



UNIVERSIDAD NACIONAL DE ASUNCIÓN

FACULTAD POLITÉCNICA

REDES DE COMPUTADORAS II

ING. JORGE MOLINAS, ING. GUSTAVO AMARILLA

Capa de Transporte

OBJETIVOS

- Comprender el principio de funcionamiento de los Protocolos TCP y UDP.
- Reconocer los procesos de negociaciones para el inicio, transferencia y final de datos en la capa de transporte.
- Comprender la importancia de la capa de transporte en las redes de telecomunicaciones.

ÍNDICE GENERAL

- 1- Capa de Transporte
- 2- Introducción a los Protocolos TCP y UDP
- 3- Comunicación TCP
- 4- Confiabilidad y Control de Flujo
- 5- Comunicación UDP
- 6- Cuando utilizar TCP o UDP

1- CAPA DE TRANSPORTE

Habilitación de aplicaciones en dispositivos para establecer la comunicación

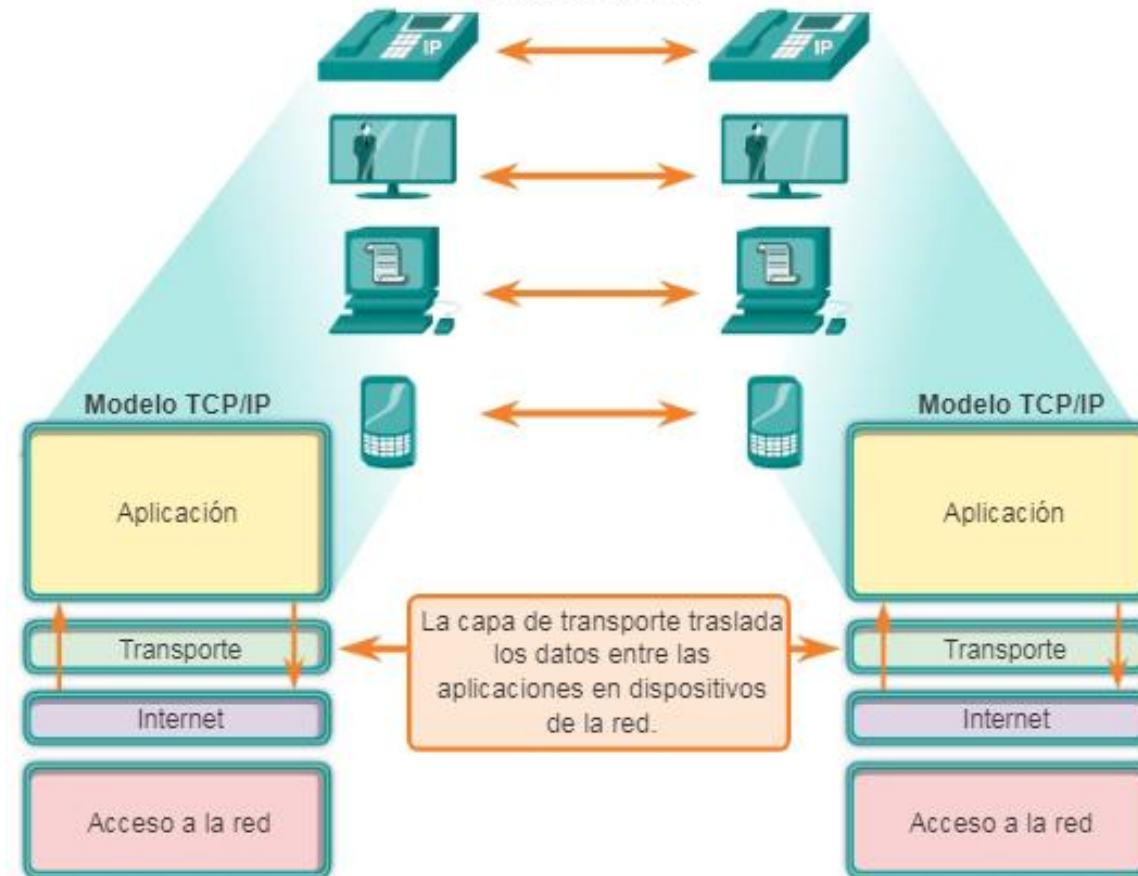


Figura 1: Comunicación en la Capa de Transporte[2].

1-CAPA DE TRANSPORTE

Las principales responsabilidades de los protocolos de la capa de transporte son las siguientes:

- 1) Rastreo de comunicación individual entre aplicaciones en los hosts de origen y destino.
- 2) División de los datos en segmentos para su administración y reunificación de los datos segmentados en streams de datos de aplicación en el destino.
- 3) Identificación de la aplicación correspondiente para cada stream de comunicación.

1.1- FUNCIONES DE LA CAPA DE TRANSPORTE

En la capa de transporte, se pueden destacar las siguiente tres funciones básicas las cuales son:

- 1) Rastreo de Conversaciones Individuales
- 2) Segmentación de Datos y Rearmado de Segmentos
- 3) Identificación de Aplicaciones

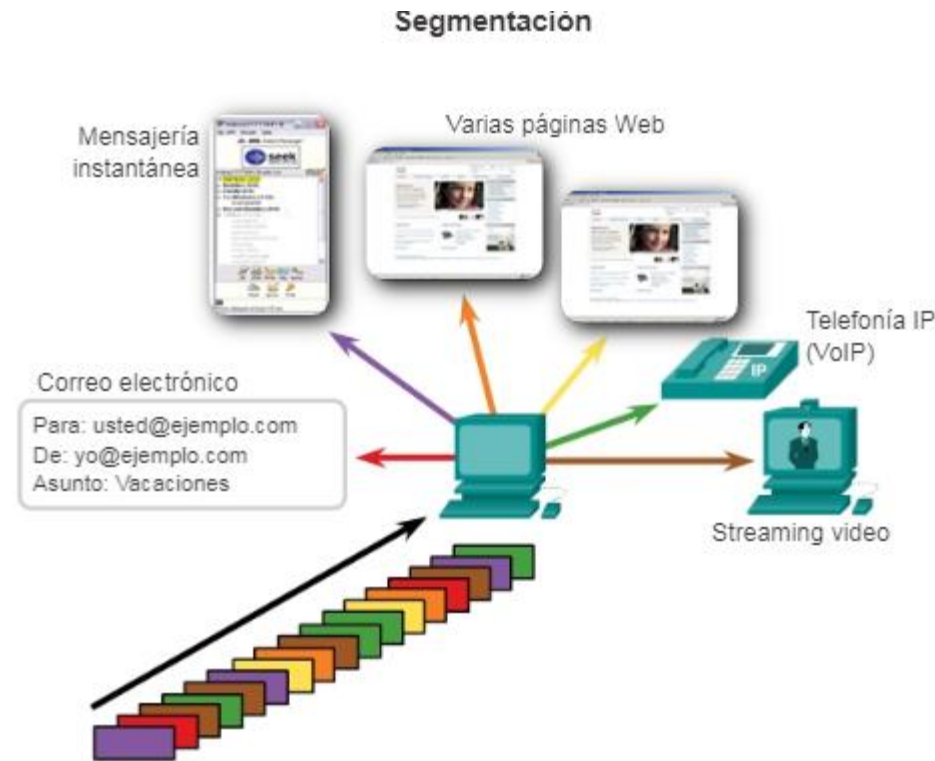
1.1- FUNCIONES DE LA CAPA DE TRANSPORTE



La capa de transporte hace un seguimiento de cada conversación individual que fluye entre una aplicación de origen y una aplicación de destino por separado.

Figura 2: Rastreo de Conversaciones Individuales [2].

1.1- FUNCIONES DE LA CAPA DE TRANSPORTE



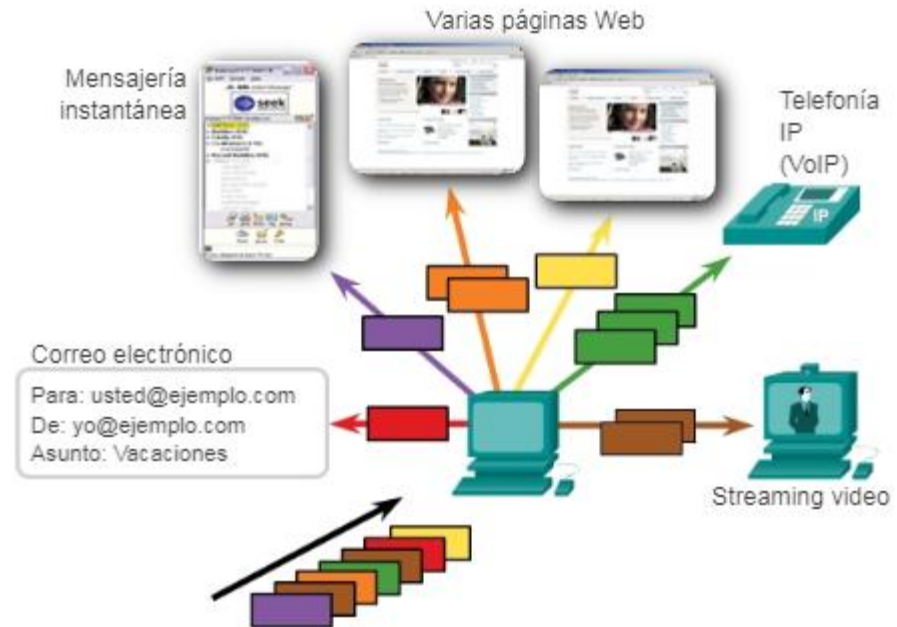
La capa de transporte divide los datos en segmentos, que son más fáciles de administrar y transportar.

Figura 3: Segmentación y Rearmado de Segmentos [2].

1.1- FUNCIONES DE LA CAPA DE TRANSPORTE



Identificación de aplicaciones



La capa de transporte garantiza que aunque sean varias las aplicaciones se ejecutan en un dispositivo, todas reciban los datos correctos.

Figura 4: Identificación de Aplicaciones [2].

1.2- MULTIPLEXACIÓN DE CONVERSACIONES

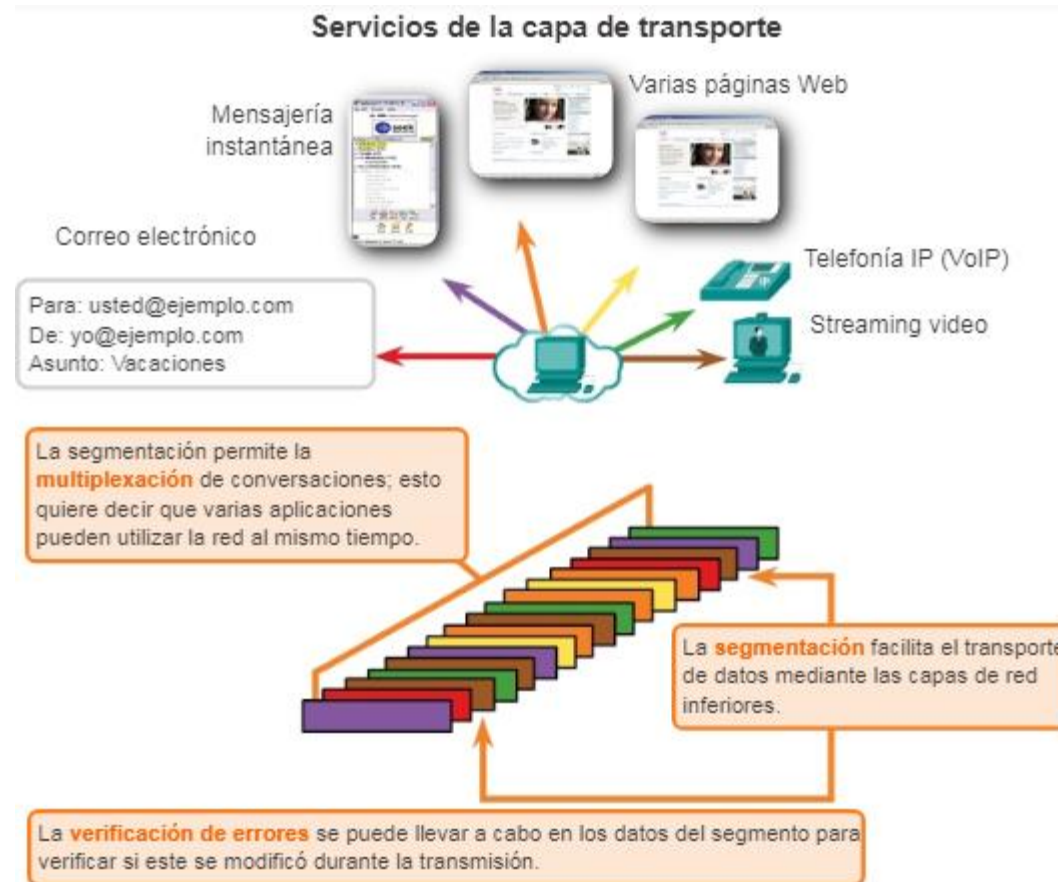


Figura 5: Multiplexación en la Capa de Transporte [2].

1.3- CONFIABILIDAD DE LA CAPA DE TRANSPORTE

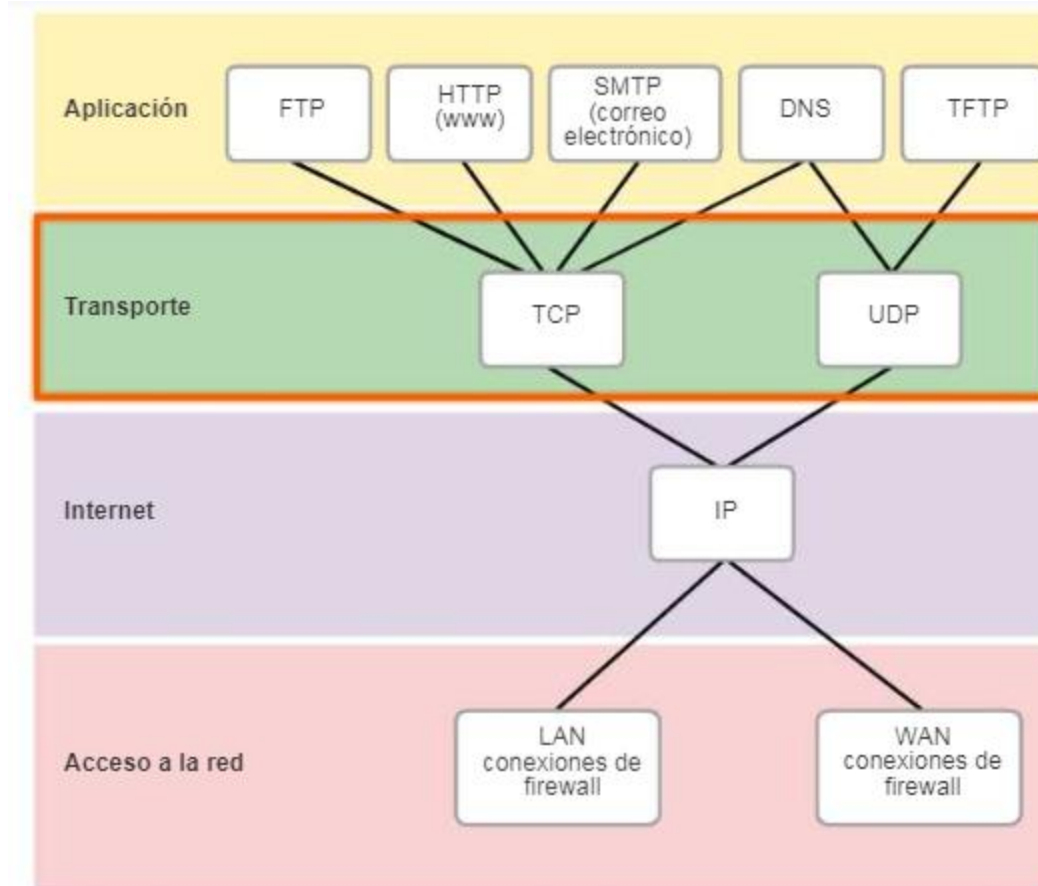


Figura 6: Confiabilidad en la Capa de Transporte [2].

1.4- PROTOCOLO TCP

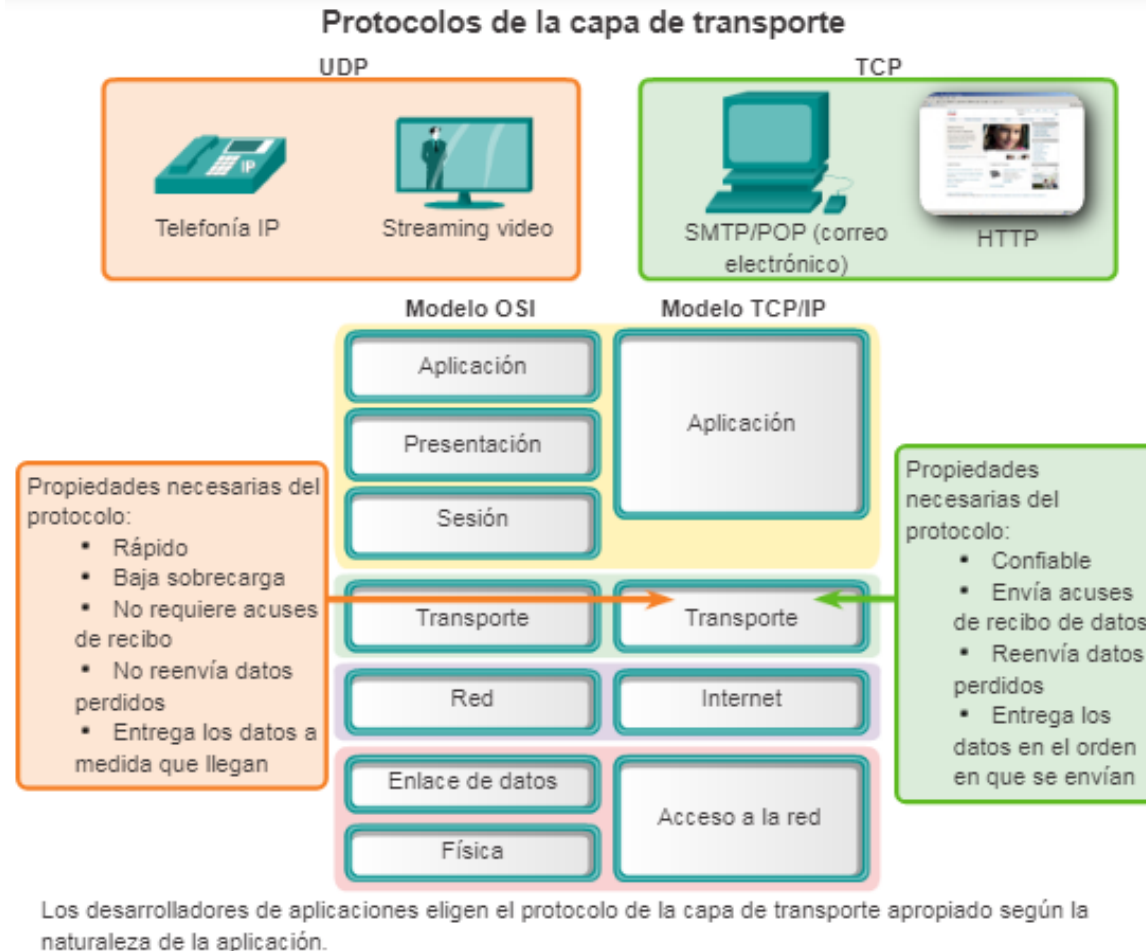
TCP se considera un protocolo de transporte confiable, lo que significa que incluye procesos para garantizar la entrega confiable entre aplicaciones mediante el uso de entrega con acuse de recibo. Con TCP, las tres operaciones básicas de confiabilidad son las siguientes:

- 1) Seguimiento de segmentos de datos transmitidos
- 2) Acuse de recibo de datos
- 3) Retransmisión de cualquier dato sin acuse de recibo

1.5- PROTOCOLO UDP

UDP proporciona solo las funciones básicas para entregar segmentos de datos entre las aplicaciones adecuadas, con muy poca sobrecarga y revisión de datos. El protocolo UDP se conoce como protocolo de entrega de máximo esfuerzo. En el contexto de redes, la entrega de máximo esfuerzo se denomina "poco confiable", porque no hay acuse de recibo que indique que los datos se recibieron en el destino. Con UDP, no existen procesos de capa de transporte que informen al emisor si la entrega se produjo correctamente.

1.6- PROTOCOLO DE LA CAPA DE TRANSPORTE ADECUADO PARA LA APLICACIÓN ADECUADA



2- INTRODUCCIÓN A LOS PROTOCOLOS TCP Y UDP



2.1- PROTOCOLO DE CONTROL DE TRANSMISIÓN (TCP)

TCP se describió inicialmente en RFC 793. Además de admitir funciones básicas de segmentación y rearmado de datos, TCP, proporciona lo siguiente:

- 1) Conversaciones orientadas a la conexión mediante el establecimiento de sesiones
- 2) Entrega Confiable
- 3) Reconstrucción de Datos Ordenada
- 4) Control del Flujo

2.2- ROL DE TCP

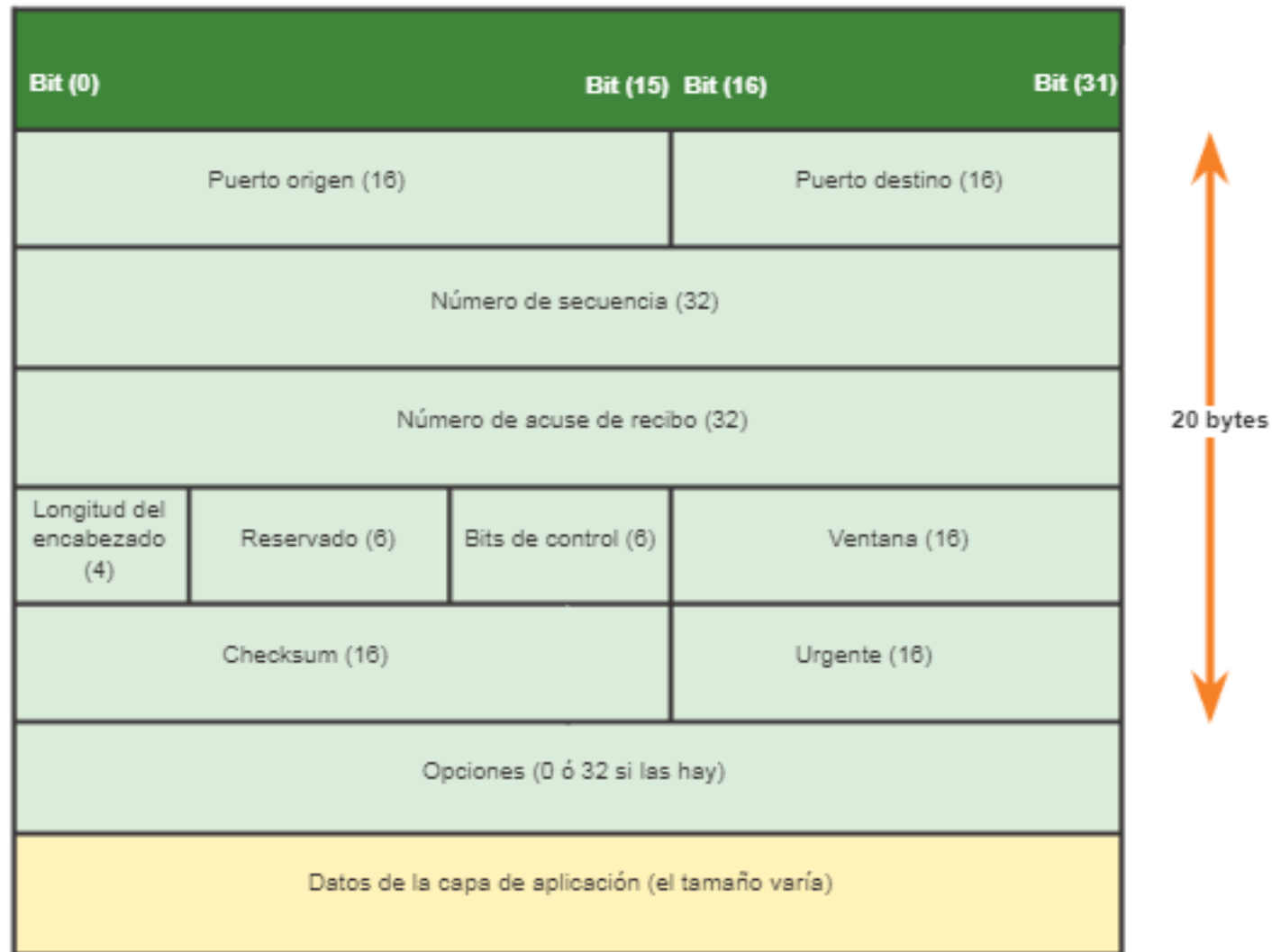


Figura 8: Encabezado del Segmento TCP[2].

2.3- PROTOCOLO DE DATAGRAMAS DE USUARIO (UDP)

UDP se considera un protocolo de transporte de máximo esfuerzo, descrito en RFC 768. UDP es un protocolo de transporte liviano. UDP es un protocolo tan simple que, por lo general, se lo describe en términos de lo que no hace en comparación con TCP. Las siguientes características describen a UDP:

- 1) Sin Conexión
- 2) Entrega no Confiable
- 3) Reconstrucción de Datos Desordenada
- 4) Sin Control del Flujo

2.4- ROL DE UDP

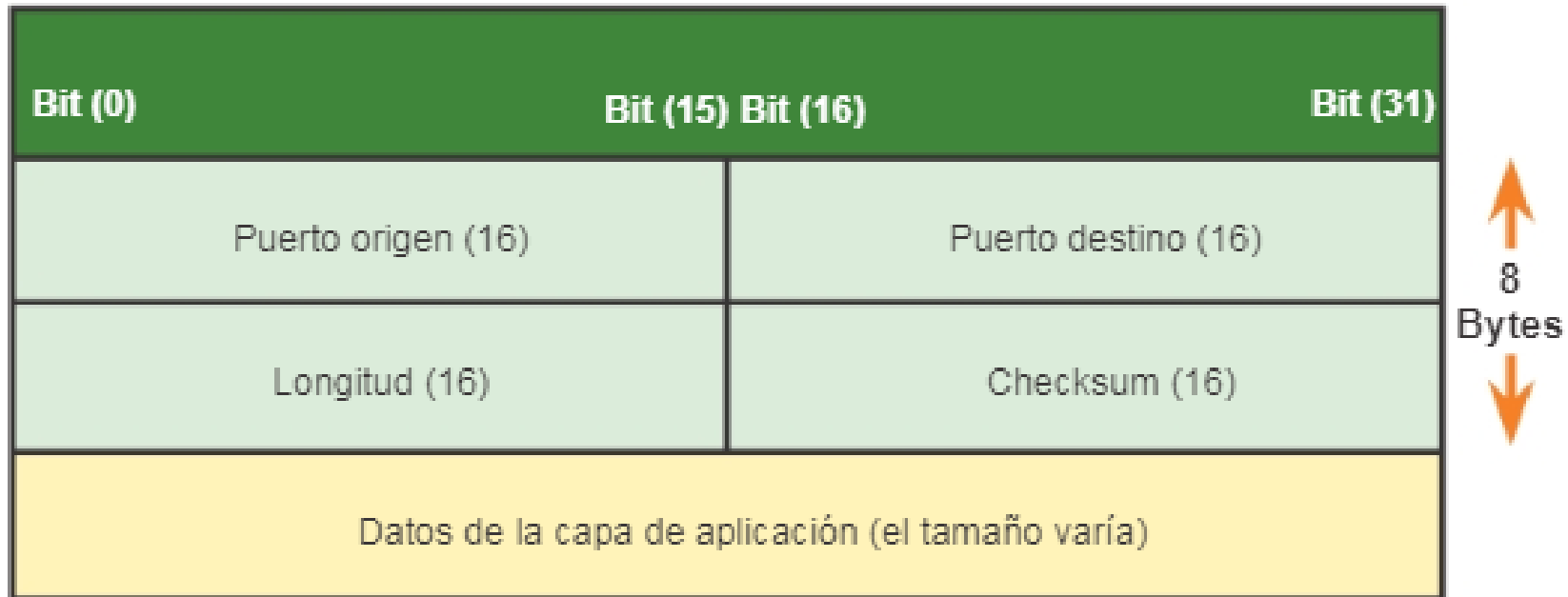


Figura 9: Encabezado del Segmento UDP[2].

2.5- SEPARACIÓN DE COMUNICACIONES MÚLTIPLES



Figura 10: Direccionamiento del Puerto [2].

2.6- DIRECCIONAMIENTO DE PUERTOS TCP Y UDP



Los datos de las distintas aplicaciones se dirigen a la aplicación correcta, ya que cada aplicación tiene un número de puerto único.

Figura 11: Direccionamiento del Puerto [2].

2.6- DIRECCIONAMIENTO DE PUERTOS TCP Y UDP

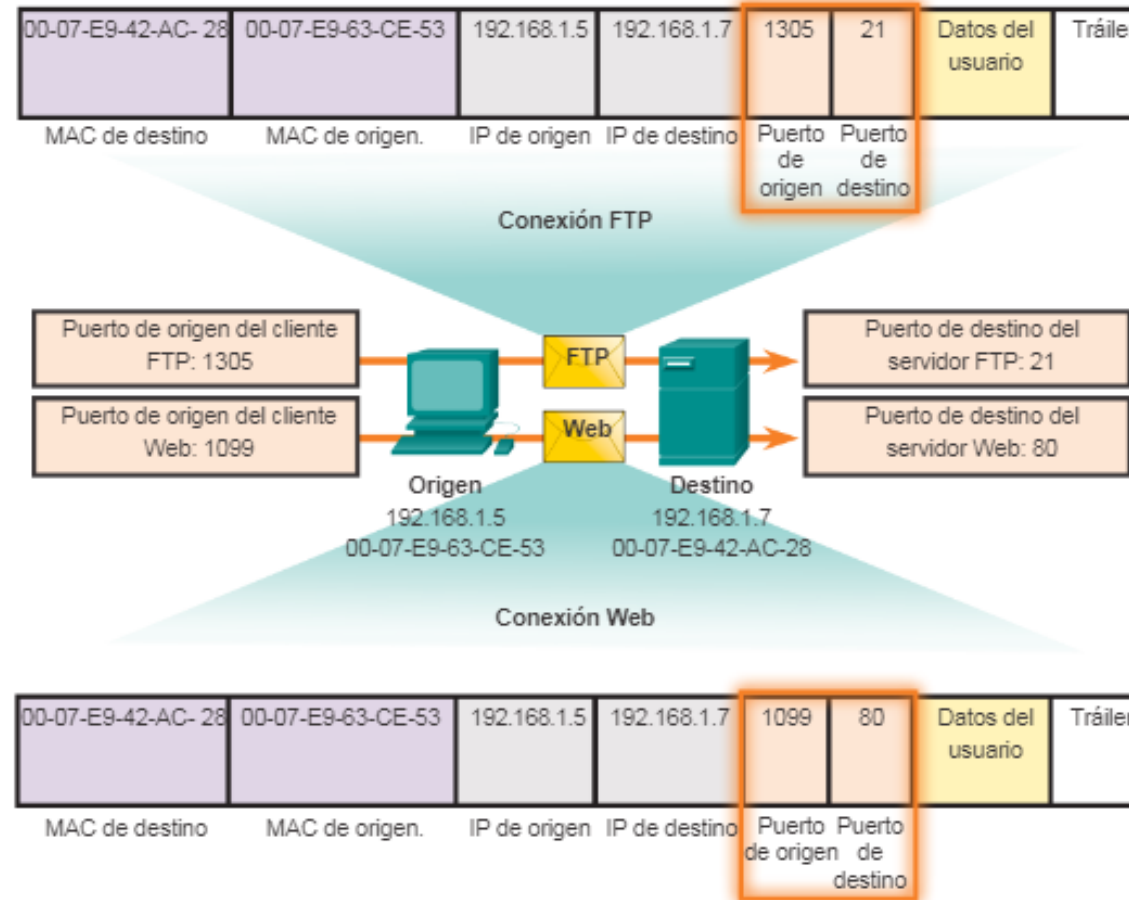


Figura 12: Ejemplo de uso de puertos entre cliente/servidor [2].

2.7- ASIGNACIÓN DE PUERTOS POR IANA

La Agencia de asignación de números por Internet (IANA) asigna números de puerto. IANA es un organismo normativo responsable de asegurar diferentes estándares de direccionamiento. Existen diferentes tipos de números de puerto:

- Puertos bien conocidos (0 al 1023)
- Puertos registrados (1024 al 49151)
- Puertos dinámicos o privados (49152 al 65535)

2.8- VERIFICACIÓN DE CONEXIONES EN CAPA DE TRANSPORTE



```
C:\Users\Jorge>netstat
```

```
Conexiones activas
```

Proto	Dirección local	Dirección remota	Estado
TCP	192.168.1.10:55081	37.156.185.135:http	ESTABLISHED
TCP	192.168.1.10:55085	64.94.18.65:https	ESTABLISHED
TCP	192.168.1.10:55091	sea03-014:http	ESTABLISHED
TCP	192.168.1.10:55144	13.107.42.11:https	ESTABLISHED
TCP	192.168.1.10:55170	ce-in-f188:https	ESTABLISHED
TCP	192.168.1.10:58309	104.208.156.39:https	ESTABLISHED
TCP	192.168.1.10:58721	whatsapp-cdn-shv-02-gru2:https	ESTABLISHED
TCP	192.168.1.10:58748	us1:https	ESTABLISHED
TCP	192.168.1.10:58758	13.89.118.161:https	TIME_WAIT
TCP	192.168.1.10:58771	13.89.118.161:https	TIME_WAIT
TCP	192.168.1.10:58772	gru09s09-in-f131:https	TIME_WAIT

Figura 13: Uso del comando **netstat** en CMD. Fuente Propia.

3- COMUNICACIÓN TCP

3.1- ENTREGA CONFIABLE

La confiabilidad de la comunicación TCP se obtiene con el uso de sesiones orientadas a la conexión. TCP inicia un proceso para crear una conexión con el destino. Esta conexión con estado permite hacer un seguimiento de una sesión o un stream de comunicación entre los hosts. Este proceso asegura que cada host tenga conocimiento del stream de comunicación y se prepare para este. Una conversación TCP requiere que se establezca una sesión entre hosts en ambas direcciones.

El destino envía acuses de recibo al origen por los segmentos que recibe. Cuando el origen recibe un acuse de recibo, reconoce que los datos se entregaron correctamente y puede dejar de rastrearlos. Si el origen no recibe el acuse de recibo dentro de un tiempo predeterminado, retransmite esos datos al destino.

3.2- PROCESOS DEL SERVIDOR TCP

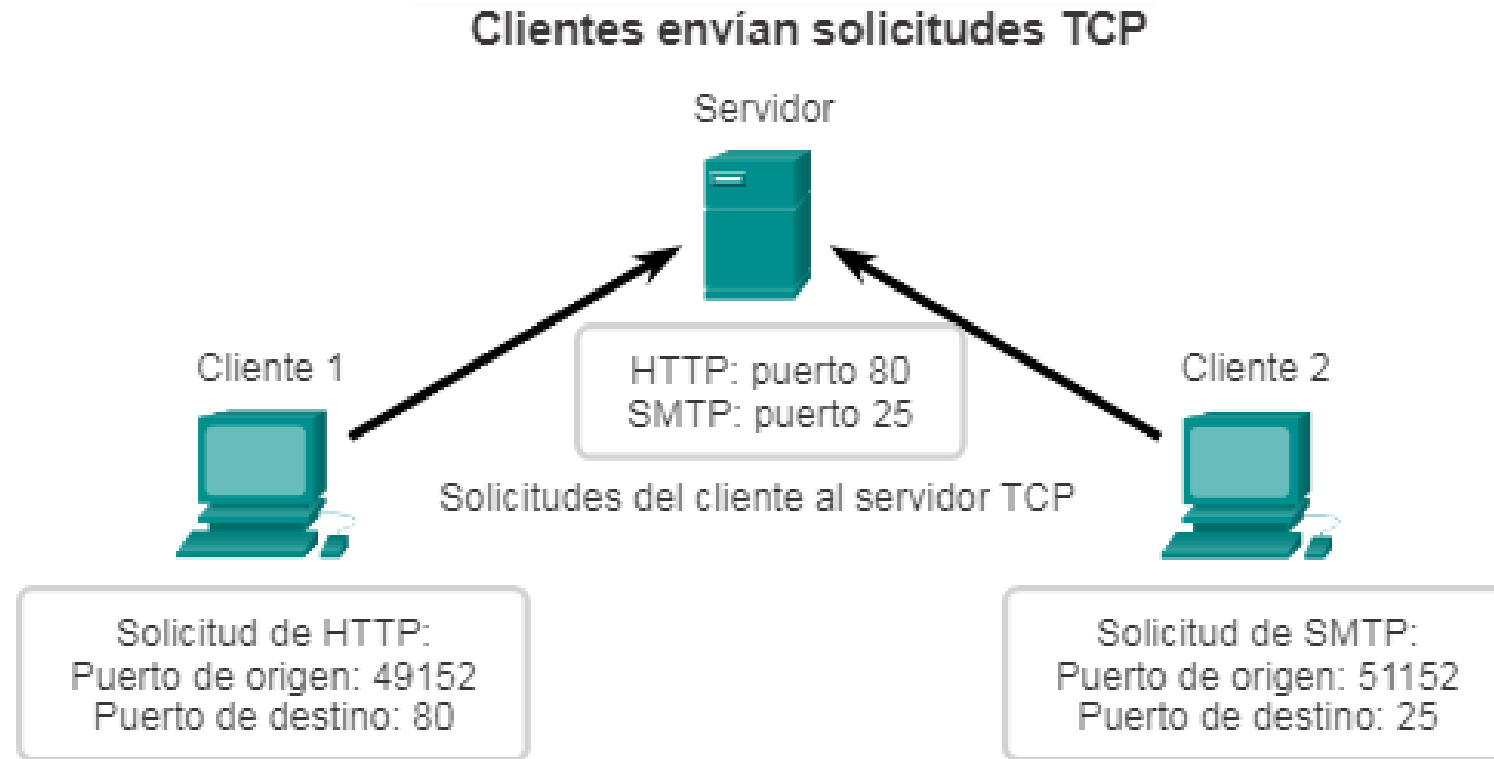


Figura 14: Clientes envían solicitud TCP [2].

3.2- PROCESOS DEL SERVIDOR TCP

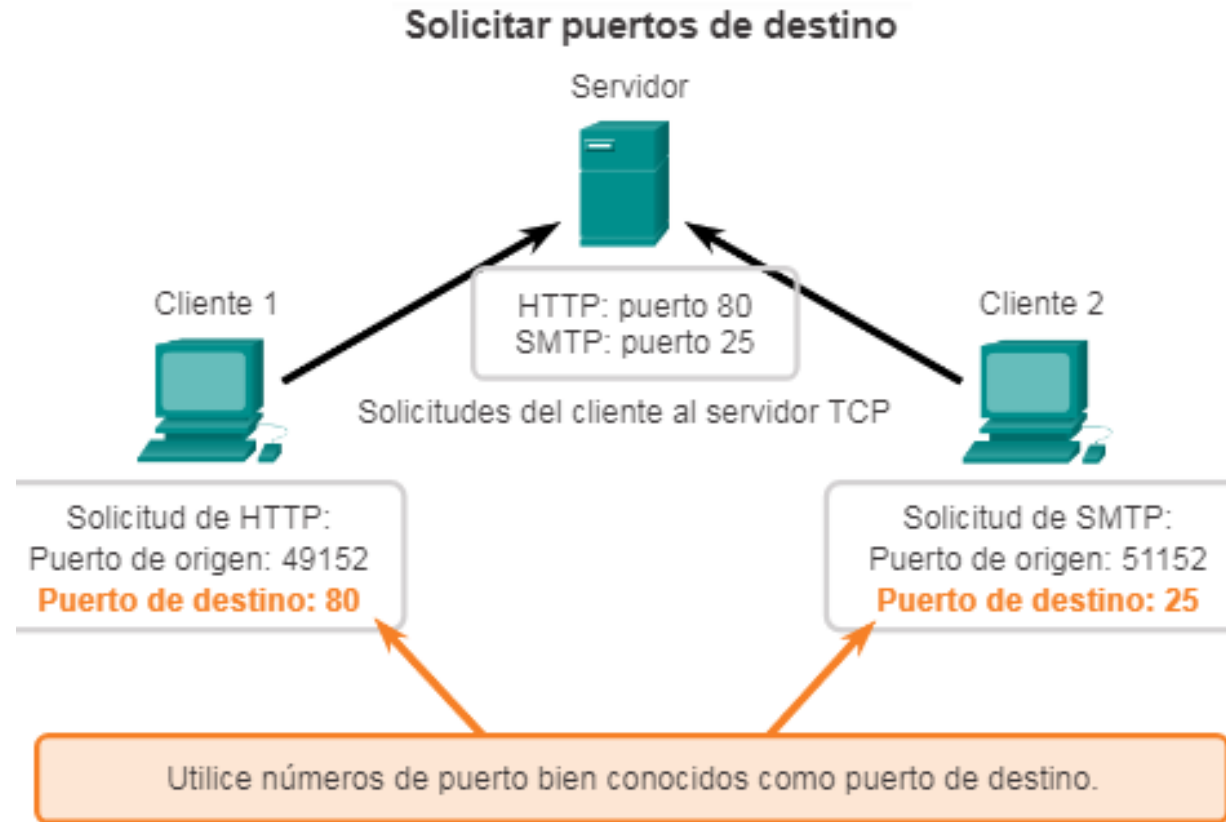


Figura 15: Se solicitan los puertos de destino[2].

3.2- PROCESOS DEL SERVIDOR TCP

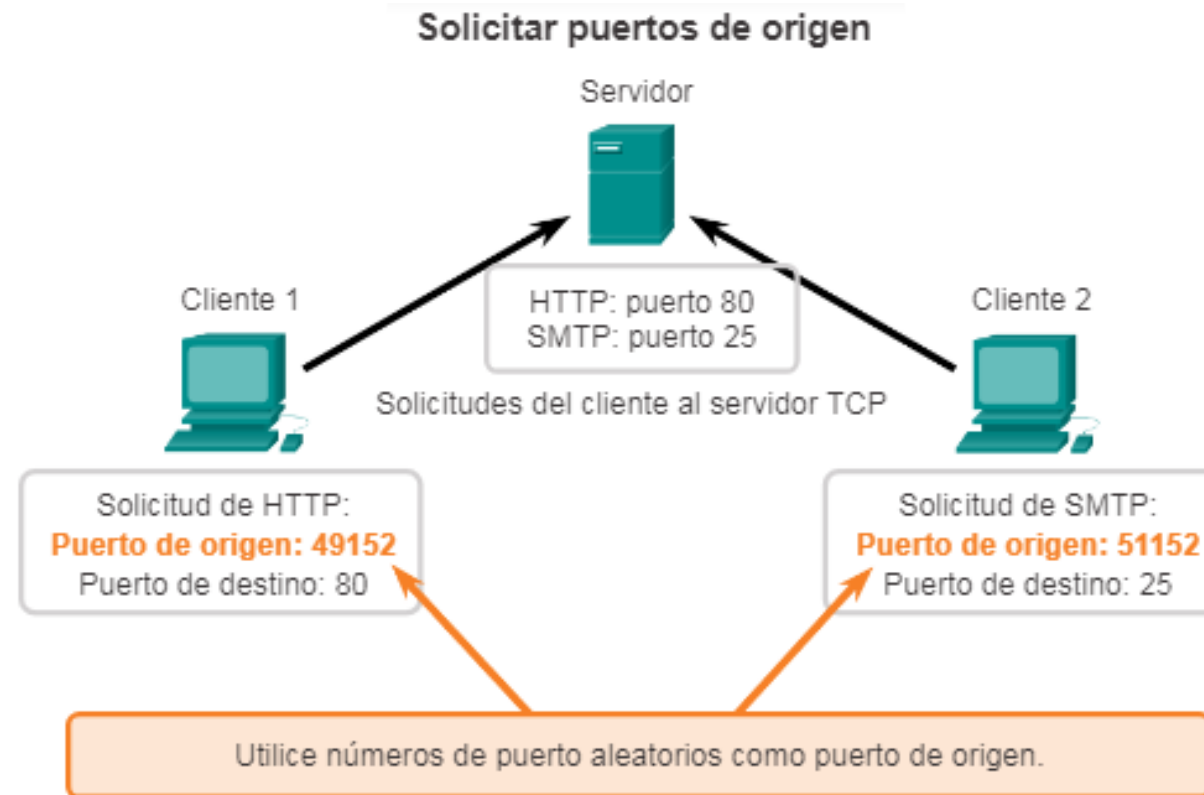


Figura 16: Se solicitan los puertos de origen[2].

3.2- PROCESOS DEL SERVIDOR TCP

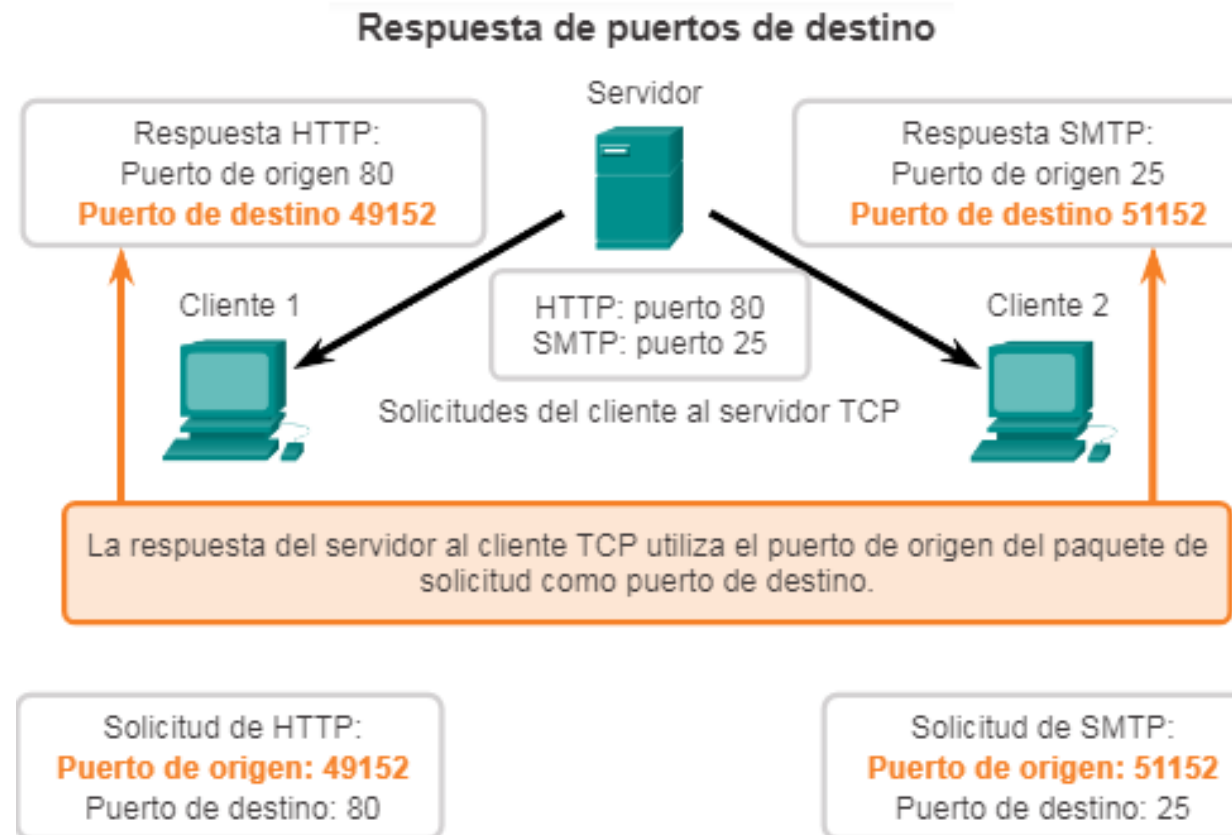


Figura 17: Respuesta de puertos de destino[2].

3.2- PROCESOS DEL SERVIDOR TCP

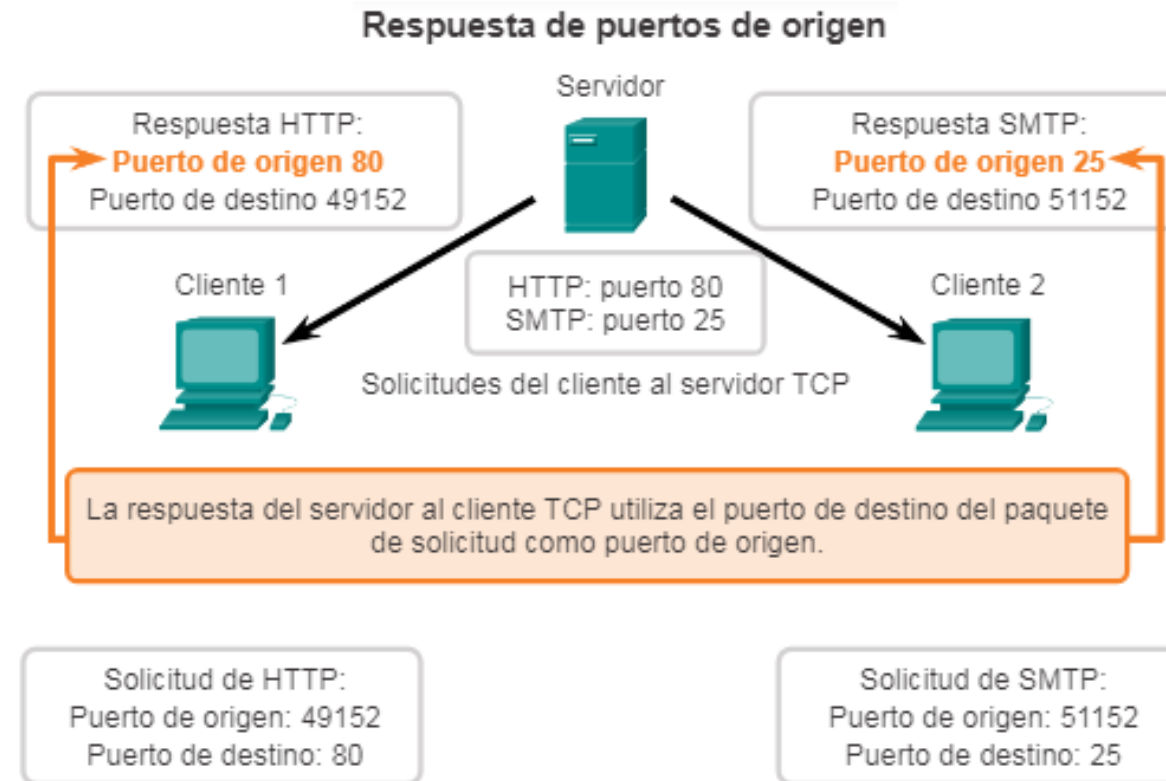


Figura 18: Respuesta de puertos de origen[2].

3.3- ESTABLECIMIENTO DE CONEXIÓN TCP



Figura 18: Establecimiento de conexión TCP [2].

3.4- FINALIZACIÓN DE CONEXIÓN TCP

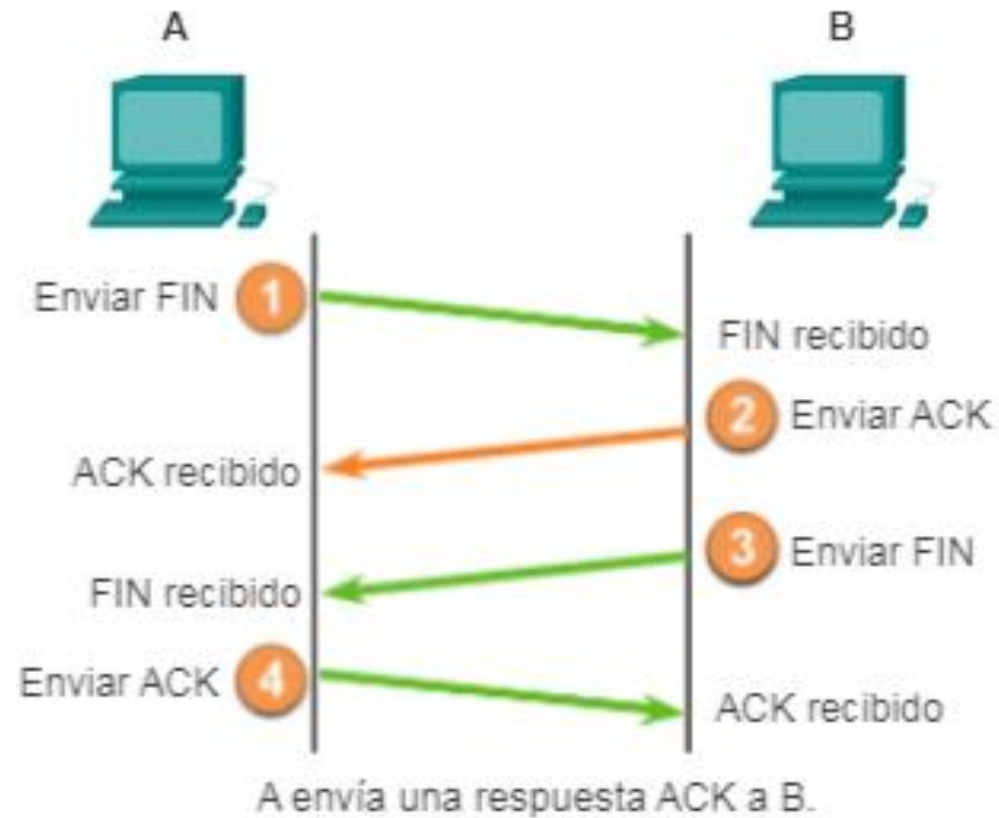
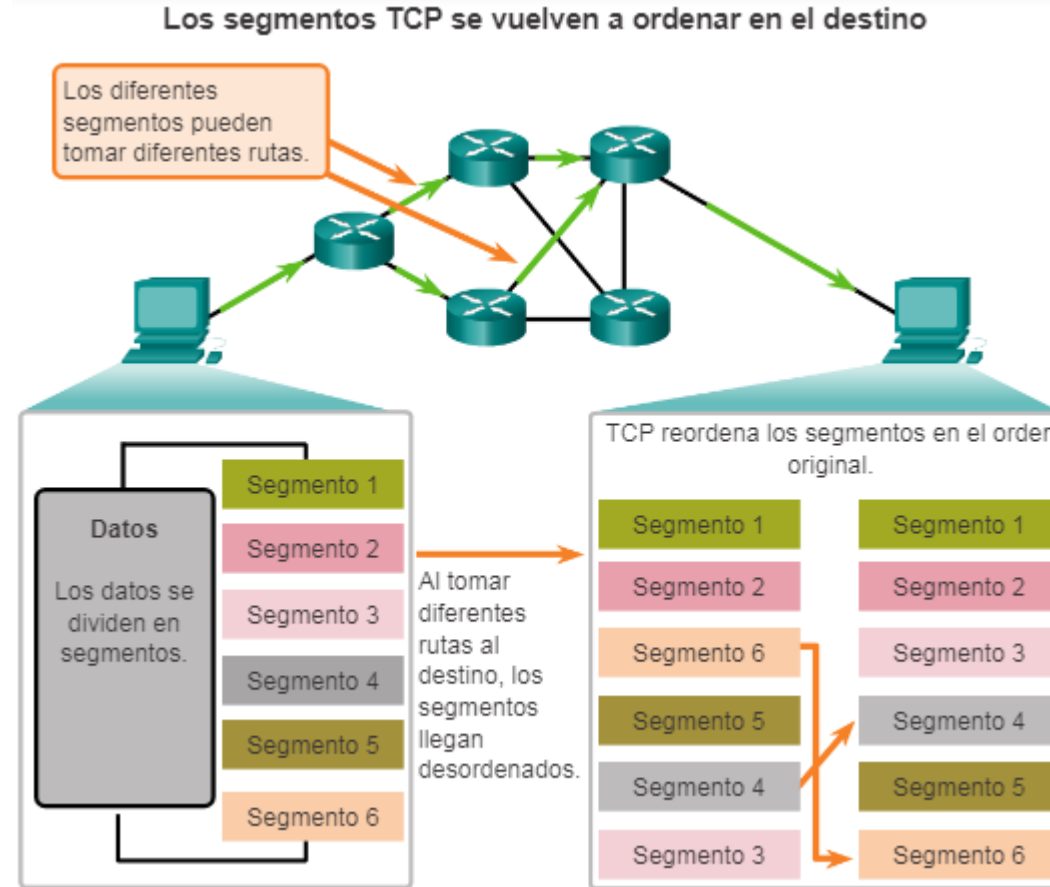


Figura 19: Finalización de conexión TCP [2].

4- CONFIABILIDAD Y CONTROL DE FLUJO

4.1 ENTREGA ORGENADA



Redes de Computadoras II **Figura 20:** Reordenamiento de Segmentos mediante TCP [2].

4.2- ACUSE DE RECIBO DE SEGMENTOS

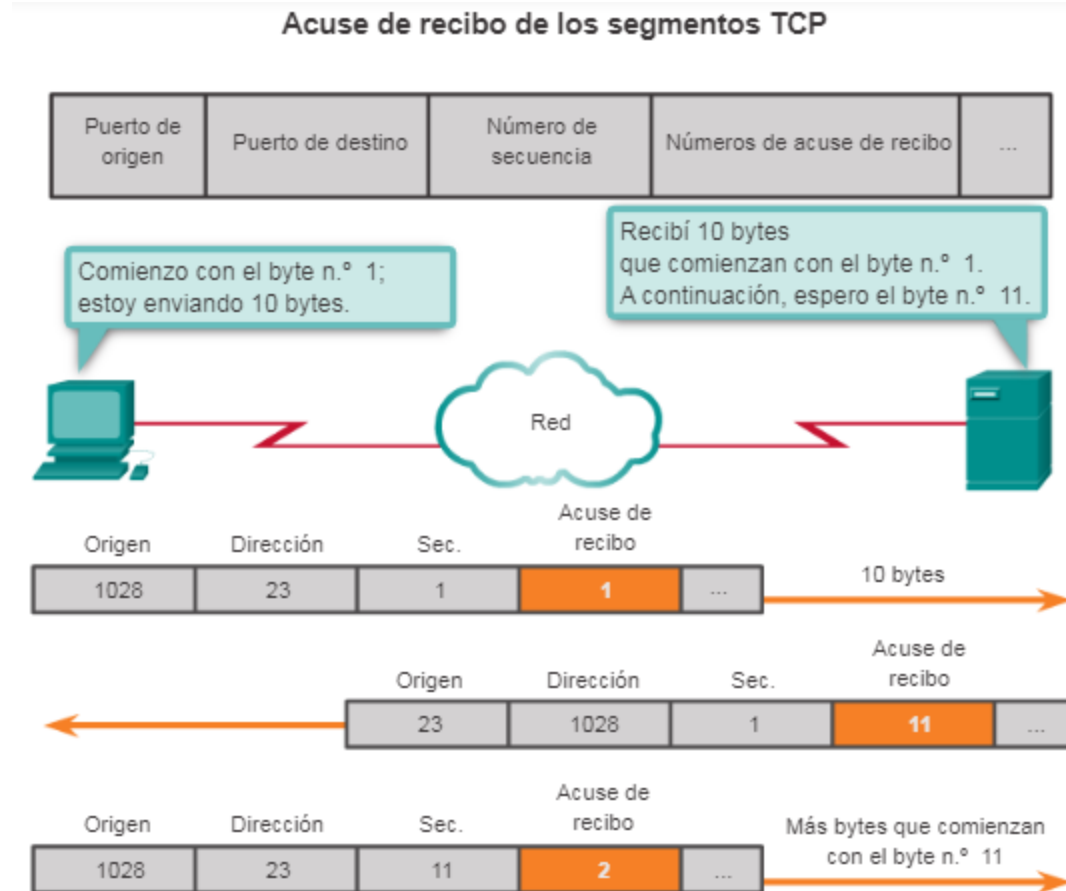


Figura 21: Acuse de Recibo de Segmentos mediante TCP [2].

4.3- PERDIDA Y REENVÍO DE SEGMENTOS

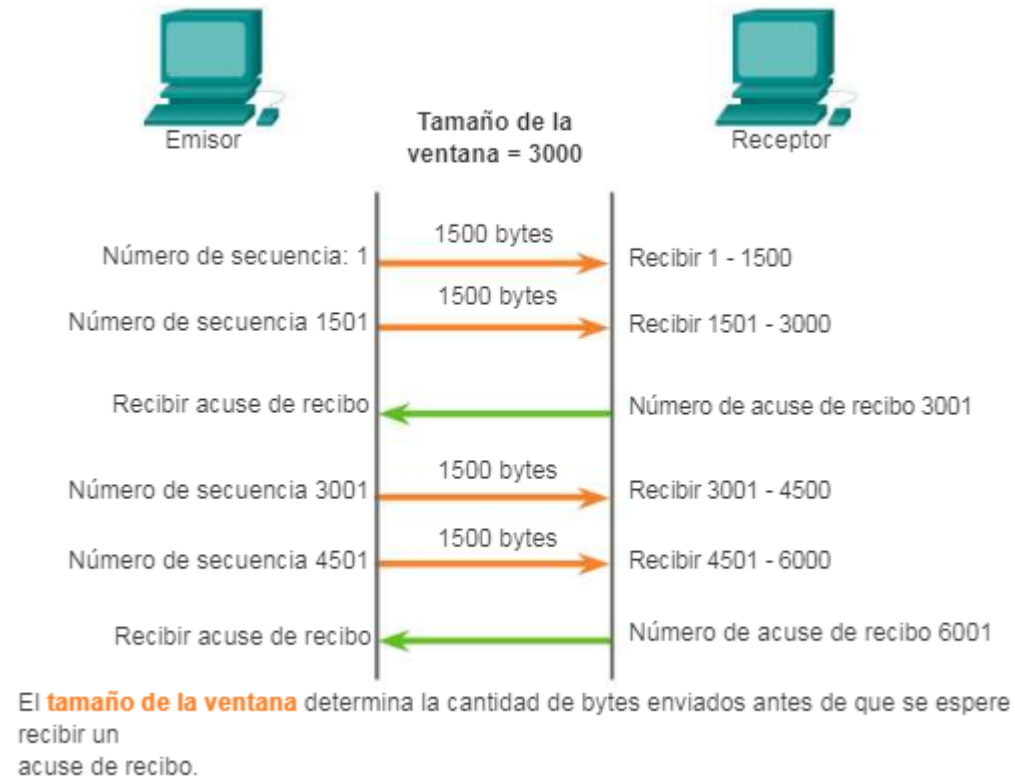
La pérdida de datos se produce en ocasiones, sin importar que tan bien diseñada este la red; por lo tanto, TCP proporciona métodos para administrar estas pérdidas de segmentos.

Cuando el TCP en el host de origen no recibe un acuse de recibo después de una cantidad de tiempo predeterminada, este vuelve al ultimo numero de ACK recibido y vuelve a transmitir los datos desde ese punto en adelante.

En la actualidad, los hosts pueden emplear también una característica optativa llamada "acuses de recibo selectivos" (SACK). Si ambos hosts admiten los SACK, es posible que el destino acuse recibo de los bytes de segmentos discontinuos, y el host solo necesitara volver a transmitir los datos perdidos.

4.4- CONTROL DE FLUJO

Acuse de recibo y tamaño de la ventana del segmento TCP

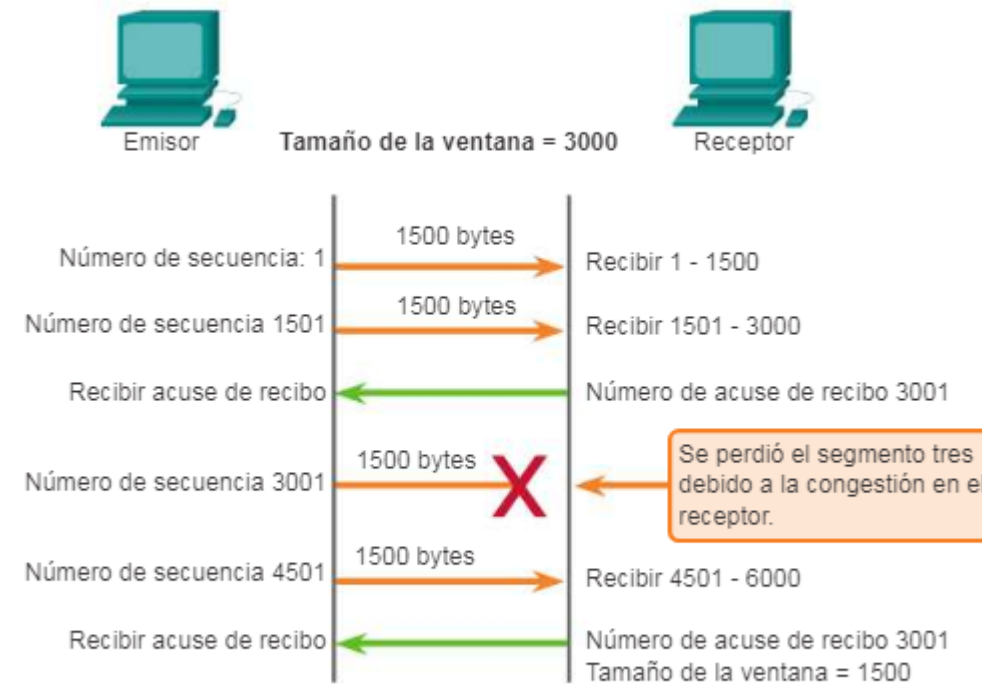


El número de **acuse de recibo** es el número del siguiente byte previsto.

Figura 22: Acuse de Recibo y Tamaño de la Ventana del Segmento TCP [2].

4.4- CONTROL DE FLUJO

Congestión y control del flujo de TCP



Si se pierden los segmentos debido a la congestión, el receptor acusará recibo del último segmento secuencial recibido y responderá con un tamaño de ventana reducido.

Figura 23: Reducción del Tamaño de la Ventana del Segmento TCP [2].

5- COMUNICACIÓN UDP

5.1- BAJA SOBRECARGA Y POCA CONFIABILIDAD

UDP es un protocolo simple que proporciona las funciones básicas de la capa de transporte. Tiene una sobrecarga mucho menor que TCP, ya que no está orientado a la conexión y no proporciona los mecanismos sofisticados de retransmisión, secuenciación y control del flujo que ofrecen confiabilidad.

Esto no significa que las aplicaciones que utilizan UDP sean siempre poco confiables ni que UDP sea un protocolo inferior. Solo quiere decir que estas funciones no las proporciona el protocolo de la capa de transporte, y se deben implementar aparte, si fuera necesario.

5.2- REENSAMBLE DE DATAGRAMAS UDP

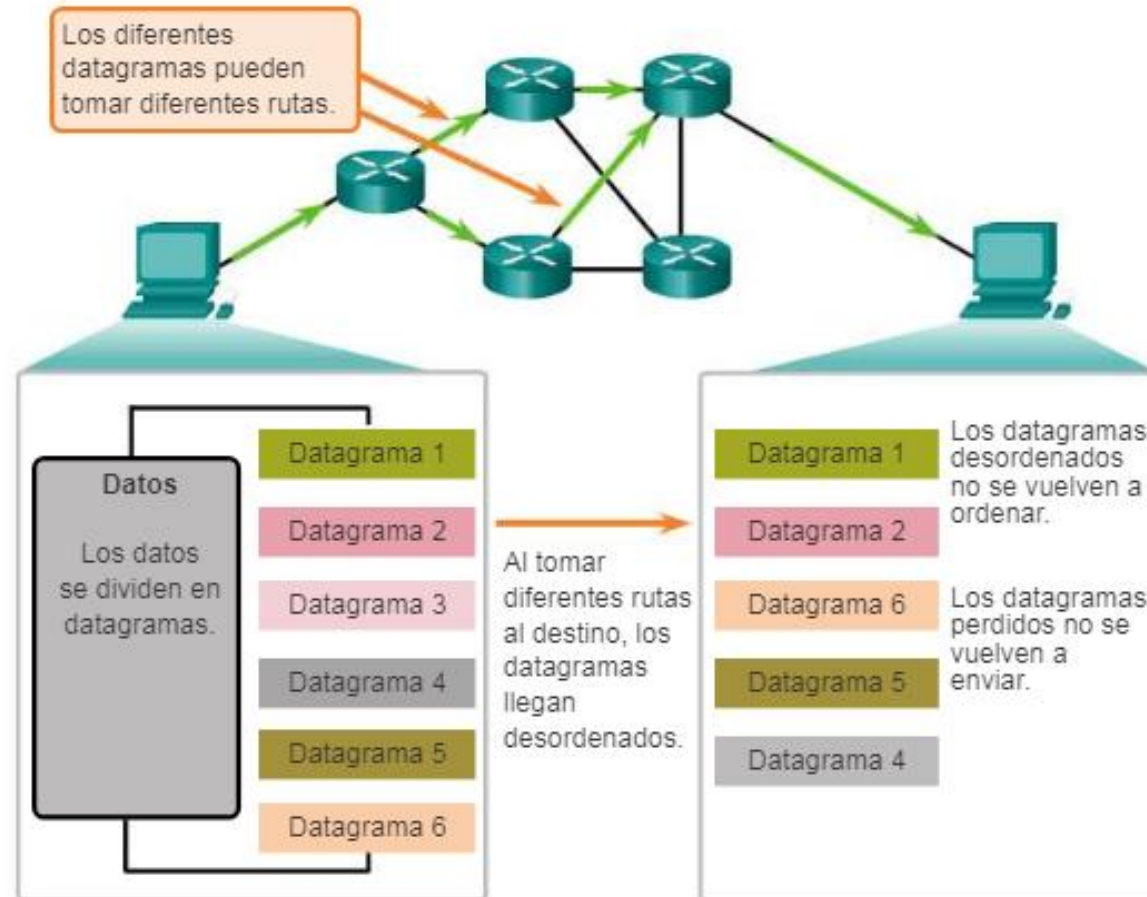


Figura 24: UDP no orientado a conexión[2].

5.3- PROCESOS Y SOLICITUDES DEL SERVIDOR UDP

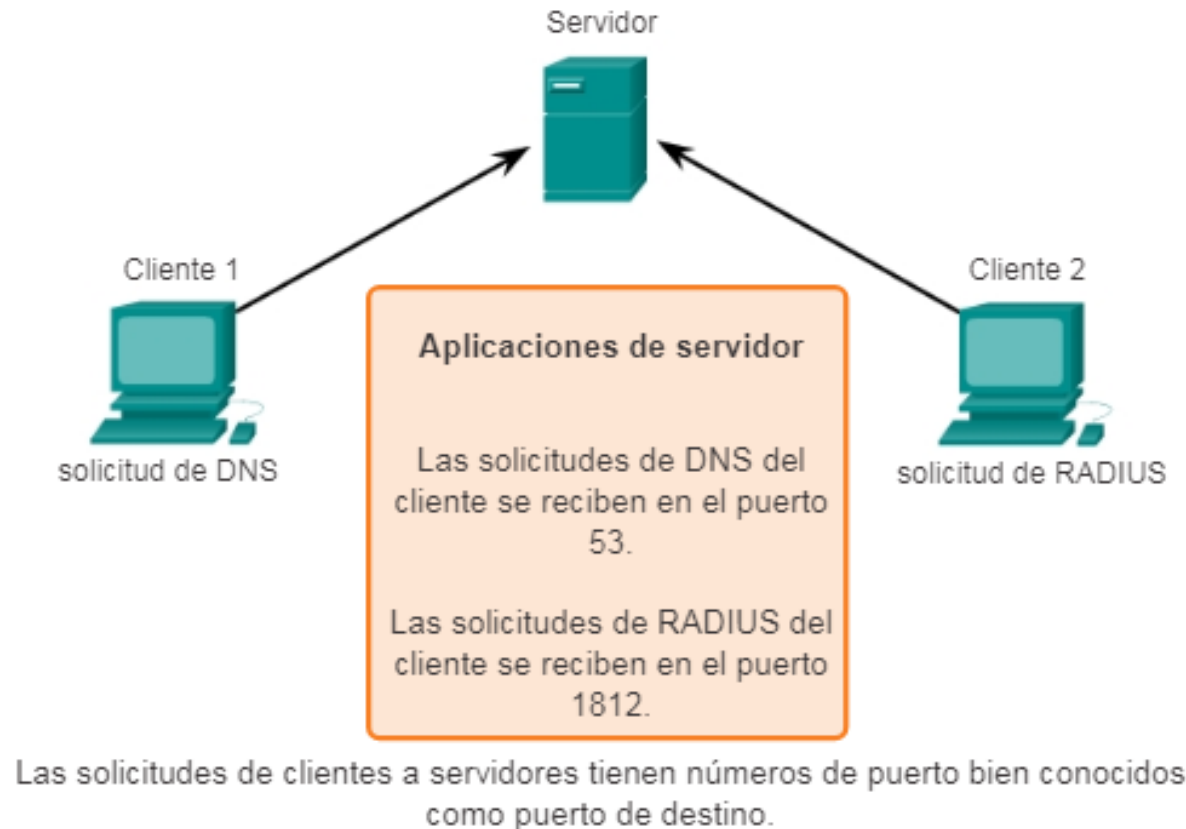


Figura 25: Servidor UDP a la escucha de clientes [2].

5.4- PROCESOS Y SOLICITUDES DEL CLIENTE UDP

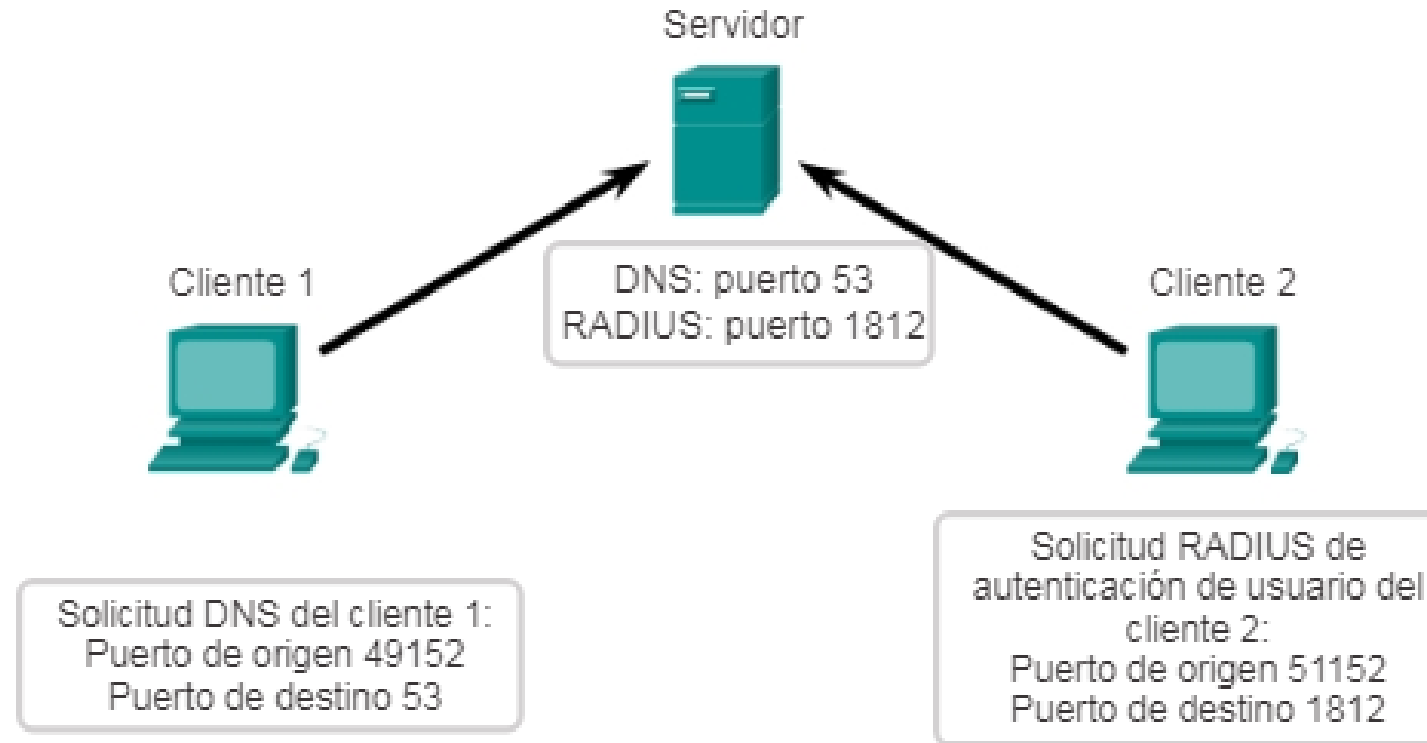


Figura 26: Clientes envían solicitudes [2].

5.4- PROCESOS Y SOLICITUDES DEL CLIENTE UDP

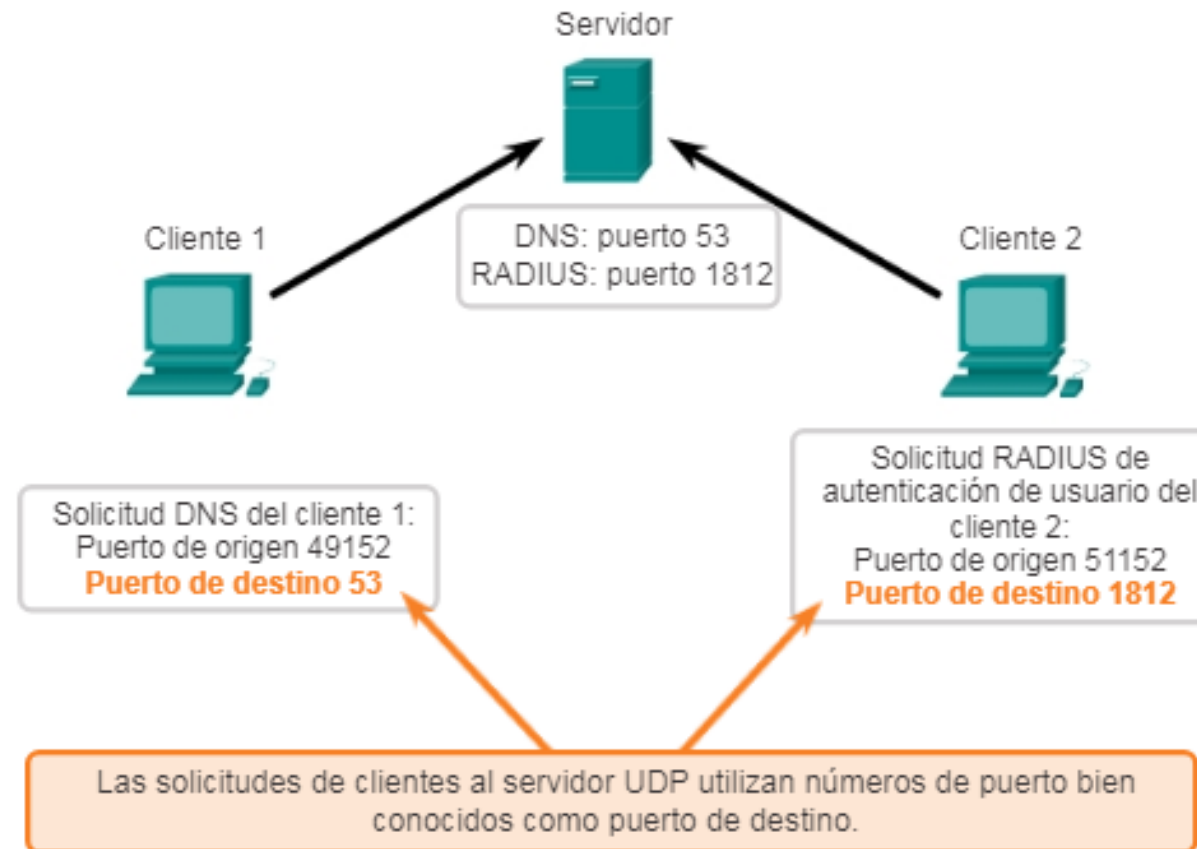


Figura 27: Clientes solicitan puertos de destino [2].

5.4- PROCESOS Y SOLICITUDES DEL CLIENTE UDP

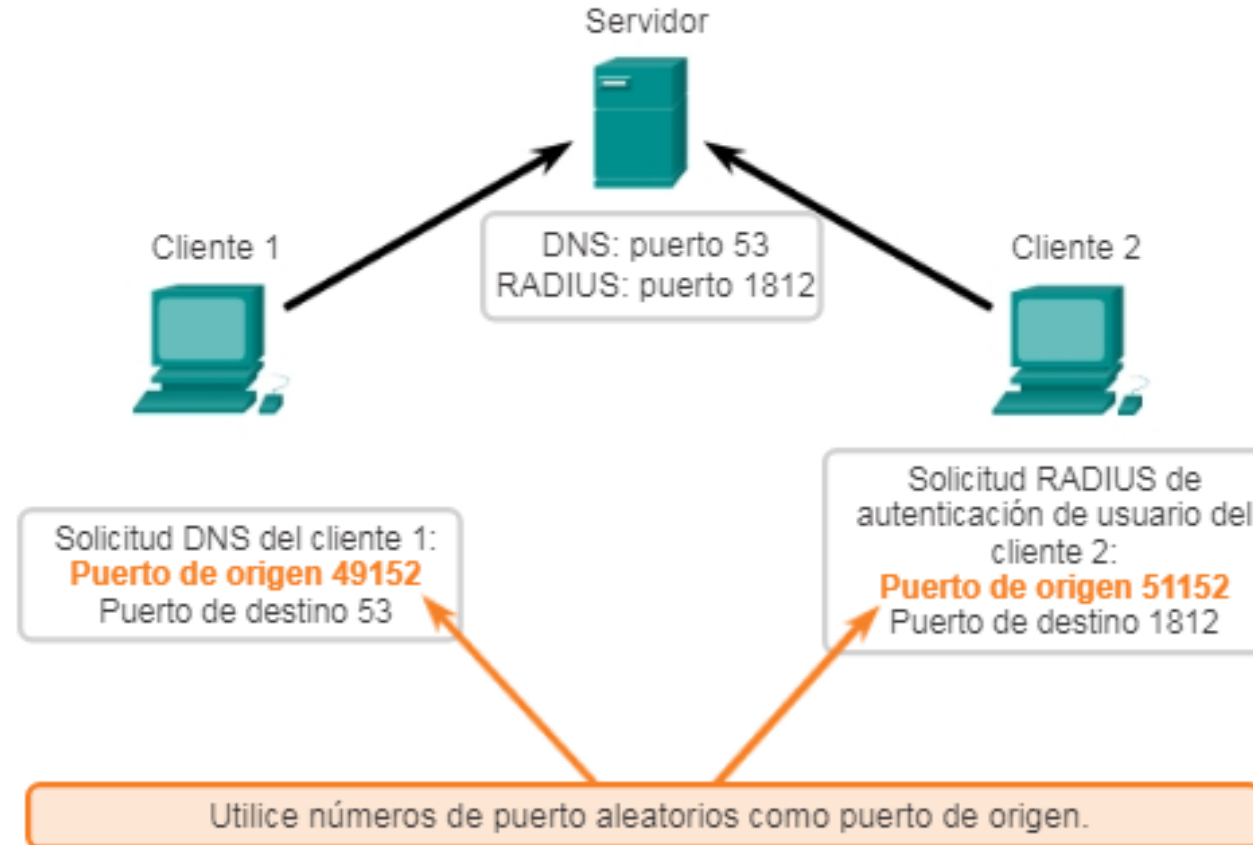


Figura 28: Se definen los puertos de origen en el cliente [2].

5.4- PROCESOS Y SOLICITUDES DEL CLIENTE UDP

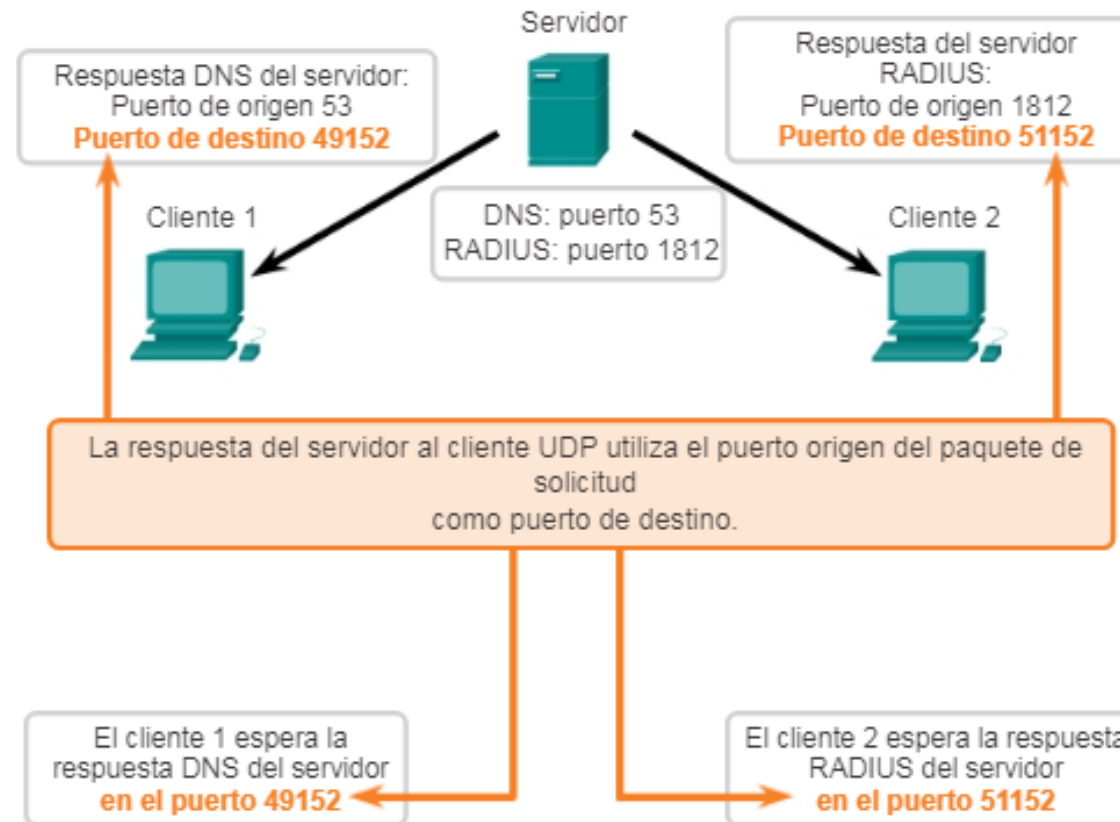


Figura 29: Respuesta de puerto de destino [2].

5.4- PROCESOS Y SOLICITUDES DEL CLIENTE UDP

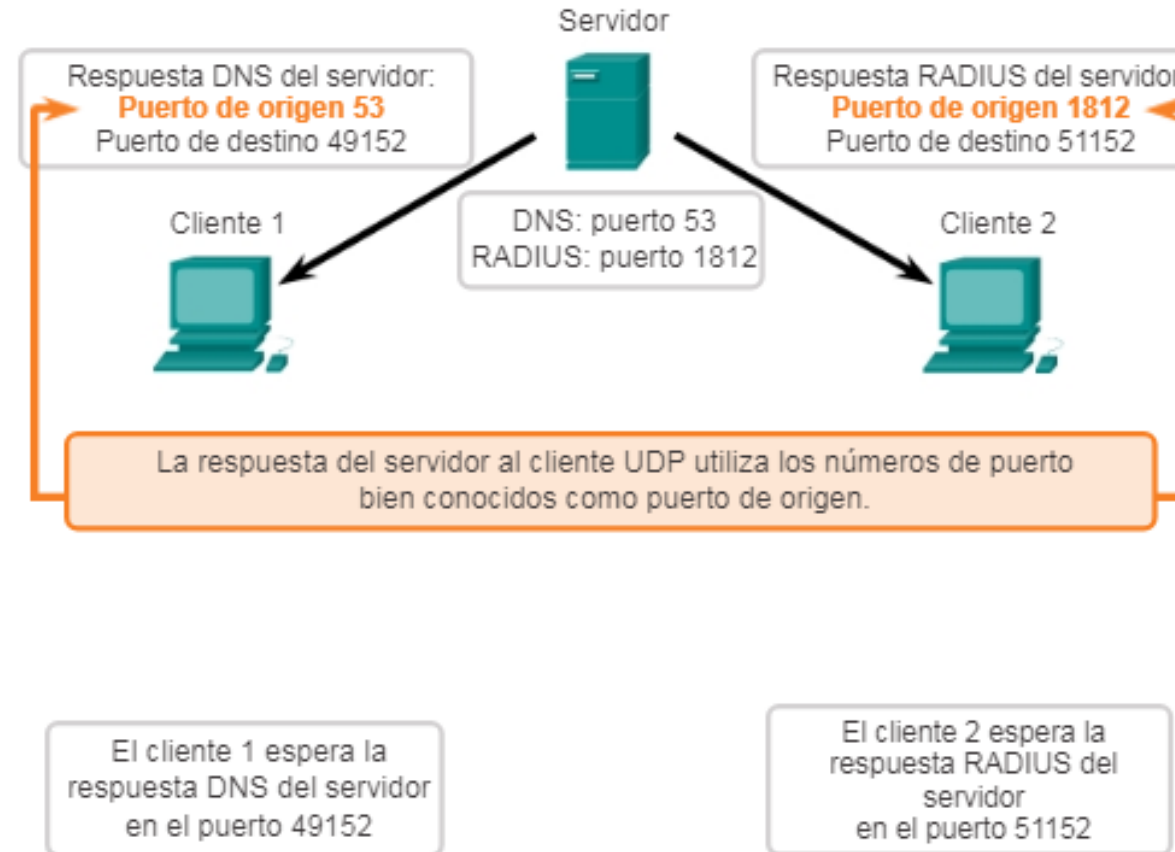


Figura 30: Respuesta de puerto de origen [2].

6.1 - CUANDO UTILIZAR TCP

Muchas aplicaciones requieren confiabilidad y otros servicios que proporciona TCP. Estas son aplicaciones que pueden tolerar cierto grado de demora o pérdida de rendimiento debido a la sobrecarga que impone TCP. Esto hace que TCP sea mas adecuado para las aplicaciones que necesitan transporte confiable y que pueden tolerar cierta demora. Algunos ejemplos de aplicaciones bien conocidas que utilizan TCP incluyen las siguientes:

- Protocolo de transferencia de hipertexto (HTTP)
- Protocolo de transferencia de archivos (FTP)
- Protocolo simple de transferencia de correo (SMTP)
- Telnet

6.1 - CUANDO UTILIZAR UDP

Existen tres tipos de aplicaciones que son las mas adecuadas para UDP:

- Aplicaciones que pueden tolerar cierta perdida de datos, pero requieren retrasos cortos o que no haya retrasos
- Aplicaciones con transacciones de solicitud y respuesta simples
- Comunicaciones unidireccionales donde no se requiere confiabilidad o donde la aplicación la pueda administrar

6.1 - CUANDO UTILIZAR UDP

Existen tres tipos de aplicaciones que son las mas adecuadas para UDP:

- Aplicaciones que pueden tolerar cierta perdida de datos, pero requieren retrasos cortos o que no haya retrasos
- Aplicaciones con transacciones de solicitud y respuesta simples
- Comunicaciones unidireccionales donde no se requiere confiabilidad o donde la aplicación la pueda administrar

Ejemplos de protocolos que usan UDP: DHCP, DNS (también puede utilizar TCP), SNMP, TFTP, VoIP, IPTV

REFERENCIAS BIBLIOGRÁFICAS

- 1) A. Tanenbaum, 2012. Redes de Computadoras. 5ta Edición
- 2) NetAcad CISCO, Material de Lectura de CCNA 1 v6.0, 2018.



**MUCHAS GRACIAS POR SU
ATENCIÓN !!!**