Objectives

- Explain the need for the Transport layer.
- Identify the role of the Transport layer as it provides the end-to-end transfer of data between applications.
- Describe the role of two TCP/IP Transport layer protocols: TCP and UDP.
- Explain the key functions of the Transport layer, including reliability, port addressing, and segmentation.

- Explain how TCP and UDP each handle key functions.
- Identify when it is appropriate to use TCP or UDP and provide examples of applications that use each protocol.

Transport Layer Layer 4 (Segments)

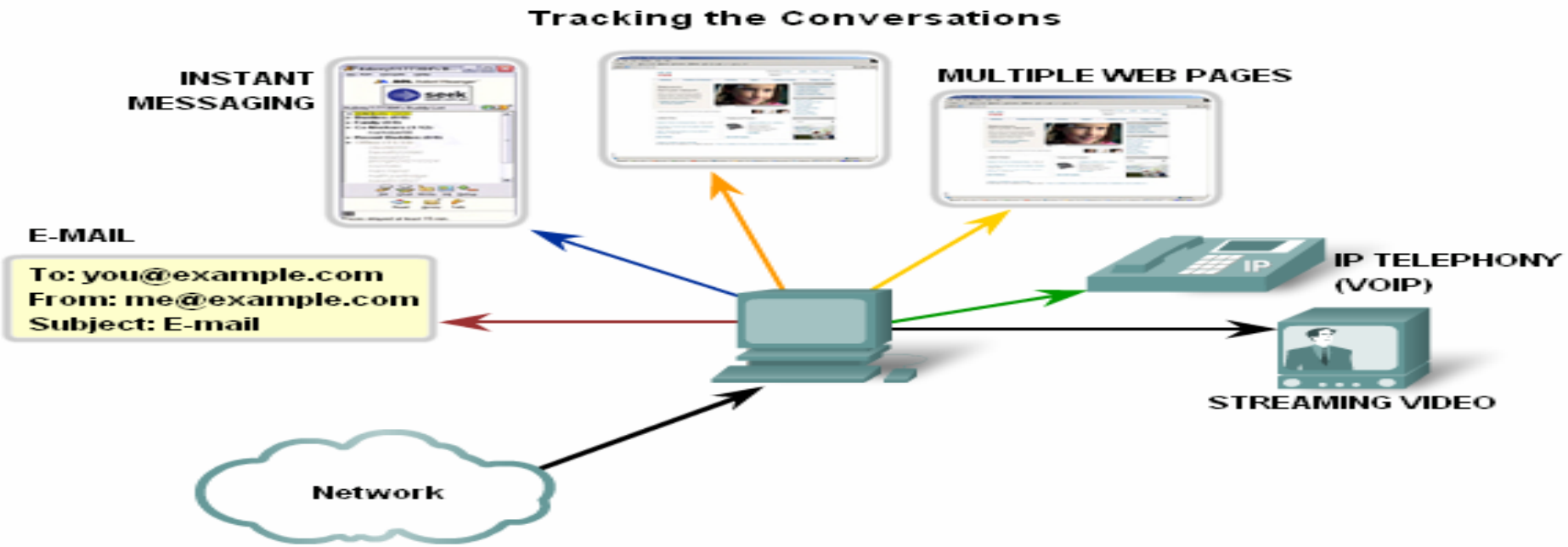
The Transport layer provides for the segmentation of data and the control necessary to reassemble these pieces into the various communication streams. Its primary responsibilities to accomplish this are:

- Tracking the individual communication between applications on the source and destination hosts.
- Segmenting data and managing each piece.
- Reassembling the segments into streams of application data
- Identifying the different applications.

Transport Layer Layer 4 (Segments)

Two main protocols are worked in this layer TCP and UDP

1-TCP : Transmission Control Protocol
2-UDP : User Datagram Protocol



The Transport layer divides the data into segments that are easier to manage and transport.

- **Tracking Individual Conversations**

Any host may have multiple applications that are communicating across the network. Each of these applications will be communicating with one or more applications on remote hosts. It is the responsibility of the Transport layer to maintain the multiple communication streams between these applications.

- **Segmenting Data**

As each application creates a stream data to be sent to a remote application, this data must be prepared to be sent across the media in manageable pieces. The Transport layer protocols describe services that segment this data from the Application layer. This includes the encapsulation required on each piece of data. Each piece of application data requires headers to be added at the Transport layer to indicate to which communication it is associated.

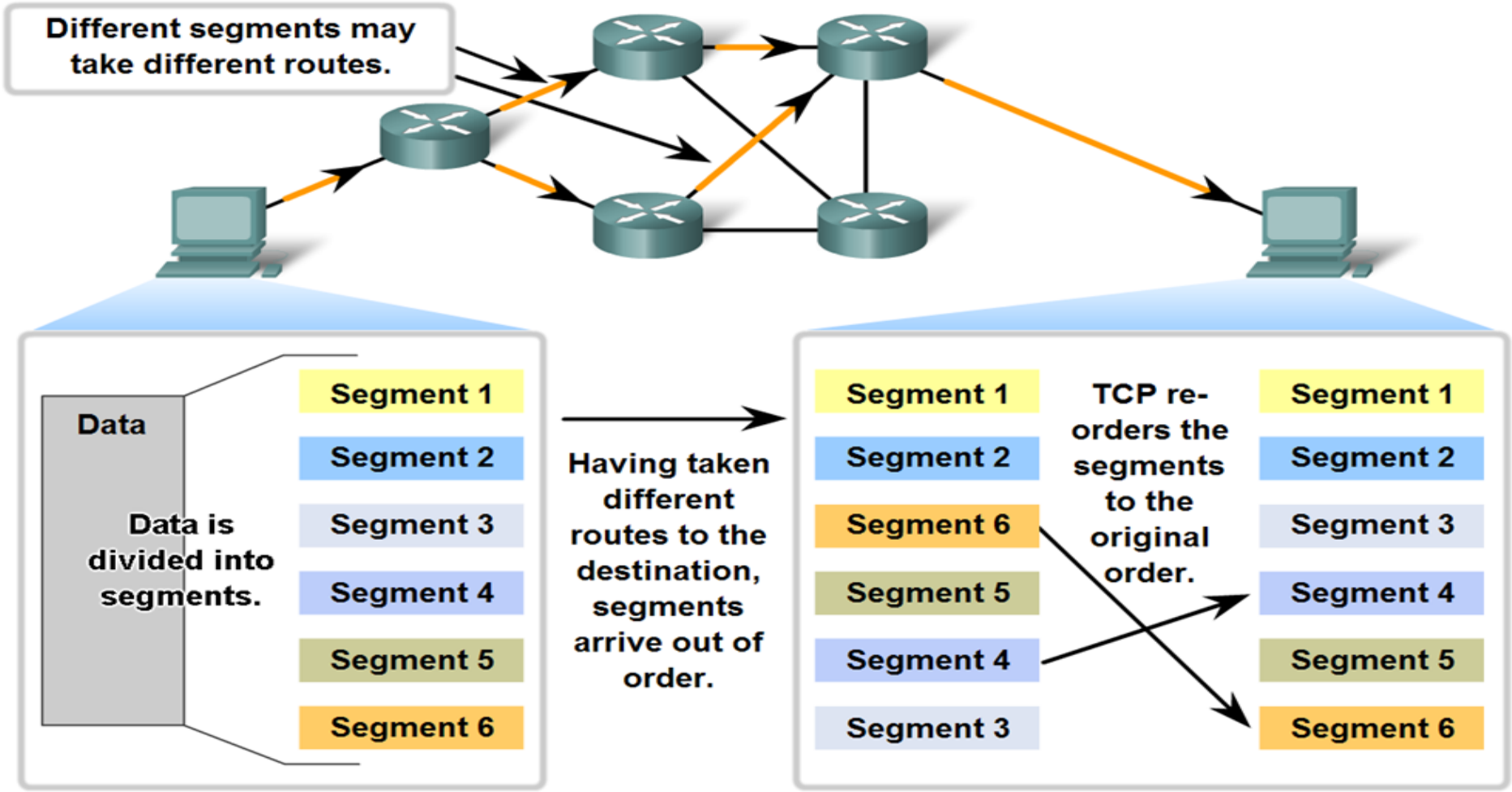
- **Reassembling Segments**

At the receiving host, each piece of data may be directed to the appropriate application. Additionally, these individual pieces of data must also be reconstructed into a complete data stream that is useful to the Application layer. The protocols at the Transport layer describe the how the Transport layer header information is used to reassemble the data pieces into streams to be passed to the Application layer.

- **Identifying the Applications**

In order to pass data streams to the proper applications, the Transport layer must identify the target application. To accomplish this, the Transport layer assigns an application an identifier. The TCP/IP protocols call this identifier a port number. Each software process that needs to access the network is assigned a port number unique in that host. This port number is used in the transport layer header to indicate to which application that piece of data is associated

TCP Segments Are Re-Ordered at the Destination



Port Addressing

The Internet Assigned Numbers Authority (IANA) assigns port numbers. IANA is a standards body that is responsible for assigning various addressing standards.
There are different types of port numbers:

1-Well Known Ports (Numbers 0 to 1023) - These numbers are reserved for services and applications.

2-Registered Ports (Numbers 1024 to 49151) - These port numbers are assigned to user processes or applications. These processes are primarily individual applications that a user has chosen to install rather than common applications that would receive a Well Known Port. When not used for a server resource, these ports may also be used dynamically selected by a client as its source port.

3-Dynamic or Private Ports (Numbers 49152 to 65535) - Also known as Ephemeral Ports, these are usually assigned dynamically to client applications when initiating a connection.

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
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 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 UDP Ports:
1812 RADIUS Authentication Protocol
2000 Cisco SCCP (VoIP)
5004 RTP (Voice and Video Transport Protocol)
5060 SIP (VoIP)

Well Known UDP Ports:
69 TFTP
520 RIP

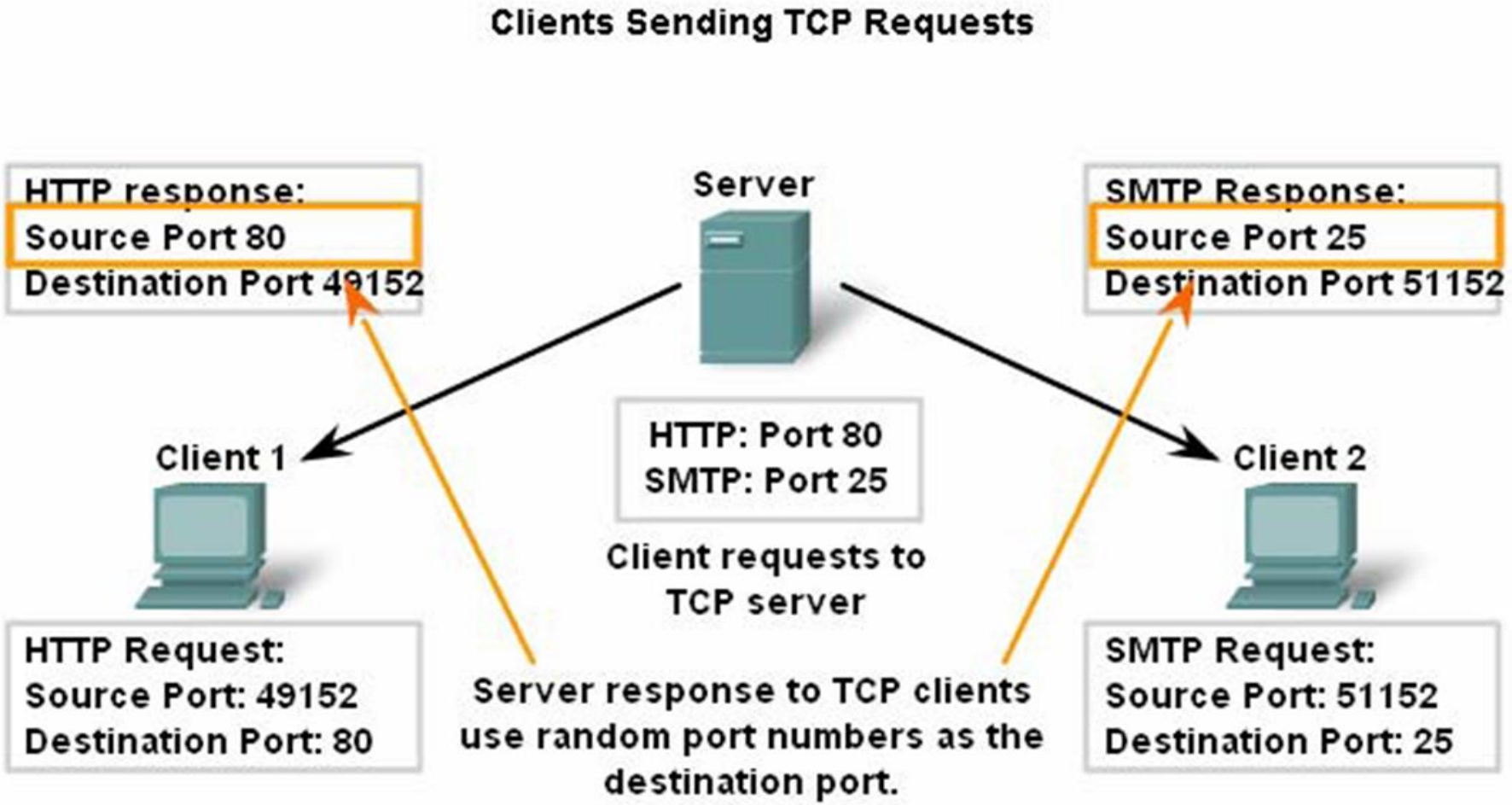
Port Addressing

Port Numbers

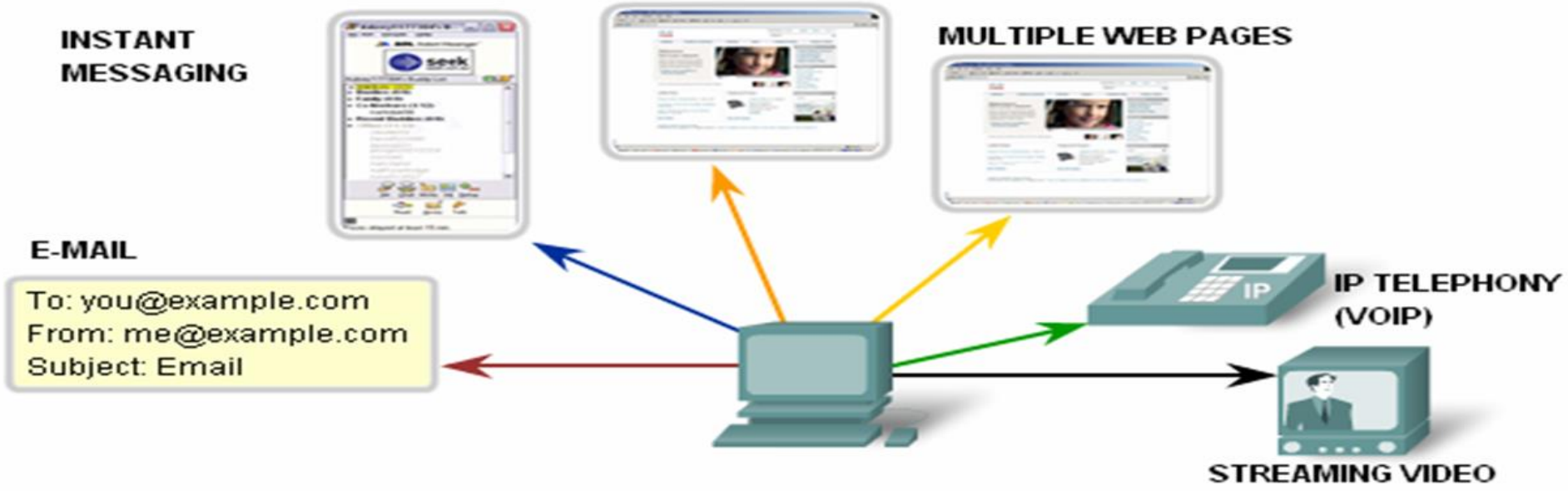
Port Number Range	Port Group
0 to 1023	Well Known (Contact) Ports
1024 to 49151	Registered Ports
49152 to 65535	Private and/or Dynamic Ports

Registered TCP/UDP Common Ports:
1433 MS SQL
2948 WAP (MMS)

Well Known TCP/UDP Common Ports:
53 DNS
161 SNMP
531 AOL Instant Messenger, IRC


Clients Sending TCP Requests
HTTP response:
Source Port 80
Destination Port 49152
Server
SMTP Response:
Source Port 25
Destination Port 51152
HTTP: Port 80
SMTP: Port 25
Client 1
Client 2
Client requests to
TCP server
HTTP Request:
Source Port: 49152
Destination Port: 80
Server response to TCP clients
use random port numbers as the
destination port.
SMTP Request:
Source Port: 51152
Destination Port: 25

Transport Layer Services



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 data is delivered sequentially as it was sent.

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

TCP Connection Establishment and Termination

TCP Three-Way Handshake

Step 1

A TCP client begins the three-way handshake by sending a segment with the SYN (Synchronize Sequence Number) control flag set, indicating an initial value in the sequence number field in the header. This initial value for the sequence number, known as the Initial Sequence Number (ISN), is randomly chosen and is used to begin tracking the flow of data from the client to the server for this session. The ISN in the header of each segment is increased by one for each byte of data sent from the client to the server as the data conversation continues.

TCP Connection Establishment and Termination

TCP Three-Way Handshake

Step 2

The TCP server needs to acknowledge the receipt of the SYN segment from the client to establish the session from the client to the server. To do so, the server sends a segment back to the client with the ACK flag set indicating that the Acknowledgment number is significant. With this flag set in the segment, the client recognizes this as an acknowledgement that the server received the SYN from the TCP client.

The value of the acknowledgment number field is equal to the client initial sequence number plus 1.

TCP Connection Establishment and Termination

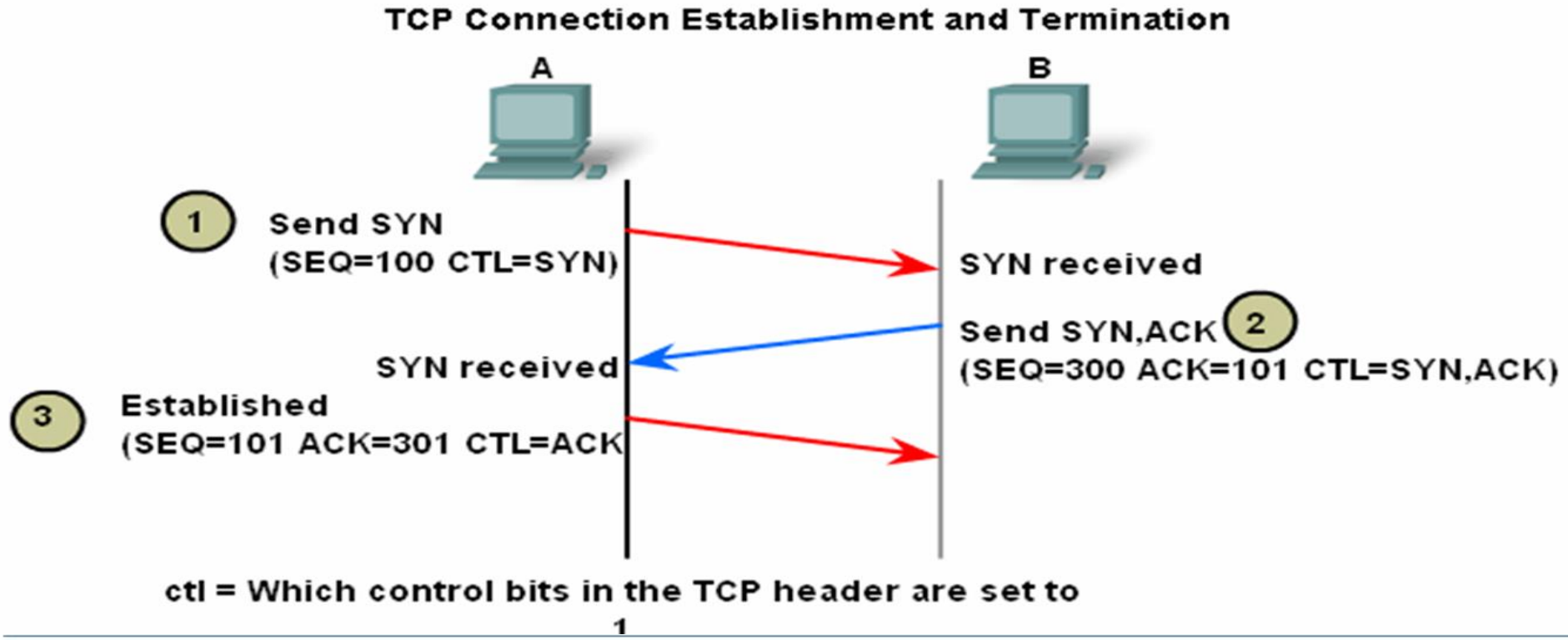
TCP Three-Way Handshake

Step 3

Finally, the TCP client responds with a segment containing an ACK that is the response to the TCP SYN sent by the server. There is no user data in this segment.

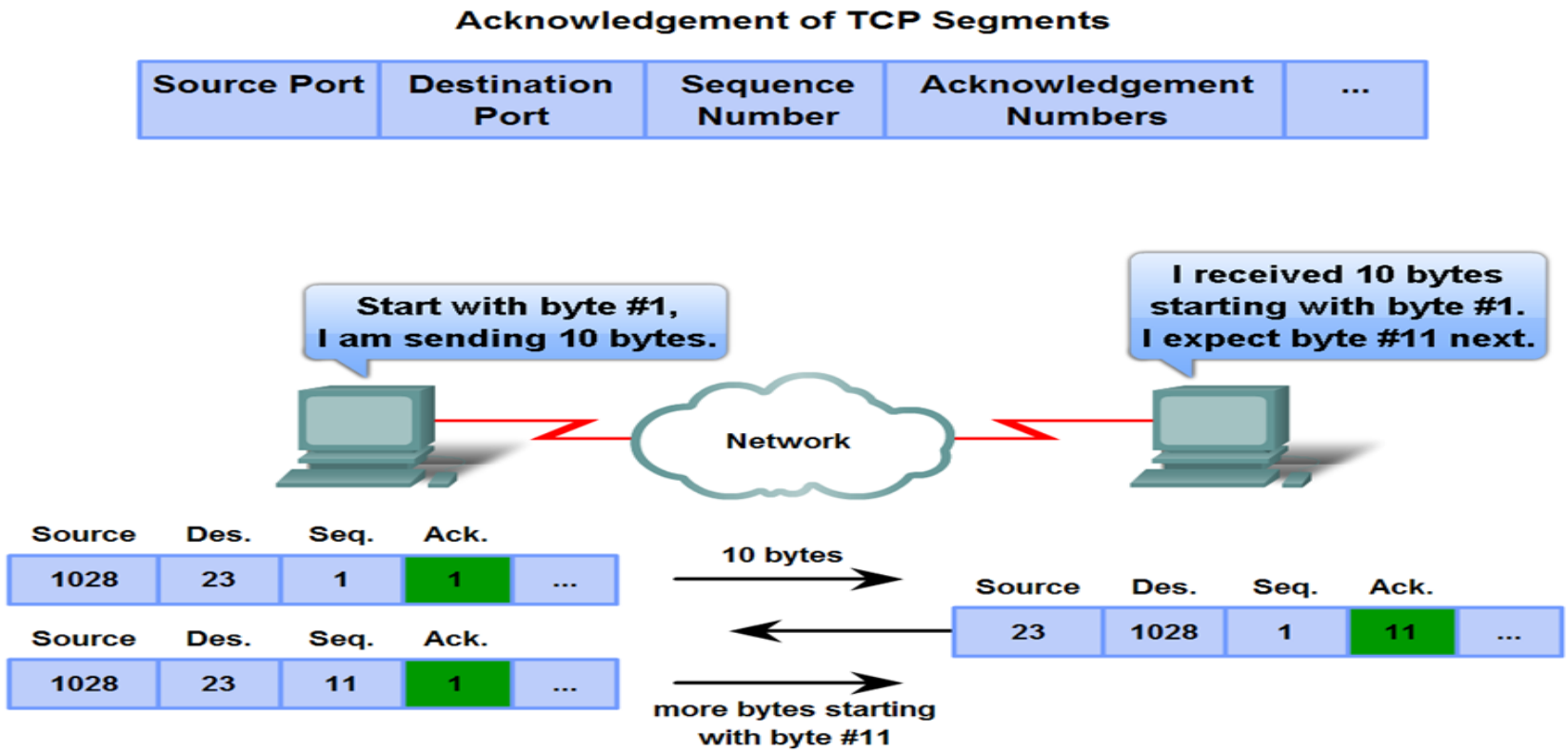
TCP Connection Establishment and Termination

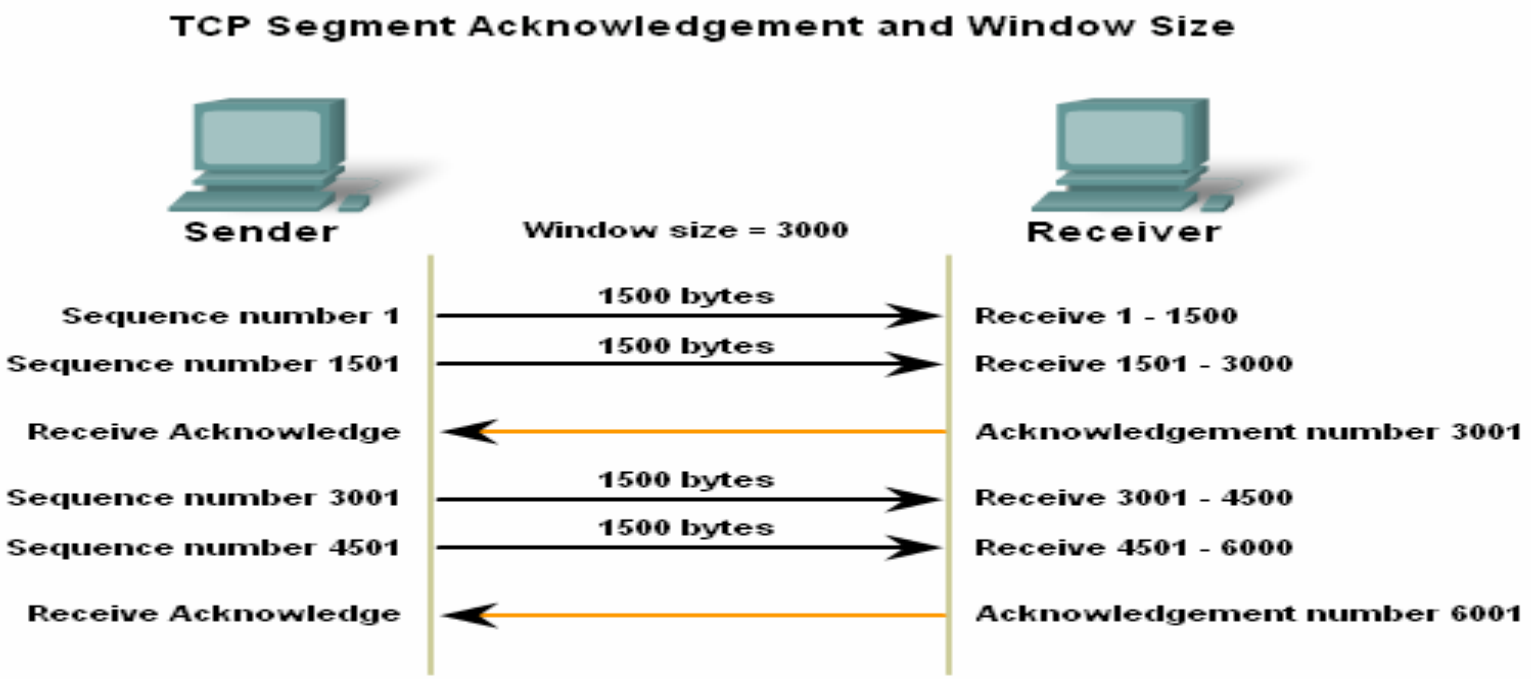
TCP Three-Way Handshake



Managing TCP Sessions

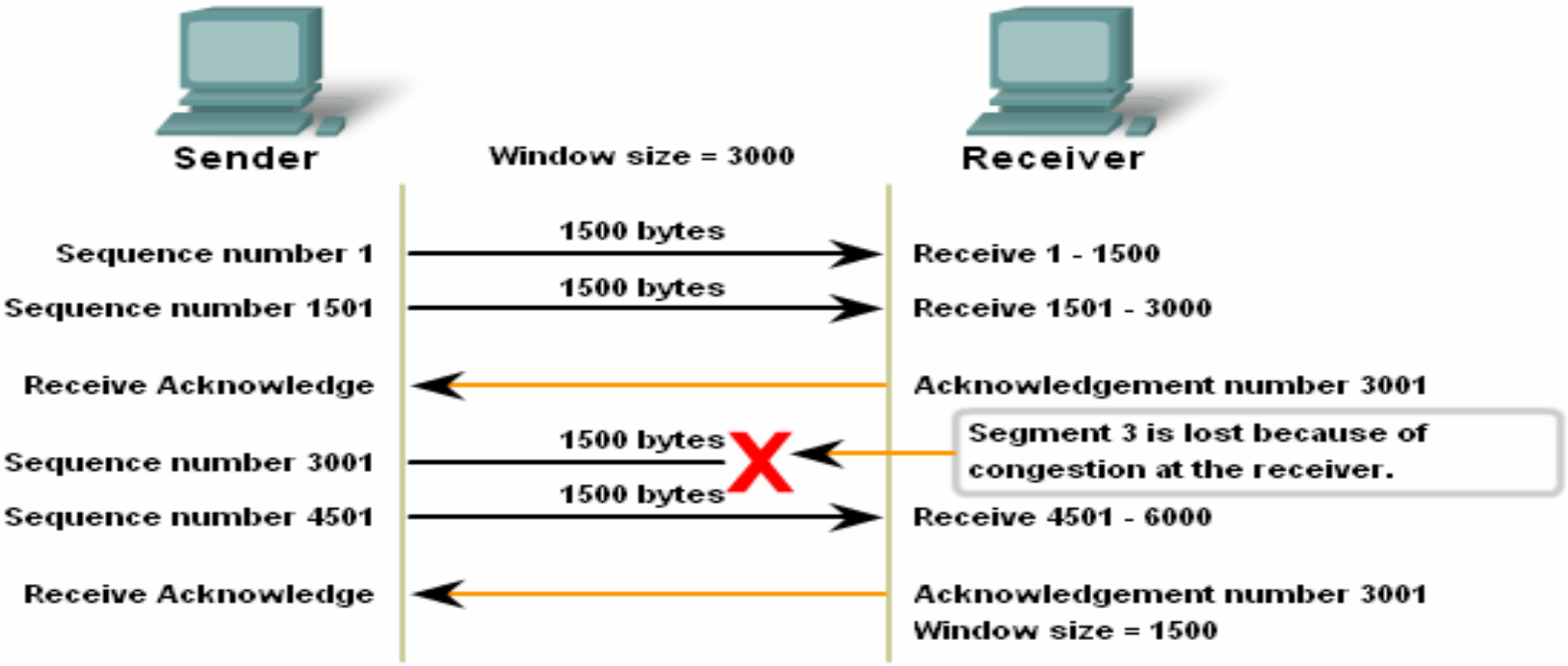
- Trace the steps used by the TCP protocol in which sequence numbers and acknowledgement numbers are used to manage exchanges in a conversation





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 and Flow Control

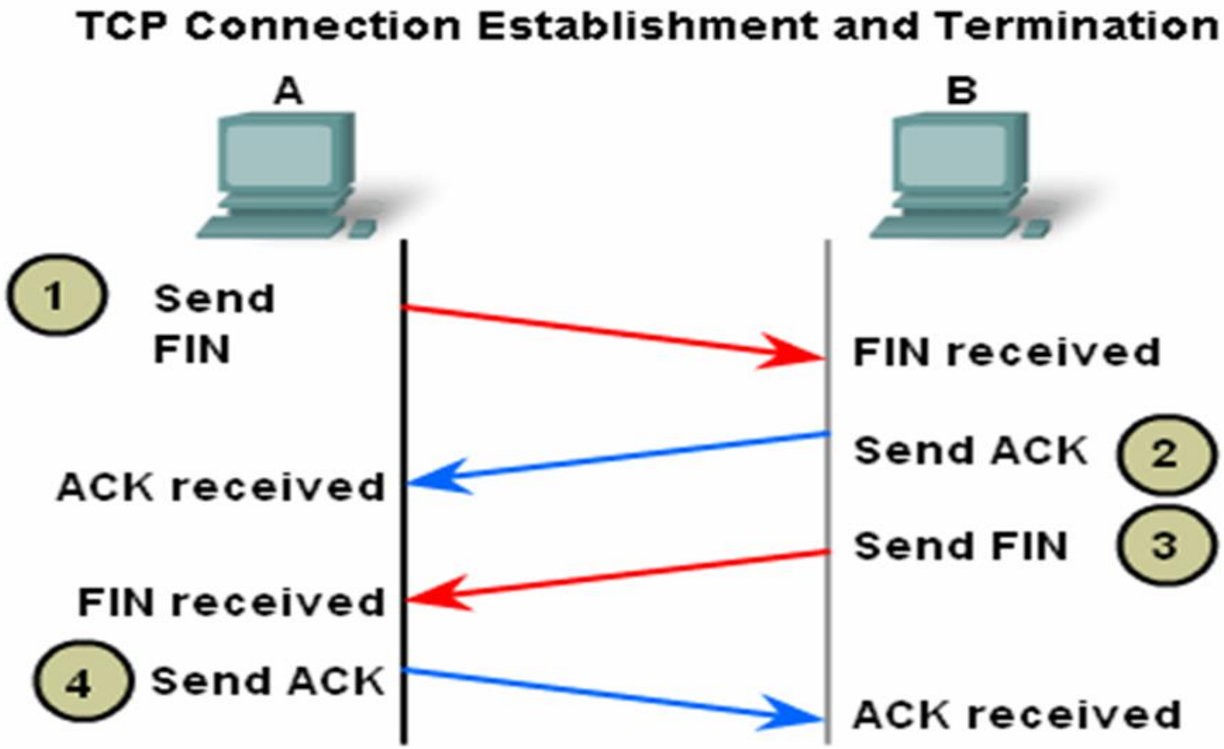


If segments are lost because of congestion, the Receiver will acknowledge the last received sequential segment and reply with a reduced window size.

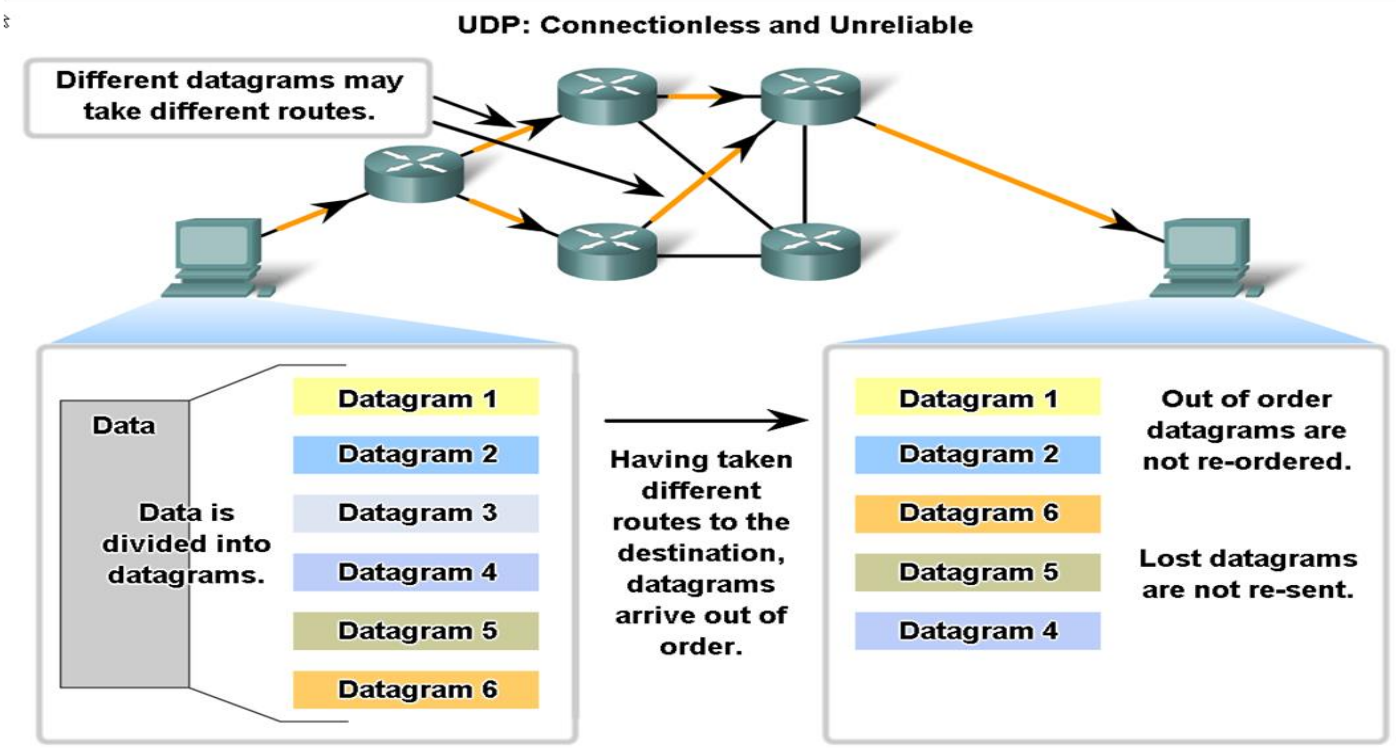
TCP Termination

1. When the client has no more data to send in the stream, it sends a segment with the FIN flag set.
2. The server sends an ACK to acknowledge the receipt of the FIN to terminate the session from client to server.
3. The server sends a FIN to the client, to terminate the server to client session.
4. The client responds with an ACK to acknowledge the FIN from the server.

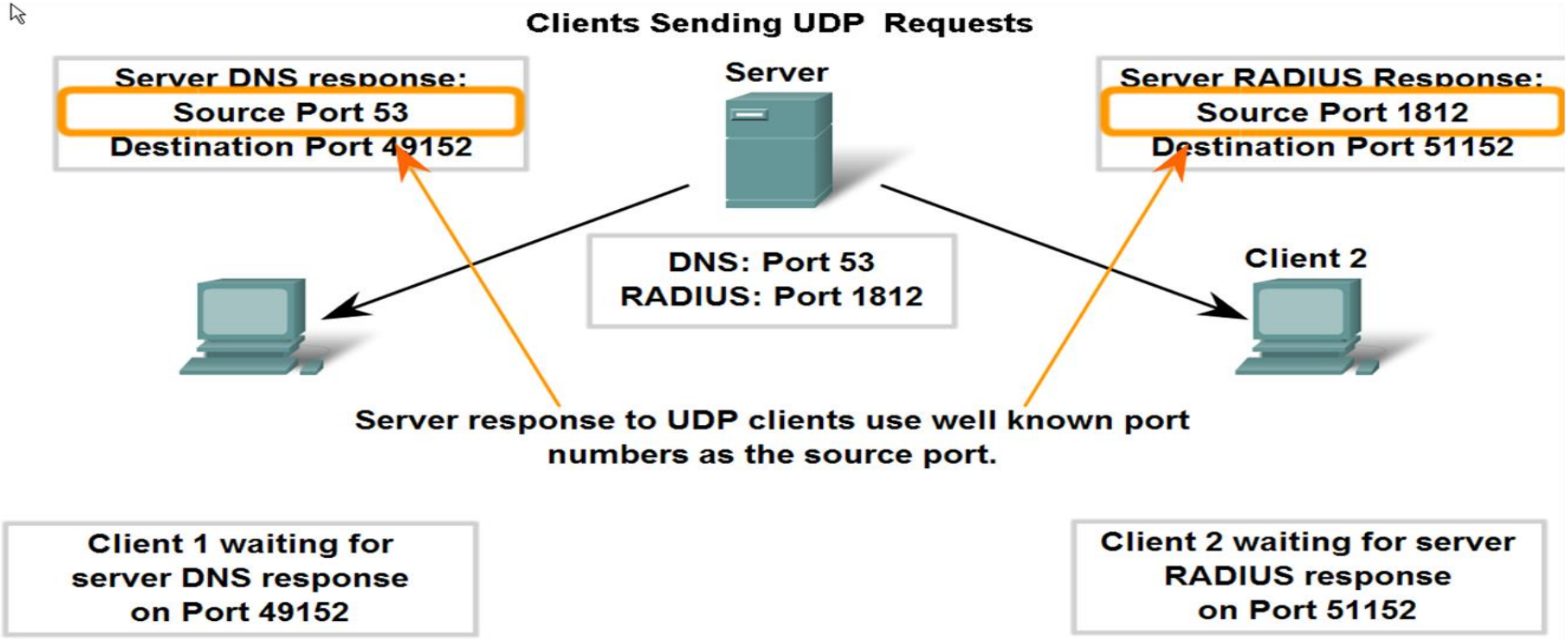
TCP Termination



UDP Protocol



UDP Protocol



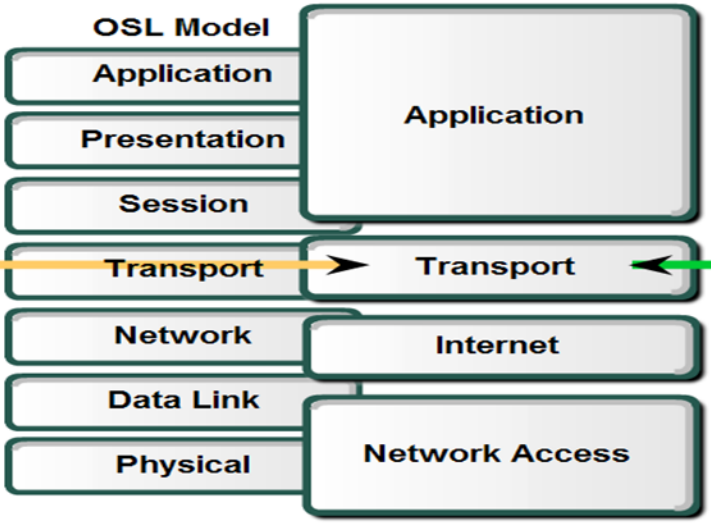
UDP

TCP

Transport Layer Protocols

- IP Telephony
- Streaming Video

- SMTP/POP (Email)
- HTTP



Required Protocol Properties

- Fast
- Low overhead
- Does not require acknowledgements
- Does not resend lost data
- Delivers data as it arrives

- **Connection Less**
- Best effort delivery

Required Protocol Properties

- Reliable
- Acknowledge data
- Resend lost data
- Delivers data in order sent

- **Connection Oriented**
- **Flow control**
- Guaranteed delivery

Application developers choose the appropriate Transport Layer protocol based on the nature of the application.

Transport Layer Role and Services

TCP and UDP Headers

TCP SEGMENT & HEADER FIELDS

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 SEGMENT (Size varies)		

20 Bytes

UDP SEGMENT & HEADER FIELDS

Bit (0)	Bit (15) Bit (16)	Bit (31)
Source Port (16)	Destination Port (16)	
Length (16)	Checksum (16)	
APPLICATION LAYER DATA SEGMENT (Size varies)		

8 Bytes