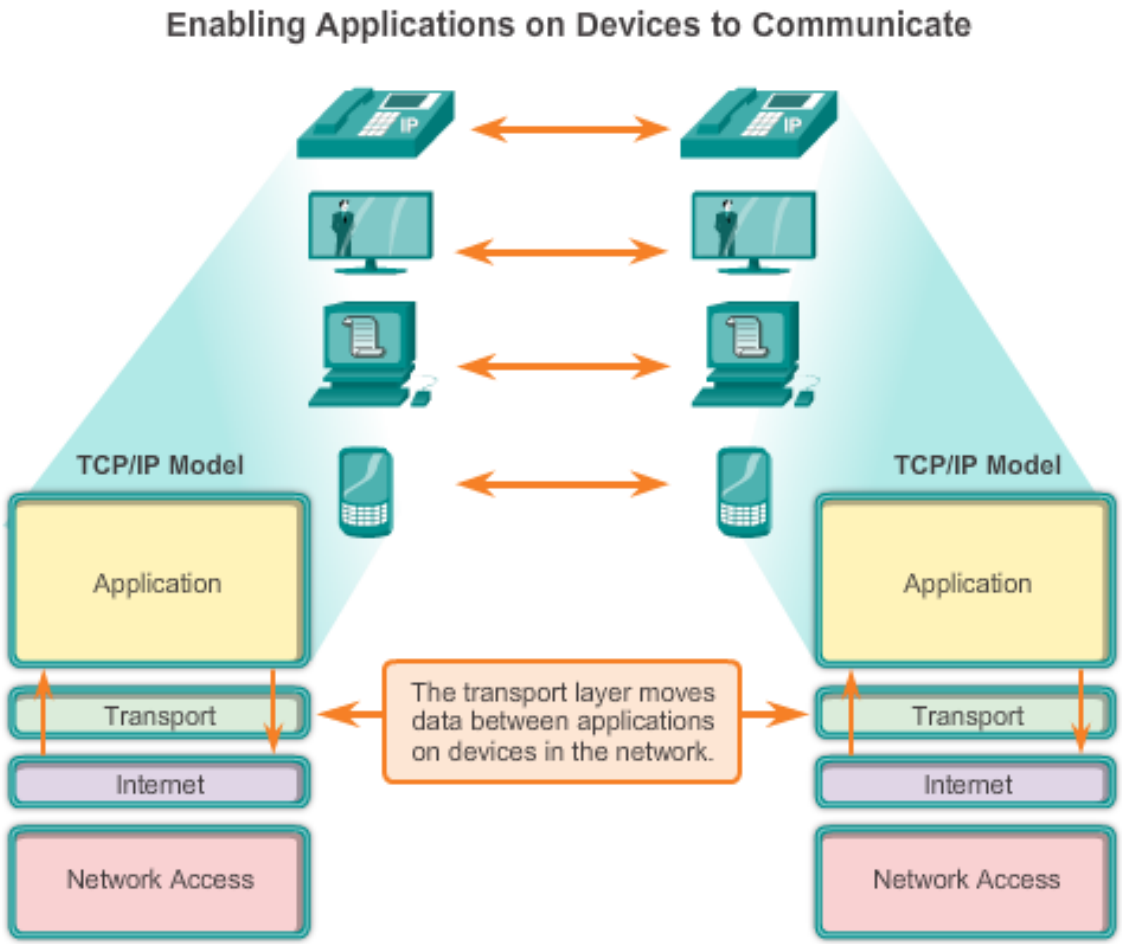
Capa: ?
Rol de la capa de transporte



Rol de la capa de transporte

La capa de transporte es responsable del establecimiento de una sesión de comunicación temporal entre dos aplicaciones y de la entrega de datos entre ellos. TCP/IP usa dos protocolos para alcanzar esto:

- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

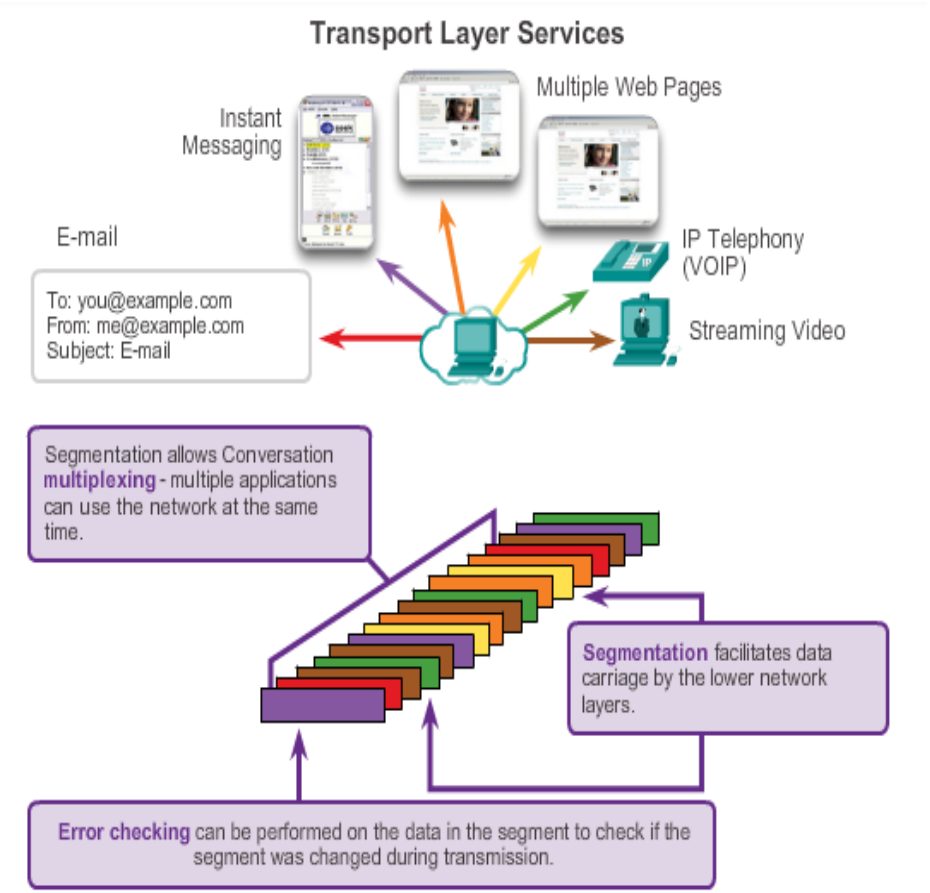
Principales responsabilidades de los protocolos de la capa de transporte

- Seguimiento de las comunicaciones individuales entre aplicaciones en los hosts origen y destino
- Segmentación de datos para la manejabilidad y reensamblado de segmentos de datos en flujos de datos de aplicación en el destino
- Identificación de la aplicación apropiada para cada flujo de comunicación

Multiplexación de las Conversaciones

Segmentación de datos

- Permite muchas comunicaciones, desde diferentes usuarios, para ser intercalada (multiplexada) en la misma red, al mismo tiempo.
- Provee los medios para enviar y recibir datos cuando están corriendo múltiples aplicaciones.
- Encabezado agregado a cada segmento para identificarlo.



Confiabilidad de capa de Transporte

Diferentes aplicaciones tienen diferentes requerimientos de confiabilidad de transporte

TCP/IP provee dos protocolos de capa de transporte, **TCP y UDP**

Transmission Control Protocol (TCP)

- Provee entrega confiable asegurando que todos los datos llegan al destino.
- Usa confirmaciones y otros procesos para asegurar la entrega
- Gran demanda en la red – mucha carga

User Datagram Protocol (UDP)

- Provee sólo las funciones básicas para la entrega – no confiable
- Menos sobrecarga

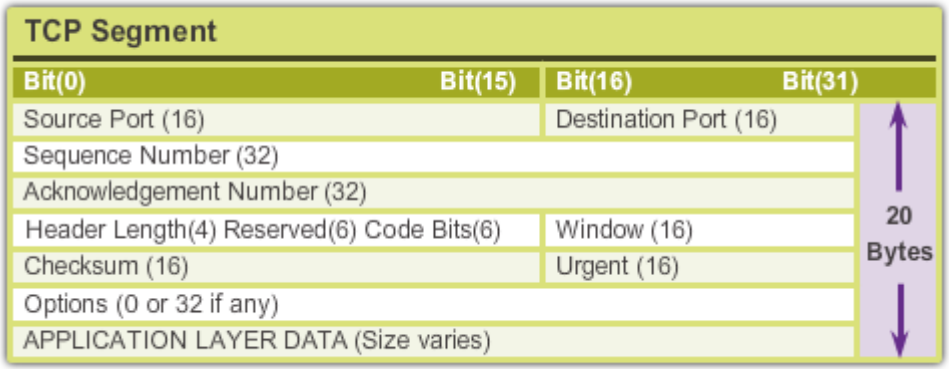
TCP o UDP

- Hay un equilibrio entre el valor de la confiabilidad y el esfuerzo en ponerlo en la red.
- Desarrolladores de aplicaciones eligen el protocolo de transporte basado en los requerimientos de sus aplicaciones.

Introducción a TCP

Transmission Control Protocol (TCP)

- RFC 793
- Orientado a la conexión – creando una sesión entre origen y destino
- Entrega confiable – retransmición de datos perdidos o corruptos
- Reconstrucción ordenada de datos – numerando y secuenciando los segmentos
- Control de flujo - regulando la cantidad de datos transmitidos
- Protocolo Stateful (con estado) – mantiene un seguimiento de la sesión



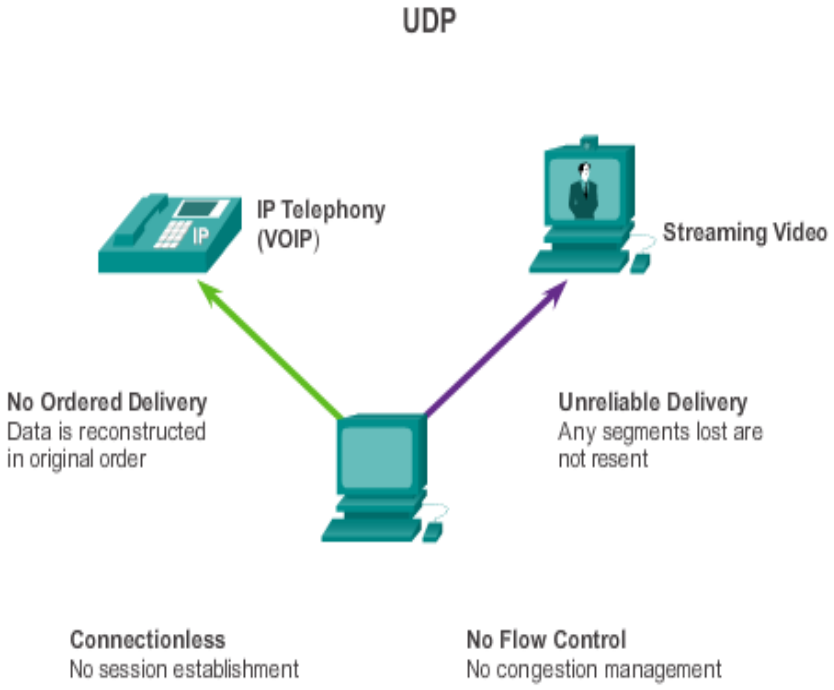
Introducción a UDP

User Datagram Protocol (UDP)

- RFC 768
- Sin conexión
- Entrega no confiable
- Reconstrucción no ordenada de datos
- Sin control de flujo
- Protocolo sin estado

Aplicaciones que usan UDP:

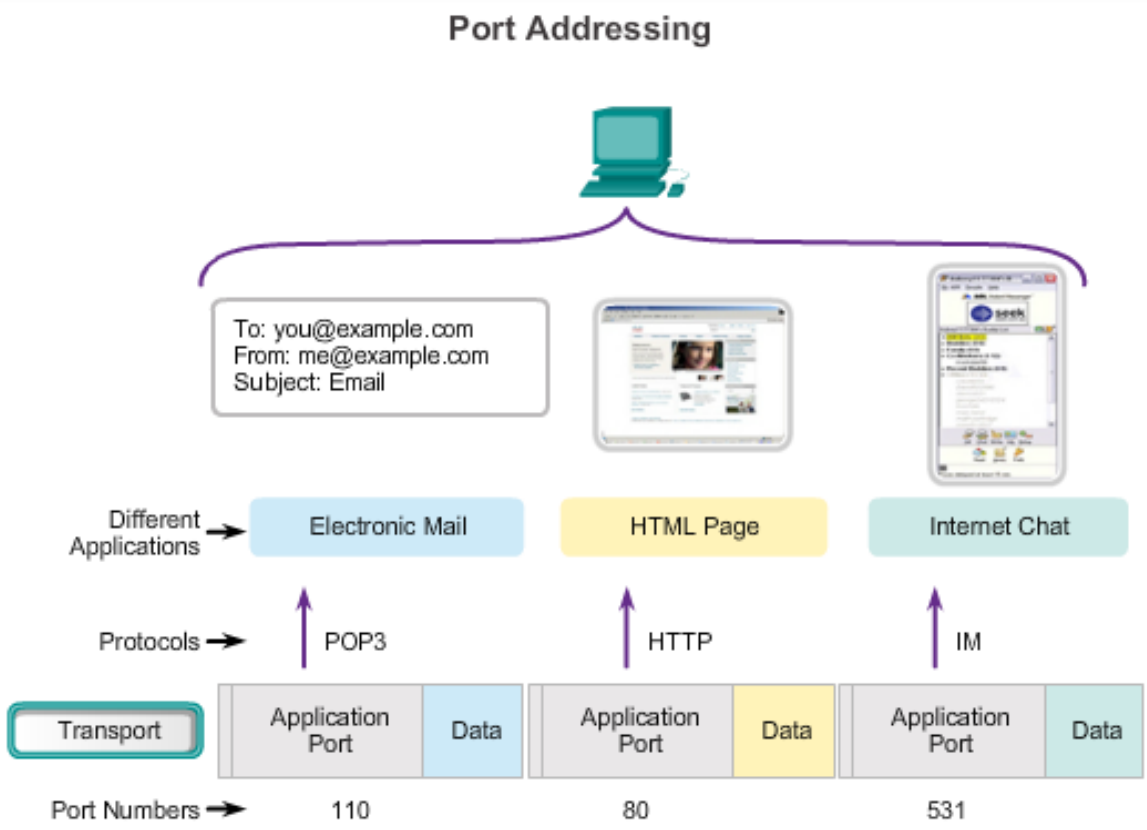
- Domain Name System (DNS)
- Video Streaming
- Voice over IP (VoIP)



Separando Múltiples Comunicaciones

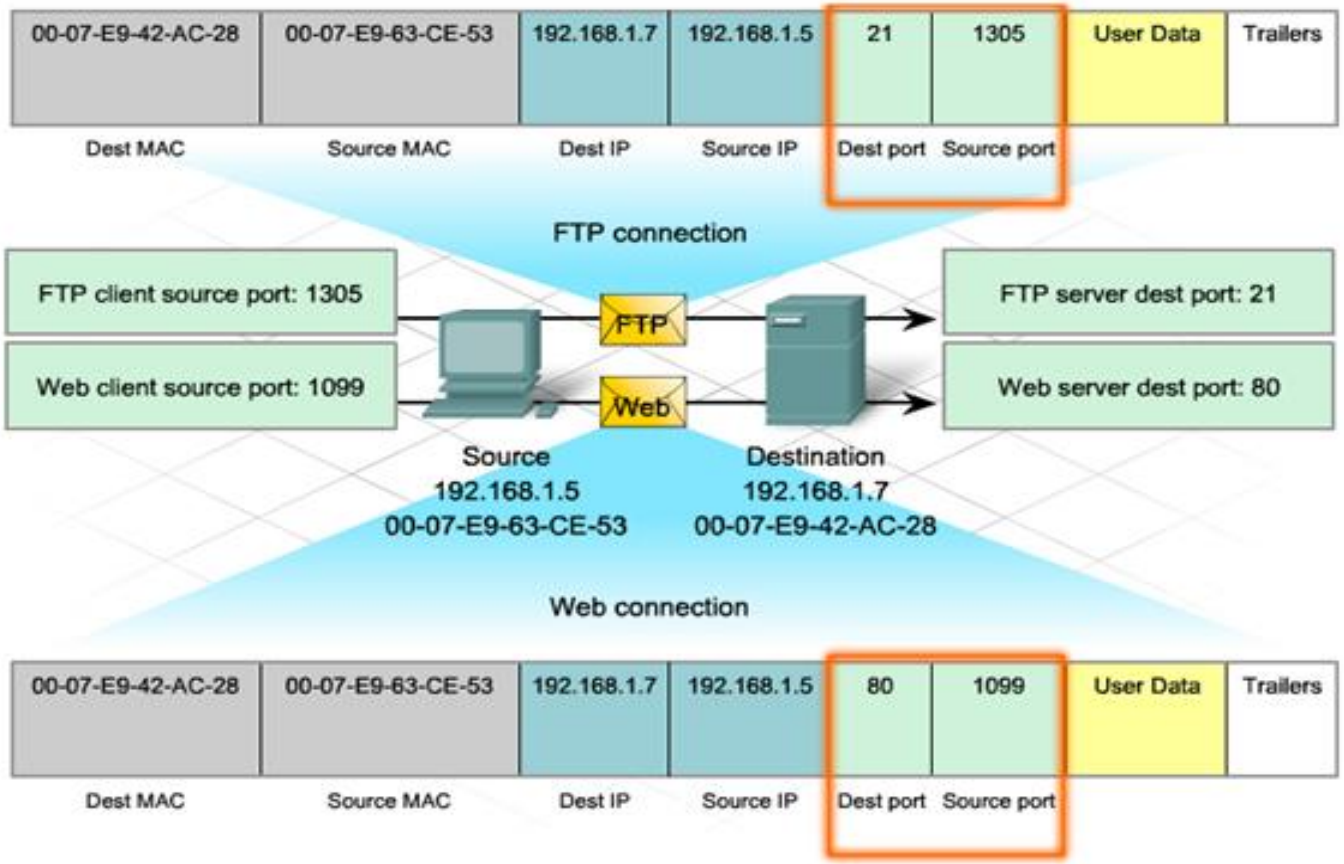
Números de puerto son usados por TCP y UDP para diferenciar entre aplicaciones.

.



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

Direccionamiento de puerto TCP y UDP



Direccionamiento de puerto TCP y UDP

Port Numbers

Port Number Range	Port Group
0 to 1023	Well Known (Contact) Ports
1024 to 49151	Registered Ports
49152 to 65533	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)

Registered UDP Ports:
1812 RADIUS Authentication Protocol
5004 RTP (Voice and Video Transport Protocol)
5040 SIP (VoIP)

Well Known UDP Ports:
69 TFTP
520 RIP

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

Direccionamiento de puerto TCP y UDP

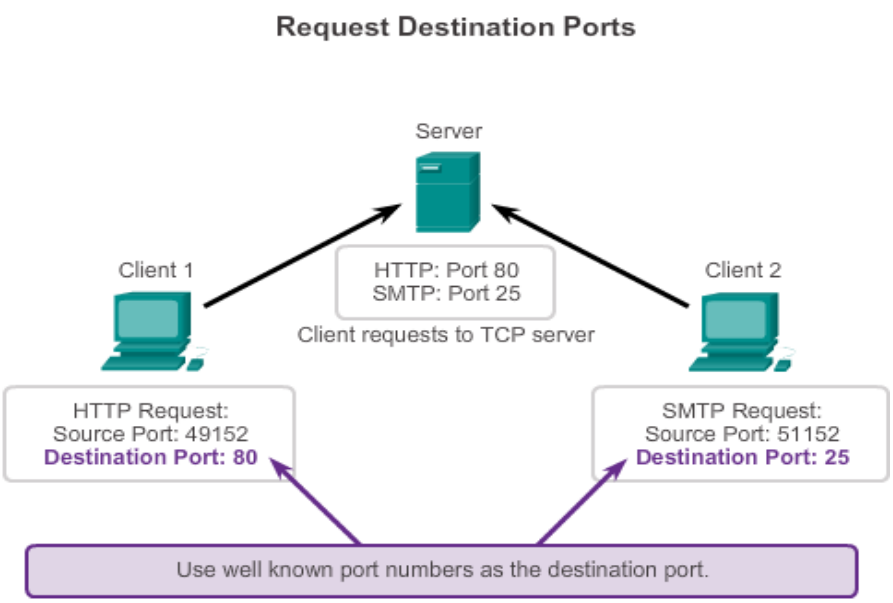
Netstat

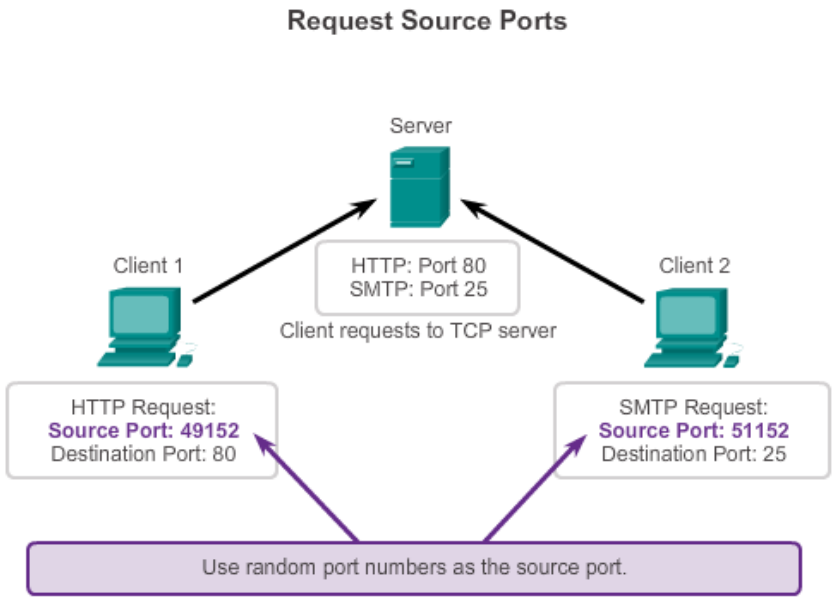
- Usado para examinar conexiones TCP que están abiertas y corriendo en un host de red

```
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:\>
```

Comunicación TCP

Procesos de servidor TCP





TCP Conexión, Establecimiento y finalización

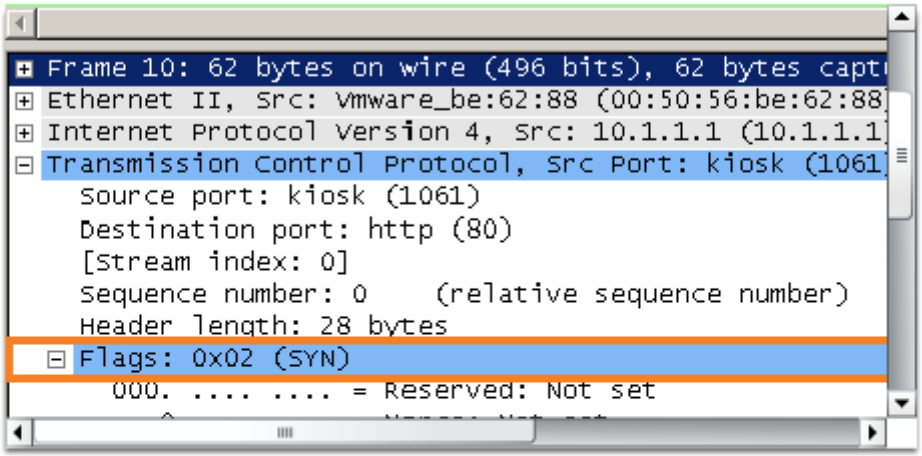
Three-Way Handshake

- Determina que el dispositivo destino está presente en la red.
- Verifica que el dispositivo destino tiene un servicio activo y está aceptando solicitudes en el número de puerto destino que el cliente que inició la sesión pretende usar.
- Informa al dispositivo destino que el cliente origen pretende establecer una sesión de comunicación en ese número de puerto.

TCP Three-Way Handshake – Paso 1

- **Paso 1: El cliente solicita una sesión de comunicación cliente-a-servidor con el servidor.**

TCP 3-way Handshake (SYN)



Protocol Analyzer shows initial client request for session in frame 14

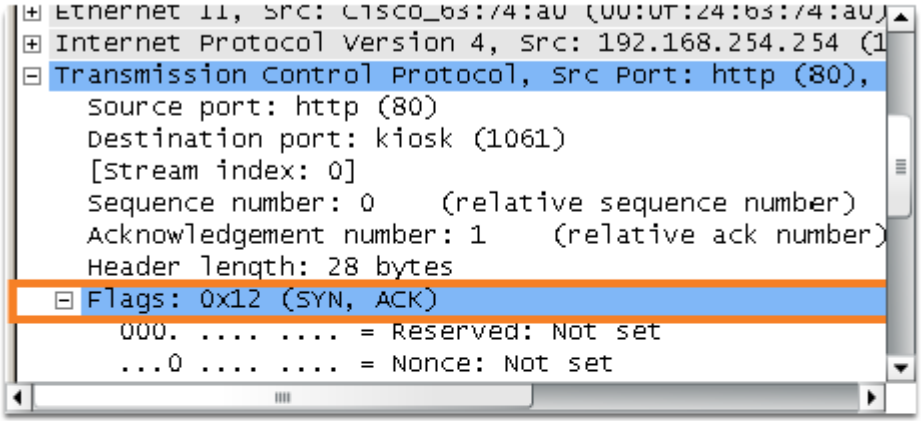
TCP segment in this frame shows:

- SYN flag set to validate an Initial Sequence Number
- Randomized sequence number valid (relative value is 0)
- Random source port 1061
- Well-known destination port is 80 (HTTP port) indicates web server (httpd)

TCP Three-Way Handshake – Paso 2

- **Paso 2 : El servidor confirma la sesión de comunicación cliente-a-servidor y solicita una sesión de comunicación servidor-a-cliente**.

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 1061 to corresponding to the clients source port
- Source port number of 80 (HTTP) indicating the web server service (httpd)

TCP Three-Way Handshake – Paso 3

- **Paso 3: El cliente confirma la sesión de comunicación servidor-a-cliente.**

TCP 3-way Handshake (ACK)

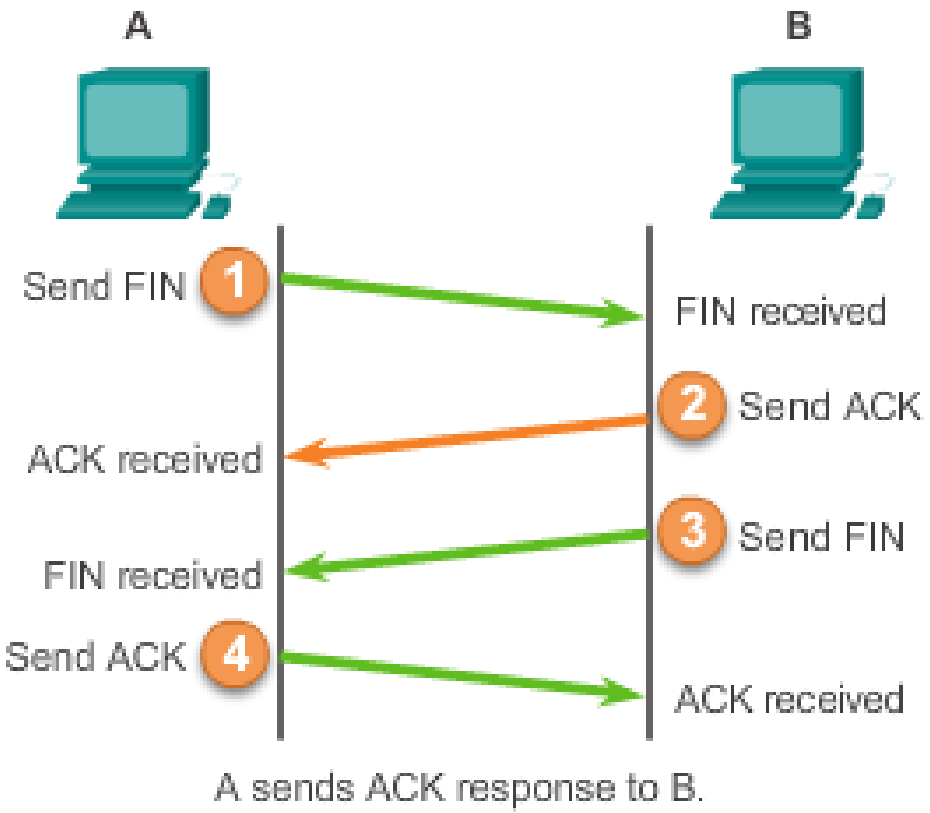
```
Source port: kiosk (1061)
Destination port: http (80)
[Stream index: 0]
Sequence number: 1    (relative sequence number)
Acknowledgement number: 1    (relative ack number)
Header Length: 20 bytes
⊟ Flags: 0x10 (ACK)
    000. .... .... = Reserved: Not set
    ...0 .... .... = Nonce: Not set
    .... 0... .... = Congestion Window Reduced (CWR)
    .... .0.. .... = ECN-Echo: Not set
    .... ..0. .... = Urgent: Not set
```

Protocol Analyzer shows client response to session in frame 16

The TCP segment in this frame shows:

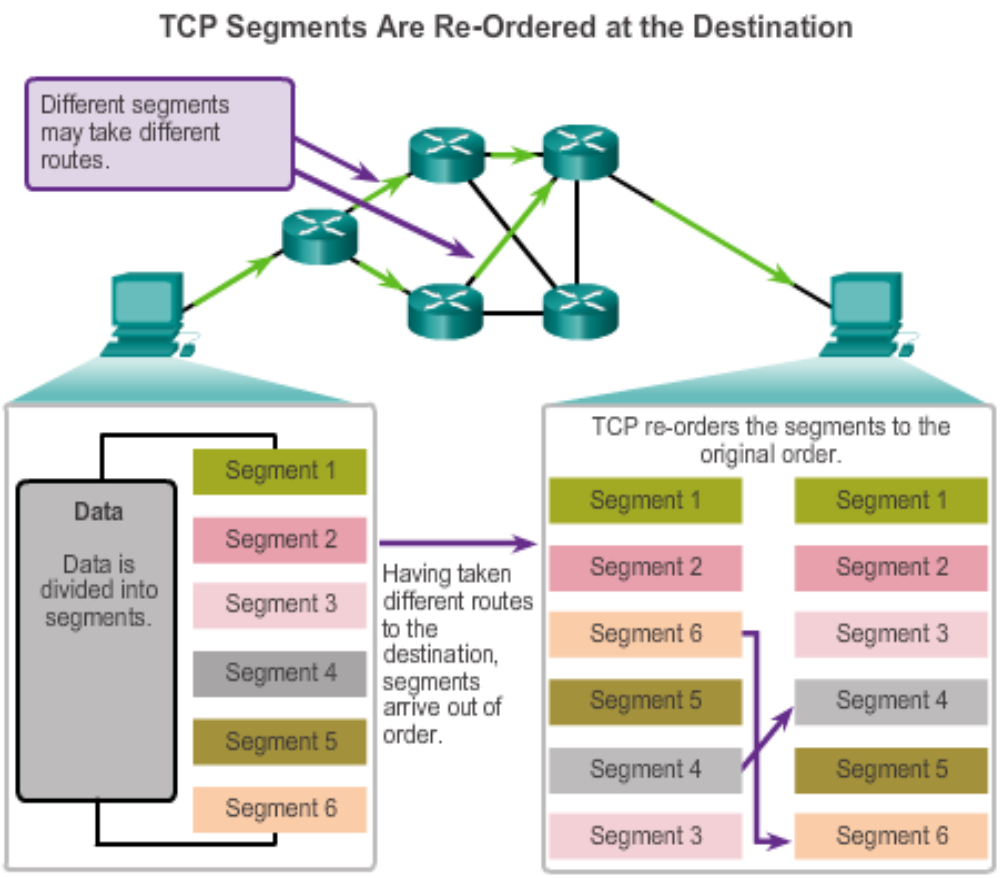
- ACK flag set to indicate a valid Acknowledgement number
- Acknowledgement number response to initial sequence number as relative value of 1
- Source port number of 1061 to corresponding
- Destination port number of 80 (HTTP) indicating the web server service (httpd)

Finalización de sesión TCP



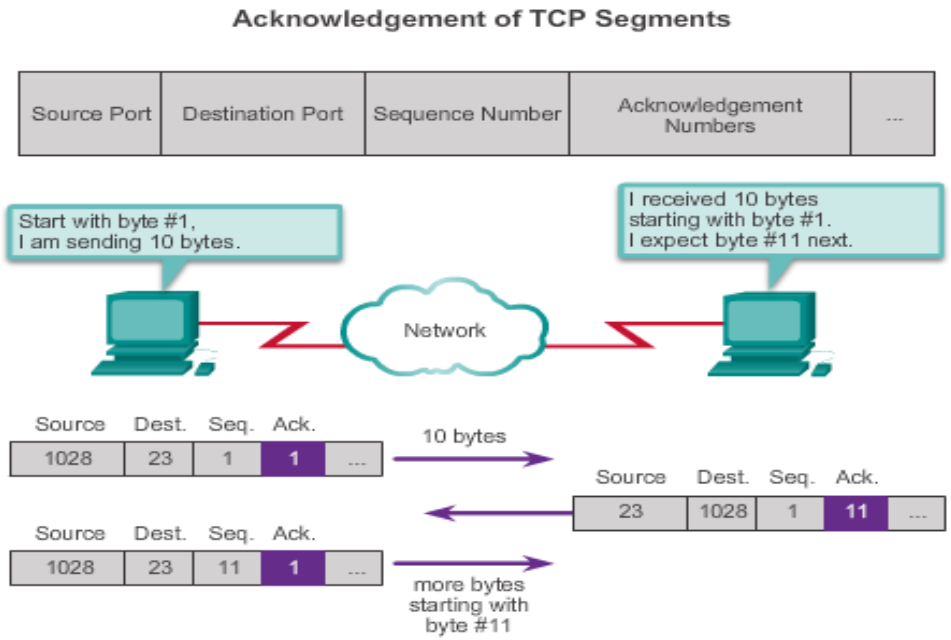
Confiabilidad TCP – Entrega Ordenada

Números de secuencia son usados para reensamblar segmentos en el orden original



Confiabilidad TCP – Confirmacines (Acknowledgement) y tamaño de ventana

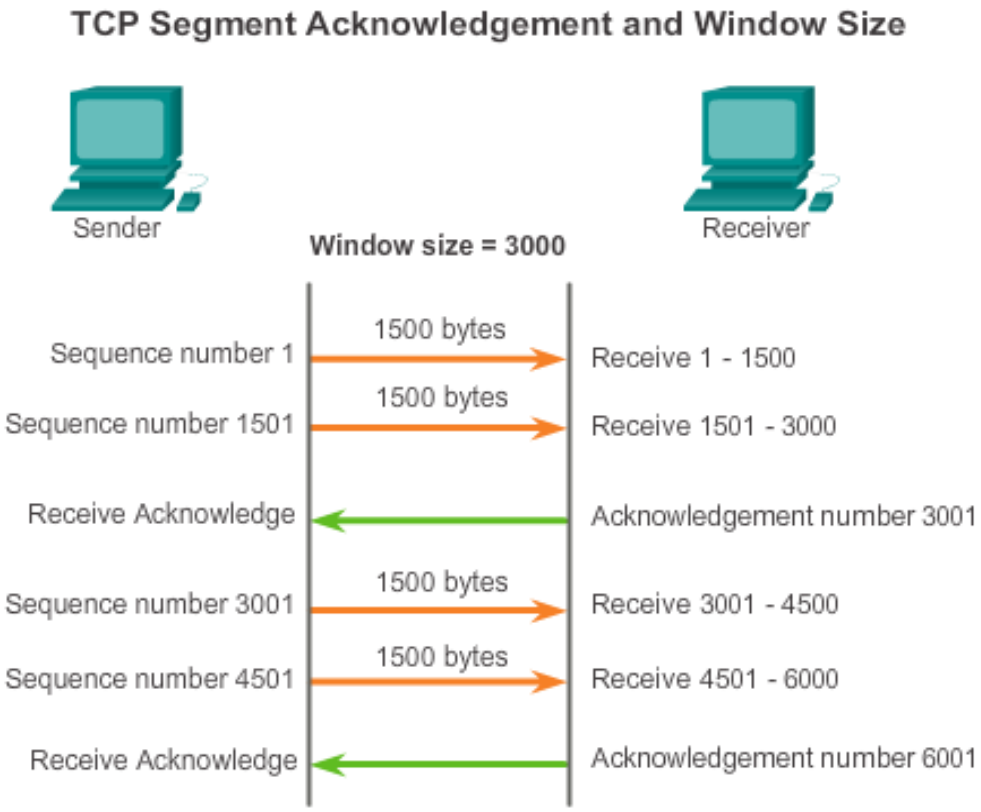
Los números de secuencia y números de acuse de recibo (acknowledgement) son usados para confirmar la recepción.



Tamaño de ventana - La cantidad de datos que un origen puede transmitir antes de que una confirmación deba ser recibida.

Confiabilidad TCP y control de flujo

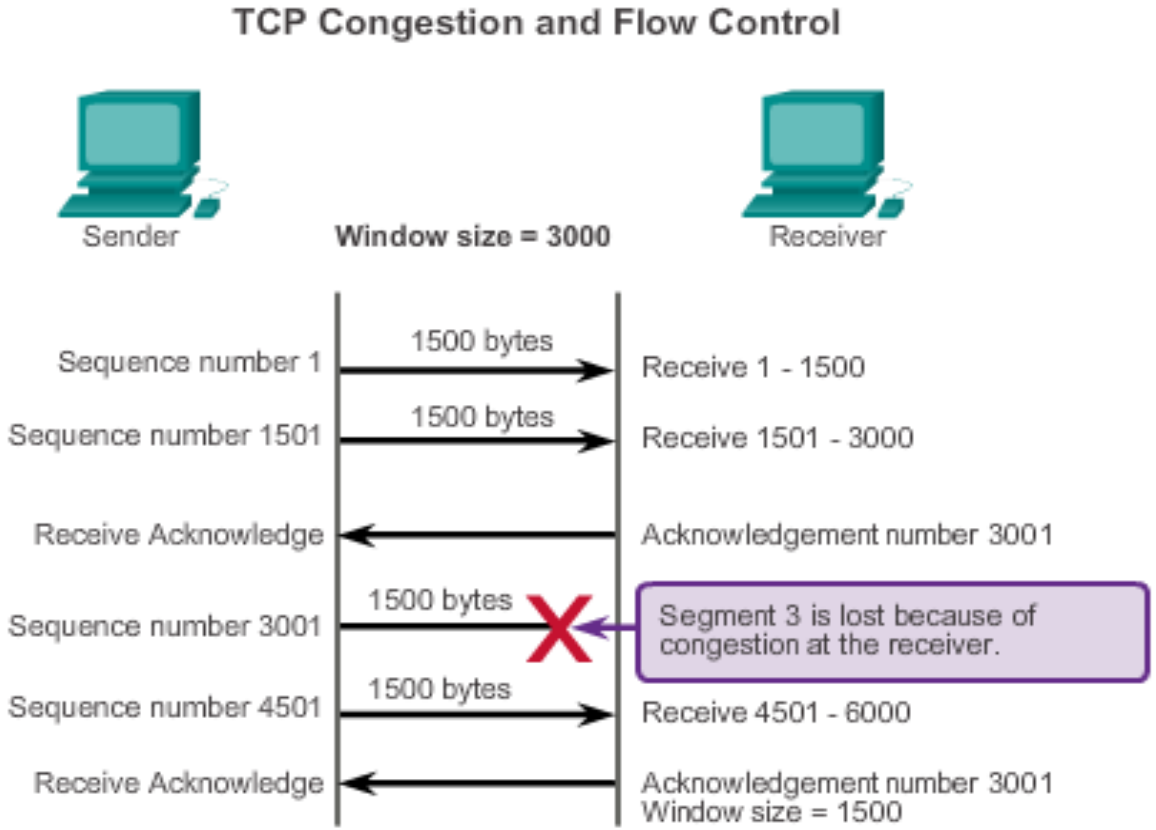
Tamaño de ventana y acuses de recibo



The **window size** determines the number of bytes sent before an acknowledgment is expected.

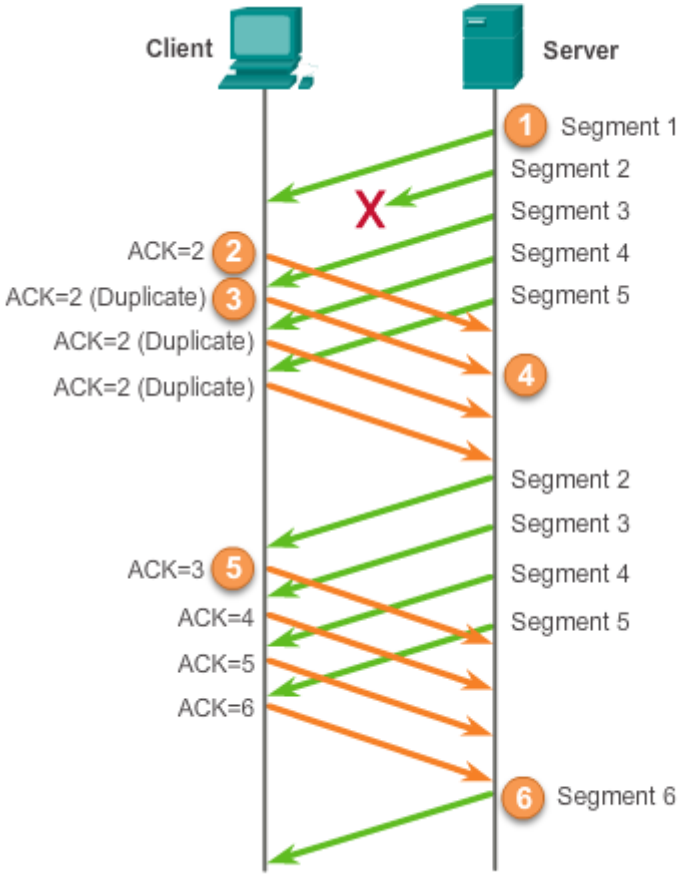
The **acknowledgement** number is the number of the next expected byte.

Control de flujo TCP – Prevención de congestión



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

Confiabilidad TCP - Acuse de recibo



Baja sobrecarga UDP vs. Confiabilidad

UDP

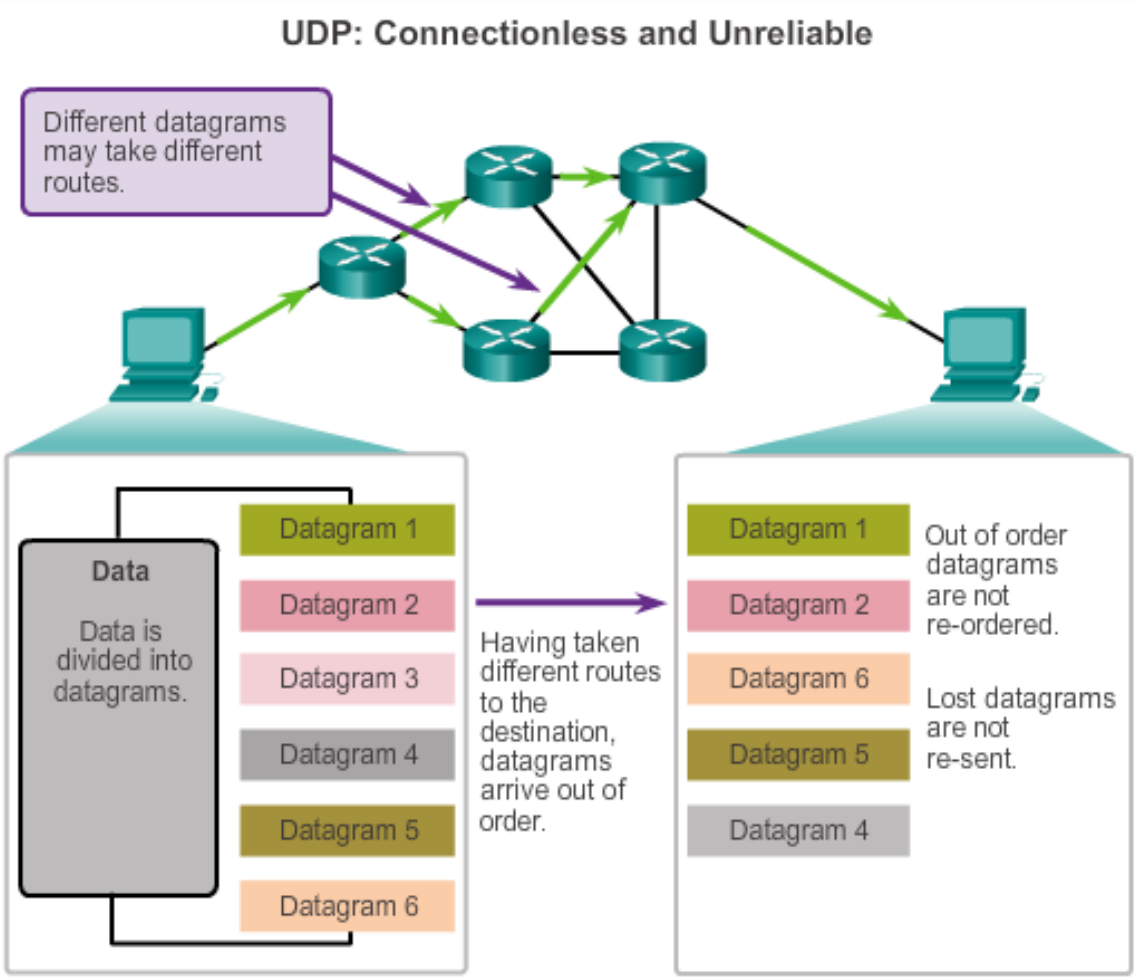
- Protocolo simple que provee las funciones básicas de la capa de transporte
- Usado por las aplicaciones que pueden tolerar pequeñas pérdidas de datos
- Usado por aplicaciones que no pueden tolerar retardos

Usado por

- Domain Name System (DNS)
- Simple Network Management Protocol (SNMP)
- Dynamic Host Configuration Protocol (DHCP)
- Trivial File Transfer Protocol (TFTP)
- IP telephony or Voice over IP (VoIP)
- Juegos online

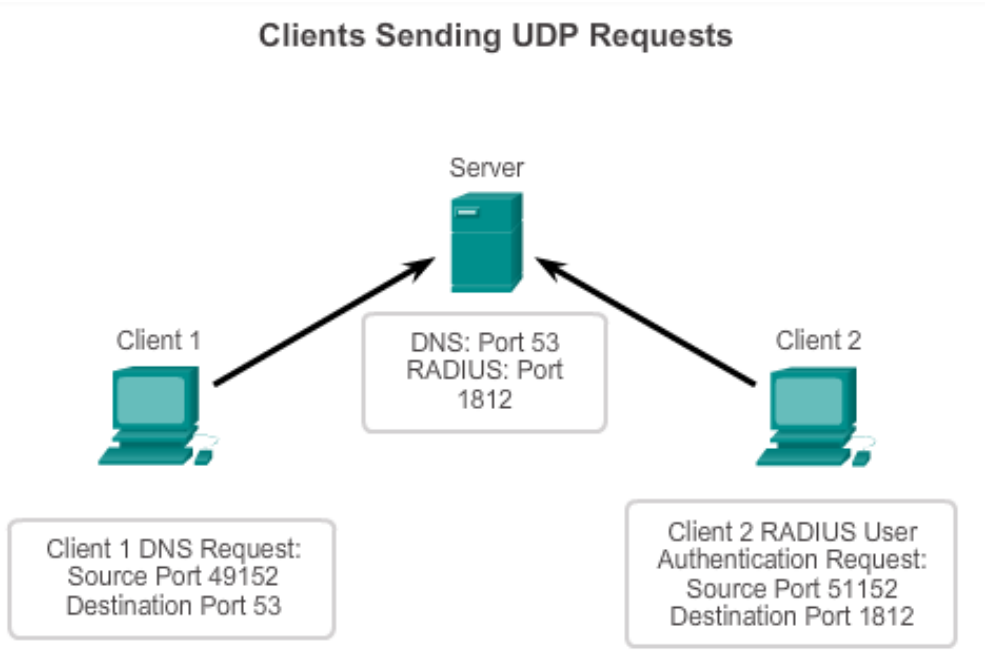
Comunicación UDP

Reensamble de Datagrama

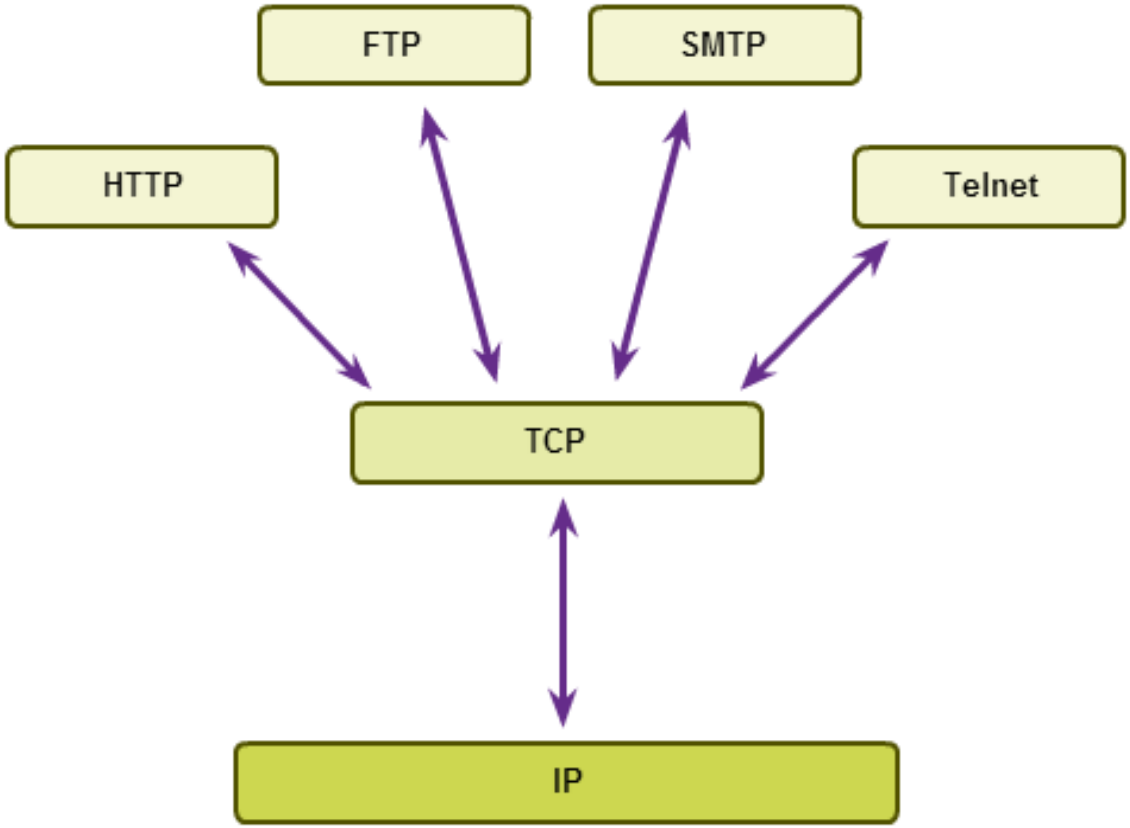


Procesos UDP de servidor y cliente

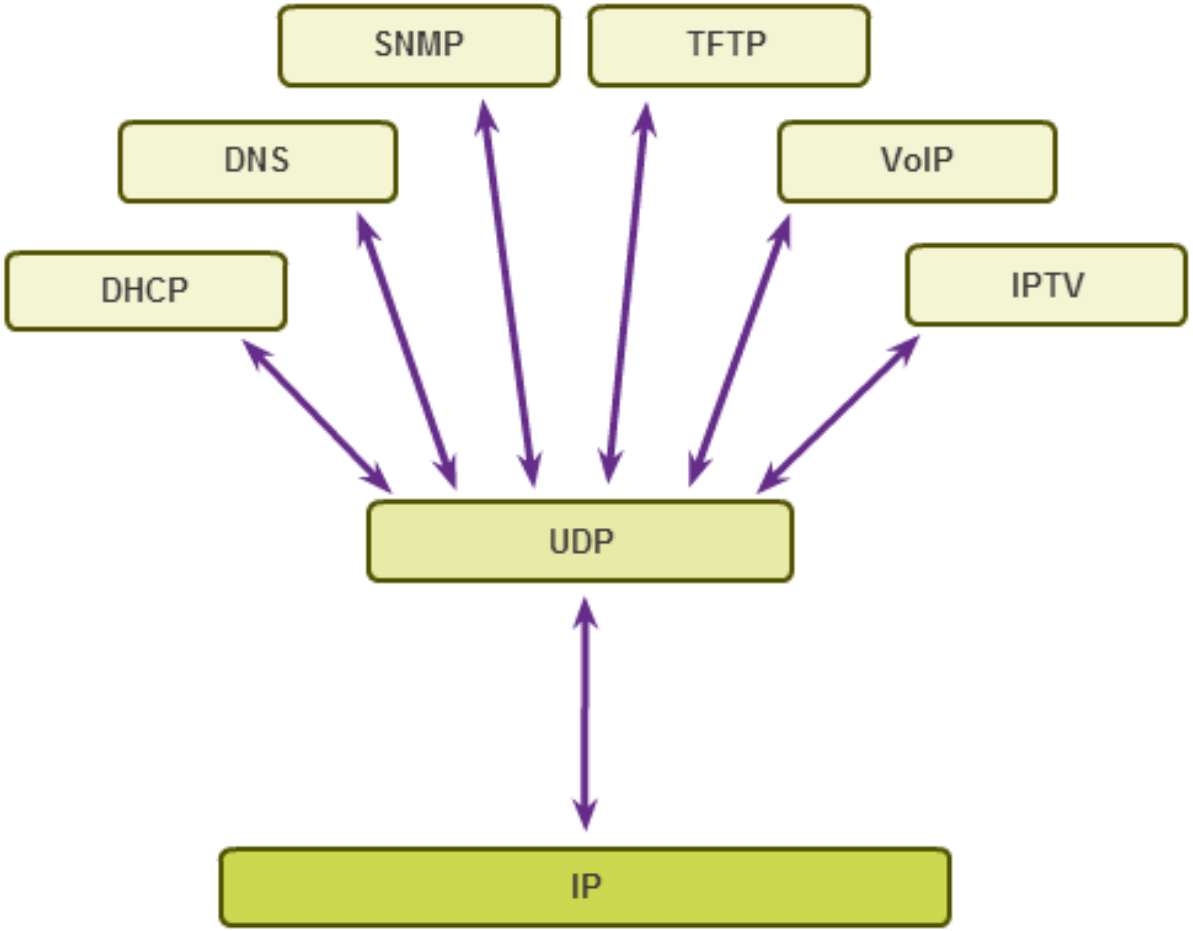
- Aplicaciones de servidor basadas en UDP son asignados a números de puertos bien conocidos o registrados.
- Los procesos clientes UDP, aleatoriamente seleccionan números de puerto, de un rango de números de puerto dinámicos como puerto origen.



Aplicaciones que usan TCP



Aplicaciones que usan UDP



Router on Stick

- Diseñe las subredes a utilizar.
- Configure los host.
- Configure las VLAN en el Capa 2

  Crea las Vlans:

  2960Sw(config)#vlan 10

  2960Sw(config-vlan)#vlan 20

  2960Sw(config-vlan)#vlan 30

  2960Sw(config-vlan)#exit

- Configure los puertos que van a estar en las VLAN

Comandos de IOS de un switch Cisco	
Ingrese al modo de configuración global.	`S1# configure terminal`
Ingrese al modo de configuración de interfaz para la SVI.	`S1(config)# interface id_interfaz`
Establezca el puerto en modo de acceso.	`S1(config-if)# switchport mode access`
Asigne el puerto a una VLAN.	`S1(config-if)# switchport access vlan id_vlan`
Vuelva al modo EXEC privilegiado.	`S1(config-if)# end`

- A la interface que va hacia el enrutador, configúrela como un enlace troncal:

  2960Sw(config)#int fa0/4

  2960Sw(config-if)#sw mode trunk

- Ahora ingrese al CLI del Router y encienda la interface que va hacia el Sw:

  Router(config-if)#int g0/0

  Router(config-if)#no shut

- Después crear las subinterfaces, para que por un único cable pueda formar la conectividad entre diferentes VLANs:

```
Router(config-if)#int g0/0.10
Router(config-subif)#encapsulation dot1q 10
Router(config-subif)#ip add 192.168.10.1 255.255.255.0
(Esta ip es el Gateway de la PC de la Vlan 10)
Router(config-subif)#int g0/0.20
Router(config-subif)#encapsulation dot1q 20
Router(config-subif)#ip add 192.168.20.1 255.255.255.0
Router(config-subif)#int g0/0.30
Router(config-subif)#encapsulation dot1q 30
Router(config-subif)#ip add 192.168.30.1 255.255.255.0
Router(config-subif)#end
```

InterVLAN Routing

Configure las Pc (o dispositivos finales) con la dirección que corresponde a la VLAN.

Crear las VLANs en Capa 2:

```
2960Sw(config)#vlan 40
2960Sw(config-vlan)#vlan 50
2960Sw(config-vlan)#vlan 60
2960Sw(config-vlan)#exit
```

En el L3 crear las mismas Vlans de los Sw capa 2:

```
Switch(config)#host L3
L3(config)#vlan 40
L3(config-vlan)#vlan 50
L3(config-vlan)#vlan 60
```

En el L2 configurar los puertos a las VLANs

En el L2 configurar el puerto trunk

En las interfaces del L3 que conectan con los Sw capa 2 configure:

```
L3(config-vlan)#int fa0/1
L3(config-if)#sw trunk encapsulation dot1q
L3(config-if)#sw mode trunk
L3(config-if)#int fa0/3
L3(config-if)#sw trunk encapsulation dot1q
L3(config-if)#sw mode trunk
L3(config-if)#exit
```

Ahora en cada vlan virtual del L3 (SVI) ingrese el gateway de cada pc a la Vlan que corresponde con su máscara de red:

```
L3(config)#int vlan 40
L3(config-if)#ip add 192.168.40.1 255.255.255.0
L3(config-if)#int vlan 50
L3(config-if)#ip add 192.168.50.1 255.255.255.0
L3(config-if)#int vlan 60
L3(config-if)#ip add 192.168.60.1 255.255.255.0
L3(config-if)#exit
L3(config)#ip routing
```

Nota: La interface que vaya a un router se le pone: no sw Con el fin de que empiece a enrutar.