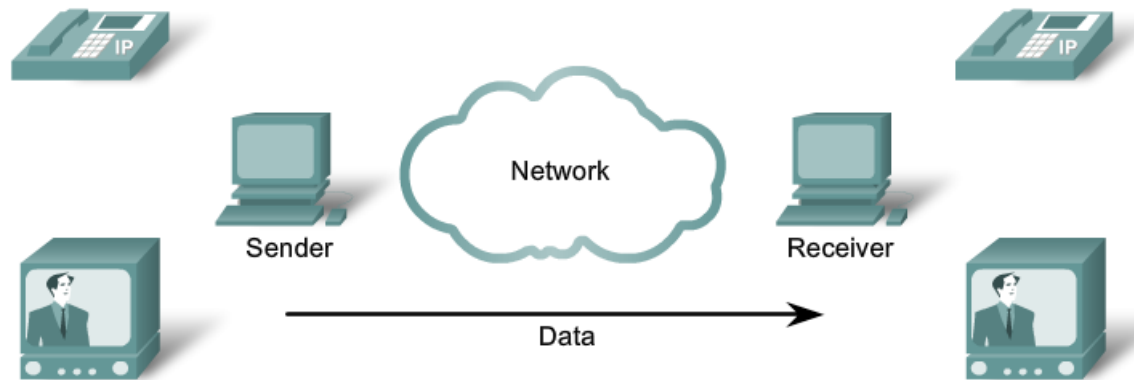
TCP Congestion and Flow Control



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

UDP

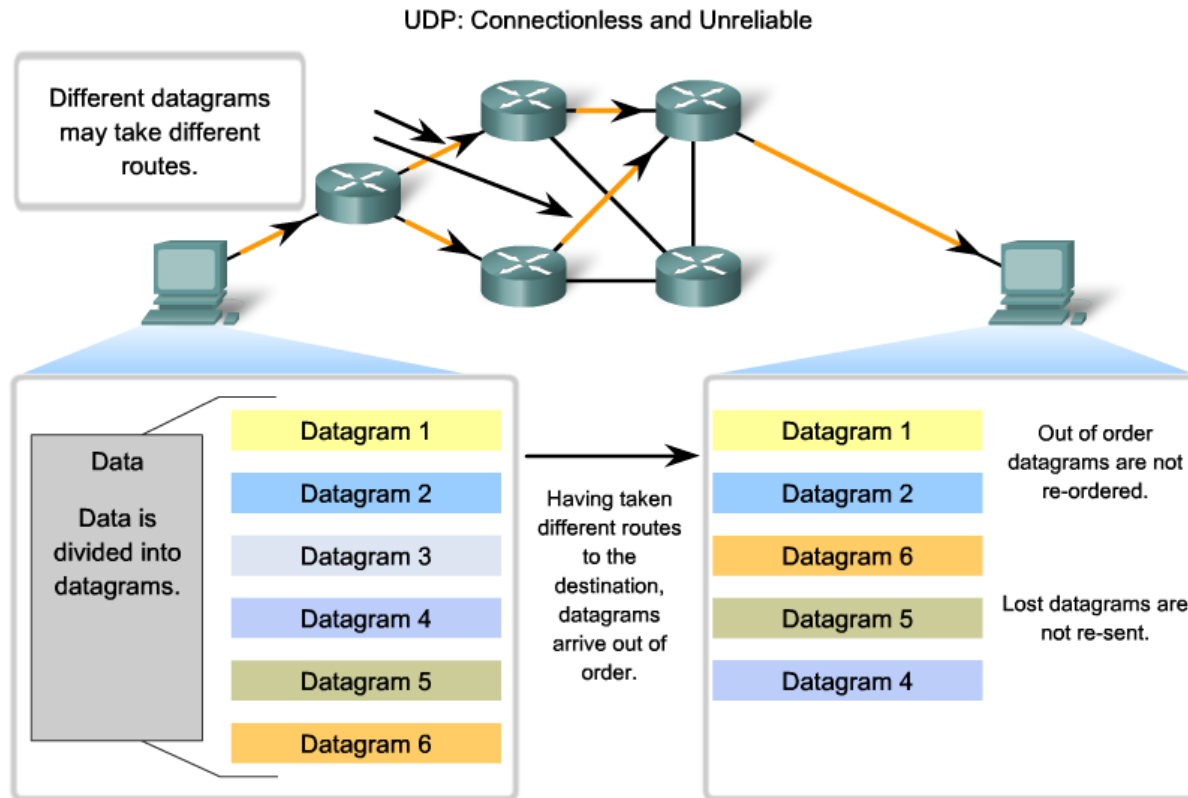
UDP Low Overhead Data Transport



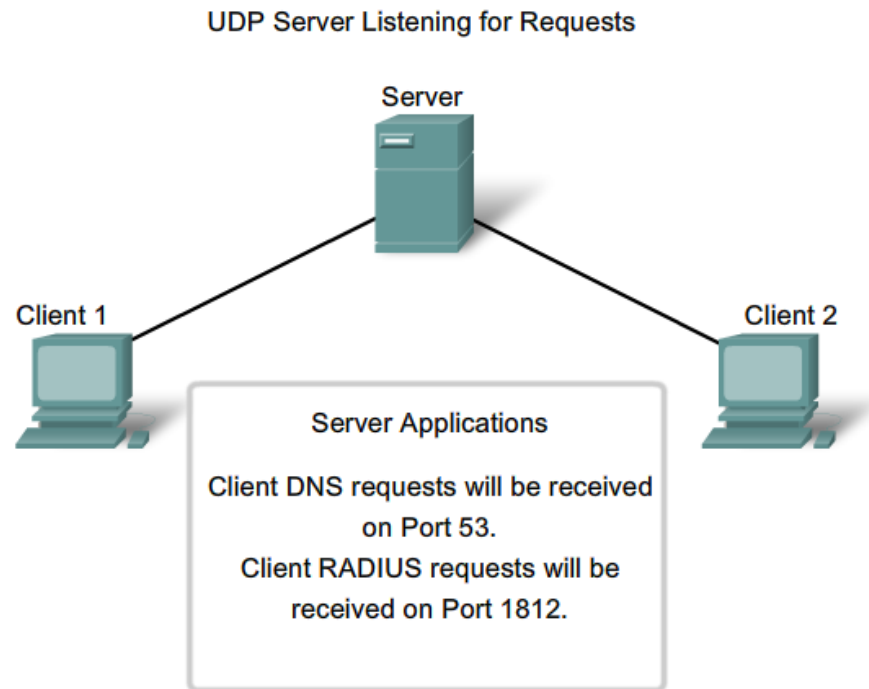
UDP does not establish a connection
before sending data.

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

UDP Datagram Reassembly

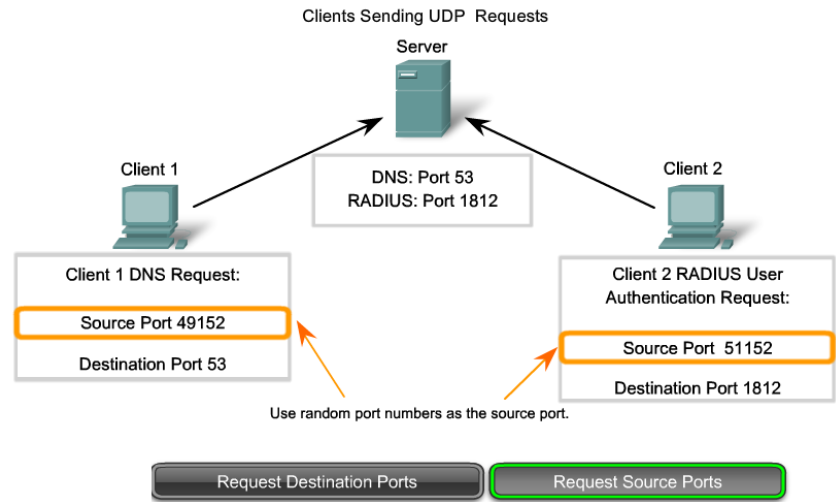
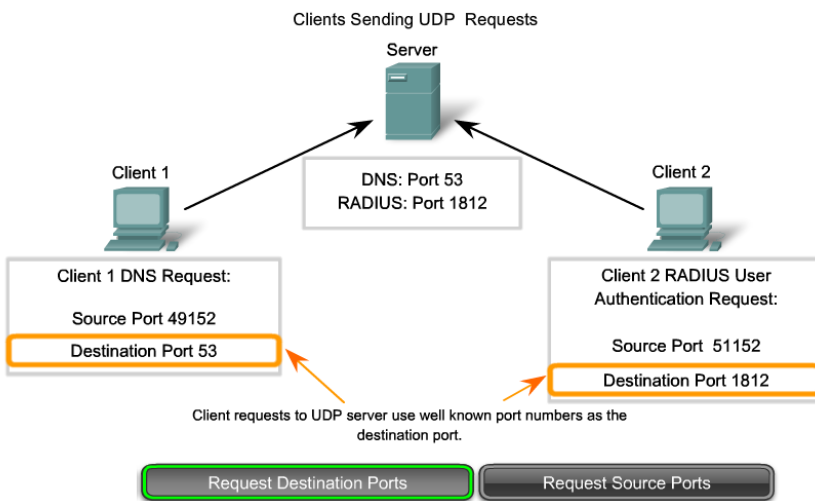


UDP server Processes and Requests

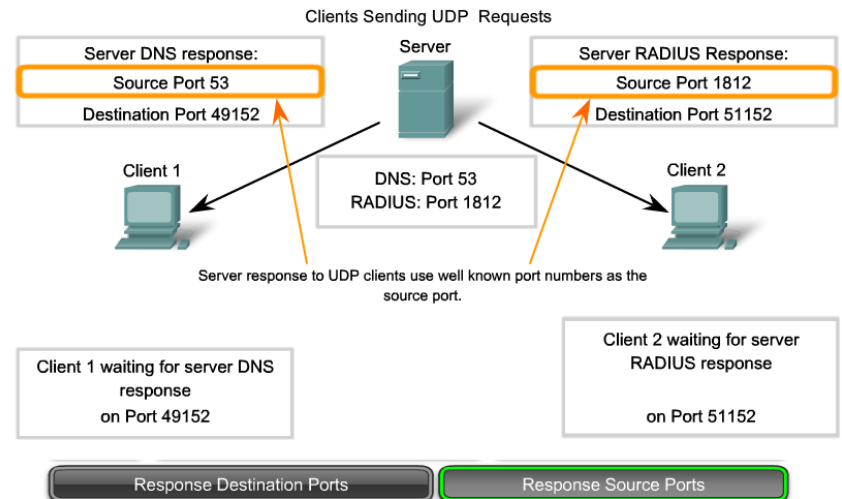
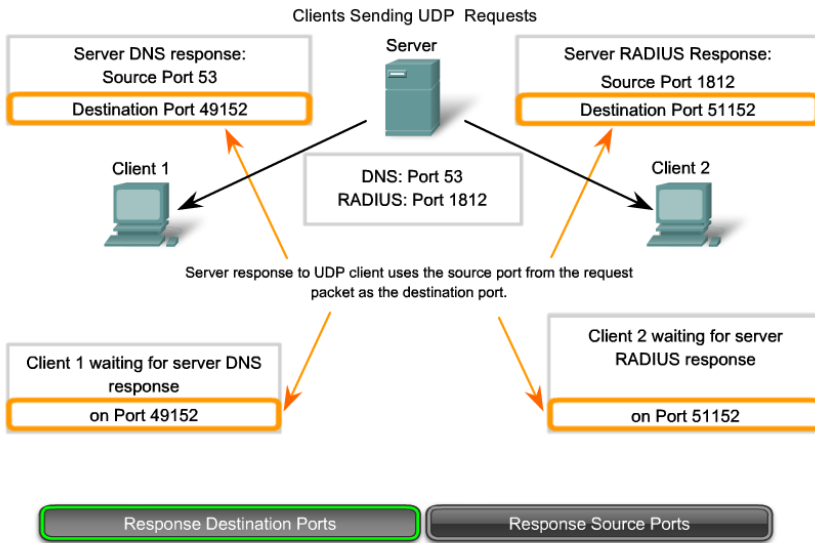


Client requests to servers have well known ports numbers as the destination port.

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