

Obligatorio de Redes

Primera entrega del obligatorio de redes, alumno Federico Alonso, 182999.

3 Aplicaciones

3.1 Telnet

1. *Escriba el comando utilizado*

Acceso al equipo usuario y contraseña redes.

```
# Por defecto nos conecta al puerto 23
telnet 192.168.56.2

# En caso de especificar el puerto se usa el siguiente comando
telnet 192.168.56.2 23
```

- El usuario y contraseña son "ort-grupo1"

2. *¿Qué tipo de tareas puede realizar en el host de destino?*

En el host destino puedo tomar control del equipo, ejecutar comandos, revisar directorios, administrar aplicaciones, etc.

En el siguiente ejemplo se muestra la conexión al equipo y la ejecución de comandos:

```
Last login: Tue Mar 22 01:00:38 UTC 2022 from 192.168.56.1 on pts/0
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-96-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Tue Mar 22 01:02:18 UTC 2022

System load:  0.04          Processes:            100
Usage of /:   89.6% of 2.68GB Users logged in:     1
Memory usage: 9%           IP address for enp0s3: 192.168.56.2
Swap usage:   0%           IP address for enp0s8: 10.0.3.15

=> / is using 89.6% of 2.68GB

0 packages can be updated.
0 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check
```

your Internet connection or proxy settings

No mail.

```
ort-grupo1@servidor_redes:~$ systemctl status ufw.service
```

```
ÛÄ ufw.service - Uncomplicated firewall
```

```
Loaded: loaded (/lib/systemd/system/ufw.service; enabled; vendor preset:
enabled)
```

```
Active: active (exited) since Tue 2022-03-22 03:48:20 UTC; 2h 45min left
```

```
Docs: man:ufw(8)
```

```
Process: 401 ExecStart=/lib/ufw/ufw-init start quiet (code=exited,
status=0/SUCCESS)
```

```
Main PID: 401 (code=exited, status=0/SUCCESS)
```

Warning: Journal has been rotated since unit was started. Log output is incomplete or unavailable.

```
ort-grupo1@servidor_redes:~$ pwd
```

```
/home/ort-grupo1
```

```
ort-grupo1@servidor_redes:~$ ls /
```

```
bin    cdrom  etc    initrd.img    lib    lost+found  mnt  proc  run  snap
sys    usr    vmlinuz
```

```
boot  dev    home  initrd.img.old  lib64  media      opt  root  sbin  srv
```

```
tmp  var    vmlinuz.old
```

```
ort-grupo1@servidor_redes:~$
```

3. ¿Qué es necesario para que pueda acceder desde un equipo a otro remoto por Telnet?

Es necesario que el equipo destino tenga habilitado tanto el servicio como el puerto a la escucha. En el archivo `/etc/inetd.conf` se encuentra la configuración:

```
#
# Internet superserver configuration database
#
#
# Lines starting with "#:LABEL:" or "#<off>#" should not
# be changed unless you know what you are doing!
#
# If you want to disable an entry so it isn't touched during
# package updates just comment it out with a single '#' character.
#
# Packages should modify this file by using update-inetd(8)
#
# <service_name> <sock_type> <proto> <flags> <user> <server_path> <args>
#
#:INTERNAL: Internal services
#discard                stream  tcp      nowait  root    internal
#discard                dgram  udp      wait    root    internal
#daytime                stream  tcp      nowait  root    internal
#time                   stream  tcp      nowait  root    internal

#:STANDARD: These are standard services.
```

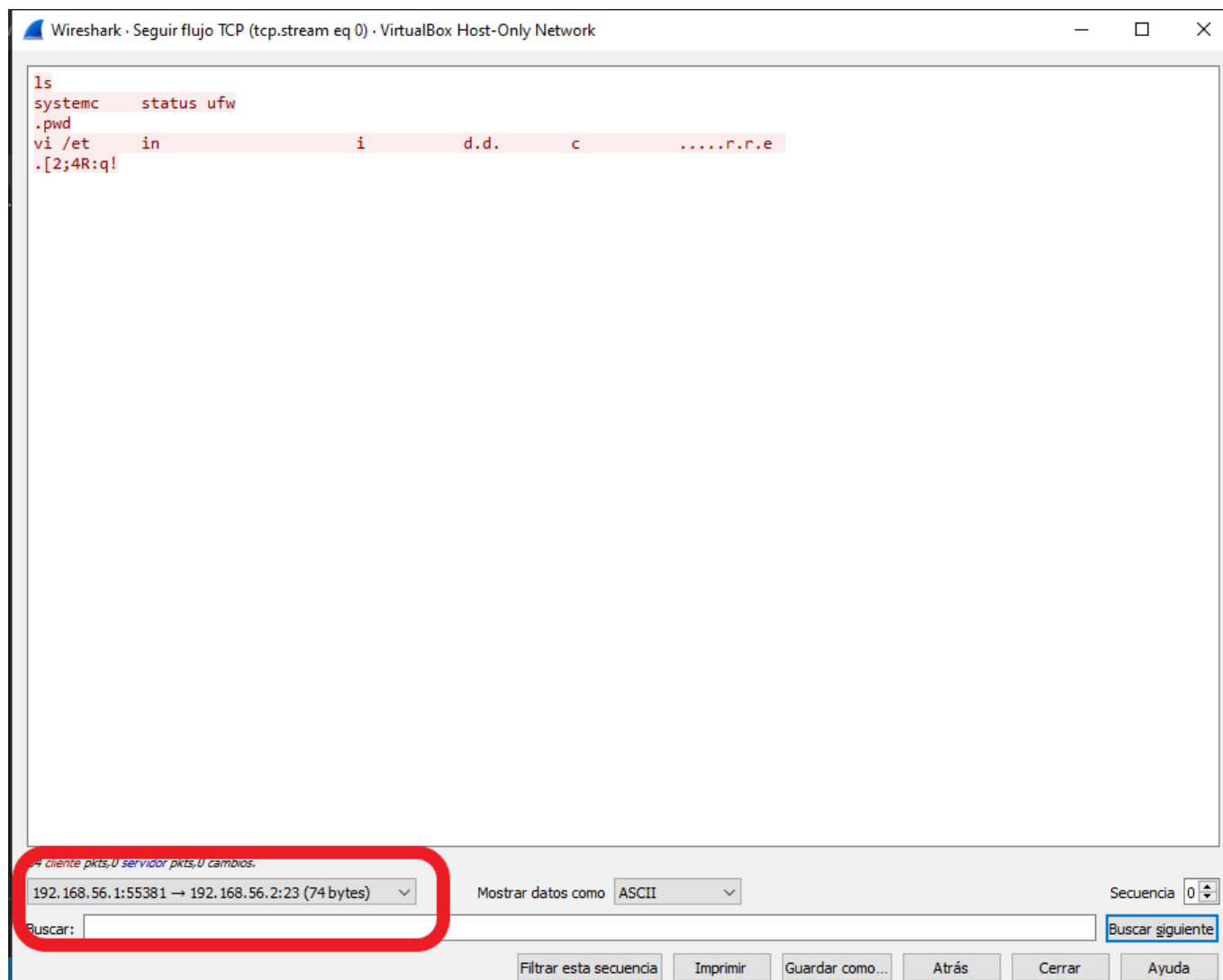
```
ftp          stream  tcp      nowait  root    /usr/sbin/tcpd
/usr/sbin/in.ftp
telnet       stream  tcp      nowait  telnetd /usr/sbin/tcpd
/usr/sbin/in.telnetd
```

4. ¿Hasta qué capa deben de entenderse los nodos entre si para que el acceso por Telnet sea exitoso?

Hasta la capa de aplicación.

5. Si analiza el tráfico capturado con Wireshark: 5.1 ¿Cuál es el número de puerto de origen y de destino con los que se está accediendo?

En la comunicación el puerto de origen está en el IP 192.168.56.1 (el pc host) y es el 55381, el puerto de destino es el estándar de Telnet, el puerto 23, en el IP 192.168.56.2 (el equipo virtual). Se puede ver en la siguiente imagen:



5.2 Identifique los paquetes Telnet de intercambio entre el cliente y el servidor. ¿Qué información contienen esos paquetes?

Contiene en texto plano toda la comunicación entre el cliente y el servidor, en la imagen anterior se pudo ver la comunicación hacia la máquina virtual, en la siguiente se puede ver la respuesta de la

misma:

```
ls
.[0m.[01;34mbin.[0m .[01;34mcdrom.[0m .[01;34metc.[0m .[40;31;01minitrd.img.[0m .[01;34mlib.[0m .
.[01;34mlost+found.[0m .[01;34mmnt.[0m .[01;34mproc.[0m .[01;34mrun.[0m .[01;34msnap.[0m .[01;34msys.[0m .[01;34musr.
.[0m .[01;36mvmlinuz.[0m
.[01;34mboot.[0m .[01;34mdev.[0m .[01;34mhome.[0m .[01;36minitrd.img.old.[0m .[01;34mlib64.[0m .[01;34media.[0m
.[01;34mopt.[0m .[01;34mroot.[0m .[01;34msbin.[0m .[01;34msrv.[0m .[30;42mtmp.[0m .[01;34mvar.[0m .[01;36mvmlinuz.old.
.[0m
ort-grupo1@servidor_redes:/# systemctl status ufw
.[0;1;32m...[0m ufw.service - Uncomplicated firewall.[m
Loaded: loaded (/lib/systemd/system/ufw.service; enabled; vendor preset: enabled).[m
Active: .[0;1;32mactive (exited).[0m since Tue 2022-03-22 04:22:13 UTC; 2h 57min left.[m
Docs: man:ufw(8).[m
Process: 392 ExecStart=/lib/ufw/ufw-init start quiet (code=exited, status=0/SUCCESS).[m
Main PID: 392 (code=exited, status=0/SUCCESS).[m
[m
Warning: Journal has been rotated since unit was started. Log output is incomplete or unavailable..[m
[7mlines 1-8.[m.[K..
ort-grupo1@servidor_redes:/# pwd
/
ort-grupo1@servidor_redes:/# vi /etc/in.
inetd.conf init.d/ initramfs-tools/ inputrc inserv.conf.d/
ort-grupo1@servidor_redes:/# vi /etc/inetd.D.[Kd..c..[D.[K.[D.[K.[D.[K.[D.[K.[D.[Kr.[D.[Ketd.conf
.[m].[m].[0m.[H.[2J.[9;1H"/etc/inetd.conf" [readonly] 40L, 1187C.[2;1H....[6n.[2;1H .[1;1H.[2;1H.[34m#:STANDARD: These are
standard services..[0m
ftp.[13Cstream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.ftpd
telnet.[10Cstream tcp nowait telnetd /usr/sbin/tcpd /usr/sbin/in.telnetd

.[34m#:BSD: Shell, login, exec and talk are BSD protocols.

#:#MAIL: Mail, news and uucp services..[0m.[9;135H24,1.[10C62%.[4;1H.[9;125H:.[4;1H.[9;1H.[K.[9;1H;q!
..[9;1H.[K.[9;1Hort-grupo1@servidor_redes:/#
```

Dentro de toda la conversación se marca en la misma el estado del ufw, el firewall del equipo.

3.2 SMTP: Simple Mail Transport Protocol - RFC 821

1. ¿Qué comando debe ejecutar para conectarse, mediante Telnet, al puerto 25 (SMTP) del servidor?

- Para conectarnos a la máquina virtual por Telnet, debemos ejecutar el comando:

```
telnet 192.168.56.2 25
```

2. Mediante el empleo del nombre asignado a su usuario (ort-grupo1) realice un diálogo SMTP a su propio usuario y al usuario ort-grupo2. Detalle cuáles fueron los comandos utilizados en cada caso.

```
helo ort-grupo1
250 servidor_redes.lan
mail from: ort-grupo1@servidor_redes
250 2.1.0 Ok
rcpt to: ort-grupo2@servidor_redes
250 2.1.5 Ok
```

```
data
354 End data with <CR><LF>.<CR><LF>
From: "Federico Alonso" <ort-grupo1@servidor_redes>
To: "Usuario Grupo 2" <ort-grupo2@servidor_redes>
Subject: Dialogo para el obligatorio
Hola ort-grupo2, esta es una prueba de envio de correo.
Saludos Federico Alonso.
```

```
220 servidor_redes.lan ESMTP Postfix (Ubuntu)
helo ort-grupo1
250 servidor_redes.lan
mail from: ort-grupo1@servidor_redes
502 5.5.2 Error: command not recognized
mail from: ort-grupo1@servidor_redes
250 2.1.0 Ok
rcpt to: ort-grupo2@servidor_redes
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
From: "Federico Alonso" <ort-grupo1@servidor_redes>
To: "Usuario Grupo 2" <ort-grupo2@servidor_redes>
Subject: Dialogo para el obligatorio
Hola ort-grupo2, esta es una prueba de envio de correo.
Saludos Federico Alonso.
.
250 2.0.0 Ok: queued as D6E72561
```

3. Identifique: el sobre, el encabezado y el cuerpo del mensaje. ¿Cuáles son las diferencias entre el sobre y el encabezado?

```
220 servidor_redes.lan ESMTP Postfix (Ubuntu)
helo ort-grupo1
250 servidor_redes.lan
mail from: ort-grupo1@servidor_redes
502 5.5.2 Error: command not recognized
mail from: ort-grupo1@servidor_redes
250 2.1.0 Ok
rcpt to: ort-grupo2@servidor_redes
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
From: "Federico Alonso" <ort-grupo1@servidor_redes>
To: "Usuario Grupo 2" <ort-grupo2@servidor_redes>
Subject: Dialogo para el obligatorio
Hola ort-grupo2, esta es una prueba de envio de correo.
Saludos Federico Alonso.
.
250 2.0.0 Ok: queued as D6E72561
```

Sobre

Encabezado

Cuerpo

En el sobre no podría mentir los datos, mientras que en el encabezado puedo poner las direcciones que desee.

3.3 POP3: Post Office Protocol version 3 - RFC 1939

1. Utilizando el protocolo Telnet, establezca una conexión al puerto estándar del protocolo. Liste el comando utilizado.

```
telnet 192.168.56.2 110
```

2. Establezca un diálogo POP3, ingresando con el usuario asignado a su grupo (ort-grupo1), liste sus mensajes y recupere los mismos. ¿Qué comandos utilizó?

```
user ort-grupo2
+OK
pass ort-grupo2
+OK Logged in
list
+OK 2 messages:
1 642
2 532
.
retr 1
+OK 642 octets
Return-Path: <ort-grupo1@servidor_redes>
X-Original-To: ort-grupo2@servidor_redes
Delivered-To: ort-grupo2@servidor_redes
Received: from ort-grupo1 (unknown [192.168.56.1])
    by servidor_redes.lan (Postfix) with SMTP id DE943561
    for <ort-grupo2@servidor_redes>; Tue, 22 Mar 2022 01:50:22 +0000
(UTC)
From: "Federico Alonso" <ort-grupo1@servidor_redes>
To: "Usuario Grupo 2" <ort-grupo2@servidor_redes>
Subject: Dialogo para el oblgatorio
Subject: Dialogo para el oblgatorio

Hola ort-grupo2, esta es una prueba de envio de correo.
Saludos Federico Alonso.
.
```

3. Verifique la correcta recepción de los mensajes que envió a su propio usuario en la parte de SMTP. Borre el último mensaje de la casilla. ¿Mediante qué comando lo hace?

```
dele 2
+OK Marked to be deleted.
list
+OK 1 messages:
```

```
1 642
```

```
.
```

3.4 HTTP: Hypertext Transfer Protocol - RFC 1945

1. Establezca una conexión al servidor a través del puerto habitual del protocolo HTTP, utilizando la aplicación Telnet. ¿Qué comando utilizó?

```
telnet 192.168.56.2 80
```

2. Recupere la página de prueba usada para el laboratorio, utilizando como URL la dirección IP del servidor. Indique el comando utilizado (cuando ejecute el comando, presione dos veces la tecla "enter"). Indique también la salida obtenida.

Como tuve problemas para poder verlo con el ejemplo, muestro la salida de una página simple levantada en mi equipo, utilizo el puerto 8082 en mi ejemplo, en la realidad el puerto HTTP es el 80.

```
# Los comandos utilizados son telnet localhost 8082 y GET / HTTP/1.0
```

```
federico@federico-PC:~/Documentos/Gitlab$ telnet localhost 8082
```

```
Trying 127.0.0.1...
```

```
Connected to localhost.
```

```
Escape character is '^['.
```

```
GET / HTTP/1.0
```

```
HTTP/1.1 200 OK
```

```
Date: Mon, 02 May 2022 23:54:17 GMT
```

```
Server: Apache/2.4.53 (Debian)
```

```
Last-Modified: Mon, 02 May 2022 23:36:53 GMT
```

```
ETag: "1b6-5de0fddf9a4ed"
```

```
Accept-Ranges: bytes
```

```
Content-Length: 438
```

```
Vary: Accept-Encoding
```

```
Connection: close
```

```
Content-Type: text/html
```

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>Document</title>
```

```
</head>
```

```
<body>
```

```
  <h1>Hola mundo desde apache!!</h1>
```

```
  <p>Página de ejemplo para que podamos ver la diferencia entre ver una  
  página por código, en contrasete con verla en un navegador.</p>
```

```
</body>
</html>Connection closed by foreign host.
```

3. Indique el lenguaje en el cual está escrita la página

La página está escrita en HTML.

4. ¿Con qué comando traería únicamente el encabezado de la página? Indique la salida obtenida y compárela con la obtenida en el punto anterior

Con el comando HEAD, se realiza de la siguiente forma:

```
# El comando sería telnet localhost 8082 para conectarnos
# HEAD / HTTP/1.0 para obtener el head
federico@federico-PC:~/Documentos/Gitlab$ telnet localhost 8082
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Date: Mon, 02 May 2022 23:56:23 GMT
Server: Apache/2.4.53 (Debian)
Last-Modified: Mon, 02 May 2022 23:36:53 GMT
ETag: "1b6-5de0fddf9a4ed"
Accept-Ranges: bytes
Content-Length: 438
Vary: Accept-Encoding
Connection: close
Content-Type: text/html

Connection closed by foreign host.
```

5. Acceda mediante un navegador web a la página y compare los resultados obtenidos.

Mediante un navegador podemos ver la página pronta para que un usuario pueda entenderla, esto se debe a que el navegador interpreta el HTTP y la muestra en pantalla correctamente.

Bienvenidos al Laboratorio de Redes

Este es el la pagina por defecto del servidor http



En la pagina hay texto plano, y tambien alguna imagen, para poder ver como se comporta un explorador al bajar la pagina por http y comparar con el modo texto.

3.5 FTP: File Transfer Protocol - RFC 959

1. ¿Cuál es el objetivo del protocolo FTP?

El objetivo de FTP es transferir archivos entre dos equipos.

2. Conéctese mediante el cliente FTP de Windows al servidor cuya IP es 192.168.56.2. Utilice el usuario ort-grupo1. ¿Qué comando utilizó?

Se utiliza el comando: ftp 192.168.56.2

```
C:\Users\fnico>ftp 192.168.56.2
Conectado a 192.168.56.2.
220 servidor_redes FTP server (Version 6.4/OpenBSD/Linux-ftpd-0.17) ready.
500 'OPTS UTF8 ON': command not understood.
Usuario (192.168.56.2:(none)): ort-grupo1
331 Password required for ort-grupo1.
Contraseña:
230 User ort-grupo1 logged in.
ftp>
```

3. Posiciónese en el directorio /home/publico y liste el contenido del mismo. ¿Qué archivos se observa?

```
ftp> pwd
257 "/home/ort-grupo1" is current directory.
ftp> cd /home/publico
250 CWD command successful.
ftp> dir /home/publico
200 PORT command successful.
150 Opening ASCII mode data connection for '/bin/ls'.
total 476
-rw-r--r-- 1 redes redes 483328 Sep  8  2011 putty.exe
-rw-r--r-- 1 redes redes   447 Sep  8  2011 texto.txt
226 Transfer complete.
ftp: 126 bytes recibidos en 0.00segundos 42.00a KB/s.
```

1. texto.txt
2. putty.exe

4. Copie en su PC el archivo correspondiente al cliente SSH (archivo putty.exe). Indique la secuencia de comandos usada, teniendo en cuenta de que se trata de un ejecutable.

```
ftp> get putty.exe
200 PORT command successful.
150 Opening ASCII mode data connection for 'putty.exe' (483328 bytes).
226 Transfer complete.
ftp: 484455 bytes recibidos en 0.02segundos 23069.29a KB/s.
```

5. ¿Qué secuencia de comandos utilizaría si deseara copiar a su equipo todo el contenido del directorio actual indicando que no se desea recibir confirmación para cada archivo a transferir?

```
ftp> prompt
Modo interactivo Desactivado .
ftp> mget *
200 Type set to A.
200 PORT command successful.
550 72364 putty.exe: No such file or directory.
200 PORT command successful.
550 72382 texto.txt: No such file or directory.
200 PORT command successful.
150 Opening ASCII mode data connection for 'putty.exe' (483328 bytes).
226 Transfer complete.
ftp: 484455 bytes recibidos en 0.02segundos 28497.35a KB/s.
200 PORT command successful.
150 Opening ASCII mode data connection for 'texto.txt' (447 bytes).
226 Transfer complete.
ftp: 447 bytes recibidos en 0.00segundos 447.00a KB/s.
ftp>
```

3.6. SSH: Security Shell

1. Usando el cliente SSH obtenido en el ejercicio anterior, establezca una conexión con el servidor al puerto estándar del servidor SSH. Indique cuál fue la configuración empleada.

En putty debo poner en el host el IP del servidor al que quiero acceder, también puedo poner el nombre de usuario, además configurar el puerto en caso de que no sea el estándar. Putty nos permite además guardar las configuraciones y configurar claves SSH para poder acceder sin colocar contraseña, haciendo que el mismo usuario del equipo se autentique en el momento de la conexión.

```
C:\Users\fnico>ssh ort-grupo1@192.168.56.2
The authenticity of host '192.168.56.2 (192.168.56.2)' can't be
established.
ECDSA key fingerprint is
SHA256:5kjNN4930KJI+GkWcPSPGtnecuHXxLkBEZLybxCZ7tU.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.2' (ECDSA) to the list of known
hosts.
ort-grupo1@192.168.56.2's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-96-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Tue Apr  5 02:07:50 UTC 2022

System load:  0.0               Processes:            97
Usage of /:   91.0% of 2.68GB   Users logged in:    0
```

```
Memory usage: 9%          IP address for enp0s3: 192.168.56.2
Swap usage: 0%           IP address for enp0s8: 10.0.3.15
```

```
=> / is using 91.0% of 2.68GB
```

```
* Super-optimized for small spaces - read how we shrank the memory
  footprint of MicroK8s to make it the smallest full K8s around.
```

```
https://ubuntu.com/blog/microk8s-memory-optimisation
```

```
0 packages can be updated.
0 updates are security updates.
```

```
No mail.
```

```
Last login: Tue Mar 22 02:12:41 2022 from 192.168.56.1
```

```
ort-grupo1@servidor_redes:~$
```

2. ¿Qué diferencias existen entre usar SSH y Telnet?

La principal diferencia es que SSH utiliza un protocolo de comunicación seguro, establece un canal de comunicación encriptado para mantener la confidencialidad, la autenticación y la integridad de la información. Esto no es posible con Telnet, que utiliza un protocolo de comunicación no seguro.

Por este motivo es que no se habilita más telnet en los servidores. Por otro lado, telnet nos da la flexibilidad de poder establecer comunicación mediante otros puertos que no sólo se utilizan para el control total del equipo, lo que se vió en los ejercicios anteriores.

4 DNS

1. Realice una consulta DNS por un registro A usando el comando nslookup. Elija un sitio que no haya sido utilizado recientemente. Indique el sitio, el comando utilizado y la respuesta.

- Sitio: mercadolibre.com.br
- Comando: nslookup mercadolibre.com.br

```
> nslookup mercadolibre.com.br
Servidor:  mercadolibre.com.br
Addresses: 18.65.48.109
           18.65.48.84
           18.65.48.27
           18.65.48.55
```

```
DNS request timed out.
  timeout was 2 seconds.
```

```
DNS request timed out.
  timeout was 2 seconds.
```

```
*** Se agotó el tiempo de espera de la solicitud a mercadolibre.com.br
```

2. Detalle el el intercambio observado en Wireshark por el servidor para resolver la consulta, poniendo énfasis en quién origina la consulta, quién responde y los posibles pasos intermedios. Puede ser necesario aplicar filtros en Wireshark para lograr reducir la cantidad de paquetes visualizados (protocolo DNS por ejemplo). Tener en cuenta que existe una gran cantidad de tráfico que se cursa habitualmente por la conexión utilizada por el PC para acceder a Internet.

The screenshot displays the Wireshark network protocol analyzer interface. The top filter bar shows the filter: `dns && (ip.addr==192.168.56.2 || ip.addr==192.168.56.1 || ip.addr==192.168.79.221)`.

The main packet list on the left shows a series of DNS queries and responses. The packet details pane on the right shows the structure of a DNS query (Frame 438) and its response (Frame 841). The packet bytes pane at the bottom shows the raw data of the response.

Frame 438: 79 bytes on wire (632 bits), 79 bytes captured on interface \Device\NPF{...}

- Ethernet II, Src: PcsCompu_73:f4:8b (08:00:27:73:f4:8b), Dst: 192.168.56.1
- Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.2
- User Datagram Protocol, Src Port: 55080, Dst Port: 53
- Domain Name System (query)
 - Transaction ID: 0x000b
 - Flags: 0x0100 Standard query
 - Questions: 1
 - Answer RRs: 0
 - Authority RRs: 0
 - Additional RRs: 0
 - Queries
 - mercadolibre.com.br: type A, class IN

Frame 841: 283 bytes on wire (2264 bits), 283 bytes captured (2264 bits) on interface \Device\NPF{...}

- Ethernet II, Src: PcsCompu_73:f4:8b (08:00:27:73:f4:8b), Dst: 0a:00:00:00:00:00
- Internet Protocol Version 4, Src: 192.168.56.2, Dst: 192.168.56.1
- User Datagram Protocol, Src Port: 53, Dst Port: 55080
- Domain Name System (response)
 - Transaction ID: 0x000b
 - Flags: 0x8100 Standard query response, No error
 - Questions: 1
 - Answer RRs: 4
 - Authority RRs: 4
 - Additional RRs: 0
 - Queries
 - mercadolibre.com.br: type A, class IN
 - Answers
 - mercadolibre.com.br: type A, class IN, addr 18.65.48.109
 - mercadolibre.com.br: type A, class IN, addr 18.65.48.84
 - mercadolibre.com.br: type A, class IN, addr 18.65.48.27
 - mercadolibre.com.br: type A, class IN, addr 18.65.48.55
 - Authoritative nameservers
 - mercadolibre.com.br: type A, class IN, addr 18.65.48.55

The packet bytes pane at the bottom shows the raw data of the response, including the domain name and IP addresses.

3. Reinicie la captura de Wireshark (puede guardar la anterior si así lo desea). Realice la misma consulta DNS y analice nuevamente el intercambio en Wireshark. ¿El servidor contesta de caché? ¿Cómo distingue si la respuesta es de caché o no? Detalle las diferencias con el caso anterior.

Como podemos ver en estas siguientes imágenes, el servidor efectivamente está respondiendo de caché, ya que el mismo no "sale" a buscar la dirección IP buscada, él anteriormente había realizado la búsqueda y "recuerda" la respuesta.

No.	Time	Source	Destination	Protocol	Length	Info
390	1.311596	192.168.79.221	193.0.14.129	DNS	102	Standard query 0x54e1 A mercadolibre.com.br OPT
405	1.355691	193.0.14.129	192.168.79.221	DNS	541	Standard query response 0x54e1 A mercadolibre.com.br NS a.dns.br NS b.dns.br NS c.dns.br NS d.dns.br NS e.dns.br NS f.dns.br DS RRSIG
431	1.431837	192.168.79.221	200.219.154.10	DNS	102	Standard query 0x6a10 A mercadolibre.com.br OPT
438	1.309658	192.168.56.1	192.168.56.2	DNS	79	Standard query 0x000b A mercadolibre.com.br
441	1.451524	200.219.154.10	192.168.79.221	DNS	422	Standard query response 0x6a10 A mercadolibre.com.br NS ns-264.awsdns-33.com NS ns-700.awsdns-23.net NS ns-1154.awsdns-16.org NS ns-15
644	1.729116	192.168.79.221	205.251.196.130	DNS	102	Standard query 0x9bce A mercadolibre.com.br OPT
656	1.778496	205.251.196.130	192.168.79.221	DNS	294	Standard query response 0x9bce A mercadolibre.com.br A 18.65.48.27 A 18.65.48.109 A 18.65.48.55 A 18.65.48.84 NS ns-1154.awsdns-16.org
841	2.022430	192.168.56.2	192.168.56.1	DNS	283	Standard query response 0x000b A mercadolibre.com.br A 18.65.48.109 A 18.65.48.84 A 18.65.48.27 A 18.65.48.55 NS ns-264.awsdns-33.com

Comentario:		Introduzca un comentario para el botón de filtro	Aceptar	Cancelar
-------------	--	--	---------	----------

No.	Time	Source	Destination	Protocol	Length	Info
179	1.196551	192.168.79.221	18.65.48.109	DNS	68	Standard query 0x0015 A nslookup
190	1.184458	192.168.56.1	192.168.56.2	DNS	79	Standard query 0x0013 A mercadolibre.com.br
191	1.185246	192.168.56.2	192.168.56.1	DNS	283	Standard query response 0x0013 A mercadolibre.com.br A 18
192	1.186485	192.168.56.1	192.168.56.2	DNS	79	Standard query 0x0014 AAAA mercadolibre.com.br
193	1.187139	192.168.56.2	192.168.56.1	DNS	166	Standard query response 0x0014 AAAA mercadolibre.com.br S
488	3.206114	192.168.79.221	18.65.48.109	DNS	68	Standard query 0x0016 AAAA nslookup

4. Obtenga la dirección IP asociada al nombre **www.lab.ort.edu.uy**. Detalle el comando y la salida obtenida.

El comando es `nslookup www.lab.ort.edu.uy 192.168.56.2`, y la salida se muestra a continuación.

```
C:\Users\fnico>nslookup www.lab.ort.edu.uy 192.168.56.2
Servidor: mail.lab.ort.edu.uy
Address: 192.168.56.2

Nombre: www.lab.ort.edu.uy
Address: 192.168.56.2
```

5. Obtenga todos los dominios asociados a la dirección IP **192.168.56.2**. Indique el comando y la respuesta obtenida.

```
C:\Users\fnico>nslookup 192.168.56.2 192.168.56.2
Servidor: www.lab.ort.edu.uy
Address: 192.168.56.2

Nombre: dns.lab.ort.edu.uy
Address: 192.168.56.2
```

6. ¿Cuál es el registro por el que se debe preguntar para conocer el servidor al cual podemos entregar correos para el dominio **lab.ort.edu.uy**? Realice la consulta y detalle los comandos utilizados.

Se debe preguntar por el registro MX, nos muestra el servidor y datos de la configuración del mismo.

```
C:\Users\fnico>nslookup -type=mx www.lab.ort.edu.uy 192.168.56.2
Servidor: mail.lab.ort.edu.uy
Address: 192.168.56.2

lab.ort.edu.uy
```

```
primary name server = dns.lab.ort.edu.uy
responsible mail addr = ubuntu.lab.ort.edu.uy
serial = 2008100100
refresh = 10800 (3 hours)
retry = 1800 (30 mins)
expire = 3600 (1 hour)
default TTL = 3600 (1 hour)
```

7. ¿Cuál es el comando para encontrar los servidores autoritativos del dominio com.uy? Indique el comando y detalle los resultados obtenidos.

Se debe preguntar por el registro SOA.

```
C:\Users\fnico>nslookup -type=soa com.uy 192.168.56.2
Servidor:  www.lab.ort.edu.uy
Address:  192.168.56.2

Respuesta no autoritativa:
com.uy
    primary name server = ns1.anteldata.com.uy
    responsible mail addr = dns-admin.antel.net.uy
    serial = 2204040545
    refresh = 10800 (3 hours)
    retry = 1800 (30 mins)
    expire = 604800 (7 days)
    default TTL = 3600 (1 hour)

com.uy  nameserver = ns1.anteldata.com.uy
com.uy  nameserver = ns2.anteldata.com.uy
com.uy  nameserver = seciu.edu.uy
```

8. Realice una consulta no recursiva, usando el registro A, correspondiente a un dominio por el cual no haya consultado anteriormente. Indique el comando utilizado y la salida obtenida. ¿Puede obtener la respuesta? ¿Por qué?

Cuando hacemos una consulta no recursiva, no conseguiremos la respuesta entera, sino una respuesta parcial correspondiente al root server a los que le debemos preguntar por el top-level domain, en nuestro caso que buscamos por movistar.com, debemos preguntarle por .com.

```
C:\Users\fnico>nslookup
Servidor predeterminado:  UnKnown
Address:  192.168.201.1

> server 192.168.56.2
Servidor predeterminado:  [192.168.56.2]
Address:  192.168.56.2

> set norecurse
> movistar.com
```

```
Servidor: [192.168.56.2]
Address: 192.168.56.2

Nombre: movistar.com
Served by:
- k.gtld-servers.net
      com
- l.gtld-servers.net
      com
- m.gtld-servers.net
      com
- e.gtld-servers.net
      com
- a.gtld-servers.net
      com
- g.gtld-servers.net
      com
- f.gtld-servers.net
      com
- h.gtld-servers.net
      com
- c.gtld-servers.net
      com
- i.gtld-servers.net
      com
```

9. Vuelva a realizar la consulta pero en modo recursivo. Indique el comando utilizado y la salida obtenida. ¿Puede obtener la respuesta ahora?¿Por qué?

Ahora sí vemos la respuesta, ya que el servidor de dns realiza toda la consulta y nos devuelve el resultado final.

```
> set recurse
> movistar.com
Servidor: [192.168.56.2]
Address: 192.168.56.2

Respuesta no autoritativa:
Nombre: movistar.com
Address: 192.0.66.32
```

10. Haga una consulta correspondiente a www.yahoo.com y repita inmediatamente la misma consulta. Detalle las consultas y las salidas obtenidas. Compare las respuestas y explique las diferencias.

Podemos ver que se realiza round robin de dns. Esto se realiza por dos motivos principales, uno de ellos repartir la carga y el otro mantener disponible el servicio en caso de falla de alguno de los servidores. En cualquiera de los casos (IPv6 o IPv4) nos alteró el orden en las direcciones que marca las flechas, para que clientes distintos pregunten a servidores distintos.

```
> www.yahoo.com
Servidor: [192.168.56.2]
Address: 192.168.56.2

Respuesta no autoritativa:
Nombre: new-fp-shed.wg1.b.yahoo.com
Addresses: 2001:4998:44:3507::8000 # <-----
           2001:4998:124:1507::f000
           2001:4998:44:3507::8001
           2001:4998:124:1507::f001
           74.6.143.26
           74.6.143.25 # <-----
           74.6.231.21
           74.6.231.20
Aliases: www.yahoo.com

> www.yahoo.com
Servidor: [192.168.56.2]
Address: 192.168.56.2

Respuesta no autoritativa:
Nombre: new-fp-shed.wg1.b.yahoo.com
Addresses: 2001:4998:44:3507::8001 # <-----
           2001:4998:124:1507::f000
           2001:4998:44:3507::8000
           2001:4998:124:1507::f001
           74.6.143.26
           74.6.231.21 # <-----
           74.6.143.25
           74.6.231.20
Aliases: www.yahoo.com
```

A su vez, podemos observar la diferencia en tiempo entre una consulta y la otra con el siguiente comando.

```
C:\Users\fnico>powershell "Measure-Command { nslookup pedidosya.com 2>
$null }" | FINDSTR "^Milliseconds"
Milliseconds      : 159

C:\Users\fnico>powershell "Measure-Command { nslookup pedidosya.com 2>
$null }" | FINDSTR "^Milliseconds"
Milliseconds      : 76
```


5 TCP/HTTP

5.1 Análisis de mensajes y secuencia TCP

1. Acceda mediante el navegador a la página del servidor (<http://192.168.56.2>). Luego de obtenida la página, detenga la captura.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.56.1	192.168.56.2	TCP	74	47220 → 80 [SYN] Seq=3418145883 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1561862148 TSecr=0 WS=128
2	0.000365999	192.168.56.2	192.168.56.1	TCP	74	80 → 47220 [SYN, ACK] Seq=1018748408 Ack=3418145884 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=1853624692 TSecr=1561862148
3	0.000389271	192.168.56.1	192.168.56.2	TCP	66	47220 → 80 [ACK] Seq=3418145884 Ack=1018748409 Win=64256 Len=0 TSval=1561862148 TSecr=1853624692
4	0.000509372	192.168.56.1	192.168.56.2	HTTP	423	GET / HTTP/1.1
5	0.000595402	192.168.56.2	192.168.56.1	TCP	66	80 → 47220 [ACK] Seq=1018748409 Ack=3418146241 Win=64896 Len=0 TSval=1853624692 TSecr=1561862148
6	0.000844699	192.168.56.2	192.168.56.1	HTTP	731	HTTP/1.1 200 OK (text/html)
7	0.000854108	192.168.56.1	192.168.56.2	TCP	66	47220 → 80 [ACK] Seq=3418146241 Ack=1018749074 Win=64128 Len=0 TSval=1561862148 TSecr=1853624693
8	0.048645731	192.168.56.1	192.168.56.2	HTTP	375	GET /logoort.gif HTTP/1.1
9	0.049218967	192.168.56.2	192.168.56.1	TCP	1514	80 → 47220 [ACK] Seq=1018749074 Ack=3418146550 Win=64640 Len=1448 TSval=1853624741 TSecr=1561862196 [TCP segm...]
10	0.049257373	192.168.56.1	192.168.56.2	TCP	66	47220 → 80 [ACK] Seq=3418146550 Ack=1018750522 Win=64128 Len=0 TSval=1561862197 TSecr=1853624741
11	0.049293353	192.168.56.2	192.168.56.1	HTTP	1493	HTTP/1.1 200 OK (GIF89a)
12	0.049303844	192.168.56.1	192.168.56.2	TCP	66	47220 → 80 [ACK] Seq=3418146550 Ack=1018751949 Win=63488 Len=0 TSval=1561862197 TSecr=1853624741
13	0.052694313	192.168.56.1	192.168.56.2	HTTP	375	GET /favicon.ico HTTP/1.1
14	0.053022483	192.168.56.2	192.168.56.1	HTTP/XML	545	HTTP/1.1 404 Not Found
15	0.053042262	192.168.56.1	192.168.56.2	TCP	66	47220 → 80 [ACK] Seq=3418146859 Ack=1018752428 Win=64128 Len=0 TSval=1561862201 TSecr=1853624745
16	3.589976389	PcsCompu_f6:...	Broadcast	ARP	42	Who has 192.168.56.101? Tell 192.168.56.100
17	3.589991676	PcsCompu_f6:...	Broadcast	ARP	42	Who has 192.168.56.101? Tell 192.168.56.100
18	5.159646178	0a:00:27:00:...	PcsCompu_25:3...	ARP	42	Who has 192.168.56.2? Tell 192.168.56.1
19	5.160327960	PcsCompu_25:...	0a:00:27:00:0...	ARP	60	192.168.56.2 is at 08:00:27:25:3d:6b
20	6.062944850	192.168.56.2	192.168.56.1	TCP	66	80 → 47220 [FIN, ACK] Seq=1018752428 Ack=3418146859 Win=64384 Len=0 TSval=1853630756 TSecr=1561862201
21	6.063260885	192.168.56.1	192.168.56.2	TCP	66	47220 → 80 [FIN, ACK] Seq=3418146859 Ack=1018752429 Win=64128 Len=0 TSval=1561868211 TSecr=1853630756
22	6.063978234	192.168.56.2	192.168.56.1	TCP	66	80 → 47220 [ACK] Seq=1018752429 Ack=3418146860 Win=64384 Len=0 TSval=1853630757 TSecr=1561868211

2. Identifique el establecimiento de conexión. Describa de la misma, los números de secuencia (SEQ) inicial de ambas partes, los números de reconocimiento (ACK), el largo del segmento, como así también qué banderas van activas durante la secuencia de segmentos intercambiados.

Los largos de segmento de dichas transmisiones es de 0. En la imagen se muestran las banderas, los números de secuencia y las banderas asociadas.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.56.1	192.168.56.2	TCP	74	47220 → 80 [SYN] Seq=3418145883 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1561862148 TSecr=0 WS=128
2	0.000365999	192.168.56.2	192.168.56.1	TCP	74	80 → 47220 [SYN, ACK] Seq=1018748408 Ack=3418145884 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=1853624692 TSecr=1561862148
3	0.000389271	192.168.56.1	192.168.56.2	TCP	66	47220 → 80 [ACK] Seq=3418145884 Ack=1018748409 Win=64256 Len=0 TSval=1561862148 TSecr=1853624692
4	0.000509372	192.168.56.1	192.168.56.2	HTTP	423	GET / HTTP/1.1
5	0.000595402	192.168.56.2	192.168.56.1	TCP	66	80 → 47220 [ACK] Seq=1018748409 Ack=3418146241 Win=64896 Len=0 TSval=1853624692 TSecr=1561862148
6	0.000844699	192.168.56.2	192.168.56.1	HTTP	731	HTTP/1.1 200 OK (text/html)
7	0.000854108	192.168.56.1	192.168.56.2	TCP	66	47220 → 80 [ACK] Seq=3418146241 Ack=1018749074 Win=64128 Len=0 TSval=1561862148 TSecr=1853624693
8	0.048645731	192.168.56.1	192.168.56.2	HTTP	375	GET /logoort.gif HTTP/1.1
9	0.049218967	192.168.56.2	192.168.56.1	TCP	1514	80 → 47220 [ACK] Seq=1018749074 Ack=3418146550 Win=64640 Len=1448 TSval=1853624741 TSecr=1561862196 [TCP segm...]
10	0.049257373	192.168.56.1	192.168.56.2	TCP	66	47220 → 80 [ACK] Seq=3418146550 Ack=1018750522 Win=64128 Len=0 TSval=1561862197 TSecr=1853624741
11	0.049293353	192.168.56.2	192.168.56.1	HTTP	1493	HTTP/1.1 200 OK (GIF89a)
12	0.049303844	192.168.56.1	192.168.56.2	TCP	66	47220 → 80 [ACK] Seq=3418146550 Ack=1018751949 Win=63488 Len=0 TSval=1561862197 TSecr=1853624741
13	0.052694313	192.168.56.1	192.168.56.2	HTTP	375	GET /favicon.ico HTTP/1.1
14	0.053022483	192.168.56.2	192.168.56.1	HTTP/XML	545	HTTP/1.1 404 Not Found
15	0.053042262	192.168.56.1	192.168.56.2	TCP	66	47220 → 80 [ACK] Seq=3418146859 Ack=1018752428 Win=64128 Len=0 TSval=1561862201 TSecr=1853624745
16	3.589976389	PcsCompu_f6:...	Broadcast	ARP	42	Who has 192.168.56.101? Tell 192.168.56.100
17	3.589991676	PcsCompu_f6:...	Broadcast	ARP	42	Who has 192.168.56.101? Tell 192.168.56.100
18	5.159646178	0a:00:27:00:...	PcsCompu_25:3...	ARP	42	Who has 192.168.56.2? Tell 192.168.56.1
19	5.160327960	PcsCompu_25:...	0a:00:27:00:0...	ARP	60	192.168.56.2 is at 08:00:27:25:3d:6b
20	6.062944850	192.168.56.2	192.168.56.1	TCP	66	80 → 47220 [FIN, ACK] Seq=1018752428 Ack=3418146859 Win=64384 Len=0 TSval=1853630756 TSecr=1561862201
21	6.063260885	192.168.56.1	192.168.56.2	TCP	66	47220 → 80 [FIN, ACK] Seq=3418146859 Ack=1018752429 Win=64128 Len=0 TSval=1561868211 TSecr=1853630756
22	6.063978234	192.168.56.2	192.168.56.1	TCP	66	80 → 47220 [ACK] Seq=1018752429 Ack=3418146860 Win=64384 Len=0 TSval=1853630757 TSecr=1561868211

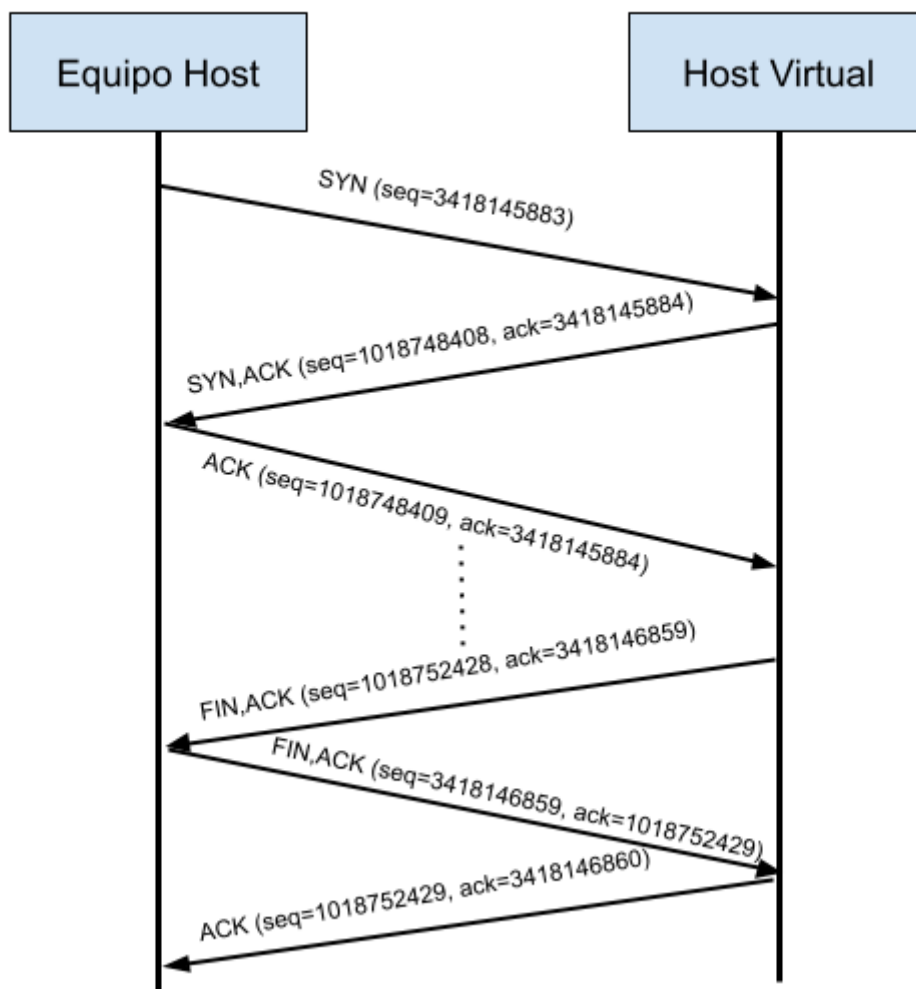
En wireshark las banderas se ven de esta forma:

```

[Next Sequence Number: 3700978392]
Acknowledgment Number: 1250777163
1010 .... = Header Length: 40 bytes (10)
▼ 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 = Acknowledgment: Set
  .... ...0 = Push: Not set
  .... ...0 = Reset: Not set
  ▶ .... ...1 = Syn: Set
  .... ...0 = Fin: Not set
[TCP Flags: .....A..S.]
Window: 65160
[Calculated window size: 65160]

```

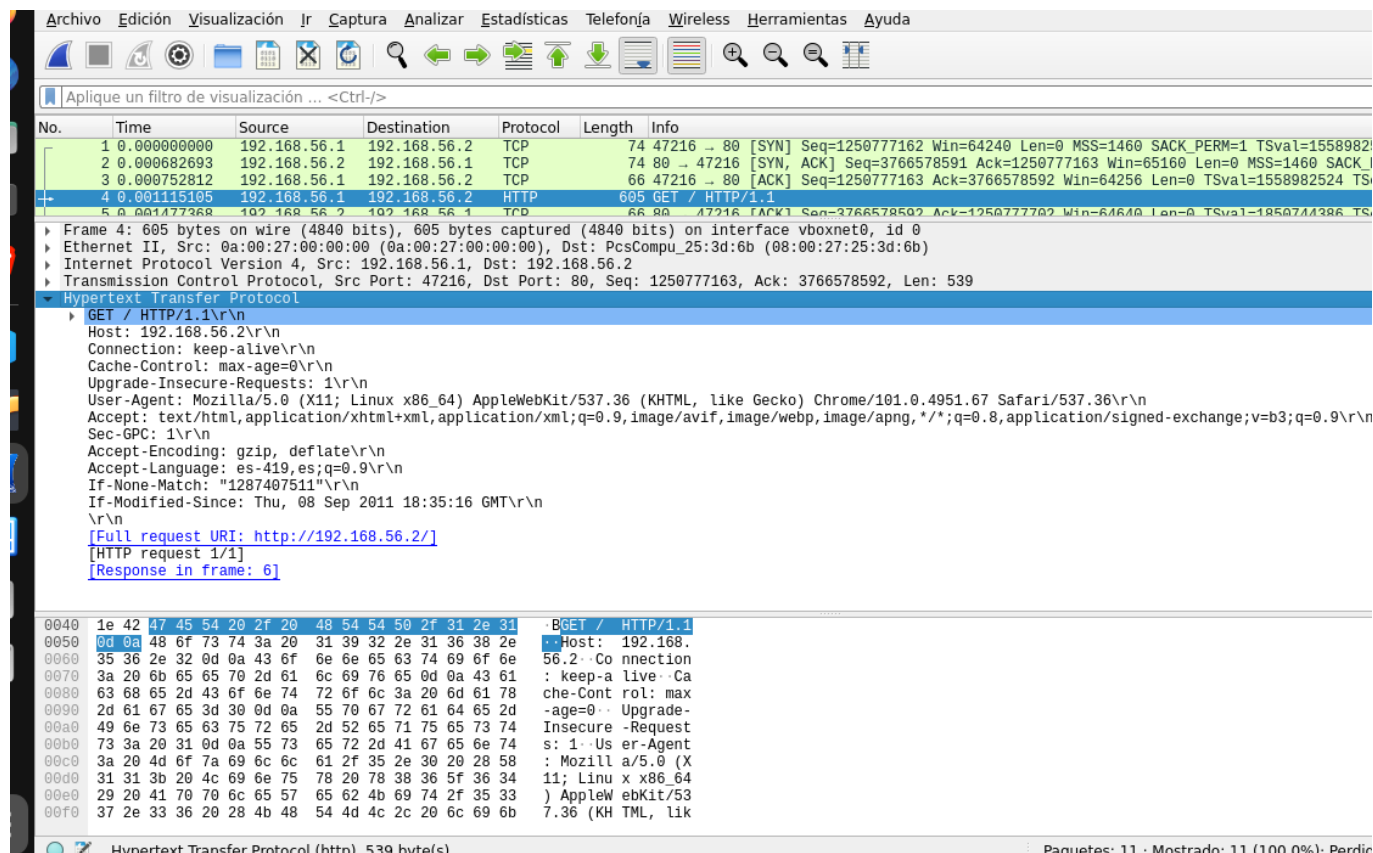
3. Identifique la finalización de la conexión, describa la secuencia de segmentos intercambiados indicando: los números de SEQ y ACK, como así también banderas activas y largo de segmentos.



Se marcan en la imagen anterior en las últimas líneas, se ven con las banderas FIN. Primero el servidor informa la finalización con la bandera FIN, el cliente le da el ACK y envía también la bandera FIN, finalmente el servidor da el ACK. Éste es un final simétrico.

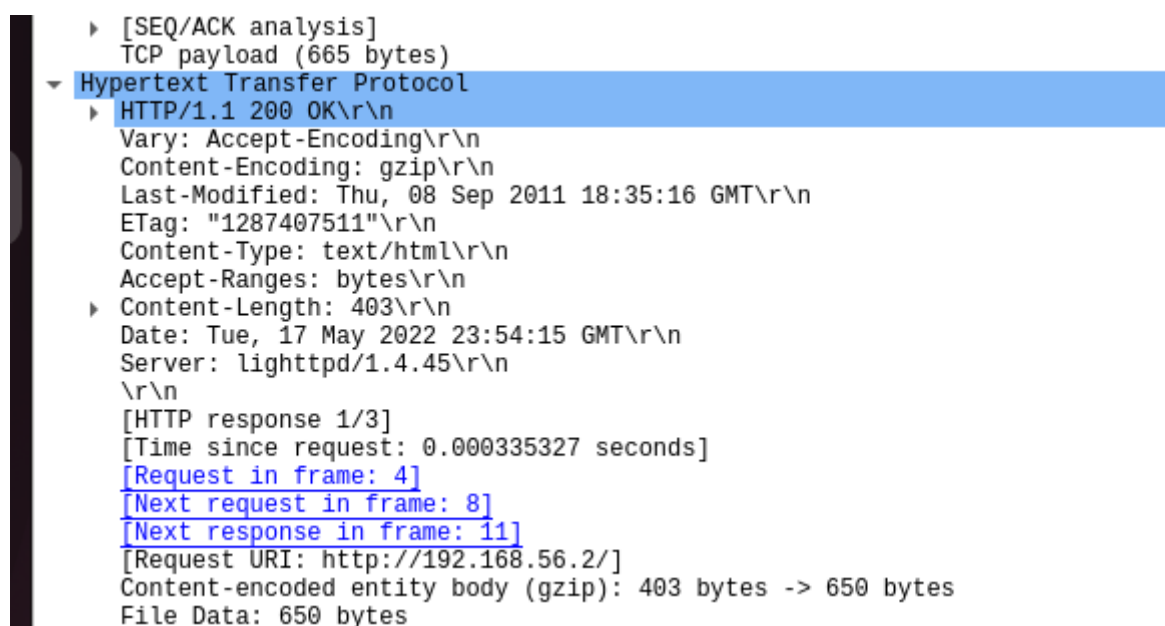
4. Identifique en el request HTTP, aquel encabezado de solicitud y su valor, que le brinda información al servidor acerca del navegador web cliente. Justifique su uso.

Los datos obtenidos a través de la cabecera, permite al servidor brindar la información de la mejor manera para que el cliente pueda recibirla sin problemas y de la forma más rápida posible, por ejemplo, If-Modified-Since permite enviar contenido sólo si éste ha sido modificado desde el momento especificado. O el User-Agent, para que el servidor brinde cierta información sólo a aquellos navegadores que puedan manejarlo.

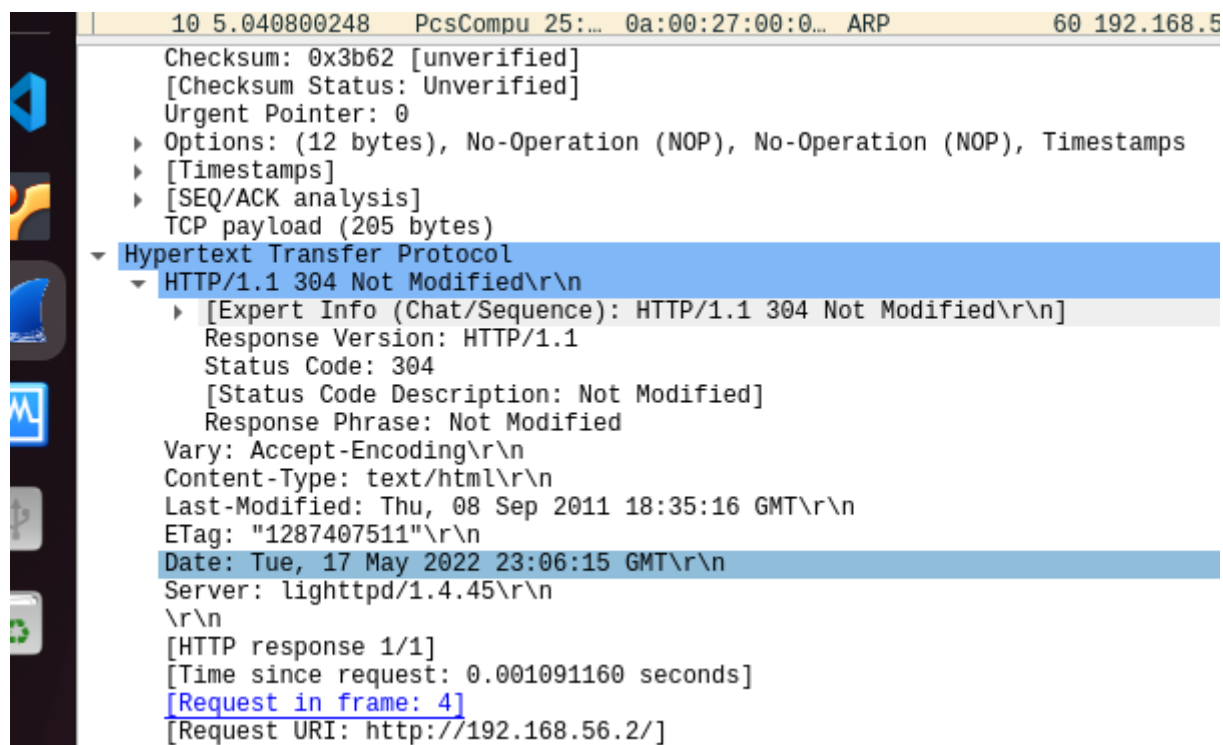


5. Identifique en el response HTTP, aquel encabezado de respuesta y su valor, que le brinda información al cliente acerca del servidor web.

Acá se ve un ejemplo en el que el navegador recibe la información por primera vez:



Acá se ve un ejemplo en el que el navegador ya poseía la información:



```

10 5.040800248 PcsCompu 25:... 0a:00:27:00:0... ARP 60 192.168.5
Checksum: 0x3b62 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
  Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  [Timestamps]
  [SEQ/ACK analysis]
  TCP payload (205 bytes)
- Hypertext Transfer Protocol
  HTTP/1.1 304 Not Modified\r\n
    [Expert Info (Chat/Sequence): HTTP/1.1 304 Not Modified\r\n]
    Response Version: HTTP/1.1
    Status Code: 304
    [Status Code Description: Not Modified]
    Response Phrase: Not Modified
    Vary: Accept-Encoding\r\n
    Content-Type: text/html\r\n
    Last-Modified: Thu, 08 Sep 2011 18:35:16 GMT\r\n
    ETag: "1287407511"\r\n
    Date: Tue, 17 May 2022 23:06:15 GMT\r\n
    Server: lighttpd/1.4.45\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.001091160 seconds]
    [Request in frame: 4]
    [Request URI: http://192.168.56.2/]

```

6. Analizando la captura realizada. ¿En qué momento se incrementan los números de secuencia y en qué valor lo hacen? Identifique todos los casos posibles.

Existen dos casos en que los números de secuencia se incrementa:

- Los números de secuencia se incrementan en 1, cuando reciben el ACK del otro host que le informa por ejemplo cuando se manda ACK=1250777702 se informa que hasta el 1250777701 recibió bien, por lo que espera el 1250777702.
- El caso de incremento mayor a 1 es cuando se envía data en el segmento, en este caso varía según el tamaño de datos enviados.

7. Analizando los números de secuencia. ¿Puede deducir cuántos bytes fueron enviados en cada sentido? Justifique su respuesta.

Sí, es más, wireshark lo puede hacer por nosotros, dentro del análisis de cada transmisión existe un campo llamado SEQ/ACK analysis, en el que se muestran los bits transmitidos entre otros datos, podríamos sumarlos todos y obtener los datos de la transmisión.

```

▶ Frame 4: 423 bytes on wire (3384 bits), 423 bytes captured (3384 bits) on interface vboxnet0, id 0
▶ Ethernet II, Src: 0a:00:27:00:00:00 (0a:00:27:00:00:00), Dst: PcsCompu_25:3d:6b (08:00:27:25:3d:6b)
▶ Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.2
▼ Transmission Control Protocol, Src Port: 47220, Dst Port: 80, Seq: 3418145884, Ack: 1018748409, Len: 357
  Source Port: 47220
  Destination Port: 80
  [Stream index: 0]
  [Conversation completeness: Complete, WITH_DATA (31)]
  [TCP Segment Len: 357]
  Sequence Number: 3418145884
  [Next Sequence Number: 3418146241]
  Acknowledgment Number: 1018748409
  1000 .... = Header Length: 32 bytes (8)
  Flags: 0x018 (PSH, ACK)
  Window: 502
  [Calculated window size: 64256]
  [Window size scaling factor: 128]
  Checksum: 0xa676 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  [Timestamps]
  [SEQ/ACK analysis]
    [RTT: 0.000389271 seconds]
    [Bytes in flight: 357]
    [Bytes sent since last PSH flag: 357]
    TCP payload (357 bytes)
  ▶ Hypertext Transfer Protocol
    0020 38 02 b8 74 00 50 cb bc c4 5c 3c b8 dd f9 80 18 8 .t .P . .\< . . . .
    0030 01 f6 a6 76 00 00 01 01 08 0a 5d 18 20 04 6e 7c . .v . . . . ] . n |
    0040 11 74 47 45 54 20 2f 20 48 54 54 50 2f 31 2e 31 .tGET / HTTP/1.1
    0050 0d 0a 48 6f 73 74 3a 20 31 39 32 2e 31 36 38 2e .Host: 192.168.
    0060 35 36 2e 32 0d 0a 55 73 65 72 2d 41 67 65 6e 74 56.2 .Us er-Agent

```

No.: 4 · Time: 0.000509372 · Source: 192.168.56.1 · Destination: 192.168.56.2 · Protocol: HTTP · Length: 423 · Info: GET / HTTP/1.1

X Cerrar

Los números de secuencia aumentan en 1 por byte enviado, por lo que haciendo la resta entre números de secuencia tenemos el total:

- host 1 (Enviados) = 3418146859 - 3418145883 = 976 bytes
- host 2 (Recibidos) = 1018752429 - 1018748408 = 4021 bytes

8. ¿Puede observar en algún momento la bandera PSH en TCP? ¿Para qué se utiliza?

Sí, se ve en las transmisiones HTTP, lo que realiza dicha bandera es el aviso a la capa de transporte de que dicha información debe ser pasada inmediatamente a las capas superiores para la capa de aplicación, no debe esperar para hacerlo.

9. Si una parte de la comunicación desea enviar solamente un reconocimiento y no datos. ¿Cuál número de secuencia debe enviar?

Debe enviar el ACK de la comunicación que recibió, por ejemplo, si recibió el paquete con el número 1018752429, y no desea enviar datos, sólo reconoce el paquete anterior, manda un ACK con el número siguiente, 1018752430.

10. Capture nuevamente e intente acceder ahora a la dirección del servidor pero en el puerto 8080. (http://192.168.56.2:8080). ¿Logró conectarse? ¿Por qué sucede esto? ¿Qué bandera se utiliza para señalar esto?

No nos pudimos conectar.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.56.1	192.168.56.2	TCP	74	49366 → 8080 [SYN] Seq=1769845349 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1564875659 TSecr=0 WS=128
2	0.000700490	192.168.56.2	192.168.56.1	TCP	60	8080 → 49366 [RST, ACK] Seq=0 Ack=1769845350 Win=0 Len=0
3	5.024484813	0a:00:27:00:00:00	PcsCompu_25:3d:6b	ARP	42	Who has 192.168.56.2? Tell 192.168.56.1
4	5.025095321	PcsCompu_25:3d:6b	0a:00:27:00:00:00	ARP	60	192.168.56.2 is at 08:00:27:25:3d:6b
5	5.177630294	PcsCompu_25:3d:6b	0a:00:27:00:00:00	ARP	60	Who has 192.168.56.1? Tell 192.168.56.2
6	5.177668623	0a:00:27:00:00:00	PcsCompu_25:3d:6b	ARP	42	192.168.56.1 is at 0a:00:27:00:00:00

Esto sucede porque el puerto se encuentra cerrado:

```

fedrico@feedrico-pc: ~
fedrico@feedrico-pc: ~

fedrico@feedrico-pc:~$ telnet 192.168.56.2 8080
Trying 192.168.56.2...
telnet: Unable to connect to remote host: Connection refused
fedrico@feedrico-pc:~$

```

Para señalar este comportamiento se utiliza la bandera RST.

11. Descargue la página del laboratorio (<http://192.168.56.2>). Indique cuál es la fecha de última modificación de la misma y cuál es el código de la respuesta HTTP.

El código de la respuesta HTTP es 200, la última modificación fue el 8 de septiembre de 2011.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.56.1	192.168.56.2	TCP	74	47234 → 80 [SYN] Seq=3377541014 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1565541036 TSecr=0 WS=128
2	0.000108171	192.168.56.1	192.168.56.2	TCP	74	47236 → 80 [SYN] Seq=3462831445 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1565541037 TSecr=0 WS=128
3	0.000399678	192.168.56.2	192.168.56.1	TCP	74	80 → 47234 [SYN, ACK] Seq=138365209 Ack=3377541015 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=1857304393 TSecr=1857304393
4	0.000444273	192.168.56.1	192.168.56.2	TCP	66	47234 → 80 [ACK] Seq=3377541015 Ack=138365210 Win=64256 Len=0 TSval=1565541037 TSecr=1857304393
5	0.000585543	192.168.56.2	192.168.56.1	TCP	74	80 → 47236 [SYN, ACK] Seq=4211105003 Ack=3462831446 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=1857304393 TSecr=1857304393
6	0.000685799	192.168.56.1	192.168.56.2	TCP	66	47236 → 80 [ACK] Seq=3462831446 Ack=4211105004 Win=64256 Len=0 TSval=1565541037 TSecr=1857304393
7	0.002865881	192.168.56.1	192.168.56.2	HTTP	488	GET / HTTP/1.1
8	0.003150650	192.168.56.2	192.168.56.1	TCP	66	80 → 47234 [ACK] Seq=138365210 Ack=3377541437 Win=64768 Len=0 TSval=1857304396 TSecr=1565541039
9	0.003600753	192.168.56.2	192.168.56.1	HTTP	731	HTTP/1.1 200 OK (text/html)
10	0.003615821	192.168.56.1	192.168.56.2	TCP	66	47234 → 80 [ACK] Seq=3377541437 Ack=138365875 Win=64128 Len=0 TSval=1565541040 TSecr=1857304397
11	0.023539172	192.168.56.1	192.168.56.2	HTTP	429	GET /logoort.gif HTTP/1.1
12	0.023916255	192.168.56.2	192.168.56.1	TCP	1514	80 → 47234 [ACK] Seq=138365875 Ack=3377541800 Win=64512 Len=1448 TSval=1857304417 TSecr=1565541060 [TCP
13	0.023938228	192.168.56.1	192.168.56.2	TCP	66	47234 → 80 [ACK] Seq=3377541800 Ack=138367323 Win=64128 Len=0 TSval=1565541060 TSecr=1857304417
14	0.023968810	192.168.56.2	192.168.56.1	HTTP	1493	HTTP/1.1 200 OK (GIF89a)
15	0.023973098	192.168.56.1	192.168.56.2	TCP	66	47234 → 80 [ACK] Seq=3377541800 Ack=138368750 Win=64128 Len=0 TSval=1565541060 TSecr=1857304417
16	0.051317773	192.168.56.1	192.168.56.2	HTTP	429	GET /favicon.ico HTTP/1.1
17	0.051650404	192.168.56.2	192.168.56.1	HTTP/XML	545	HTTP/1.1 404 Not Found
18	0.051667775	192.168.56.1	192.168.56.2	TCP	66	47234 → 80 [ACK] Seq=3377542163 Ack=138369229 Win=64128 Len=0 TSval=1565541088 TSecr=1857304445
19	0.060989458	192.168.56.2	192.168.56.1	TCP	66	80 → 47234 [FIN, ACK] Seq=138369229 Ack=3377542163 Win=64256 Len=0 TSval=1857309455 TSecr=1565541088
20	5.102930219	192.168.56.1	192.168.56.2	TCP	66	47234 → 80 [ACK] Seq=3377542163 Ack=138369230 Win=64128 Len=0 TSval=1565546139 TSecr=1857309455
21	5.246963323	0a:00:27:00:00:00	PcsCompu_25:3...	ARP	42	Who has 192.168.56.2? Tell 192.168.56.1
22	5.247177167	PcsCompu_25:3...	0a:00:27:00:00:00	ARP	60	192.168.56.2 is at 08:00:27:25:3d:6b

12. Vuelva a descargar la página e indique ahora cuál es el código de respuesta HTTP. Además, justifique el porqué de esta situación e indique cómo es el procedimiento de solicitud/respuesta HTTP con los datos de la captura.

El nuevo código es 304, esto se debe a que la misma se encuentra almacenada en caché. Esto le dice al cliente que al no ser actualizada la página desde 2011, puede utilizar la que ya posee en caché.

El navegador pregunta con el campo If-Modified-Since: ... para comparar con lo que posee el navegador, por lo que el servidor responde Not Modified, en caso de que la página sea la misma.

Archivo Edición Visualización Jr Captura Analizar Estadísticas Telefonía Wireless Herramientas Ayuda									
Aplique un filtro de visualización ... <Ctrl-/>									
Archivos	Time	Source	Destination	Protocol	Length	Info			
2	0.686234717	192.168.56.1	239.255.255.2...	SSDP	214	M-SEARCH * HTTP/1.1			
3	0.686312515	192.168.56.1	192.168.56.2	TCP	66	47238 → 80 [FIN, ACK] Seq=1359776196 Ack=4202064765 Win=501 Len=0 TSval=1565637117 TSecr=1857382526			
4	0.686498321	192.168.56.2	192.168.56.1	TCP	74	47242 → 80 [SYN] Seq=2796859350 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1565637117 TSecr=0 WS=128			
5	0.686549998	192.168.56.2	192.168.56.1	TCP	66	80 → 47238 [ACK] Seq=4202064765 Ack=1359776197 Win=505 Len=0 TSval=1857400494 TSecr=1565637117			
6	0.686564047	192.168.56.1	192.168.56.2	TCP	74	80 → 47242 [SYN, ACK] Seq=2864994347 Ack=2796859351 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=1857400494			
7	0.687404663	192.168.56.1	192.168.56.2	HTTP	593	GET / HTTP/1.1			
8	0.687604828	192.168.56.2	192.168.56.1	TCP	66	80 → 47240 [ACK] Seq=4097554919 Ack=3273753924 Win=505 Len=0 TSval=1857400495 TSecr=1565637118			
9	0.687774005	192.168.56.2	192.168.56.1	HTTP	271	HTTP/1.1 304 Not Modified			
10	0.687779835	192.168.56.1	192.168.56.2	TCP	66	47240 → 80 [ACK] Seq=3273753924 Ack=4097555124 Win=501 Len=0 TSval=1565637118 TSecr=1857400495			
11	1.000979161	192.168.56.1	239.255.255.2...	SSDP	214	M-SEARCH * HTTP/1.1			
12	2.001773450	192.168.56.1	239.255.255.2...	SSDP	214	M-SEARCH * HTTP/1.1			
13	5.773369357	PcsCompu_25:...	0a:00:27:00:0...	ARP	60	Who has 192.168.56.1? Tell 192.168.56.2			
14	5.773409607	0a:00:27:00:0...	PcsCompu_25:3...	ARP	42	192.168.56.1 is at 0a:00:27:00:00:00			
15	6.702094301	192.168.56.2	192.168.56.1	TCP	66	80 → 47240 [FIN, ACK] Seq=4097555124 Ack=3273753924 Win=505 Len=0 TSval=1857406511 TSecr=1565637118			
16	6.748986097	192.168.56.1	192.168.56.2	TCP	66	47240 → 80 [ACK] Seq=3273753924 Ack=4097555125 Win=501 Len=0 TSval=1565643179 TSecr=1857406511			

5.2 Análisis de las conexiones

1. Ejecute el comando `netstat -na` desde una consola de Windows. Detalle brevemente la salida observada.

Lo que se puede observar son las conexiones del equipo. Con el comando `netstat -h` nos muestra la ayuda del mismo, vemos que la `n` es para que nos muestre las direcciones en formato numérico y la `a` es para que muestre todas las conexiones.

Active Connections

Proto	Local Address	Foreign Address	State
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING
TCP	0.0.0.0:808	0.0.0.0:0	LISTENING
TCP	0.0.0.0:3306	0.0.0.0:0	LISTENING
TCP	0.0.0.0:5040	0.0.0.0:0	LISTENING
TCP	0.0.0.0:5357	0.0.0.0:0	LISTENING
TCP	0.0.0.0:7680	0.0.0.0:0	LISTENING
TCP	0.0.0.0:9001	0.0.0.0:0	LISTENING
TCP	0.0.0.0:33060	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49664	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49665	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49666	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49667	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49668	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49670	0.0.0.0:0	LISTENING
TCP	127.0.0.1:1434	0.0.0.0:0	LISTENING
TCP	127.0.0.1:49671	127.0.0.1:49672	ESTABLISHED
TCP	127.0.0.1:49672	127.0.0.1:49671	ESTABLISHED
TCP	127.0.0.1:49673	127.0.0.1:49674	ESTABLISHED
TCP	127.0.0.1:49674	127.0.0.1:49673	ESTABLISHED
TCP	127.0.0.1:49675	127.0.0.1:49676	ESTABLISHED
TCP	127.0.0.1:49676	127.0.0.1:49675	ESTABLISHED
TCP	127.0.0.1:49684	0.0.0.0:0	LISTENING
TCP	127.0.0.1:49697	127.0.0.1:49698	ESTABLISHED
TCP	127.0.0.1:49698	127.0.0.1:49697	ESTABLISHED
TCP	127.0.0.1:49860	127.0.0.1:49861	ESTABLISHED
TCP	127.0.0.1:49861	127.0.0.1:49860	ESTABLISHED
TCP	127.0.0.1:61328	127.0.0.1:61329	ESTABLISHED
TCP	127.0.0.1:61329	127.0.0.1:61328	ESTABLISHED
TCP	192.168.114.200:139	0.0.0.0:0	LISTENING
TCP	192.168.114.200:61300	52.226.139.185:443	ESTABLISHED
TCP	192.168.114.200:61330	82.202.184.220:443	ESTABLISHED
TCP	192.168.114.200:61425	40.102.35.114:443	TIME_WAIT
TCP	192.168.114.200:61426	40.102.35.114:443	TIME_WAIT
TCP	192.168.114.200:61427	52.97.2.130:443	ESTABLISHED
TCP	192.168.114.200:61440	152.199.55.200:443	LAST_ACK
TCP	192.168.114.200:61443	152.199.55.200:443	LAST_ACK

2. ¿Qué significan los estados “ESTABLISHED” y “LISTENING” que observa?

Una conexión established significa que la conexión fue establecida, o sea, que se encuentra conectado, mientras que listening significa que está a la escucha en el puerto mostrado.

3. Establezca una conexión Telnet al servidor en otra consola y ejecute nuevamente netstat -na. Describa qué diferencia hay con la salida anterior.

Como vemos marcado en la siguiente imagen, tras establecer una conexión telnet con el equipo virtual, podremos ver la conexión a la salida en la consola. Se muestra que nuestro equipo, con dirección 192.168.79.211 y puerto 48026, se conectó con el equipo remoto en la dirección 192.168.56.2 en el puerto de telnet, el 23, y la conexión se encuentra establecida.

udp6	0	0	:::57622	:::*		64)
fedrico@feedrico-pc:~\$ netstat -natu						
Conexiones activas de Internet (servidores y establecidos)						
Proto	Recib	Enviad	Dirección local	Dirección remota	Estado	
tcp	0	0	127.0.0.1:40049	0.0.0.0:*	ESCUCHAR	* Docume
tcp	0	0	127.0.0.53:53	0.0.0.0:*	ESCUCHAR	* Manage
tcp	0	0	127.0.0.1:631	0.0.0.0:*	ESCUCHAR	* Suppor
tcp	0	0	192.168.79.221:57482	200.40.28.46:80	ESTABLECIDO	System
tcp	0	0	192.168.79.221:35396	44.240.140.78:443	ESTABLECIDO	System
tcp	0	0	192.168.79.221:49384	13.227.92.11:443	ESTABLECIDO	Usage o
tcp	0	0	192.168.79.221:52108	34.117.237.239:443	TIME_WAIT	Memory
tcp	0	0	192.168.56.1:48026	192.168.56.2:23	ESTABLECIDO	68.56.2
tcp	0	0	192.168.79.221:38308	34.98.75.36:443	ESTABLECIDO	Swap us
tcp	1	0	127.0.0.1:39780	127.0.0.1:40049	CLOSE_WAIT	3.15
tcp	0	0	192.168.79.221:36378	31.13.94.52:443	ESTABLECIDO	
tcp	0	0	192.168.79.221:49726	13.227.92.34:443	ESTABLECIDO	=> / is
tcp	0	0	192.168.79.221:36374	31.13.94.52:443	ESTABLECIDO	
tcp6	0	0	:::1:631	:::*	ESCUCHAR	* Super-

5.3 Throughput de una conexión TCP

2. Descargue el archivo archivogrande.zip del servidor mediante HTTP, usando:

http://192.168.56.2/archivogrande.zip

3. Inicie una nueva captura y comience a descargar el archivo. En la captura identifique el comienzo y el fin de conexión y el número de secuencia inicial y final. Indique:

- **La cantidad de bytes enviados.**

Número de secuencia inicial: 1024118411 Número de secuencia final: 1069304140 Bytes: 45.185.729

- **El tiempo transcurrido.**

Tiempo de transmisión: 38.518476539 s

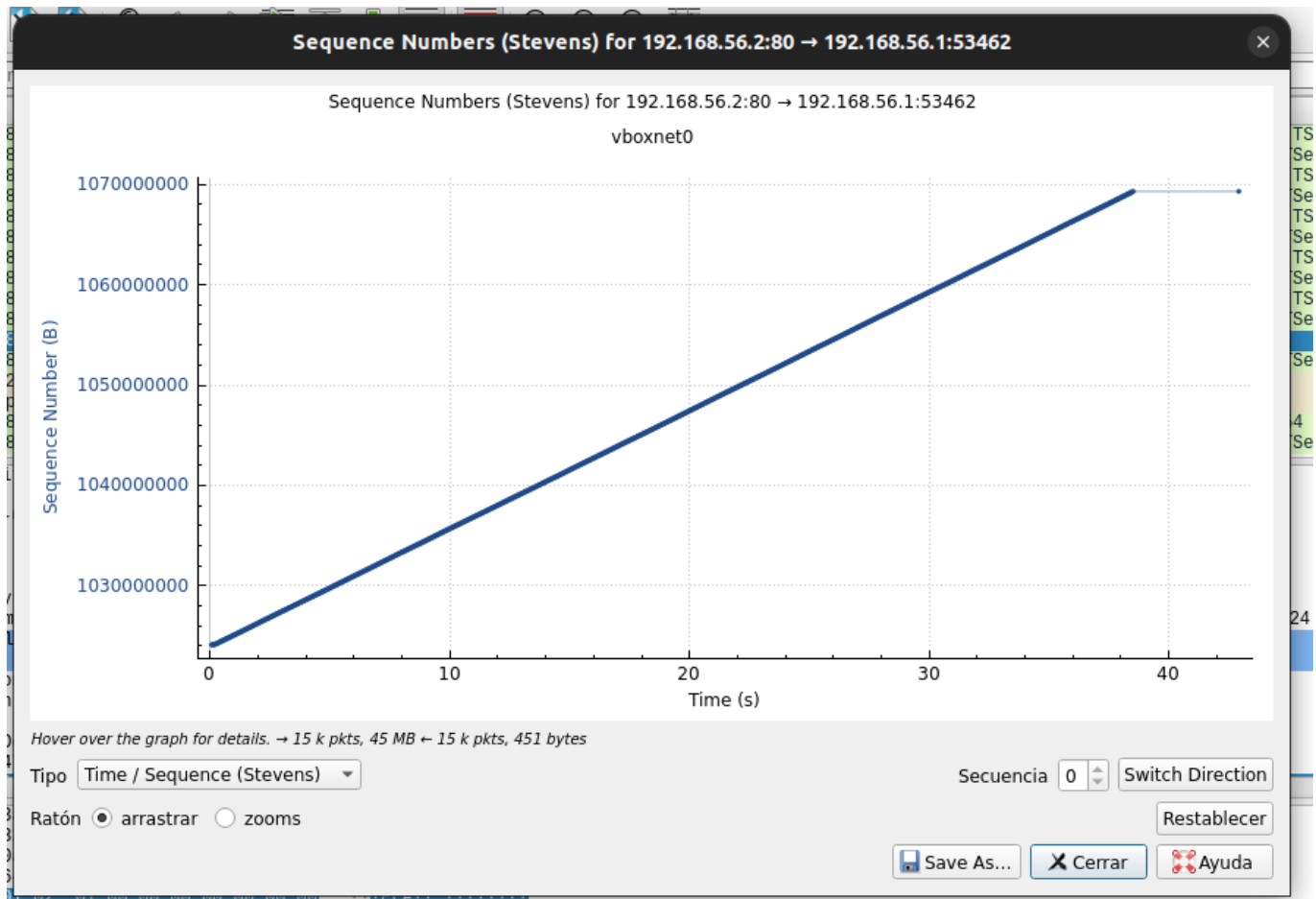
- **Con los datos anteriores, calcule el throughput en Mbps y compárelo con el configurado como límite, utilizando la aplicación tc.**

Throughput: 45.185.729 byte / 38.518476539 s Throughput: 1.173.092 byte/s -> 1,1 Mbps

Se puede ver que el resultado es menor al límite establecido.

4. Seleccione el flujo TCP relativo a la descarga. Usando la opción Statistics/TCP Stream Graph/time-sequence graph (Stevens) observe la evolución del número de secuencia en función del tiempo y verifique el cálculo anterior.

Se verifica el cálculo realizado anteriormente.



5. Identifique si TCP finaliza la conexión en forma simétrica o asimétrica. Justifique brevemente su respuesta.

En este caso se terminó la conexión de forma asimétrica, el servidor informa el corte de la conexión con la flag FIN y el cliente le da el ACK de dicha transmisión, sin mandar una finalización.

31275	38.518556868	192.168.56.1	192.168.56.2	TCP	66 53462 → 80 [ACK] Seq=915432553 Ack=1069304802 Win=
31276	42.821703465	0a:00:27:00:...	PcsCompu_25:3...	ARP	42 Who has 192.168.56.2? Tell 192.168.56.1
31277	42.872592351	PcsCompu_25:...	0a:00:27:00:0...	ARP	60 192.168.56.2 is at 08:00:27:25:3d:6b
31278	42.922895632	192.168.56.2	192.168.56.1	TCP	66 80 → 53462 [FIN, ACK] Seq=1069304802 Ack=915432555
31279	42.966016263	192.168.56.1	192.168.56.2	TCP	66 53462 → 80 [ACK] Seq=915432553 Ack=1069304803 Win=

[Checksum Status: Unverified]
Urgent Pointer: 0

En otro ejemplo se muestra la finalización simétrica, en la que además de lo ocurrido anteriormente, el cliente además le envía para finalizar la conexión.

Actividades Wireshark 17 de may 21:16

*vboxnet0

Archivo Edición Visualización Ir Captura Analizar Estadísticas Telefonía Wireless Herramientas Ayuda

Aplique un filtro de visualización ... <Ctrl-F>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.56.1	192.168.56.2	TCP	74	47220 → 80 [SYN] Seq=3418145883 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1561862148 TSecr=0 WS=128
2	0.000365999	192.168.56.2	192.168.56.1	TCP	74	80 → 47220 [SYN, ACK] Seq=1018748408 Ack=3418145884 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=1853624692 TSe...
3	0.000389271	192.168.56.1	192.168.56.2	TCP	66	47220 → 80 [ACK] Seq=3418145884 Ack=1018748409 Win=64256 Len=0 TSval=1561862148 TSecr=1853624692
4	0.000509372	192.168.56.1	192.168.56.2	HTTP	423	GET / HTTP/1.1
5	0.000595402	192.168.56.2	192.168.56.1	TCP	66	80 → 47220 [ACK] Seq=1018748409 Ack=3418146241 Win=64896 Len=0 TSval=1853624692 TSecr=1561862148
6	0.000844699	192.168.56.2	192.168.56.1	HTTP	731	HTTP/1.1 200 OK (text/html)
7	0.000854108	192.168.56.1	192.168.56.2	TCP	66	47220 → 80 [ACK] Seq=3418146241 Ack=1018749074 Win=64128 Len=0 TSval=1561862148 TSecr=1853624693
8	0.048645731	192.168.56.1	192.168.56.2	HTTP	375	GET /logoort.gif HTTP/1.1
9	0.049218967	192.168.56.2	192.168.56.1	TCP	1514	80 → 47220 [ACK] Seq=1018749074 Ack=3418146550 Win=64640 Len=1448 TSval=1853624741 TSecr=1561862196 [TCP segm...
10	0.049257373	192.168.56.1	192.168.56.2	TCP	66	47220 → 80 [ACK] Seq=3418146550 Ack=1018750522 Win=64128 Len=0 TSval=1561862197 TSecr=1853624741
11	0.049293353	192.168.56.2	192.168.56.1	HTTP	1493	HTTP/1.1 200 OK (GIF89a)
12	0.049303844	192.168.56.1	192.168.56.2	TCP	66	47220 → 80 [ACK] Seq=3418146550 Ack=1018751949 Win=63488 Len=0 TSval=1561862197 TSecr=1853624741
13	0.052694313	192.168.56.1	192.168.56.2	HTTP	375	GET /favicon.ico HTTP/1.1
14	0.053022483	192.168.56.2	192.168.56.1	HTTP/XML	545	HTTP/1.1 404 Not Found
15	0.053042262	192.168.56.1	192.168.56.2	TCP	66	47220 → 80 [ACK] Seq=3418146859 Ack=1018752428 Win=64128 Len=0 TSval=1561862201 TSecr=1853624745
16	3.589976389	PcsCompu_r6:...	Broadcast	ARP	42	Who has 192.168.56.101? Tell 192.168.56.100
17	3.589991676	PcsCompu_r6:...	Broadcast	ARP	42	Who has 192.168.56.101? Tell 192.168.56.100
18	5.159646178	0a:00:27:00:...	PcsCompu_25:3...	ARP	42	Who has 192.168.56.2? Tell 192.168.56.1
19	5.160327960	PcsCompu_25:...	0a:00:27:00:0...	ARP	60	192.168.56.2 is at 08:00:27:25:3d:6b
20	6.062944850	192.168.56.2	192.168.56.1	TCP	66	80 → 47220 [FIN, ACK] Seq=1018752428 Ack=3418146859 Win=64384 Len=0 TSval=1853630756 TSecr=1561862201
21	6.063260885	192.168.56.1	192.168.56.2	TCP	66	47220 → 80 [FIN, ACK] Seq=3418146859 Ack=1018752429 Win=64128 Len=0 TSval=1561862211 TSecr=1853630756
22	6.063978234	192.168.56.2	192.168.56.1	TCP	66	80 → 47220 [ACK] Seq=1018752429 Ack=3418146860 Win=64384 Len=0 TSval=1853630757 TSecr=1561862211

6. Aplicando ahora ./enlace2.sh: Repita ahora las pruebas de los puntos 2 y 3. ¿Qué cambios observa? ¿Por qué ocurren los mismos?

En este caso se puede ver que se pierden paquetes durante la transmisión:

No.	Time	Source	Destination	Protocol	Length	Info
31332	70.392067921	192.168.56.1	192.168.56.2	TCP	78	[TCP Dup ACK 31318#6] 53466 → 80 [ACK] Seq=3649862992
31333	70.394956619	192.168.56.2	192.168.56.1	HTTP	2962	Continuation
31334	70.396167878	192.168.56.1	192.168.56.2	TCP	78	[TCP Dup ACK 31318#7] 53466 → 80 [ACK] Seq=3649862992
31335	70.405384609	192.168.56.2	192.168.56.1	HTTP	2962	Continuation
31336	70.406223307	192.168.56.2	192.168.56.1	HTTP	2962	Continuation
31337	70.406552514	192.168.56.1	192.168.56.2	TCP	78	[TCP Dup ACK 31318#8] 53466 → 80 [ACK] Seq=3649862992
31338	70.408673713	192.168.56.2	192.168.56.1	HTTP	2962	Continuation
31339	70.409815477	192.168.56.1	192.168.56.2	TCP	78	[TCP Dup ACK 31318#9] 53466 → 80 [ACK] Seq=3649862992
31340	70.416909131	192.168.56.2	192.168.56.1	HTTP	2962	Continuation
31341	70.418076476	192.168.56.1	192.168.56.2	TCP	78	[TCP Dup ACK 31318#10] 53466 → 80 [ACK] Seq=3649862992
31342	70.419862489	192.168.56.2	192.168.56.1	HTTP	2962	[TCP Fast Retransmission] Continuation
31343	70.419917176	192.168.56.1	192.168.56.2	TCP	66	53466 → 80 [ACK] Seq=3649862992 Ack=18113821 Win=31219
31344	70.425226129	192.168.56.2	192.168.56.1	HTTP	1514	Continuation
31345	70.430504579	192.168.56.2	192.168.56.1	HTTP	2962	Continuation

- **La cantidad de bytes enviados.**

Número de secuencia inicial: 1619180962 Número de secuencia final: 1664367353 Bytes: 45.186.391

- **El tiempo transcurrido.**

Tiempo de transmisión: 71,34 s

- **Con los datos anteriores, calcule el throughput en Mbps y compárelo con el configurado como límite, utilizando la aplicación tc.**

Throughput: 45.186.391 byte / 71,34 s Throughput: 633.395 byte/s -> 0,6 Mbps

Se puede ver que el resultado es menor al anterior, debido a los paquetes perdidos, aproximadamente la mitad.

