Redes de Computadoras
Práctica: La capa de transporte

Problema 1
¿Qué es UDP? Dibuje el esquema básico del paquete. Ejemplifique para qué puede ser utilizado.

UDP, o Protocolo de Datagramas de Usuario, es un protocolo de comunicación en la capa de transporte del modelo OSI que proporciona un servicio de comunicación sin conexión y no orientado a la conexión. Es decir, UDP no establece una conexión antes de enviar datos y no garantiza la entrega ordenada o fiable de los datos. UDP es más ligero que TCP y se utiliza en aplicaciones donde la velocidad y la eficiencia son más importantes que la integridad de los datos.

El esquema básico de un paquete UDP consta de los siguientes campos:

Puerto de origen	Puerto de destino	Longitud del datagrama	Suma de verificación
Datos	Datos	Datos	Datos

1. Puerto de origen: Identifica el puerto de origen del emisor.
2. Puerto de destino: Identifica el puerto de destino del receptor.
3. Longitud del datagrama: Indica la longitud total del datagrama UDP, incluyendo el encabezado y los datos.
4. Suma de verificación: Proporciona una suma de verificación opcional para la integridad de los datos.
5. Datos: Contiene los datos transmitidos.

UDP puede ser utilizado en aplicaciones donde la pérdida ocasional de datos no es crítica o donde se requiere una comunicación rápida y eficiente, como:

- Transmisión de audio y video en tiempo real.
- Aplicaciones de juegos en línea.
- Protocolos de enrutamiento en redes de computadoras.
- Transmisión de datos en aplicaciones de Internet de las cosas (IoT).
- Resolución de nombres de dominio (DNS).

Problema 2
¿Qué diferencias hay entre UDP y TCP con respecto a los servicios que ofrece? ¿En qué escenarios es preferible cada uno?

Diferencias entre UDP y TCP

.

Confiabilidad:

TCP es confiable, asegura que los paquetes lleguen sin errores y en orden, usando acuses de recibo y retransmisiones.

UDP es no confiable, no garantiza la entrega ni el orden de los paquetes, pero tiene menos sobrecarga.

Velocidad y Sobrecarga:

TCP es más lento y tiene más sobrecarga debido a sus mecanismos de control.

UDP es más rápido y eficiente, ideal para situaciones donde la velocidad es crítica.

Escenarios Preferibles

UDP es preferible cuando la rapidez es crítica y se pueden tolerar pérdidas de paquetes. Ejemplos: Streaming de video y audio, juegos en línea, DNS.

TCP es preferible cuando se necesita garantizar que todos los datos lleguen completos y en orden. Ejemplos: Aplicaciones web, transferencia de archivos, correos electrónicos.

Por lo tanto, el uso de UDP o TCP depende de si la prioridad es la velocidad o la confiabilidad.

Problema 3

Ofrezca algunos ejemplos de parámetros de protocolo que podrían negociarse al establecerse una conexión TCP

Tamaño de la ventana (Window Size): Define cuántos datos se pueden enviar antes de necesitar confirmación. Es ajustado para optimizar el uso del ancho de banda.

Tamaño máximo del segmento (MSS): Es el tamaño máximo de los segmentos de datos que se pueden enviar sin necesidad de dividirlos en partes más pequeñas. Ayuda a hacer la transmisión más eficiente.

Opciones de escala de ventana: Permite manejar tamaños de ventana más grandes de lo normal, para soportar altas velocidades de transmisión.

Opción de timestamps: Incluye un sello de tiempo en los paquetes para medir el tiempo de ida y vuelta y mejorar el control de la congestión.

Selección de algoritmo de control de congestión: Algunos sistemas permiten elegir entre diferentes métodos para manejar la congestión de la red.

Selección rápida (Fast Open): Permite enviar datos durante el proceso de establecimiento de la conexión, reduciendo la latencia para conexiones sucesivas.

Estos parámetros ayudan a optimizar la conexión TCP para que sea eficiente y confiable según las condiciones de la red y las capacidades de los dispositivos.

Problema 4
¿Por qué existe UDP? ¿No sería suficiente dejar que el proceso de nivel 7 simplemente envíe paquetes IP?

UDP (Protocolo de Datagramas de Usuario) existe porque ofrece ventajas específicas que el envío directo de paquetes solo a nivel IP no proporciona:

Multiplexación: Permite a varias aplicaciones usar la misma dirección IP simultáneamente mediante puertos diferenciados.
Chequeo de Errores: Incluye una suma de verificación para asegurar la integridad de los datos enviados.

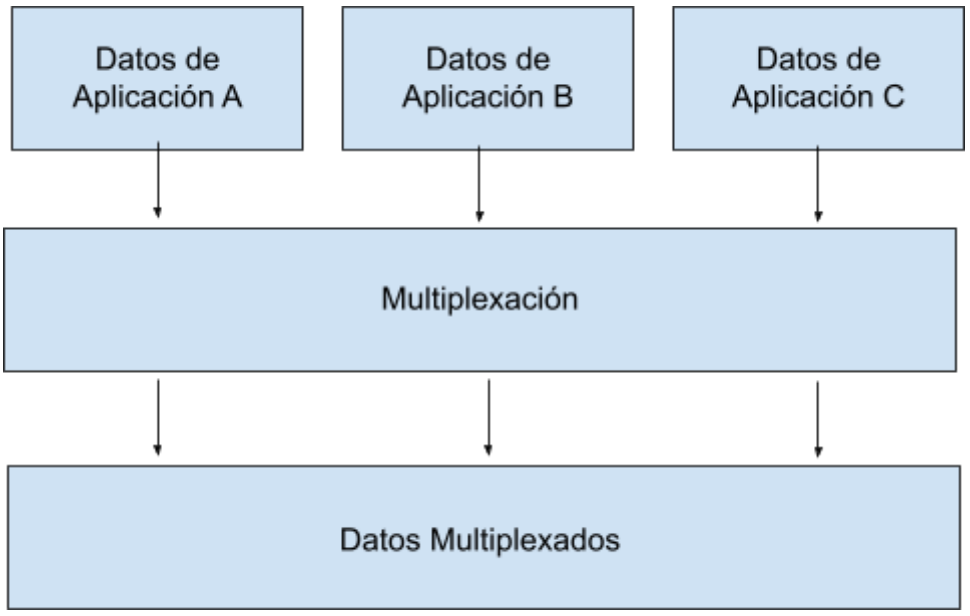
Eficiencia: UDP es más rápido porque no establece conexión ni maneja estados, ideal para aplicaciones que requieren alta velocidad y pueden tolerar pérdidas de datos, como el streaming de video o juegos en línea.

Control Flexible: Ofrece a las aplicaciones la libertad de implementar su propio control de transmisión adaptado a sus necesidades específicas.

Estas características hacen que UDP sea valioso para situaciones donde la eficiencia y la flexibilidad son más críticas que la fiabilidad absoluta.

Problema 5
Explique con la ayuda de un diagrama qué se entiende por multiplexación en el nivel de transporte

La multiplexación en el nivel de transporte se refiere al proceso de combinar múltiples flujos de datos de aplicaciones diferentes en un único flujo de datos para su transmisión a través de un canal de comunicación compartido, como una conexión de red.



En este diagrama, los datos de las aplicaciones A, B y C se multiplexan juntos para formar un único flujo de datos multiplexados. Estos datos pueden luego ser transmitidos a través de un canal de comunicación compartido, como una conexión de red. La multiplexación permite que múltiples aplicaciones utilicen eficientemente el mismo canal de comunicación sin interferir entre sí.

Problema 6
Completar el siguiente cuadro para los protocolos TCP y UDP

Elemento del Protocolo	**TCP**	**UDP**
Establecimiento de conexiones	Sí	No
Particionamiento de mensajes largos en distintos segmentos	Sí	No
Transferencia de segmentos	Sí	Sí
Numeración de los segmentos	Sí	No
Control de flujo de nivel de transporte	Sí	No
Multiplexación	Sí	Sí
Retransmisión debida a timeout	Sí	No
Re secuenciamiento de segmentos	Sí	No
Checksum de nivel de transporte	Sí	Sí

Problema 7
A un proceso en el host 1 le ha sido asignado el puerto p, y a un proceso en el host 2 el puerto q. ¿Es posible que existan dos o más conexiones TCP entre estos dos puertos al mismo tiempo?

Sí, es posible tener dos o más conexiones TCP activas simultáneamente entre el mismo par de puertos en dos hosts, siempre que las conexiones se distingan por sus direcciones IP. En TCP, una conexión es identificada de manera única por un conjunto de cuatro elementos: dirección IP de origen, puerto de origen, dirección IP de destino y puerto de destino. Por lo tanto, aunque los puertos sean los mismos, las conexiones pueden diferenciarse si las direcciones IP de origen y/o destino son diferentes. Esto permite múltiples conexiones simultáneas entre los mismos puertos en diferentes hosts.

Problema 8
¿Con qué tipo de servicio de red está diseñado para trabajar el protocolo TCP? ¿Y UDP?

TCP Tipo de servicio: Orientado a la conexión y confiable.
Características: Garantiza la entrega de datos sin errores y en orden, adecuado para aplicaciones donde la precisión y la confiabilidad son críticas (por ejemplo, correo electrónico, transferencia de archivos web).

UDP Tipo de servicio: No orientado a la conexión y menos confiable.
Características: Ofrece entrega de datos rápida y con mínima sobrecarga, adecuado para aplicaciones que requieren velocidad y pueden tolerar pérdidas de datos (por ejemplo, streaming de video, juegos en línea).

Problema 9
¿Por qué cree que el protocolo UDP no utiliza números de secuencia en los paquetes? ¿O sí los utiliza? ¿Con qué fin los utiliza?

UDP no utiliza números de secuencia en sus paquetes porque está diseñado para ser un protocolo simple y rápido, enfocado en minimizar la sobrecarga. La omisión de números de secuencia reduce la complejidad y mejora la velocidad de transmisión, sacrificando la capacidad de garantizar la entrega y el orden de los paquetes. Esto es adecuado para aplicaciones que priorizan la baja latencia, como streaming de video o juegos en línea, donde la pérdida de algunos datos es aceptable.

Problema 10
El protocolo UDP brinda a su nivel superior servicios:
a. Sin conexión
b. Orientado a conexión
c. Sin conexión u orientado a conexión
d. Ninguna de los anteriores
e. Cualquiera de las anteriores

a. Sin conexión

El protocolo UDP brinda servicios sin conexión. Esto significa que no se establece una conexión antes de enviar los datos y no se garantiza la entrega ni el orden de los paquetes. UDP es un protocolo ligero y eficiente que se utiliza para aplicaciones donde la velocidad y la simplicidad son más importantes que la fiabilidad, como en transmisiones de audio y video en tiempo real, juegos en línea y consultas de DNS.

Problema 11
¿Por qué cree que el protocolo TCP no utiliza números de secuencia en los paquetes? ¿O sí los utiliza? ¿Con qué fin los utiliza?

El protocolo TCP sí utiliza números de secuencia en los paquetes. Estos números cumplen funciones críticas.

Ordenar los datos: Aseguran que los datos lleguen al receptor en el mismo orden en que fueron enviados.
Control de errores: Permiten detectar pérdidas de paquetes y solicitar retransmisiones.
Control de flujo: Ayudan a gestionar cuánto dato puede recibir el receptor sin saturarse.
Los números de secuencia son esenciales en TCP para garantizar una transmisión de datos confiable y ordenada.

Problema 12
Esta es la salida simplificada del comando netstat –an, ejecutado en un servidor. La salida ha sido modificada de manera de introducir algunos errores. Indique los errores que encuentre. Explique para qué puede ser utilizado el comando netstat.

Local Address	Remote Address	State
200.11.163.35.110	200.11.163.155.1454	ESTABLISHED
200.11.163.35.25	200.11.163.19.1626	TIME_WAIT
200.11.163.35.110	172.18.105.129.1579	TIME_WAIT
200.11.163.35.110	200.114.139.238.2751	ESTABLISHED
200.11.163.35.110	200.11.163.110.1924	TIME_WAIT
200.11.163.35.110	200.5.114.77.3888	ESTABLISHED
200.11.163.35.110	64.76.45.189.1348	TIME_WAIT
200.11.163.35.110	200.5.114.77.3888	ESTABLISHED
200.11.163.35.25	200.11.163.19.1624	TIME_WAIT
200.11.163.35.110	200.11.163.135.1454	ESTABLISHED

Los errores en la salida del comando netstat –an son en la columna "Local Address" y "Remote Address", las direcciones IP y los números de puerto deben estar separados por ":". Por ejemplo, en lugar de "200.11.163.35.110" debería ser "200.11.163.35:110".

El comando netstat se utiliza para mostrar las conexiones de red, las tablas de enrutamiento y una variedad de estadísticas de red. Proporciona información sobre las conexiones activas, los puertos escuchados y otros detalles relacionados con la actividad de red en el sistema. Esto puede ser útil para diagnosticar problemas de red, monitorear la actividad de la red y administrar las conexiones de red.

Problema 13
Respuesta
a) Separe las conexiones TCP, explicando el criterio usado para separarlas.

Para separar las conexiones TCP, se utiliza el criterio de observar el número de secuencia (Seq) y el número de reconocimiento (ACK). Cada conexión TCP se caracteriza por tener un conjunto único de direcciones IP y números de puerto de origen y destino, así como un flujo de datos que se refleja en los números de secuencia y de reconocimiento.

Separación de las conexiones TCP:
- Conexión 1:
  - Paquete 1: Inicio de conexión desde 10.1.0.1:1234 a 10.3.2.4:80
  - Paquete 2: Respuesta al inicio de conexión desde 10.3.2.4:80 a 10.1.0.1:1234
  - Paquete 4: Fin de conexión desde 10.1.0.1:1234 a 10.3.2.4:80

- Conexión 2:
  - Paquete 3: Inicio de conexión desde 3.14.15.92:654 a 2.71.82.81:82
  - Paquete 5: Respuesta al inicio de conexión desde 2.71.82.81:82 a 3.14.15.92:654
  - Paquete 7: Fin de conexión desde 3.14.15.92:654 a 2.71.82.81:82
  - Paquete 8: Retransmisión del paquete 7

- Conexión 3:
  - Paquetes 6, 9, 11, 13, 14, 16, 19, 22: Establecimiento de conexión y transmisión de datos entre 10.1.0.1:1234 y 10.3.2.4:80

- Conexión 4:
  - Paquetes 10, 12, 15, 17, 18, 20, 21, 23: Establecimiento de conexión y transmisión de datos entre 2.71.82.81:82 y 3.14.15.92:654

b) Para cada conexión, indique los segmentos que componen el inicio y cierre de la misma.

Para cada conexión TCP:

- Inicio de la conexión: Se identifica cuando se envía un paquete con la bandera SYN en el campo Flags.
- Cierre de la conexión: Se identifica cuando se envía un paquete con la bandera FIN en el campo Flags.

Segmentos de inicio y cierre de cada conexión:

- Conexión 1:
  - Inicio: Paquete 1
  - Cierre: Paquete 4

- Conexión 2:
  - Inicio: Paquete 3
  - Cierre: Paquete 7 , Paquete 8

- Conexión 3:
  - Inicio: Paquete 6
  - Cierre: Paquete 16

- Conexión 4:
  - Inicio: Paquete 10
  - Cierre: Paquete 21

c) En una conexión ocurrió una retransmisión. Identifique el segmento original y la retransmisión.

La retransmisión ocurre en la conexión 2, donde el paquete 8 es una retransmisión del paquete 7 que es el segmento original. Esto se evidencia al observar que ambos tienen el mismo número de secuencia y número de reconocimiento.

Problema 14

Indique cuántas conexiones TCP hay en esta secuencia de segmentos. Asocie cada segmento con cada una de las conexiones.

Indique el criterio usado para determinar a qué conexión pertenece un segmento.

Paq	Orig	Dest	Flags	Seq	Ack	Length
1	157.92.75.5:1024	157.92.23.3:801	S	1	---	0
2	157.92.23.3:801	157.92.75.5:1024	SA	1000000	2	0
3	190.30.132.239:3623	157.92.23.3:801	S	42	---	0
4	157.92.75.5:1024	157.92.23.3:801	S	2	1000001	0
5	157.92.23.3:801	190.30.132.239:3623	SA	204	43	0
6	190.30.132.239:3623	157.92.23.3:801	A	43	205	0
7	157.92.75.5:1024	157.92.23.3:801	A	2	1000001	100
8	157.92.23.3:801	157.92.75.5:1024	A	1000001	102	0
9	157.92.23.3:801	190.30.132.239:3623	—	205	---	40
10	157.92.75.5:1024	157.92.23.3:801	A	202	1000001	100
11	190.30.132.239:3623	157.92.23.3:801	A	43	245	100
12	157.92.23.3:801	157.92.75.5:1024	A	1000001	102	0
13	157.92.75.5:1024	157.92.23.3:801	A	302	1000001	100
14	157.92.23.3:801	190.30.132.239:3623	A	245	143	0
15	190.30.132.239:3623	157.92.23.3:801	F	143	---	0
16	157.92.23.3:801	157.92.75.5:1024	A	1000001	102	0
17	157.92.75.5:1024	157.92.23.3:801	A	402	1000001	100
18	157.92.23.3:801	190.30.132.239:3623	A	245	144	0
19	157.92.23.3:801	157.92.75.5:1024	A	1000001	102	0
20	157.92.75.5:1024	157.92.23.3:80	A	102	1000001	100
21	157.92.23.3:801	190.30.132.239:3623	F	245	---	0
22	157.92.23.3:801	157.92.75.5:1024	A	1000001	502	0
23	157.92.75.5:1024	157.92.23.3:801	A	502	1000001	100
24	157.92.23.3:801	157.92.75.5:1024	A	1000001	602	0
25	157.92.75.5:1024	157.92.23.3:801	FA	602	1000001	0
26	190.30.132.239:3623	157.92.23.3:801	A	144	246	0
27	157.92.23.3:801	157.92.75.5:1024	FA	1000001	603	0
28	157.92.75.5:1024	157.92.23.3:801	A	602	1000002	0

Para determinar cuántas conexiones TCP hay en esta secuencia de segmentos y asociar cada segmento con cada conexión, se utiliza el criterio de observar las direcciones IP de origen y destino, así como los números de puerto de origen y destino.

Dado que cada conexión TCP está definida por una combinación única de direcciones IP y puertos de origen y destino, podemos agrupar los segmentos que comparten la misma combinación en una conexión TCP.

Las conexiones TCP identificadas en la secuencia de segmentos proporcionada:

1. Conexión TCP 1:
   - Dirección IP de origen: 157.92.75.5
   - Puerto de origen: 1024
   - Dirección IP de destino: 157.92.23.3
   - Puerto de destino: 801

2. Conexión TCP 2:
   - Dirección IP de origen: 190.30.132.239
   - Puerto de origen: 3623
   - Dirección IP de destino: 157.92.23.3
   - Puerto de destino: 801

Con base en esta información, podemos asociar cada segmento con su conexión TCP correspondiente:

- Segmentos pertenecientes a la conexión TCP 1:
1, 2, 4, 7, 8, 10, 12, 13, 16, 17, 19, 22, 23, 24, 28
- Segmentos pertenecientes a la conexión TCP 2:
3, 5, 6, 9, 11, 14, 15, 18, 20, 21, 25, 26, 27

Cada uno de estos grupos de segmentos representa una conexión TCP distinta.

Problema 15
Un servidor Web cuya dirección IP es 168.83.72.5 levanta el servicio en el puerto TCP/80. En una PC cuya dirección es 157.92.27.33 alguien abre dos ventanas de un navegador y en cada una de ellas abre simultáneamente una página del mismo servidor. ¿Es esto posible? ¿Cómo se distinguen los paquetes que son para una ventana de los que son para la otra?

Sí, es posible que dos ventanas de un navegador en una misma PC abran simultáneamente páginas desde el mismo servidor web (168.83.72.5) en el puerto TCP/80. Esto es factible porque cada ventana del navegador mantiene una conexión independiente con el servidor.

Los paquetes que pertenecen a una ventana específica se distinguen mediante la dirección IP y el número de puerto de origen y destino, así como el número de secuencia TCP. Cada conexión TCP desde la PC hacia el servidor web tendrá una combinación única de estos valores. El sistema operativo del cliente se encarga de manejar las conexiones simultáneas y de asegurar que los paquetes de cada conexión estén correctamente asociados con la ventana del navegador correspondiente.

Problema 16
En una conexión TCP, durante el three-way handshake (saludo a dos vías) se produce lo siguiente: El cliente envía un segmento con el flag SYN activado y número de secuencia 100. El servidor envía un segmento con los flags SYN y ACK activados, número de secuencia 200 y número de ACK 100. ¿Qué sucede después?

Después de que el servidor envía un segmento con los flags SYN y ACK activados, así como el número de secuencia 200 y el número de ACK 100 durante el three-way handshake:

1. El cliente recibe el segmento SYN/ACK del servidor.
2. El cliente responde al servidor con un segmento que tiene el flag ACK activado y un número de secuencia 101 (que es el número de secuencia del servidor más 1).

3. Una vez que el servidor recibe el segmento ACK del cliente, la conexión TCP se establece correctamente y ambas partes pueden comenzar a intercambiar datos de manera bidireccional.

Por lo tanto, después de recibir el segmento SYN/ACK del servidor, el cliente envía un segmento ACK al servidor para finalizar el three-way handshake y establecer la conexión TCP.

Problema 17
Se desea usar Internet para realizar llamadas telefónicas en tiempo real. ¿Si debe diseñar el protocolo para las llamadas, sobre qué protocolo existente lo implementaría, TCP o UDP? ¿Por qué?

Si se desea realizar llamadas telefónicas en tiempo real a través de Internet, es más adecuado implementar el protocolo sobre UDP.

Porque, debido a su menor latencia UDP no establece una conexión antes de enviar datos, lo que reduce la latencia.
No se realizan retransmisiones automáticas, UDP no tiene mecanismos de control de flujo ni de retransmisión automática de paquetes perdidos, lo que reduce la posibilidad de retrasos y distorsiones en el audio de las llamadas.
Priorización de la velocidad sobre la fiabilidad, en aplicaciones de tiempo real como las llamadas telefónicas, es más importante la velocidad de transmisión de los datos que la fiabilidad en la entrega de todos los paquetes. La pérdida ocasional de paquetes es aceptable en este contexto, siempre y cuando no afecte significativamente la calidad de la llamada.

Problema 18
¿En TCP, cómo identifica el host A a cuál de las dos conexiones iniciadas por él corresponde la respuesta del host B?
Host A: TCP SYN → Host B
Host A: TCP SYN → Host B
Host A ← Host B: TCP SYN + ACK

En TCP, cada conexión se identifica de manera única mediante un conjunto de atributos conocidos como "cinco tuplas", que consisten en:

1. Dirección IP de origen.
2. Puerto de origen.
3. Dirección IP de destino.
4. Puerto de destino.
5. Protocolo (en este caso, TCP).

Después de que el host A envía los segmentos SYN para iniciar las dos conexiones, el host B responde con segmentos SYN+ACK. En cada respuesta SYN+ACK, el host B incluirá un número de secuencia y un número de reconocimiento. Estos números de secuencia y de reconocimiento son parte de un proceso de negociación durante el establecimiento de la conexión TCP y se utilizan para sincronizar las secuencias de datos entre los hosts.

El host A utilizará estos números de secuencia y de reconocimiento para identificar a cuál de las dos conexiones corresponde la respuesta del host B. Cada conexión tendrá su propio conjunto de números de secuencia y de reconocimiento, lo que permite al host A distinguir entre ellas.

Problema 19
Indique cuáles de las siguientes secuencias de finalización de una conexión TCP entre dos hosts pueden ser correctas:
a. SYN - SYN/ACK
b. SYN - ACK – SYN – ACK
c. SYN - SYN/ACK - ACK
d. SYN/ACK - ACK
e. SYN - ACK - SYN – ACK- SYN/ACK
f. SYN – ACK
g. Cualquiera de las anteriores
h. Ninguna de las anteriores

La secuencia correcta para finalizar una conexión TCP entre dos hosts es la opción:
d. SYN/ACK - ACK.

La secuencia SYN/ACK - ACK corresponde al proceso de finalización de una conexión TCP, donde el host que inició la conexión (con SYN) envía un segmento SYN/ACK como respuesta al primer segmento SYN y luego recibe un ACK del host receptor, indicando que la conexión puede ser finalizada.

Problema 20
Respuesta

a) Explique cómo se eligen esos números de puertos en los segmentos TCP.

Los números de puerto en los segmentos TCP se eligen de manera aleatoria por el sistema operativo del host que inicia la conexión. Estos puertos son asignados dinámicamente y están disponibles para su uso durante la duración de la conexión. Los puertos permiten que múltiples aplicaciones en el mismo host se comuniquen simultáneamente a través de la red sin interferir entre sí.

b) ¿Qué realizan los segmentos 3, 4 y 5 en función de TCP? ¿Por qué se usan esos números de SEQ y ACK?

Los segmentos 3, 4 y 5 en función de TCP están relacionados con el establecimiento de la conexión TCP mediante el proceso de three-way handshake.
- En el segmento 3, el cliente (209.13.34.94) envía un segmento SYN al servidor (odin.sinectis.com) con un número de secuencia (SEQ) aleatorio de 3405653374.
- En el segmento 4, el servidor responde con un segmento SYN ACK confirmando la recepción del segmento SYN del cliente. El número de secuencia (SEQ) es 2978859689 y el número de ACK es 3405653375.

- En el segmento 5, el cliente envía un segmento ACK al servidor confirmando la recepción del segmento SYN ACK del servidor. El número de secuencia (SEQ) es 2978859690 y el número de ACK es 3405653376.

Los números de SEQ y ACK se utilizan para sincronizar y confirmar el intercambio de datos entre el cliente y el servidor durante el establecimiento de la conexión TCP.

c) ¿Qué significan las líneas 1 y 2?

Las líneas 1 y 2 corresponden a consultas DNS. En la línea 1, el cliente (209.13.34.94) realiza una consulta DNS para resolver la dirección IP asociada al nombre de dominio "mail.sinectis.com.ar". En la línea 2, el servidor DNS responde con la dirección IP asociada al nombre de dominio consultado. Estas líneas representan el proceso de resolución de nombres de dominio a direcciones IP antes de establecer la conexión TCP con el servidor SMTP.