

Web: 80

SSH: 22

DNS: 53

Web seguro: 443

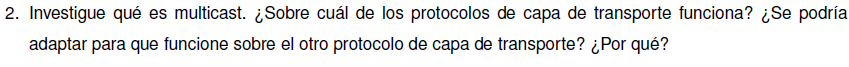
POP3: 110

IMAP: 143

SMTP: 25

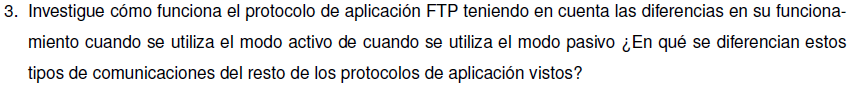
Linux: /etc/services

Windows: C:\WINDOWS\system32\drivers\etc\services



Multicast permite el envío simultáneo de información a varios usuarios de una red desde un punto o nodo. Sin embargo, a diferencia de la difusión amplia o broadcast, los destinatarios son previamente seleccionados por el emisor. Funciona con el protocolo UDP.

No se utiliza en TCP, porque en TCP el emisor requiere sincronizarse con el receptor en un handshake 1 a 1. El multicast tiene múltiples receptores, por lo que el emisor no puede realizar las sincronizaciones y no puede responder en caso de que se pierdan algunos paquetes durante la transferencia.

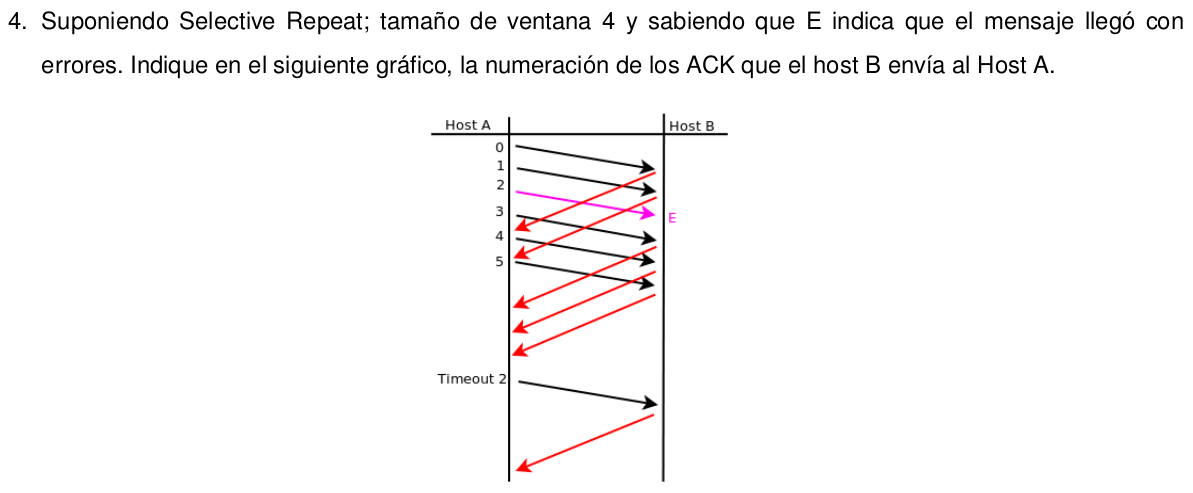


FTP es un protocolo de la capa de aplicación que utiliza los puertos 20 y 21. Permite transmitir archivos de un dispositivo a otro. Para poder transmitir datos se necesitan dos conexiones: una de control (establecer la conexion) y otra de datos (envio y recepcion de archivos); esta es la principal diferencia sobre los demás protocolos de la capa de aplicación debido a que los demás sólo tienen 1 conexión.

FTP tiene dos modos:

- Modo pasivo: el cliente inicia la conexión de control a través del puerto 21 y el servidor le responde indicándole en qué puerto debe conectarse para realizar la conexión de datos. Luego el cliente se conecta al puerto que le enviaron para iniciar la conexión de datos.

- Modo activo: el cliente se inicia la conexión de control hacia el puerto 21 con el servidor enviándole un puerto aleatorio en el que va a realizar la conexión de datos. Luego, el servidor es el encargado de iniciar la conexión hacia el puerto que le envió el cliente (el servidor siempre usa el puerto 20 para la conexión de datos en modo activo).



ack 0

ack 1

ack 3

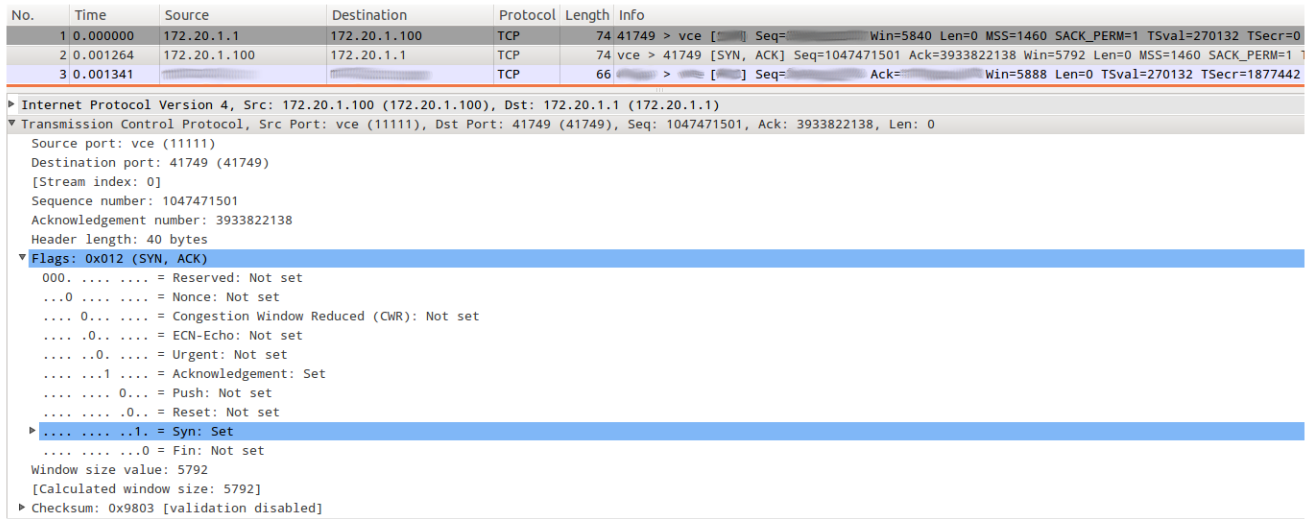
ack 4

ack 5

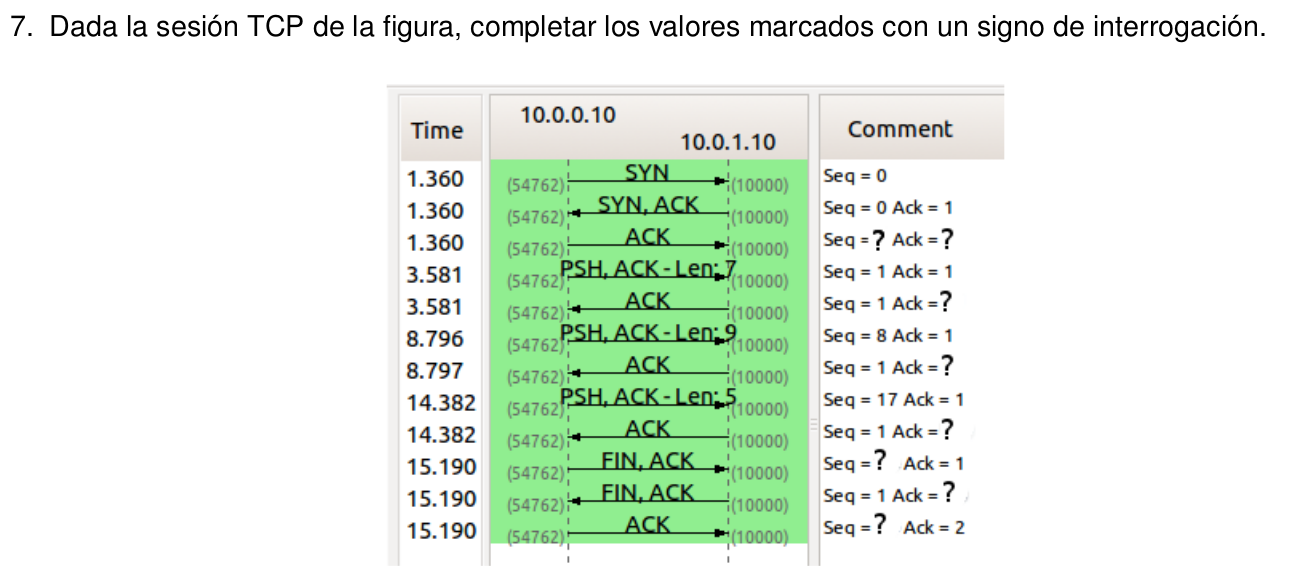
ack 2



La restricción de Selective Repeat que existe sobre el tamaño de la ventana es que, el tamaño de esta, debe ser menor o igual que la mitad del tamaño del espacio de números de secuencia. Esto se debe a que la ventana se implementa como un buffer circular, entonces se podría dar la situación en que el receptor no puede determinar si para un número de secuencia X, se trata del segmento X o el segmento X + Y (donde Y es una vuelta completa en el buffer circular). Esta situación se da cuando se pierden mensajes de ACK dirigidos al emisor.

1) Syn 2)3933822137 3) Source: 172.20.1.1 4)Destination 172.20.1.100

5 y 6) 41749 > vce 7) seq=3933822138 8) ack = 1047471502



Seq=1ack=1

ack=8

ack=17

Ack=22

Seq = 22

Ack = 23

seq=23

Es el tiempo que transcurre desde que se envía un segmento hasta que se recibe su reconocimiento. Puede definirse como el tiempo de ida y vuelta, y se mide en mili-segundos.

La opción *timestamp* permite que los endpoints puedan medir el RTT.

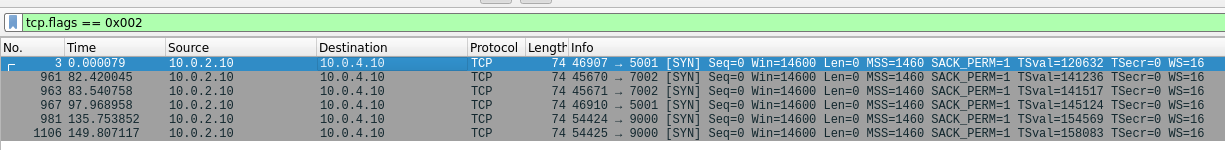
*TSval* y *TSecr* contienen información de timestamp que es enviada de ida y vuelta por los hosts.

*TSval* es un valor que se incluye en cada segmento para activar el timestamp y poder medir el RTT de cada paquete. Este valor es repetido por el lado contrario de la conexión en el valor *TSecr*. Entonces, cuando el segmento está confirmado, el emisor de ese segmento puede simplemente substraer su actual timestamp del valor de TSecr para computar un buen calculo de RTT.

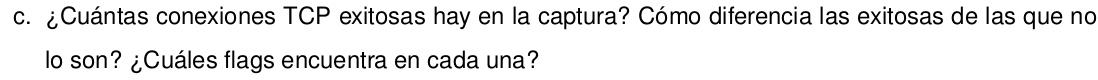
Fuente: https://www.qacafe.com/resources/tcp-timestamp-option/

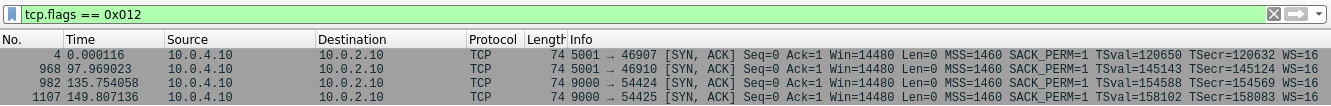


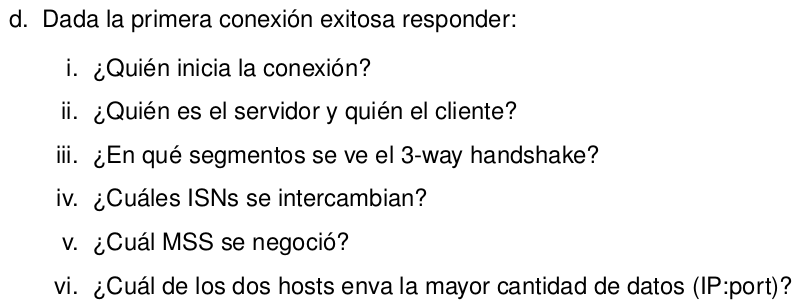
Hay 6 intentos de conexión. Estos se identifican por tener el bit SYN en 1.



Para todos los intentos de conexión la fuente es la IP: 10.0.2.10 mientras que el destino es 10.0.4.10.



Hay 4 conexiones exitosas. Diferencio las exitosas de las que no lo son, porque las exitosas poseen los flags SYN,ACK en 1.

i) ip 10.0.2.10

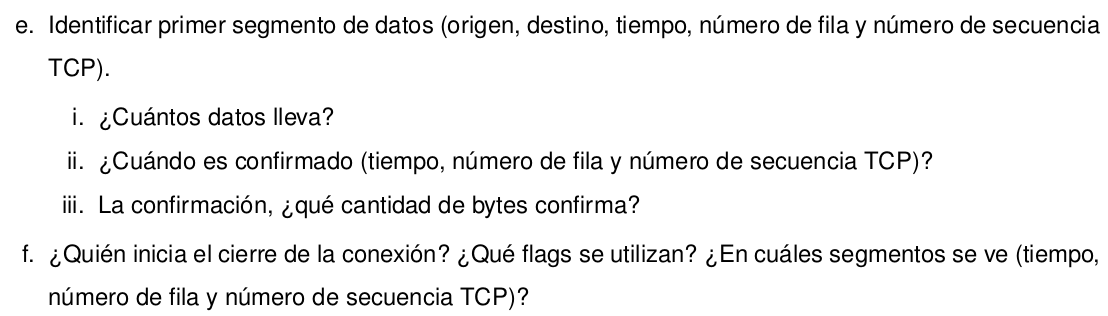
ii) Servidor: 10.0.4.10 Cliente: 10.0.2.10

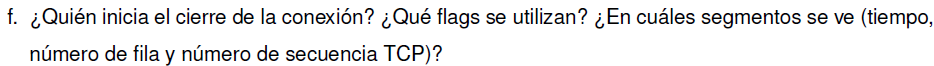
iii) Se ve en los primeros 3 segmentos el 3-way handshake, esto es posible de observar gracias a los flags.

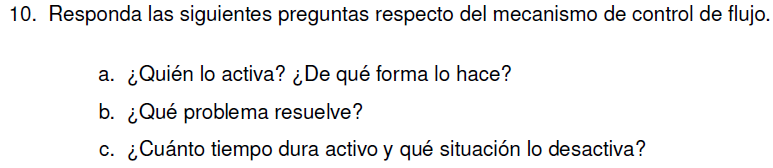
iv) 2218428254 del lado del cliente y 1292618479 del lado del servidor.

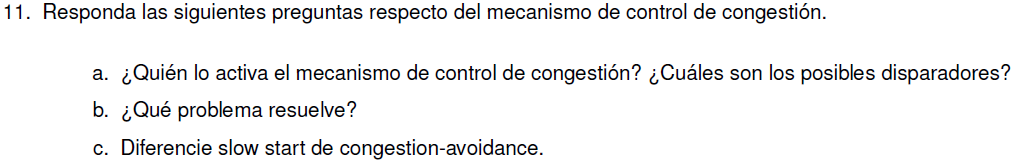
v) 1460. Se envia solo en los primeros 2 segmentos de la conexión.

vi)













9 comunicaciones



Con los mensajes ICMP



No necesariamente. Depende de como utilice la aplicación las comunicaciones udp.



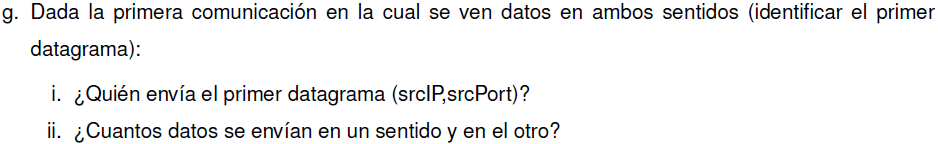
* Protocolo de Transferencia de Ficheros Trivial (TFTP)
* Sistema de Nombres de Dominio (**DNS**) servidor de nombres
* Llamada a Procedimiento Remoto (RPC), usado por el Sistema de Ficheros en Red (NFS)
* Sistema de Computación de Redes (NCS)
* Protocolo de Gestión Simple de Redes (SNMP)
* Aplicaciones de streaming de video o voz

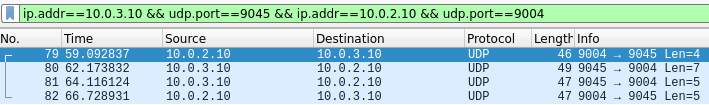


No tiene mecanismos de control de errores, sino de detección de errores. Como mecanismo de detección de errores, ofrece el “Checksum”, el cual es de uso opcional.



UDP no mantiene estados ni flags.



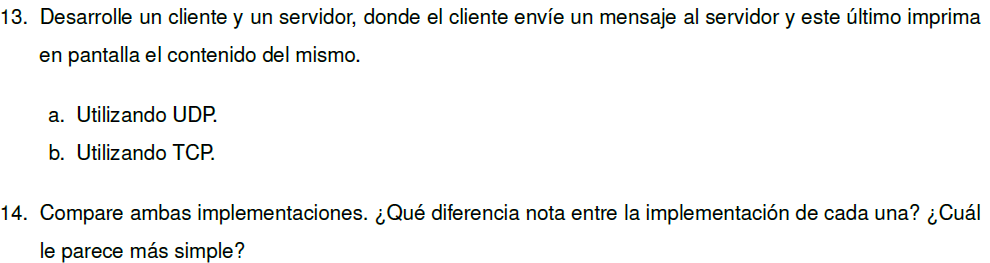


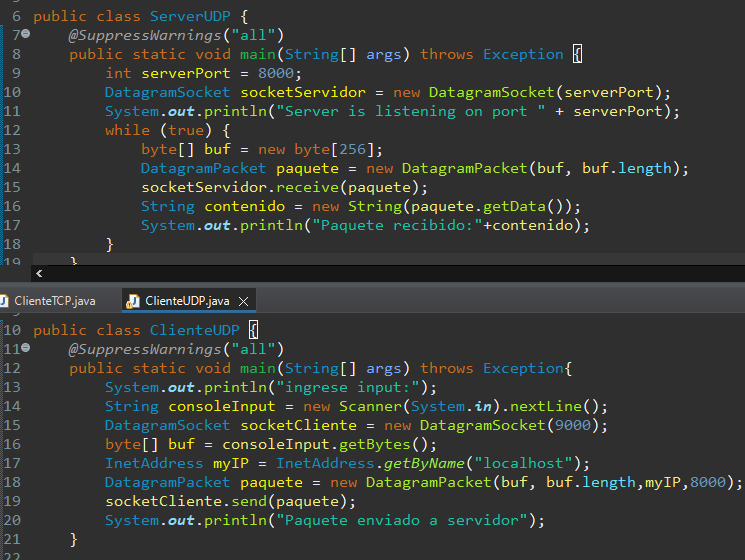
i) Primer datagrama:10.0.2.10:9004

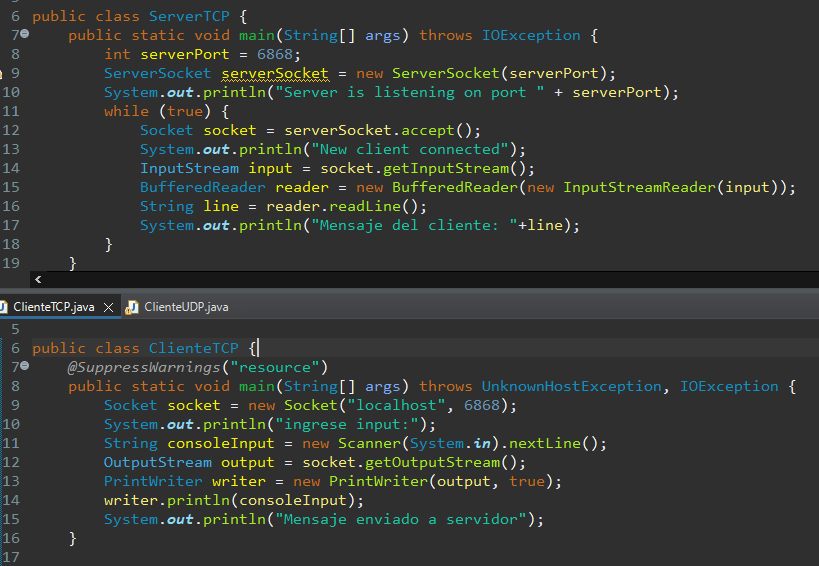
ii)10.0.2.10:9004 manda 9 bytes de datos y 10.0.3.10:9045 manda 12 bytes

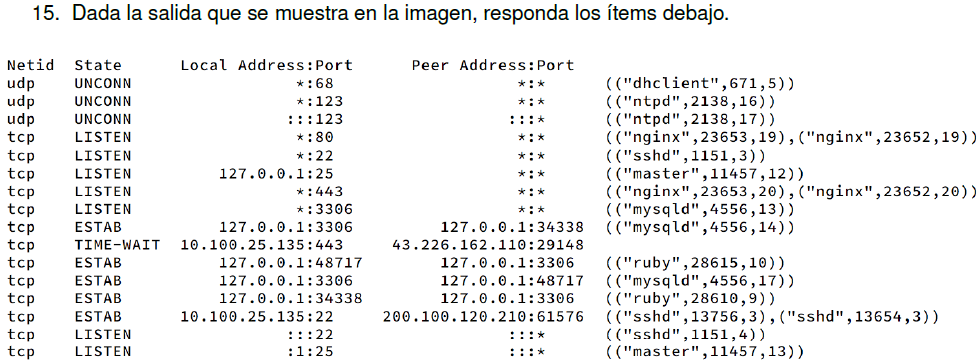


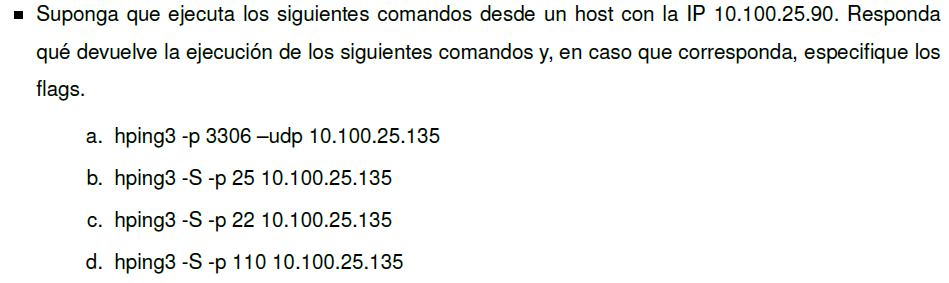
A nivel del protocolo UDP no se puede, ya que no se espera una respuesta por cada segmento enviado. Se podría hacer a nivel de capa de aplicación.











a) ICMP Port Unreachable. No hay ningún socket udp escuchando en 10.100.25.135:3306

b) Devuelve RST/ACK. No hay ningun socket tcp escuchando en 10.100.25.135:25

c) devuelve syn/ack

d) RST/ACK



3