Práctica 6

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 descripta la asociación utilizada por defecto para cada servicio.

Servicio	Puerto por defecto
Web	80 (HTTP)
SSH	22
DNS	53
Web Seguro	443 (HTTPS)
POP3	110
IMAP	143

   En Linux, la asociación de puertos por defecto para los servicios se encuentra en el archivo /etc/services. Este archivo es un archivo de texto que contiene una lista de servicios registrados, junto con sus números de puerto, protocolos y nombres de dominio.

   En Windows, la asociación de puertos por defecto para los servicios se encuentra en el archivo %SystemRoot%\System32\drivers\etc\services. Este archivo es similar al archivo /etc/services de Linux.

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?**

   El multicast es una técnica que permite enviar un mensaje a un grupo de destinatarios de forma simultánea. A diferencia del broadcast, que envía un mensaje a todos los dispositivos de una red, el multicast solo envía el mensaje a los dispositivos que están interesados en recibirlo.

   La técnica del multicast funciona sobre UDP, ya que no necesita establecer una conexión y se podría usar un mismo socket (un proceso tiene asociado un socket) para recibir datos de varios procesos que se quieren comunicar con un proceso a la vez.

   Teóricamente podría intentarse adaptar multicast sobre TCP, pero sería demasiado complejo e iría en contra de la naturaleza del modelo ya TCP establece una conexión punto a punto entre un único emisor y receptor.

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?**

FTP requiere dos conexiones TCP. Una conexión de control y otra para la transferencia de datos**.** El cliente escoge cualquier puerto no privilegiado, (n > 1023) y genera conexión de control contra el puerto 21 del servidor. El servidor recibe los comandos por dicha conexión y responde/recibe por la conexión de datos aquellos que lo requieran. La conexión de datos se crea y de cierra bajo demanda. El estado de cada operación se transmite por el canal de control.

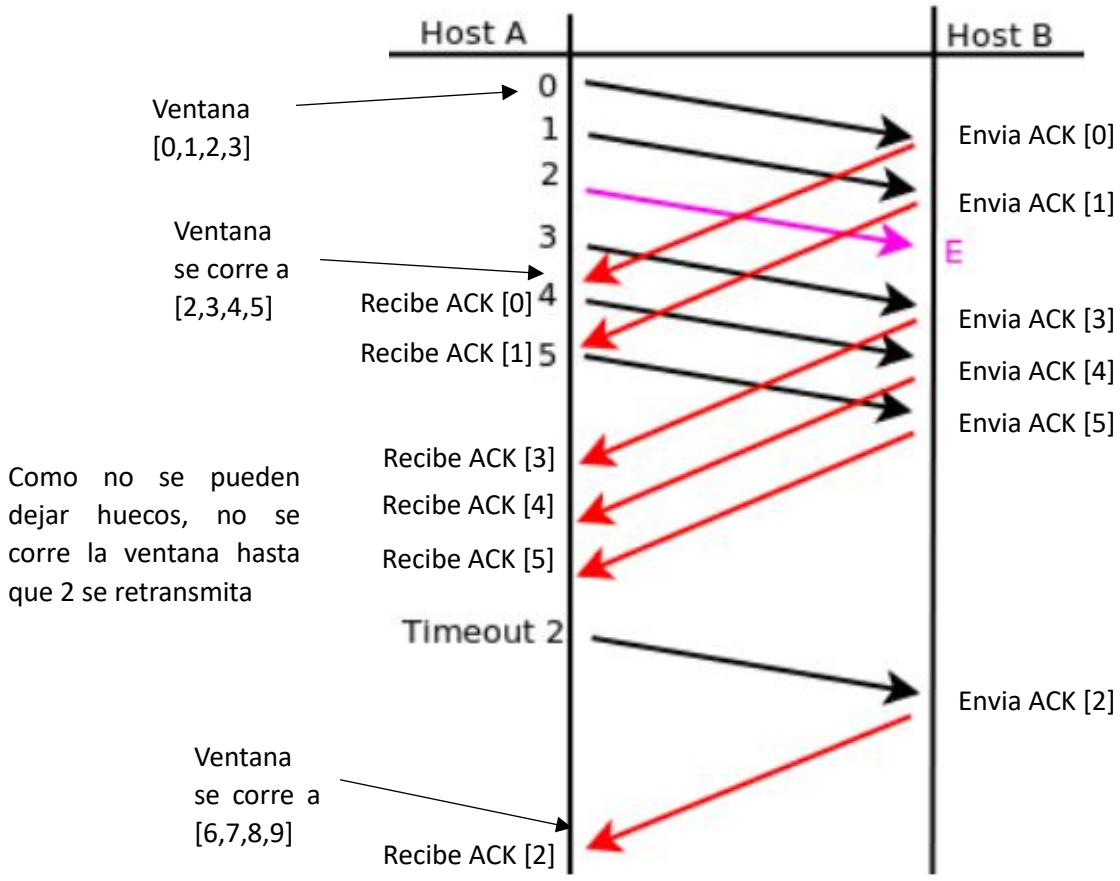
Modo Activo

- Conexión de control: port 21.
- Conexión de datos: port 20.
- El servidor de forma activa se conecta al cliente para generar la conexión de datos.

Modo Pasivo

- Conexión de control: port 21.
- Conexión de datos: port no privilegiado.
- El servidor de forma pasiva indica al cliente a que nuevo puerto debe conectarse. La conexión de datos la abre el cliente.

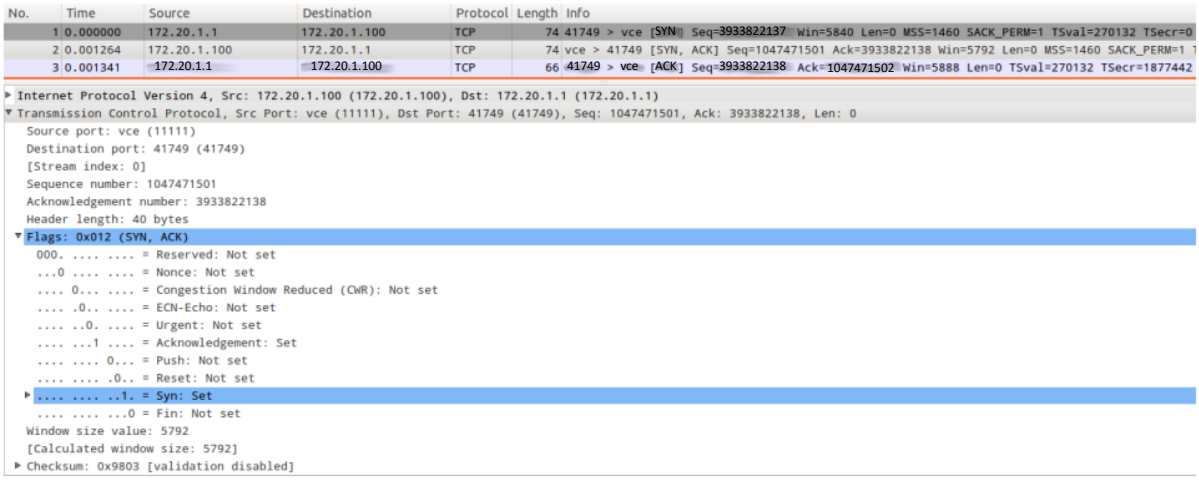
4. **Suponiendo Selective Repeat; tamaño de ventana 4 y sabiendo que E indica que el mensaje llegó con errores. Indique en el siguiente gráfico, la numeración de los ACK que el host B envía al Host A.**



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

   El tamaño de la ventana no debe exceder la mitad del tamaño total del espacio de números de secuencia. La razón detrás de esta restricción es evitar la posibilidad de que un número de secuencia se reutilice antes de que el ACK correspondiente haya llegado, ya que la ventana se implementa como un buffer circular, entonces si fuese más grande podría haber paquetes representados por la misma posición en el buffer lo que podría llevar a confusiones en la correcta interpretación de los frames.

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



SYN ➔Comienzo de 3WH

3933822137 ➔Se que es ese porque el receptor (linea 2) me indica que espera (ACK) que se le envie el segmento 3833822138, por lo tanto el que le envie en 1 es 3833822138 – 1

172.20.1.1 ➔ IP Origen

172.20.1.100 ➔ IP Destino

41749 ➔ Puerto Destino

vce ➔ Puerto Origen

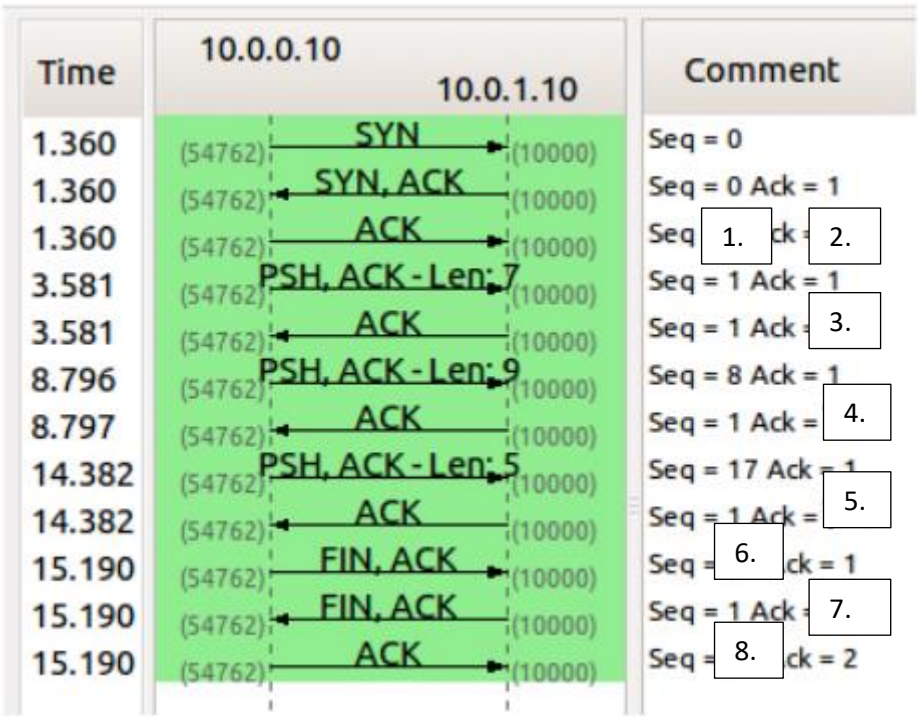
ACK ➔ Fin de 3WH

3933822138 ➔ En el segmento anterior a este se indicó que se esperaba 3933822138

1047471502 ➔ Se recibió 1047471501 , por lo que se espera recibir 1047471503

Las confirmaciones son “anticipativas”, indican el nro. de byte que esperan.

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



1. 1
2. 1
3. 8
4. 17
5. 22
6. 22
7. 23
8. 23

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

   El RTT es el tiempo que tarda un paquete en viajar desde un host a otro y recibir un ACK de vuelta.

   La opción de marcas de tiempo en TCP permite a los endpoints mantener una medición más precisa del tiempo de ida y vuelta (RTT) de la red entre ellos. Este valor ayuda a cada pila TCP a configurar y ajustar su temporizador de retransmisión. Hay otros beneficios, pero la medición RTT es el principal.

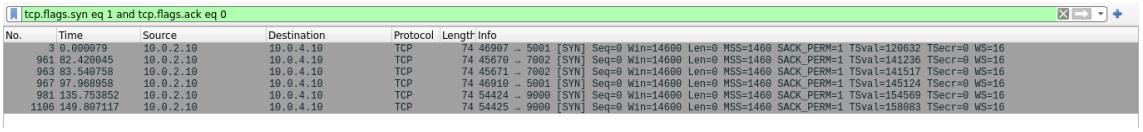
   Para ello se incluye un Timestamp Value TSval en cada segmento que se envía. Los valores TSval se repiten en el lado opuesto de la conexión en el campo Timestamp Echo Reply TSecr . Entonces, cuando se confirma un segmento, el remitente de ese segmento puede simplemente restar su marca de tiempo actual del valor TSecr para calcular una medición precisa del tiempo de ida y vuelta (RTT).

   RTT=TSecr − TSval

9. **Para la captura dada, responder las siguientes preguntas.**

a. ¿Cuántos intentos de conexiones TCP hay?

6

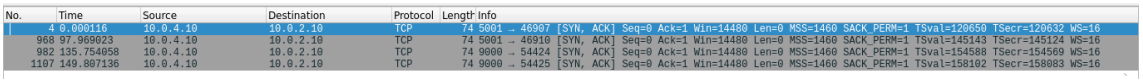


b. ¿Cuáles son la fuente y el destino (IP:port) para c/u?

Fuente	Destino
10.0.2.10:46907	10.0.4.10:5001
10.0.2.10:45670	10.0.4.10:7002
10.0.2.10:45671	10.0.4.10:7002
10.0.2.10:46910	10.0.4.10:5001
10.0.2.10:54424	10.0.4.10:9000
10.0.2.10:54425	10.0.4.10:9000

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?

4



Las exitosas tienen los flags SYN/ACK en 1, las fallidas tienen los flags RST/ACK en 1

d. Dada la primera conexión exitosa responder:

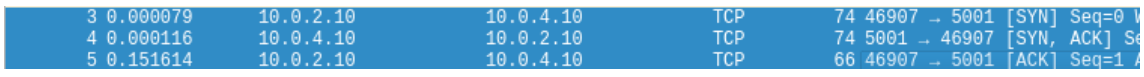
i. ¿Quién inicia la conexión?

La conexión es iniciada por 10.0.2.10:46907

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

El cliente es el que inicia la conexión 10.0.2.10:46907 y el servidor es el destino 10.0.4.10:7002

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



iv. ¿Cuáles ISNs se intercambian?

1 –

```
Sequence Number: 0    (relative sequence number)
Sequence Number (raw): 2218428254
[Next Sequence Number: 1    (relative sequence number)]
Acknowledgment Number: 0
Acknowledgment number (raw): 0
```

2 –

```
Sequence Number: 0    (relative sequence number)
Sequence Number (raw): 1292618479
[Next Sequence Number: 1    (relative sequence number)]
Acknowledgment Number: 1    (relative ack number)
Acknowledgment number (raw): 2218428255
```

3 –

```
Sequence Number: 1    (relative sequence number)
Sequence Number (raw): 2218428255
[Next Sequence Number: 1    (relative sequence number)]
Acknowledgment Number: 1    (relative ack number)
Acknowledgment number (raw): 1292618480
```

Se intercambian los ISN 2218428254, 1292618479 y 2218428255.

v. ¿Cuál MSS se negoció?

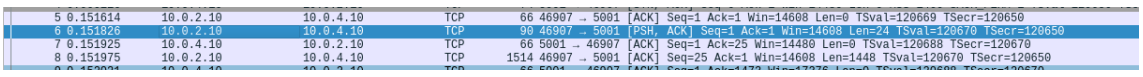
```
▾ Options: (20 bytes), Maximum segment size, SACK permitted,
  ▾ TCP Option - Maximum segment size: 1460 bytes
      Kind: Maximum Segment Size (2)
      Length: 4
      MSS Value: 1460
```

Se puede ver el segmento Nro 4 (siguiendo el orden de Wireshark)

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

10.0.2.10:46907, se incrementa su numero de secuencia (se incrementa cuando se envían datos), mientras que 10.0.4.10:7002 nunca lo incrementa (salvo en el 3WH). 10.0.2.10:46907 termina con el ISN relativo de 786458 y 10.0.4.10:7002 con 1.

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



(el azul)

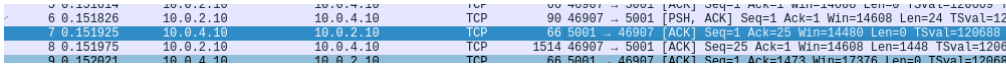
Origen: 10.0.2.10:46907
Destino: 10.0.4.10:7002
Tiempo: 0.151826

Nro. de fila: 6
Nro. de secuencia TCP: 1 (*221842855*)

i. **¿Cuántos datos lleva?**

Lleva 24 bytes.

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



(el azul)

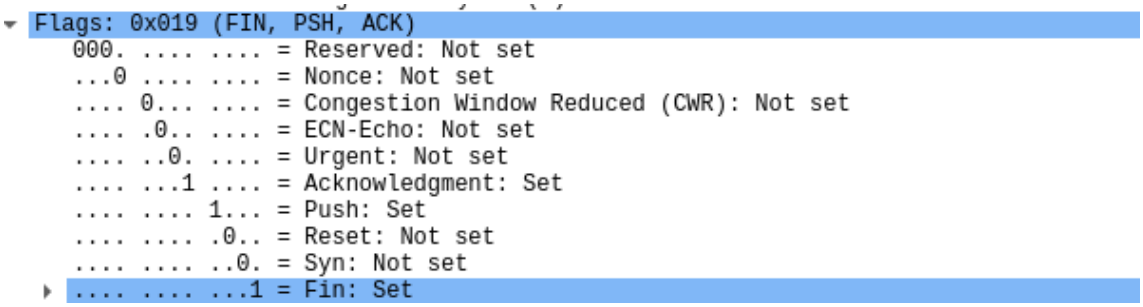
Tiempo: 0.151925
Nro. de fila: 7
Nro. de secuencia TCP: 1 (*1292618480*)

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

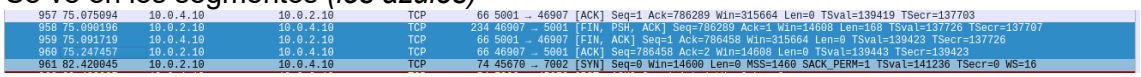
Confirma los 24 bytes, ya que indica que espera el byte nro 25.

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)?**

La inicia 10.0.2.10:46907. Utiliza los flags FIN, PSH y ACK



Se ve en los segmentos *(los azules)*



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

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

El control de flujo lo activa el receptor enviando ventanas más chicas. Esto deja en evidencia que el receptor tiene poco espacio (o no tiene más lugar) para seguir recibiendo datos. Esto se realiza a través del

campo de tamaño de ventana en los encabezados de los segmentos TCP.

b. ¿Qué problema resuelve?

Resuelve el problema de la posible saturación o congestión de los buffers en los endpoints. Al indicar al emisor que reduzca la cantidad de datos que está enviando, evita que el receptor se sobrecargue.

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

Cuanto tiempo dura activo depende del receptor (más que nada la velocidad en que lee la aplicación). El control de flujo está activo mientras el receptor envíe ventanas más pequeñas (indicando capacidad limitada). Durará activo hasta que el receptor envíe ventanas más grandes, lo que indica que tiene más capacidad para recibir datos.

En todo momento ambos extremos están actualizando su propia ventana.