

# Practica 6 RyC

<b>1. ¿Cuál es el puerto por defecto que se utiliza en los siguientes servicios? Web / SSH / DNS / Web Seguro / POP3 / IMAP / SMTP? Investigue en qué lugar en Linux y en Windows está descrita la asociación utilizada por defecto para cada servicio.....</b>	<b>1</b>
Ubicación de la Descripción de la Asociación de Puertos.....	2
<b>2. Investigue qué es multicast. ¿Sobre cuál de los protocolos de capa de transporte funciona? ¿Se podría adaptar para que funcione sobre el otro protocolo de capa de transporte? ¿Por qué?.....</b>	<b>3</b>
Protocolo de Transporte Subyacente.....	3
Adaptación a otros Protocolos de Transporte.....	3
<b>3. Investigue cómo funciona el protocolo de aplicación FTP teniendo en cuenta las diferencias en su funcionamiento cuando se utiliza el modo activo de cuando se utiliza el modo pasivo ¿En qué se diferencian estos tipos de comunicaciones del resto de los protocolos de aplicación vistos?.....</b>	<b>4</b>
Modos Activo y Pasivo.....	4
Diferencias con Otros Protocolos.....	5
<b>4. Suponiendo Selective Repeat; tamaño de ventana 4 y sabiendo que E indica que el mensaje llegó con errores. Indique en el siguiente gráfico, la numeración de los ACK que el host B envía al Host A.....</b>	<b>6</b>
<b>5. ¿Qué restricción existe sobre el tamaño de ventanas en el protocolo Selective Repeat?.....</b>	<b>7</b>
<b>6. De acuerdo a la captura TCP de la siguiente figura, indique los valores de los campos borroneados.....</b>	<b>7</b>
<b>7. Dada la sesión TCP de la figura, completar los valores marcados con un signo de interrogación.....</b>	<b>8</b>
<b>8. ¿Qué es el RTT y cómo se calcula? Investigue la opción TCP timestamp y los campos TSval y TSecr.....</b>	<b>9</b>
Cálculo del RTT.....	9
Variabilidad del RTT.....	9
Intervalo de Fin de Temporización.....	10
<b>9. Para la captura tcp-captura.pcap, responder las siguientes preguntas.....</b>	<b>10</b>
a. ¿Cuántos intentos de conexiones TCP hay?.....	10
b. ¿Cuáles son la fuente y el destino (IP:port) para c/u?.....	10
c. ¿Cuántas conexiones TCP exitosas hay en la captura? ¿Cómo diferencia las exitosas de las que no lo son? ¿Cuáles flags encuentra en cada una?.....	11
d. Dada la primera conexión exitosa responder:.....	11
i. ¿Quién inicia la conexión?.....	11
ii. ¿Quién es el servidor y quién el cliente?.....	11
iii. ¿En qué segmentos se ve el 3-way handshake?.....	11
iv. ¿Cuáles ISNs se intercambian?.....	11
v. ¿Cuál MSS se negoció?.....	11
vi. ¿Cuál de los dos hosts envía la mayor cantidad de datos (IP:port)?.....	11
e. Identificar el primer segmento de datos (origen, destino, tiempo, número de fila y número de secuencia TCP).....	11

i. ¿Cuántos datos lleva?.....	12
ii. ¿Cuándo es confirmado (tiempo, número de fila y número de secuencia TCP)?..	12
iii. La confirmación, ¿qué cantidad de bytes confirma?.....	12
f. ¿Quién inicia el cierre de la conexión? ¿Qué flags se utilizan? ¿En cuáles segmentos se ve (tiempo, número de fila y número de secuencia TCP)?.....	12
<b>10. Responda las siguientes preguntas respecto del mecanismo de control de flujo.</b>	<b>12</b>
a. ¿Quién lo activa? ¿De qué forma lo hace?.....	12
b. ¿Qué problema resuelve?.....	12
c. ¿Cuánto tiempo dura activo y qué situación lo desactiva?.....	13
<b>11. Responda las siguientes preguntas respecto del mecanismo de control de congestión.....</b>	<b>13</b>
a. ¿Quién activa el mecanismo de control de congestión? ¿Cuáles son los posibles disparadores?.....	13
b. ¿Qué problema resuelve?.....	13
c. Diferencie slow start de congestion-avoidance.....	14
<b>12. Para la captura udp-captura.pcap, responder las siguientes preguntas.....</b>	<b>15</b>
a. ¿Cuántas comunicaciones (srcIP,srcPort,dstIP,dstPort) UDP hay en la captura?.....	15
b. ¿Cómo se podrían identificar las exitosas de las que no lo son?.....	15
c. ¿UDP puede utilizar el modelo cliente/servidor?.....	15
d. ¿Qué servicios o aplicaciones suelen utilizar este protocolo?¿Qué requerimientos tienen?.....	15
e. ¿Qué hace el protocolo UDP en relación al control de errores?.....	16
f. Con respecto a los puertos vistos en las capturas, ¿observa algo particular que lo diferencie de TCP?.....	17
g. Dada la primera comunicación en la cual se ven datos en ambos sentidos (identificar el primer datagrama):.....	17
i. ¿Cuál es la dirección IP que envía el primer datagrama?,¿desde cuál puerto?.....	17
ii. ¿Cuántos datos se envían en un sentido y en el otro?.....	17
<b>13. Dada la salida que se muestra en la imagen, responda los ítems debajo.....</b>	<b>18</b>
• Suponga que ejecuta los siguientes comandos desde un host con la IP 10.100.25.90. Responda qué devuelve la ejecución de los siguientes comandos y, en caso que corresponda, especifique los flags.....	18
a. hping3 -p 3306 –udp 10.100.25.135.....	18
b. hping3 -S -p 25 10.100.25.135.....	18
c. hping3 -S -p 22 10.100.25.135.....	18
d. hping3 -S -p 110 10.100.25.135.....	18
• ¿Cuántas conexiones distintas hay establecidas? Justifique.....	18

1. ¿Cuál es el puerto por defecto que se utiliza en los siguientes servicios? Web / SSH / DNS / Web Seguro / POP3 / IMAP / SMTP? Investigue en qué lugar en Linux y en Windows está descrita la asociación utilizada por defecto para cada servicio.

Aquí tienes una lista de los puertos por defecto utilizados por los servicios mencionados:

- **Web:** El puerto por defecto para el servicio web (HTTP) es el **80**. Para el servicio web seguro (HTTPS), el puerto por defecto es el **443**.
- **SSH:** El puerto por defecto para SSH es el **22**. *No se menciona el puerto por defecto para SSH en las fuentes.*
- **DNS:** El puerto por defecto para DNS es el **53**. El protocolo DNS se ejecuta sobre UDP.
- **POP3:** El puerto por defecto para POP3 es el **110**.
- **IMAP:** El puerto por defecto para IMAP es el **143**. *No se menciona el puerto por defecto para IMAP en las fuentes, pero 143 es el puerto estándar utilizado para IMAP.*
- **SMTP:** El puerto por defecto para SMTP es el **25**.

## Ubicación de la Descripción de la Asociación de Puertos

En cuanto a la ubicación donde se describe la asociación de puertos por defecto en Linux y Windows, la información varía según el sistema operativo y la versión.

Generalmente, la información sobre los servicios y sus puertos asociados se encuentra en los siguientes archivos:

### Linux:

- **/etc/services:** Este archivo contiene una lista de servicios de red conocidos y sus puertos correspondientes.
- Archivos de configuración específicos de cada servicio: Muchos servicios tienen sus propios archivos de configuración donde se define el puerto de escucha. Estos archivos suelen estar en directorios como **/etc/apache2/** para Apache o **/etc/ssh/** para SSH.

### Windows:

- **Registro de Windows:** La información sobre los puertos y servicios se almacena en el registro de Windows. Se puede acceder a esta información utilizando el editor de registro (regedit).
- **Herramientas de administración:** Herramientas como el "Administrador de Servicios" o "netstat" pueden mostrar los servicios en ejecución y los puertos que están utilizando.

## 2. Investigue qué es multicast. ¿Sobre cuál de los protocolos de capa de transporte funciona? ¿Se podría adaptar para que funcione sobre el otro protocolo de capa de transporte? ¿Por qué?

La **multidifusión (multicast)** es una técnica de comunicación en red que permite a un emisor enviar un único paquete de datos a un grupo específico de receptores. En contraste con la unidifusión (unicast), donde se establece una comunicación individual entre el emisor y cada receptor, la multidifusión optimiza el uso del ancho de banda al enviar una sola copia del paquete que es replicado solo para los miembros del grupo.

### Protocolo de Transporte Subyacente

En Internet, la **multidifusión funciona a nivel de la capa de red**, por lo que se implementa con el protocolo IP. El protocolo **IGMP (Internet Group Management Protocol)** se encarga de gestionar la membresía a los grupos de multidifusión, permitiendo a los hosts unirse y abandonar grupos.

### Adaptación a otros Protocolos de Transporte

Si bien la multidifusión en Internet se basa en IP, cabe preguntarse si es posible adaptarla a los protocolos de transporte como TCP y UDP.

- **Adaptación a TCP:** TCP es un protocolo orientado a conexión que garantiza la entrega confiable de datos a través de mecanismos como el control de flujo y la retransmisión. La naturaleza de la multidifusión, que implica la entrega a múltiples receptores simultáneos, dificulta la implementación de estos mecanismos de control y confiabilidad inherentes a TCP.
- **Adaptación a UDP:** UDP, por otro lado, es un protocolo sin conexión que no ofrece garantías de entrega. Si bien la multidifusión se implementa en la capa de red, teóricamente sería posible usar UDP para transportar datos a través de un grupo de multidifusión. Sin embargo, UDP por sí solo no proporciona mecanismos de control de membresía o confiabilidad.

### 3. Investigue cómo funciona el protocolo de aplicación FTP teniendo en cuenta las diferencias en su funcionamiento cuando se utiliza el modo activo de cuando se utiliza el modo pasivo ¿En qué se diferencian estos tipos de comunicaciones del resto de los protocolos de aplicación vistos?

El protocolo de transferencia de archivos (FTP) permite a los usuarios transferir archivos entre un sistema de archivos local y uno remoto. Para esto, FTP utiliza dos conexiones TCP paralelas: una de control y otra de datos.

- **Conexión de Control:** se utiliza para enviar información de control, como la identificación del usuario, la contraseña, comandos para cambiar el directorio remoto y comandos para subir (**PUT**) y descargar (**GET**) archivos. Esta conexión se establece en el puerto 21 del servidor.
- **Conexión de Datos:** se utiliza para la transferencia del archivo en sí.

FTP difiere de otros protocolos de transferencia de archivos, como HTTP, en la forma en que envía su información de control. FTP envía la información de control "fuera de banda", es decir, a través de una conexión separada (la conexión de control). En cambio, HTTP envía la información de control "en banda", junto con los datos del archivo a través de la misma conexión TCP.

#### **Modos Activo y Pasivo**

La principal diferencia entre el modo activo y el pasivo reside en cómo se establece la conexión de datos.

##### **Modo Activo:**

1. El cliente FTP establece la conexión de control con el servidor en el puerto 21.
2. El cliente informa al servidor el puerto en el que estará escuchando para la conexión de datos (un puerto aleatorio por encima del 1023).
3. **El servidor FTP inicia la conexión de datos con el cliente en el puerto especificado.**

##### **Modo Pasivo:**

1. El cliente FTP establece la conexión de control con el servidor en el puerto 21.
2. El cliente envía un comando **PASV** al servidor.
3. **El servidor FTP abre un puerto aleatorio (por encima del 1023) y le comunica al cliente la dirección IP y el puerto en el que estará escuchando.**
4. El cliente FTP inicia la conexión de datos con el servidor en la dirección IP y puerto indicados.

El modo pasivo se diseñó para solucionar problemas que pueden surgir con el modo activo cuando el cliente se encuentra detrás de un firewall. En el modo activo, el firewall del cliente puede bloquear la conexión de datos iniciada por el servidor. En el modo pasivo, el cliente inicia la conexión de datos, lo que generalmente permite que el firewall la permita.

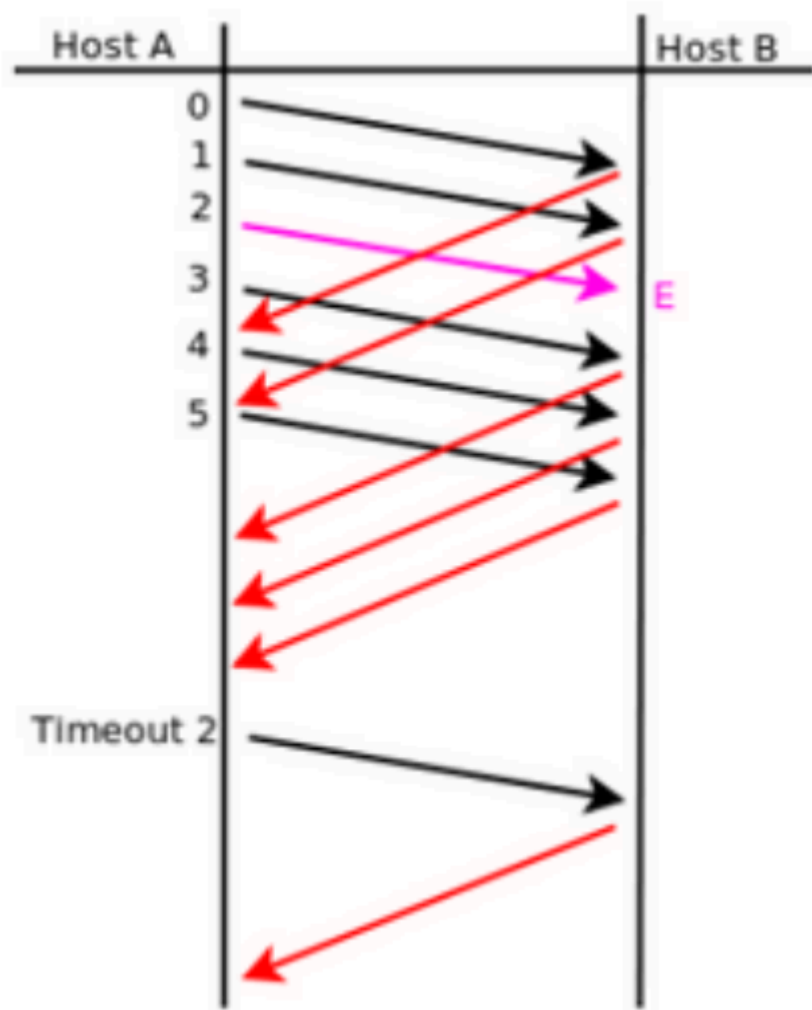
## Diferencias con Otros Protocolos

FTP se diferencia de otros protocolos de aplicación como HTTP y SMTP en varios aspectos:

- **Conexiones Dedicadas:** FTP utiliza dos conexiones separadas para control y datos, mientras que HTTP y SMTP utilizan una sola conexión para enviar tanto la información de control como los datos.
- **Estado del Usuario:** El servidor FTP mantiene un estado del usuario durante la sesión, recordando la cuenta del usuario y el directorio actual. HTTP, por otro lado, es un protocolo sin estado.

Estos aspectos hacen que FTP sea un protocolo más complejo que HTTP o SMTP, pero también le otorgan mayor flexibilidad y control sobre la transferencia de archivos.

4. Suponiendo Selective Repeat; tamaño de ventana 4 y sabiendo que E indica que el mensaje llegó con errores. Indique en el siguiente gráfico, la numeración de los ACK que el host B envía al Host A.



El Host B enviará:

ACK: 0,1,3,4,5,2

Ya que el segundo no llega y es reenviado al final.

## 5. ¿Qué restricción existe sobre el tamaño de ventanas en el protocolo Selective Repeat?

En los protocolos Selective Repeat (SR), el tamaño de la ventana no puede ser arbitrariamente grande. Para evitar ambigüedad en el receptor sobre si un paquete recibido es uno nuevo o una retransmisión, el tamaño de la ventana debe ser menor o igual a la mitad del tamaño del espacio de números de secuencia.

## 6. De acuerdo a la captura TCP de la siguiente figura, indique los valores de los campos borroneados

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.20.1.1	172.20.1.100	TCP	74	41749 > vce [ ] Seq= Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=270132 TSecr=0
2	0.001264	172.20.1.100	172.20.1.1	TCP	74	vce > 41749 [SYN, ACK] Seq=1047471501 Ack=3933822138 Win=5792 Len=0 MSS=1460 SACK_PERM=1
3	0.001341			TCP	66	> [ ] Seq= Ack= Win=5888 Len=0 TSval=270132 TSecr=1877442

Internet Protocol Version 4, Src: 172.20.1.100 (172.20.1.100), Dst: 172.20.1.1 (172.20.1.1)

Transmission Control Protocol, Src Port: vce (11111), Dst Port: 41749 (41749), Seq: 1047471501, Ack: 3933822138, Len: 0

Source port: vce (11111)  
Destination port: 41749 (41749)  
[Stream index: 0]  
Sequence number: 1047471501  
Acknowledgement number: 3933822138  
Header length: 40 bytes

Flags: 0x012 (SYN, ACK)

000. .... = Reserved: Not set  
...0 .... = Nonce: Not set  
.... 0... = Congestion Window Reduced (CWR): Not set  
.... .0.. = ECN-Echo: Not set  
.... ..0. = Urgent: Not set  
.... ...1 = Acknowledgement: Set  
.... ....0.. = Push: Not set  
.... .....0.. = Reset: Not set

.... ....1. = Syn: Set  
.... .......0 = Fin: Not set  
Window size value: 5792  
[Calculated window size: 5792]  
Checksum: 0x9803 [validation disabled]

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.20.1.1	172.20.1.100	TCP	74	41749 > vce [SYN Seq=3933822137 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=270132 TSecr=0
2	0.001264	172.20.1.100	172.20.1.1	TCP	74	vce > 41749 [SYN, ACK] Seq=1047471501 Ack=3933822138 Win=5792 Len=0 MSS=1460 SACK_PERM=1
3	0.001341	172.20.1.1	172.20.1.100	TCP	66	41749 > vce [ACK Seq=3933822138 Ack=1047471502 Win=5888 Len=0 TSval=270132 TSecr=1877442

Internet Protocol Version 4, Src: 172.20.1.100 (172.20.1.100), Dst: 172.20.1.1 (172.20.1.1)

Transmission Control Protocol, Src Port: vce (11111), Dst Port: 41749 (41749), Seq: 1047471501, Ack: 3933822138, Len: 0

Source port: vce (11111)  
Destination port: 41749 (41749)  
[Stream index: 0]  
Sequence number: 1047471501  
Acknowledgement number: 3933822138  
Header length: 40 bytes

Flags: 0x012 (SYN, ACK)

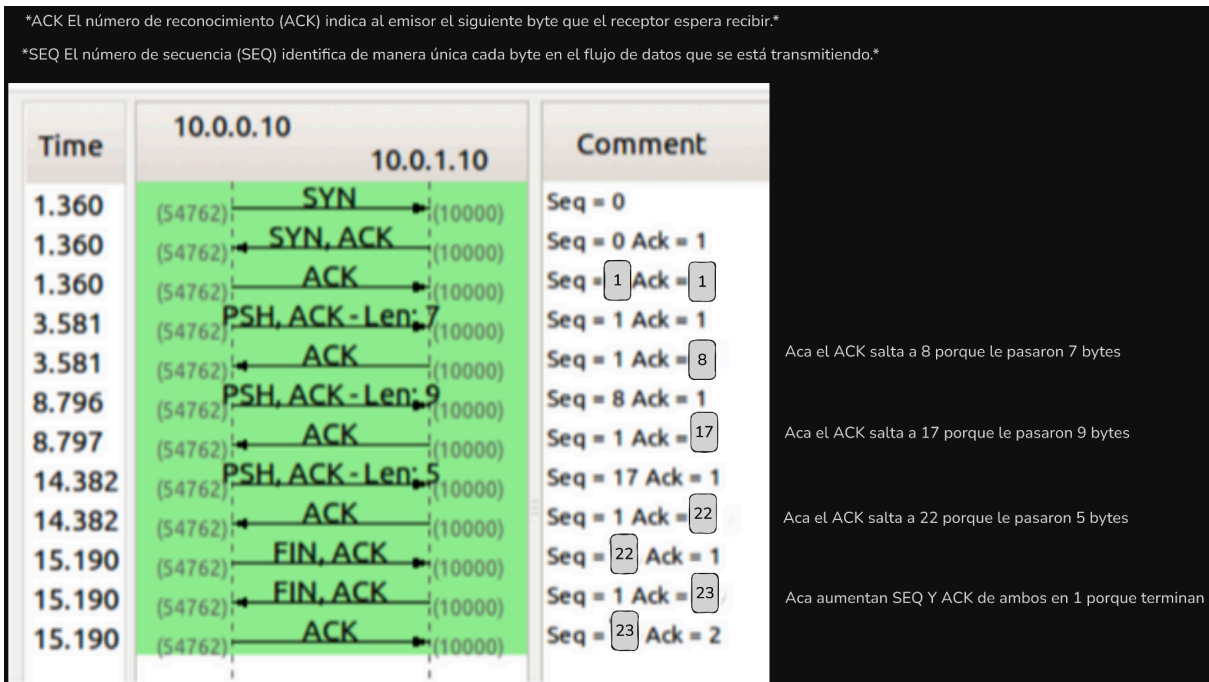
000. .... = Reserved: Not set  
...0 .... = Nonce: Not set  
.... 0... = Congestion Window Reduced (CWR): Not set  
.... .0.. = ECN-Echo: Not set  
.... ..0. = Urgent: Not set  
.... ...1 = Acknowledgement: Set  
.... ....0.. = Push: Not set  
.... .....0.. = Reset: Not set

.... ....1. = Syn: Set  
.... .......0 = Fin: Not set  
Window size value: 5792  
[Calculated window size: 5792]  
Checksum: 0x9803 [validation disabled]



7. Dada la sesión TCP de la figura, completar los valores marcados con un signo de interrogación.

Time	10.0.0.10	10.0.1.10	Comment
1.360	(54762) →	SYN (10000)	Seq = 0
1.360	(54762) ←	SYN, ACK (10000)	Seq = 0 Ack = 1
1.360	(54762) →	ACK (10000)	Seq = ? Ack = ?
3.581	(54762) →	PSH, ACK - Len: 7 (10000)	Seq = 1 Ack = 1
3.581	(54762) ←	ACK (10000)	Seq = 1 Ack = ?
8.796	(54762) →	PSH, ACK - Len: 9 (10000)	Seq = 8 Ack = 1
8.797	(54762) ←	ACK (10000)	Seq = 1 Ack = ?
14.382	(54762) →	PSH, ACK - Len: 5 (10000)	Seq = 17 Ack = 1
14.382	(54762) ←	ACK (10000)	Seq = 1 Ack = ?
15.190	(54762) →	FIN, ACK (10000)	Seq = ? Ack = 1
15.190	(54762) ←	FIN, ACK (10000)	Seq = 1 Ack = ?
15.190	(54762) →	ACK (10000)	Seq = ? Ack = 2



## 8. ¿Qué es el RTT y cómo se calcula? Investigue la opción TCP timestamp y los campos TSval y TSecr.

El **RTT (Round-Trip Time o Tiempo de Ida y Vuelta)** en una conexión TCP es el tiempo que transcurre desde que se envía un segmento hasta que se recibe su reconocimiento (ACK). Es un valor crucial para el correcto funcionamiento de TCP, ya que determina el tiempo de espera para retransmisiones y afecta directamente al rendimiento de la conexión.

### Cálculo del RTT

TCP calcula el RTT mediante la medición de **RTTMuestra (SampleRTT)**, que es el tiempo entre el envío de un segmento y la recepción de su ACK. Para evitar tomar medidas de segmentos retransmitidos o atípicos, TCP solo mide RTTMuestra para un segmento no reconocido a la vez.

Para obtener un valor representativo del RTT, se calcula un promedio de los valores de RTTMuestra, llamado **RTTEstimado (EstimatedRTT)**. La actualización de RTTEstimado se realiza mediante una media móvil exponencial (EWMA), teniendo en cuenta el valor anterior de RTTEstimado y el nuevo valor de RTTMuestra.

### Variabilidad del RTT

Además de RTT Estimado, TCP calcula la **RTTDesv (DevRTT)**, una medida de la variabilidad del RTT. RTTDesv se obtiene mediante una EWMA de la diferencia entre RTTMuestra y RTTEstimado. Si los valores de RTTMuestra fluctúan mucho, RTTDesv será grande, y viceversa.

### Intervalo de Fin de Temporización

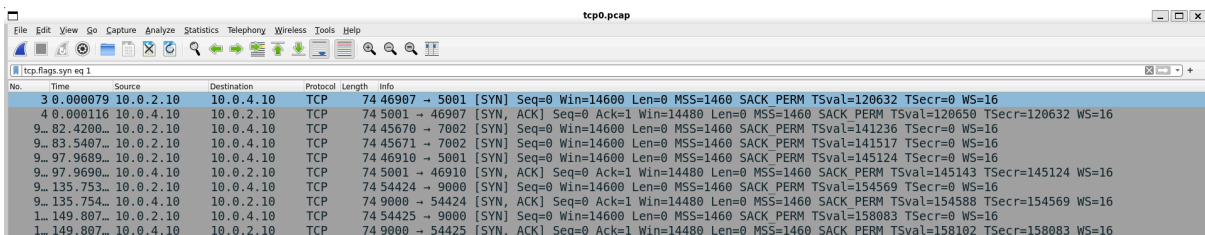
El **IntervaloFindeTemporización (TimeoutInterval)** es el tiempo que TCP espera antes de retransmitir un segmento no reconocido. Se calcula en base a RTTEstimado y RTTDesv, sumando un margen que depende de la variabilidad del RTT. La fórmula es:

**TimeoutInterval = RTTEstimado + 4 \* RTTDesv**

Si se produce un fin de temporización, TimeoutInterval se duplica para evitar retransmisiones prematuras.

## 9. Para la captura tcp-captura.pcap, responder las siguientes preguntas.

a. ¿Cuántos intentos de conexiones TCP hay?



6, todos los SYN=1 y ACK=0

b. ¿Cuáles son la fuente y el destino (IP:port) para c/u?

10.0.2.10	10.0.4.10
46907	5001
45670	7002
45671	7002
46910	5001
54424	9000
54425	9000

c. ¿Cuántas conexiones TCP exitosas hay en la captura? ¿Cómo diferencia las exitosas de las que no lo son? ¿Cuáles flags encuentra en cada una?

tcp.flags.syn eq 1 && tcp.flags.ack eq 1						
No.	Time	Source	Destination	Protocol	Length	Info
4	0.000116	10.0.4.10	10.0.2.10	TCP	74	5001 → 46907 [SYN, ACK] Seq=0 Ack=1
9...	97.9690...	10.0.4.10	10.0.2.10	TCP	74	5001 → 46910 [SYN, ACK] Seq=0 Ack=1
9...	135.754...	10.0.4.10	10.0.2.10	TCP	74	9000 → 54424 [SYN, ACK] Seq=0 Ack=1
1...	149.807...	10.0.4.10	10.0.2.10	TCP	74	9000 → 54425 [SYN, ACK] Seq=0 Ack=1

d. Dada la primera conexión exitosa responder:

i. ¿Quién inicia la conexión?

10.0.2.10:46907

ii. ¿Quién es el servidor y quién el cliente?

10.0.2.10:46907 es el cliente, y 10.0.4.10:5001 es el server

iii. ¿En qué segmentos se ve el 3-way handshake?

No.	Time	Source	Destination	Protocol	Length	Info
3	0.000079	10.0.2.10	10.0.4.10	TCP	74	46907 → 5001 [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM TSval=120632 TSecr=0 WS=16
4	0.000116	10.0.4.10	10.0.2.10	TCP	74	5001 → 46907 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM TSval=120650 TSecr=120632 WS=16
5	0.151614	10.0.2.10	10.0.4.10	TCP	66	46907 → 5001 [ACK] Seq=1 Ack=1 Win=14608 Len=0 TSval=120669 TSecr=120650

iv. ¿Cuáles ISNs se intercambian?

3. Sequence Number (raw): 2218428254

4. Sequence Number (raw): 1292618479

5. Sequence Number (raw): 2218428255

v. ¿Cuál MSS se negoció?

**MSS=1460**

vi. ¿Cuál de los dos hosts envía la mayor cantidad de datos (IP:port)?

10.0.2.10 Aumenta su número de secuencia una re banda (es decir envía datos).

e. Identificar el primer segmento de datos (origen, destino, tiempo, número de fila y número de secuencia TCP).

6	0.151826	10.0.2.10	10.0.4.10	TCP	90	46907 → 5001 [PSH, ACK] Seq=1 Ack=1 Win=14608 Len=24 TSval=120670 TSecr=120650
---	----------	-----------	-----------	-----	----	--

Es este porque el siguiente el ACK aumenta banda.

i. ¿Cuántos datos lleva?

24 bytes (len)

ii. ¿Cuándo es confirmado (tiempo, número de fila y número de secuencia TCP)?

```
7 0.151925 10.0.4.10 10.0.2.10 TCP 66 5001 → 46907 [ACK] Seq=1 Ack=25 Win=14480 Len=0 TSval=120688 TSecr=120670
```

iii. La confirmación, ¿qué cantidad de bytes confirma?

24 y espera el 25.

f. ¿Quién inicia el cierre de la conexión? ¿Qué flags se utilizan? ¿En cuáles segmentos se ve (tiempo, número de fila y número de secuencia TCP)?

```
tcp.flags.fin eq 1
No.    Time          Source      Destination  Protocol Length  Info
9... 75.0901... 10.0.2.10  10.0.4.10    TCP        234 46907 → 5001 [FIN, PSH, ACK] Seq=786289 Ack=1 Win=14608 Len=168 TSval=137726 TSecr=137707
```

La inicia 10.0.2.10

## 10. Responda las siguientes preguntas respecto del mecanismo de control de flujo.

a. ¿Quién lo activa? ¿De qué forma lo hace?

El mecanismo de control de flujo en TCP lo activa el receptor. Lo hace anunciando el tamaño de su ventana de recepción (VentanaRecepcion) al emisor en cada segmento que envía

b. ¿Qué problema resuelve?

El control de flujo resuelve el problema del desbordamiento del buffer del receptor. El emisor podría enviar datos a una velocidad mayor de la que el receptor puede procesar, lo que provocaría que el buffer del receptor se llene y se pierdan datos.

- El control de flujo permite al receptor limitar la velocidad de transmisión del emisor para que no se desborde su buffer.
- El receptor ajusta dinámicamente el tamaño de la ventana de recepción en función del espacio disponible en su buffer.

### c. ¿Cuánto tiempo dura activo y qué situación lo desactiva?

El mecanismo de control de flujo permanece activo durante toda la duración de la conexión TCP. No existe una situación específica que lo "desactive".

- El tamaño de la ventana de recepción se ajusta constantemente a medida que el receptor procesa los datos y libera espacio en su buffer.
- Si el receptor tiene suficiente espacio en su buffer, la ventana de recepción será grande, permitiendo al emisor transmitir más datos.
- Si el buffer del receptor se llena, la ventana de recepción se reducirá, lo que obligará al emisor a disminuir su velocidad de transmisión.
- En resumen, el control de flujo en TCP es un mecanismo dinámico y continuo que previene la pérdida de datos por desbordamiento del buffer del receptor.

## 11. Responda las siguientes preguntas respecto del mecanismo de control de congestión.

### a. ¿Quién activa el mecanismo de control de congestión? ¿Cuáles son los posibles disparadores?

El control de congestión lo activa el emisor. El emisor limita la velocidad de transmisión de tráfico a través de su conexión en función de la congestión de red percibida. Este proceso es dinámico y adaptativo, y el emisor ajusta su velocidad de transmisión en respuesta a las condiciones cambiantes de la red. Los posibles disparadores son:

- Fin de Temporización: La expiración del temporizador asociado con el envío de un segmento TCP puede ser interpretada como una señal de pérdida, indicando posiblemente congestión en la ruta.
- Recepción de TRES ACK Duplicados: La recepción de paquetes ACK duplicados procedentes del receptor también se interpreta como un suceso de pérdida. Este evento puede sugerir la pérdida de un paquete en la red debido a la congestión.

### b. ¿Qué problema resuelve?

El objetivo es que no se desborde la propia red. Esto ocurre cuando hay más tráfico de red del que la red puede manejar eficientemente, lo que puede resultar en la pérdida de paquetes, retrasos elevados y un rendimiento de red deficiente. El control de congestión busca evitar que la red se sobrecargue ajustando la tasa de transmisión de datos del emisor para que sea compatible con la capacidad de la red.

## c. Diferencie slow start de congestion-avoidance.

**Slow start** y **congestion-avoidance** son dos fases importantes del algoritmo de control de congestión de TCP. Se diferencian en la forma en que incrementan el tamaño de la ventana de congestión (VentanaCongestion) en respuesta a los ACK recibidos:

### Slow Start (Arranque Lento)

- **Inicio:** Se inicia al comienzo de la conexión TCP o después de un evento de pérdida de paquetes.
- **Comportamiento:** VentanaCongestion se incrementa de forma exponencial: se duplica por cada ACK recibido. Esto permite a TCP "sondear" rápidamente el ancho de banda disponible.
- **Transición:** La fase de slow start termina cuando:
  - VentanaCongestion alcanza un valor umbral (umbralAL), que se establece en la mitad del valor de VentanaCongestion cuando se detectó la última congestión.
  - Se recibe un evento de pérdida de paquetes.
  - Se detectan tres ACK duplicados.
- **Objetivo:** Alcanzar rápidamente un tamaño de ventana cercano al óptimo para la conexión.

### Congestion Avoidance (Evitación de la Congestión)

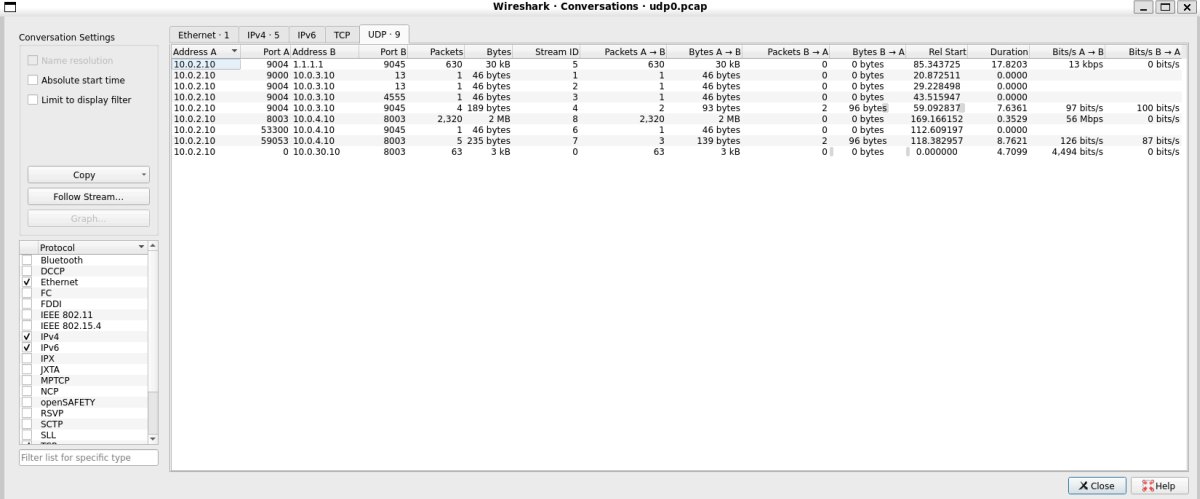
- **Inicio:** Se inicia después de la fase de slow start o después de una recuperación rápida.
- **Comportamiento:** VentanaCongestion se incrementa de forma lineal, más cautelosa: aumenta en 1 MSS (Maximum Segment Size) por cada RTT.
- **Objetivo:** Mantener un tamaño de ventana estable y evitar la congestión de la red.

En resumen:

Fase	Incremento de VentanaCongestion	Objetivo
Slow start	Exponencial (duplica por ACK)	Sondear rápidamente el ancho de banda
Congestion avoidance	Lineal (1 MSS por RTT)	Mantener un tamaño de ventana estable

12. Para la captura udp-captura.pcap, responder las siguientes preguntas.

a. ¿Cuántas comunicaciones (srcIP,srcPort,dstIP,dstPort) UDP hay en la captura?



The screenshot shows the Wireshark interface with the 'Conversations' pane open for 'udp0.pcap'. The 'UDP' tab is selected, showing 9 conversations. The table below summarizes the data visible in the screenshot.

Address A	Port A	Address B	Port B	Packets	Bytes	Stream ID	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
10.0.2.10	9004	1.1.1.1	9045	630	30 kB	5	630	30 kB	0	0 bytes	85.343725	17.8203	13 kbps	0 bits/s
10.0.2.10	9000	10.0.3.10	13	1	46 bytes	1	1	46 bytes	0	0 bytes	20.872511	0.0000		
10.0.2.10	9004	10.0.3.10	13	1	46 bytes	2	1	46 bytes	0	0 bytes	29.228498	0.0000		
10.0.2.10	9004	10.0.3.10	4555	1	46 bytes	3	1	46 bytes	0	0 bytes	43.515947	0.0000		
10.0.2.10	9004	10.0.3.10	9045	4	189 bytes	4	2	93 bytes	2	96 bytes	59.092837	7.6361	97 bits/s	100 bits/s
10.0.2.10	8003	10.0.4.10	8003	2,320	2 MB	8	2,320	2 MB	0	0 bytes	169.166152	0.3529	56 Mbps	0 bits/s
10.0.2.10	53300	10.0.4.10	9045	1	46 bytes	6	1	46 bytes	0	0 bytes	112.609197	0.0000		
10.0.2.10	59053	10.0.4.10	8003	5	235 bytes	7	3	139 bytes	2	96 bytes	118.382957	8.7621	126 bits/s	87 bits/s
10.0.2.10	0	10.0.30.10	8003	63	3 kB	0	63	3 kB	0	0 bytes	0.000000	4.7099	4,494 bits/s	0 bits/s

9

b. ¿Cómo se podrían identificar las exitosas de las que no lo son?

Se puede determinar con los mensajes ICMP

c. ¿UDP puede utilizar el modelo cliente/servidor?

Si bien UDP puede ser utilizado en arquitecturas cliente/servidor, no está intrínsecamente ligado a este modelo.

d. ¿Qué servicios o aplicaciones suelen utilizar este protocolo? ¿Qué requerimientos tienen?

#### Aplicaciones que Suelen Utilizar UDP

- **DNS (Domain Name System):** DNS utiliza UDP para las consultas de nombres de dominio a direcciones IP. Esta elección se debe a que las consultas DNS son generalmente pequeñas y no requieren la fiabilidad que ofrece TCP. Además, la velocidad es crucial en las consultas DNS, y el mecanismo de control de congestión de TCP podría agregar latencia.



- **Streaming multimedia (audio y video):** Aplicaciones como la telefonía por Internet y la videoconferencia a menudo utilizan UDP debido a sus requisitos de baja latencia. Aunque la pérdida de algunos paquetes puede degradar la calidad, la retransmisión de los mismos (como en TCP) introduciría un retardo inaceptable. Sin embargo, en la Internet actual, el tráfico de voz y video se envía cada vez más a través de TCP debido a las políticas de gestión de tráfico de los ISP, que a menudo favorecen a TCP sobre UDP.
- **Juegos en línea:** Los juegos en línea que requieren respuestas rápidas a menudo utilizan UDP para minimizar la latencia. La pérdida ocasional de paquetes es generalmente tolerable en este contexto.
- **SNMP (Simple Network Management Protocol):** SNMP, un protocolo utilizado para la gestión de dispositivos de red, típicamente utiliza UDP para el envío de mensajes de control y monitoreo.
- **Protocolos de enrutamiento:** Algunos protocolos de enrutamiento, como RIP (Routing Information Protocol), utilizan UDP para el intercambio de información de enrutamiento entre routers.

#### Requerimientos de las Aplicaciones que Usan UDP

- **Tolerancia a la pérdida de datos:** Las aplicaciones pueden tolerar cierta pérdida de paquetes sin comprometer su funcionalidad.
- **Baja latencia:** La velocidad es crucial para estas aplicaciones, y la sobrecarga de una conexión orientada a la conexión como TCP sería perjudicial.
- **Control de flujo a nivel de aplicación:** Si se necesita la fiabilidad, la aplicación debe implementar sus propios mecanismos de control de flujo y retransmisión de datos.

### e. ¿Qué hace el protocolo UDP en relación al control de errores?

El protocolo UDP (User Datagram Protocol) proporciona un mecanismo básico de **detección de errores**, pero **no implementa mecanismos de recuperación**.

- UDP utiliza una **suma de comprobación (checksum)** para verificar la integridad de los datos transmitidos. Esta suma se calcula sobre todo el segmento UDP, incluyendo la cabecera y los datos.
- El receptor verifica la suma de comprobación para determinar si los bits del segmento se alteraron durante la transmisión.
- Si se detecta un error, **UDP no realiza ninguna acción para recuperarse**. Algunas implementaciones simplemente descartan el segmento dañado, mientras que otras lo pasan a la aplicación con una advertencia.

#### Limitaciones de la Detección de Errores en UDP

- **No se garantiza la detección de todos los errores:** La suma de comprobación puede no detectar errores de múltiples bits o errores específicos que se cancelen entre sí.
- **No hay retransmisión:** UDP no retransmite los segmentos perdidos o dañados. La responsabilidad de manejar la pérdida de datos recae en la aplicación.

f. Con respecto a los puertos vistos en las capturas, ¿observa algo particular que lo diferencie de TCP?

En UDP el puerto origen puede ser 0 si no necesita una respuesta, puede ser simplemente un envío.

g. Dada la primera comunicación en la cual se ven datos en ambos sentidos (identificar el primer datagrama):

i. ¿Cuál es la dirección IP que envía el primer datagrama?, ¿desde cuál puerto?

10.0.2.1:9004

82 66.7289... 10.0.2.10 10.0.3.10 UDP 47 9004 → 9045 Len=5

ii. ¿Cuántos datos se envían en un sentido y en el otro?

udp.stream eq 4						
No.	Time	Source	Destination	Protocol	Length	Info
82	66.7289...	10.0.2.10	10.0.3.10	UDP	47	9004 → 9045 Len=5
79	59.0928...	10.0.2.10	10.0.3.10	UDP	46	9004 → 9045 Len=4
81	64.1161...	10.0.3.10	10.0.2.10	UDP	47	9045 → 9004 Len=5
80	62.1738...	10.0.3.10	10.0.2.10	UDP	49	9045 → 9004 Len=7

9 y 12 bytes

13. Dada la salida que se muestra en la imagen, responda los ítems debajo.

Netid	State	Local Address:Port	Peer Address:Port	
udp	UNCONN	*:68	:::	(( "dhclient", 671, 5))
udp	UNCONN	*:123	:::	(( "ntpd", 2138, 16))
udp	UNCONN	:::123	:::	(( "ntpd", 2138, 17))
tcp	LISTEN	*:80	:::	(( "nginx", 23653, 19), ("nginx", 23652, 19))
tcp	LISTEN	*:22	:::	(( "sshd", 1151, 3))
tcp	LISTEN	127.0.0.1:25	:::	(( "master", 11457, 12))
tcp	LISTEN	*:443	:::	(( "nginx", 23653, 20), ("nginx", 23652, 20))
tcp	LISTEN	*:3306	:::	(( "mysqld", 4556, 13))
tcp	ESTAB	127.0.0.1:3306	127.0.0.1:34338	(( "mysqld", 4556, 14))
tcp	TIME-WAIT	10.100.25.135:443	43.226.162.110:29148	
tcp	ESTAB	127.0.0.1:48717	127.0.0.1:3306	(( "ruby", 28615, 10))
tcp	ESTAB	127.0.0.1:3306	127.0.0.1:48717	(( "mysqld", 4556, 17))
tcp	ESTAB	127.0.0.1:34338	127.0.0.1:3306	(( "ruby", 28610, 9))
tcp	ESTAB	10.100.25.135:22	200.100.120.210:61576	(( "sshd", 13756, 3), ("sshd", 13654, 3))
tcp	LISTEN	:::22	:::	(( "sshd", 1151, 4))
tcp	LISTEN	:1:25	:::	(( "master", 11457, 13))

● Suponga que ejecuta los siguientes comandos desde un host con la IP 10.100.25.90. Responda qué devuelve la ejecución de los siguientes comandos y, en caso que corresponda, especifique los flags.

a. `hping3 -p 3306 -udp 10.100.25.135`

b. `hping3 -S -p 25 10.100.25.135`

c. `hping3 -S -p 22 10.100.25.135`

d. `hping3 -S -p 110 10.100.25.135`

● ¿Cuántas conexiones distintas hay establecidas? Justifique.

● **Resultados de los Comandos hping3:**

- a. `hping3 -p 3306 -udp 10.100.25.135`

Este comando enviará paquetes UDP al puerto 3306 del host 10.100.25.135. Dado que la salida muestra que el puerto 3306 está asociado con conexiones TCP establecidas y no hay información sobre un servicio UDP en ese puerto, lo más probable es que **no se reciba respuesta** o que se reciba un **mensaje ICMP de puerto inalcanzable (tipo 3, código 3)**. Esto se debe a que el host de destino no tiene un socket UDP activo en el puerto 3306.

- b. `hping3 -S -p 25 10.100.25.135`

Este comando envía un paquete TCP SYN (flag **-S**) al puerto 25 del host 10.100.25.135. La salida **no muestra** un servicio escuchando en el puerto 25 en 10.100.25.135. Por lo tanto, es probable que la respuesta sea un **segmento TCP RST/ACK**, indicando que no hay un servicio en ese puerto.

- **c. `hping3 -S -p 22 10.100.25.135`**

Este comando envía un paquete TCP SYN al puerto 22 del host 10.100.25.135. La salida muestra que hay un servicio **TCP LISTEN** en el puerto 22 (sshd). Por lo tanto, la respuesta esperada es un **segmento TCP SYN/ACK**, indicando que el servidor está listo para establecer una conexión TCP.

- **d. `hping3 -S -p 110 10.100.25.135`**

Este comando envía un paquete TCP SYN al puerto 110 del host 10.100.25.135. Similar al caso del puerto 25, la salida no muestra un servicio escuchando en el puerto 110. La respuesta esperada es un **segmento TCP RST/ACK**.

- **Conexiones Establecidas:**

Para determinar el número de conexiones distintas establecidas, se debe buscar el estado **"ESTAB"** en la columna "State" de la salida. En este caso, se observan **seis** conexiones TCP establecidas:

1. **`127.0.0.1:3306`** con PID 671 (dhclient)
2. **`127.0.0.1:3306`** con PID 2138 (ntpd)
3. **`127.0.0.1:3306`** con PID 2138 (ntpd)
4. **`127.0.0.1:3306`** con PID 23653 y 23652 (nginx)
5. **`127.0.0.1:3306`** con PID 1151 (sshd)
6. **`127.0.0.1:3306`** con PID 4556 (mysqld)

Cada una de estas entradas representa una conexión TCP única en estado establecido.