

Universidad ORT Uruguay

Facultad de Ingeniería



Redes

Obligatorio 2024

Federica Bonomi - 278347
Bruno Odella - 231665
Martín Salaberry - 294238

Índice

Índice.....	2
Primera parte.....	3
Aplicaciones.....	3
Parte 1 - Telnet.....	3
Parte 2 - SMTP: Simple Mail Transport Protocol - RFC 821.....	9
Parte 3 - POP3: Post Office Protocol version 3 - RFC 1939.....	13
Parte 4 - HTTP: Hypertext Transfer Protocol - RFC 1945.....	16
Parte 5 - FTP: File Transfer Protocol - RFC 959.....	18
Parte 6 - SSH: Security Shell.....	21
DNS.....	24
TCP/HTTP.....	35
Parte 1 - Análisis de mensajes y secuencia TCP.....	35
Parte 2 - Análisis de las conexiones.....	43
Parte 3 - Throughput de una conexión TCP.....	45
Segunda parte.....	50
6. Asignación de direccionamiento, configuración del router e interfaces.....	50
6.1 - Topología.....	50
6.2 - Asignación de direcciones IP.....	51
6.3 - Configuración de interfaces Ethernet.....	53
Parte 4 - Prueba de conectividad.....	56
7. Ruteo estático.....	58
8. Ruteo dinámico.....	62
8.1 - Protocolo RIP.....	62
8.2 - Protocolo OSPF.....	68
9. Protocolo ARP.....	74

Primera parte

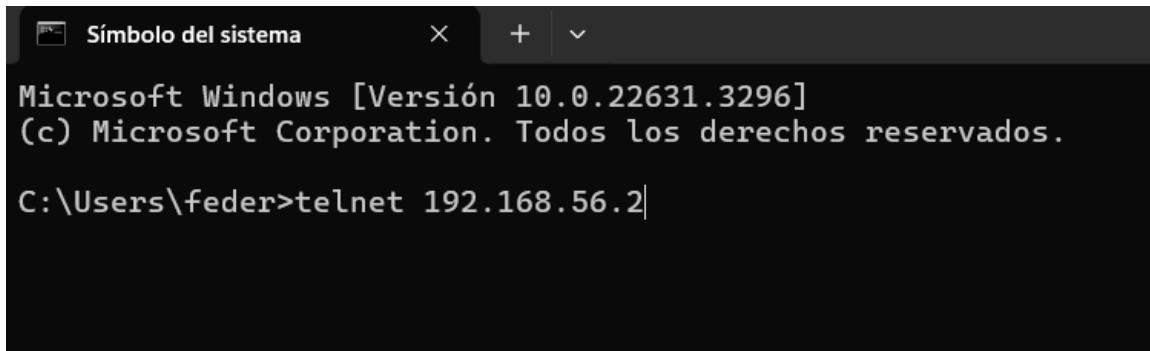
Aplicaciones

Parte 1 - Telnet

Conéctese mediante una consola (cmd) de su sistema operativo al servidor, usando el protocolo Telnet, con el usuario y contraseña “ort-grupo1”.

1. Escriba el comando utilizado y la salida obtenida.

El comando utilizado es telnet 192.168.56.2, siendo esa la ip del servidor al que nos queremos conectar. El número de puerto no fue necesario especificarlo ya que por defecto va al puerto 23, que es el que permite acceder al escritorio remoto mediante telnet.



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.22631.3296]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\feder>telnet 192.168.56.2
```

Salida

obtenida:

```
Telnet 192.168.56.2      x + 
servidor_redes login: ort-grupo1
Password:
Last login: Tue Apr  2 01:03:30 UTC 2024 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 Apr  2 01:10:46 UTC 2024

System load:  0.0          Processes:      96
Usage of /:   78.3% of 3.11GB  Users logged in:  0
Memory usage: 8%           IP address for enp0s3: 192.168.56.2
Swap usage:   0%           IP address for enp0s8: 10.0.3.15

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

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

New release '20.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

No mail.
ort-grupo1@servidor_redes:~$ |
```

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

Las tareas que se pueden realizar dependen de los permisos asignados al usuario con el que nos conectamos. Si hay limitaciones dependen de esto y no del protocolo telnet.

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

Tiene que estar habilitado telnet en ambos equipos. En Windows, por ejemplo, viene desactivado por defecto, ya que es un protocolo inseguro, por lo tanto para conectarnos hubo que habilitarlo manualmente. También es necesario que el equipo servidor esté escuchando y tenga abierto el puerto 23, y también se debe conocer la IP de este servidor para poder conectarnos.

4. ¿Hasta qué capa y qué debe realizar cada una de ellas para que la comunicación entre los nodos por Telnet sea exitoso?

Hasta la capa más baja, o sea hasta la física, por más que sea una máquina virtual, siempre hay una parte física.

Todos los peers se comunican entre sí, pero para llegar al peer del otro equipo, primero se debe bajar hasta la capa física.

La capa de aplicación inicia la conexión entre el cliente y el servidor y se comunica mediante protocolo telnet con la de su peer.

La capa de transporte es la que establece la conexión y lo hace mediante TCP.

La capa de red a través de la red los paquetes de datos generados, usando direcciones IP.

La capa de enlace transmite los datos en tramas y utiliza ethernet al igual que lo hace la capa física.

5. Si analiza el tráfico capturado con Wireshark:

¿Cuál es el número de puerto de origen y de destino con los que se está accediendo? Justifique el por qué se seleccionan los mismos.

El de origen es el puerto 52609. Este puerto se elige de los puertos libres que son los que están entre el 49151 y el tope, que es 65535.

El de destino es el 23 por defecto, que en el estándar de telnet es el que está reservado para la conexión remota.

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

Para ver solamente las comunicaciones tcp y telnet usamos el filtro `tcp.stream eq 0`

tcp.stream eq 0						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.56.1	192.168.56.2	TCP	66	52609 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
2	0.000734	192.168.56.2	192.168.56.1	TCP	66	23 → 52609 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
3	0.000874	192.168.56.1	192.168.56.2	TCP	54	52609 → 23 [ACK] Seq=1 Ack=1 Win=262656 Len=0

La captura de wireshark corresponde al establecimiento de conexión, el cual consta de 3 mensajes.

El primer mensaje, con el texto [SYN] (de synchronization) es el que enviamos nosotros como clientes para solicitar la conexión.

El segundo mensaje, con el texto [SYN, ACK] (de synchronization acknowledged) es la respuesta del servidor, en la cual nos acepta la conexión.

Por último, el tercer mensaje ([ACK]) es como una “confirmación” de parte del cliente de que recibió la conexión.

En estos 3 mensajes se pueden ver los puertos de origen y de destino que se mencionaron en la pregunta anterior.

3 0.000874	192.168.56.1	192.168.56.2	TCP	54 52609 → 23 [ACK] Seq=1 Ack=1 Win=262656 Len=0
4 0.008373	192.168.56.2	192.168.56.1	TELNET	66 Telnet Data ...
5 0.008707	192.168.56.1	192.168.56.2	TELNET	60 Telnet Data ...
6 0.009120	192.168.56.2	192.168.56.1	TCP	60 23 → 52609 [ACK] Seq=13 Ack=7 Win=64256 Len=0
7 0.009142	192.168.56.1	192.168.56.2	TELNET	63 Telnet Data ...
8 0.009622	192.168.56.2	192.168.56.1	TCP	60 23 → 52609 [ACK] Seq=13 Ack=16 Win=64256 Len=0
9 0.010006	192.168.56.2	192.168.56.1	TELNET	69 Telnet Data ...
10 0.010142	192.168.56.1	192.168.56.2	TELNET	63 Telnet Data ...
11 0.053282	192.168.56.2	192.168.56.1	TCP	60 23 → 52609 [ACK] Seq=28 Ack=25 Win=64256 Len=0
12 0.053356	192.168.56.1	192.168.56.2	TELNET	70 Telnet Data ...
13 0.053697	192.168.56.2	192.168.56.1	TCP	60 23 → 52609 [ACK] Seq=28 Ack=41 Win=64256 Len=0
14 0.054726	192.168.56.2	192.168.56.1	TELNET	66 Telnet Data ...
15 0.054992	192.168.56.1	192.168.56.2	TELNET	57 Telnet Data ...
16 0.096996	192.168.56.2	192.168.56.1	TCP	60 23 → 52609 [ACK] Seq=40 Ack=44 Win=64256 Len=0
17 0.097051	192.168.56.1	192.168.56.2	TELNET	63 Telnet Data ...
18 0.097412	192.168.56.2	192.168.56.1	TCP	60 23 → 52609 [ACK] Seq=40 Ack=53 Win=64256 Len=0
19 0.098555	192.168.56.2	192.168.56.1	TELNET	60 Telnet Data ...
20 0.098812	192.168.56.1	192.168.56.2	TELNET	57 Telnet Data ...
21 0.099604	192.168.56.2	192.168.56.1	TELNET	74 Telnet Data ...
22 0.099645	192.168.56.1	192.168.56.2	TELNET	57 Telnet Data ...
23 0.100224	192.168.56.2	192.168.56.1	TELNET	76 Telnet Data ...
24 0.139444	192.168.56.1	192.168.56.2	TCP	54 52609 → 23 [ACK] Seq=59 Ack=88 Win=262656 Len=0
29 4.612360	192.168.56.1	192.168.56.2	TELNET	55 Telnet Data ...
30 4.613403	192.168.56.2	192.168.56.1	TELNET	60 Telnet Data ...
31 4.654081	192.168.56.1	192.168.56.2	TCP	54 52609 → 23 [ACK] Seq=60 Ack=89 Win=262656 Len=0
36 25.014625	192.168.56.1	192.168.56.2	TELNET	55 Telnet Data ...
37 25.015838	192.168.56.2	192.168.56.1	TELNET	60 Telnet Data ...
38 25.055776	192.168.56.1	192.168.56.2	TCP	54 52609 → 23 [ACK] Seq=61 Ack=90 Win=262656 Len=0
39 27.337591	192.168.56.1	192.168.56.2	TELNET	55 Telnet Data ...
40 27.338529	192.168.56.2	192.168.56.1	TELNET	60 Telnet Data ...
41 27.379351	192.168.56.1	192.168.56.2	TCP	54 52609 → 23 [ACK] Seq=62 Ack=91 Win=262656 Len=0
42 28.306505	192.168.56.1	192.168.56.2	TELNET	55 Telnet Data ...
43 28.307560	192.168.56.2	192.168.56.1	TELNET	60 Telnet Data ...
44 28.348715	192.168.56.1	192.168.56.2	TCP	54 52609 → 23 [ACK] Seq=63 Ack=92 Win=262656 Len=0
45 28.618287	192.168.56.1	192.168.56.2	TELNET	55 Telnet Data ...
46 28.619294	192.168.56.2	192.168.56.1	TELNET	60 Telnet Data ...
47 28.658701	192.168.56.1	192.168.56.2	TCP	54 52609 → 23 [ACK] Seq=64 Ack=93 Win=262656 Len=0
48 28.783033	192.168.56.1	192.168.56.2	TELNET	55 Telnet Data ...

Captura correspondiente al intercambio de datos.

Cuando se envia algo por telnet, siempre hay un intercambio TCP por detrás, por eso es que hay muchos TCPs intercalados con TELNETs.

273 685.362935	192.168.56.2	192.168.56.1	TCP	60 23 → 52609 [FIN, ACK] Seq=1143 Ack=89 Win=64256 Len=0
274 685.363035	192.168.56.1	192.168.56.2	TCP	54 52609 → 23 [ACK] Seq=89 Ack=1144 Win=2097920 Len=0
275 685.363448	192.168.56.1	192.168.56.2	TCP	54 52609 → 23 [FIN, ACK] Seq=89 Ack=1144 Win=2097920 Len=0
276 685.363868	192.168.56.2	192.168.56.1	TCP	60 23 → 52609 [ACK] Seq=1144 Ack=90 Win=64256 Len=0

Captura correspondiente al fin de conexión, el cual lo inicia el servidor porque estoy remotamente en el servidor cuando lo solicito.

6. Investigue y pruebe como Telnet envía el texto Ok por consola hacia el nodo remoto. ¿Qué sucede en caso de errores de teclado, puede solucionarlo en el origen?

```
.[0m.[01;34mmail.[0m
ort-grupo1@servidor_redes:~$ ookk
```

Cada carácter que se envía mediante el protocolo Telnet es enviado inmediatamente al servidor, por lo tanto si ocurre un error de teclado no se puede solucionar, ya que el carácter escrito ya fue enviado al servidor. En la captura podemos ver que se muestran duplicadas las letras ya que se muestra la que fue enviada al servidor y la que devuelve.

También, si hacemos captura de los datos al iniciar sesión podemos ver que nos muestra (duplicadas) las credenciales con las que nos logueamos, esto significa que telnet es un protocolo inseguro ya que los mensajes no se envían con algún tipo de encriptación.

```

Wireshark - Seguir secuencia TCP (tcp.stream eq 0) · Ethernet

.... .#.'..... .#.'.....N.....'.....ANSI.....!.....!.....Ubuntu 18.04.4 LTS
...servidor_redes login: orrtt--grruuppo11

Password: ort-grupo1

Last login: Tue Apr  2 01:43:37 UTC 2024 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 Apr  2 01:58:18 UTC 2024

 System load: 0.0          Processes:      97
 Usage of /: 78.2% of 3.11GB  Users logged in:  0
 Memory usage: 8%          IP address for enp0s3: 192.168.56.2
 Swap usage:  0%          IP address for enp0s8: 10.0.3.15

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
 just raised the bar for easy, resilient and secure K8s cluster deployment.

 https://ubuntu.com/engage/secure-kubernetes-at-the-edge

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

New release '20.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

No mail.
ort-grupo1@servidor_redes:~$
```

30 client pkt(s), 23 server pkt(s), 34 turn(s).

Conversación completa (1210 bytes) Mostrar datos como ASCII Secuencia 0

Buscar: Buscar siguiente Filtrar secuencia Imprimir Guardar como... Atrás Cerrar Ayuda

7. Sin cerrar la conexión Telnet anterior, genere una nueva y demuestre como el servidor puede identificarlas.

```

ort-grupo1@servidor_redes:~$ netstat -an | grep ':23'
tcp        0      0 0.0.0.0:23              0.0.0.0:*                  LISTEN
tcp        0      0 192.168.56.2:23         192.168.56.1:58402      ESTABLISHED
tcp        0      0 192.168.56.2:23         192.168.56.1:58412      ESTABLISHED
ort-grupo1@servidor_redes:~$
```

Con este comando vemos las conexiones telnet activas y podemos saber en qué puerto está establecida cada una (58402 y 58412).

Explicación del comando utilizado:

- netstat muestra información de las conexiones de red, y al usarlos con la opción '-an' muestra las conexiones activas y puertos de escucha.
- Usando | grep ':23' filtramos las salidas del comando netstat para que solo nos muestre las que contienen el texto ':23' que es la notación para reconocer el puerto 23, el que usa telnet por defecto.

No.	Time	Source	Destination	Protocol	Length	Info
2	13.216854	192.168.56.1	192.168.56.2	TCP	66	58402 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
3	13.217663	192.168.56.2	192.168.56.1	TCP	66	23 → 58402 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128

No.	Time	Source	Destination	Protocol	Length	Info
104	43.804053	192.168.56.1	192.168.56.2	TCP	66	58412 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
105	43.804713	192.168.56.2	192.168.56.1	TCP	66	23 → 58412 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128

Capturas de wireshark con el establecimiento de cada conexión en las que podemos ver que los puertos coinciden con los que obtuvimos con el comando netstat y que Wireshark les asigna un *stream index* para identificarlas.

▼	Transmission Control Protocol, Src Port: 58402, Dst Port: 23, Seq: 0, Len: 0
	Source Port: 58402
	Destination Port: 23
	[Stream index: 0]
▼	Transmission Control Protocol, Src Port: 58412, Dst Port: 23, Seq: 0, Len: 0
	Source Port: 58412
	Destination Port: 23
	[Stream index: 1]

Parte 2 - SMTP: Simple Mail Transport Protocol - RFC 821

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

El comando utilizado es telnet 192.168.56.2 25

```
telnet 192.168.56.2 25|
```

Y la respuesta del servidor es la siguiente:

```
220 servidor_redes.lan ESMTP Postfix (Ubuntu)
```

donde 220 es el código de respuesta del servidor SMTP para indicar que la conexión del cliente fue exitosa.

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.*

Los comandos utilizados son

- HELO
- MAIL FROM
- RCPT TO
- DATA
- QUIT

Dentro de DATA es importante enviar los campos *from*, *to* y *subject*.

En la captura a continuación se puede ver el diálogo que realizamos al usuario ort-grupo2, y el que realizamos a ort-grupo1 es análogo, pero cambia el *RCPT TO* y el *To*.

```
[root@servidor_redes ~]# Símbolo del sistema
220 servidor_redes.lan ESMTP Postfix (Ubuntu)
HELO servidor_redes
250 servidor_redes.lan
MAIL FROM: ort-grupo1@servidor_redes
250 2.1.0 Ok
RCPT TO:
501 5.5.4 Syntax: RCPT TO:<address>
RCPT TO: ort-grupo2@servidor_redes
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
From: ort-grupo1@servidor_redes
To: ort-grupo2@servidor_redes
Subject: Test mail

Hello, ort-grupo2 im ort-grupo1
.
250 2.0.0 Ok: queued as 351353C0C
QUIT
221 2.0.0 Bye
```

Se ha perdido la conexión con el host.

Y en esta captura podemos ver que llegó el correo al usuario ort-grupo1.

```
servidor_redes login: ort-grupo1
Password:
Last login: Tue Apr 16 01:01:53 UTC 2024 from 192.168.56.1 on pts/1
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 16 01:37:37 UTC 2024

System load:  0.0          Processes:           104
Usage of /:   78.3% of 3.11GB   Users logged in:      1
Memory usage: 9%
Swap usage:   0%          IP address for enp0s3: 192.168.56.2
                           IP address for enp0s8: 10.0.3.15

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

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

New release '20.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

You have new mail.
ort-grupo1@servidor_redes:~$
```

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

El sobre de un mensaje SMTP incluye la información utilizada por los servidores de correo para enrutar y entregar el mensaje a sus destinatarios. No es parte del contenido del mensaje que ve el usuario final. Se compone principalmente de los comandos SMTP que especifican el remitente y los destinatarios del mensaje:

MAIL FROM: especifica la dirección de correo del remitente.

RCPT TO: especifica la(s) dirección(es) de correo de los destinatarios.

Los **encabezados** del mensaje son parte del contenido del mensaje que el usuario final puede ver, y contienen metadatos sobre el mensaje. Estos incluyen, pero no están limitados:

From: quién envía el mensaje.

To: quién recibe el mensaje.

Subject: el asunto del mensaje.

Date: la fecha de envío del mensaje.

El **cuerpo** del mensaje es el contenido textual del mensaje que el usuario escribe y que el destinatario lee. Este puede incluir texto plano, HTML y, en correos más complejos, formatos como imágenes y otros tipos de medios, aunque estos usualmente se adjuntan y no forman parte directa del cuerpo textual para los usuarios y forman parte del contenido del mensaje que se entrega y almacena.

4. *Si tuviera que realizar la parte dos (2), pero en un sólo envío, indique cómo modificaría el sobre para tal fin.*

Para realizar esto ponemos dos destinatarios en el apartado de *RCPT TO* y en el encabezado *To*, como se puede ver en la captura a continuación.

```
[root] Símbolo del sistema
RCPT TO: ort-grupo1_
550 5.1.1 <ort-grupo1_>: Recipient address rejected: User unknown in local recipient table
502 5.5.2 Error: command not recognized

500 5.5.2 Error: bad syntax
HELO servidor_redes
250 servidor_redes.lan
MAIL FROM: ort-grupo1@servidor_redes
250 2.1.0 Ok
RCPT TO: ort-grupo1@servidor_redes
250 2.1.5 Ok
RCPT TO: ort-grupo2@servidor_redes
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
From: ort-grupo1@servidor_redes
To: ort-grupo1@servidor_redes, ort-grupo2@servidor_redes
Subject: Test email ytwo people

Hello!
.
250 2.0.0 Ok: queued as F16613C0C
QUIT
221 2.0.0 Bye

Se ha perdido la conexión con el host.
```

Parte 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
```

Nos conectamos por telnet al servidor pero en el puerto para el servicio de lectura de emails, o sea POP3. El puerto asignado a este servicio es el 110.

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

Para loguearnos utilizamos los comandos

- user ort-grupo1
- pass ort-grupo1

Luego, para saber si tenemos mail usamos

- list

Y para ver ese mail utilizamos

- retr 1

Esto se puede ver en la captura a continuación.

```
[root@ORT ~]# Telnet 192.168.56.2
+OK Dovecot (Ubuntu) ready.
ort-grupo1
-ERR Unknown command.
user ort-grupo1
+OK
list
-ERR Unknown command.
stat
-ERR Unknown command.
pass ort-grupo1
+OK Logged in.
list
+OK 1 messages:
1 433
.
retr 1
+OK 433 octets
Return-Path: <ort-grupo1@servidor_redes>
X-Original-To: ort-grupo1@servidor_redes
Delivered-To: ort-grupo1@servidor_redes
Received: from servidor_redes (unknown [192.168.56.1])
        by servidor_redes.lan (Postfix) with SMTP id F16613C0C;
        Sun, 28 Apr 2024 21:30:53 +0000 (UTC)
From: ort-grupo1@servidor_redes
To: ort-grupo1@servidor_redes, ort-grupo2@servidor_redes
Subject: Test email ytwo people

Hello!
.
```

3. Verifique la correcta recepción del mensaje que envió a ort-grupo2 en la parte de SMTP. Borre el mismo indicando con qué comando lo hace.

Para loguearnos con el usuario ort-grupo2 usamos los mismos comandos que en el punto 2. Luego el comando *list* nos devuelve dos correos y utilizamos *retr 2* para quedarnos con el segundo correo. Luego para borrarlo usamos *dele 2* y nos desconectamos usando *quit*, como se ve en la captura.

```
Telnet 192.168.56.2
+OK Dovecot (Ubuntu) ready.
user ort-grupo2
+OK
pass ort-grupo2
+OK Logged in.
list
+OK 2 messages:
1 478
2 433
.
retr 2
+OK 433 octets
Return-Path: <ort-grupo1@servidor_redes>
X-Original-To: ort-grupo2@servidor_redes
Delivered-To: ort-grupo2@servidor_redes
Received: from servidor_redes (unknown [192.168.56.1])
    by servidor_redes.lan (Postfix) with SMTP id F16613C0C;
    Sun, 28 Apr 2024 21:30:53 +0000 (UTC)
From: ort-grupo1@servidor_redes
To: ort-grupo1@servidor_redes, ort-grupo2@servidor_redes
Subject: Test email ytwo people

Hello!
.
DELE 2
+OK Marked to be deleted.
QUIT
```

Nos desconectamos por que el mail no se borra hasta que se cierra la sesión, por eso dice marcado para eliminarse.

```
Telnet 192.168.56.2
+OK Dovecot (Ubuntu) ready.
user ort-grupo2
+OK
pass ort-grupo2
+OK Logged in.
list
+OK 1 messages:
1 478
.
```

Nos volvemos a conectar y vemos que ahora sí se borró.

Parte 4 - HTTP: Hypertext Transfer Protocol - RFC 1945

- 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
```

- 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.*

Ejecutamos los comandos

GET / HTTP/1.1

Host: 192.168.56.2

Luego doble enter y recuperaremos la pagina

```
HTTP/1.1 200 OK
Vary: Accept-Encoding
Content-Type: text/html
Accept-Ranges: bytes
ETag: "789668775"
Last-Modified: Thu, 08 Sep 2011 18:35:16 GMT
Content-Length: 650
Date: Mon, 29 Apr 2024 23:45:07 GMT
Server: lighttpd/1.4.45

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
        <title>Bienvenidos al Laboratorio de Redes</title>
    </head>
    <body>
        <h1>Bienvenidos al Laboratorio de Redes</h1>
        <h2>Este es el la pagina por defecto del servidor http</h2>
        <hr />
        
        <hr />
        <p>En la pagina hay texto plano, y tambien alguna imagen, para poder ver como se comporta un explorador al bajar la pagina por ht
        p y comparar con el modo texto.</p>
        <hr />
    </body>
</html>

Se ha perdido la conexión con el host.
```

- Indique el lenguaje en el cual está escrita la página.*

HTML(HyperText Markup Language)

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

HEAD / HTTP/1.1

Host: 192.168.56.2

Doble enter.

A continuación podemos ver la captura y notar que trae muchos menos datos que la anterior, ya que no trae todo el contenido de la página.

```
HTTP/1.1 200 OK
Content-Type: text/html
Accept-Ranges: bytes
ETag: "789668775"
Last-Modified: Thu, 08 Sep 2011 18:35:16 GMT
Content-Length: 650
Date: Mon, 29 Apr 2024 23:37:57 GMT
Server: lighttpd/1.4.45
```

Se ha perdido la conexión con el host.

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

Accediendo mediante el navegador web vemos la página con los estilos aplicados, en lugar de ver solamente texto plano.



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.

En la captura a continuación vemos el código fuente de la página, que coincide con el antes visto.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Bienvenidos al Laboratorio de Redes</title>
</head>

<body>
<h1>Bienvenidos al Laboratorio de Redes</h1>
<h2>Este es el la pagina por defecto del servidor http</h2>
<hr />

<hr />
<p>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.</p>
<hr />
</body>
</html>
```

Parte 5 - FTP: File Transfer Protocol - RFC 959

1. ¿Cuál es el objetivo del protocolo FTP?

Enviar y recibir archivos de mayor tamaño, y también administrar archivos remotos, entre sistemas conectados por una red.

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?

```
C:\Users\59899>ftp 192.168.56.2
Connected to 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.
User (192.168.56.2:(none)): ort-grupo1
331 Password required for ort-grupo1.
Password:
230 User ort-grupo1 logged in.
ftp> |
```

Con el comando ftp 192.168.56.2 nos conectamos al servidor con cliente ftp y las siguientes líneas constituyen el login.

3. ¿FTP encripta la información? Investigue y justifique

Si capturamos la tarjeta de red de VirtualBox y vemos qué sucede entre el servidor y nosotros (cliente) veremos que el mensaje no está cifrado, ya que aparece mismo en la columna de "info" nuestra contraseña enviada y nuestro usuario

3 0.000834	192.168.56.1	192.168.56.2	TCP	54 54613 → 21 [ACK] Seq=1 Ack=1 Win=8192 Len=0
4 0.011738	192.168.56.2	192.168.56.1	FTP	130 Response: 220 servidor_redes FTP server (Version 6.4/OpenBSD/Linux-ftpd-0.17) ready.
5 0.026630	192.168.56.1	192.168.56.2	FTP	68 Request: OPTS UTF8 ON
6 0.027150	192.168.56.2	192.168.56.1	TCP	60 21 → 54613 [ACK] Seq=77 Ack=15 Win=64256 Len=0
7 0.027960	192.168.56.2	192.168.56.1	FTP	99 Response: 500 'OPTS UTF8 ON': command not understood.
8 0.068047	192.168.56.1	192.168.56.2	TCP	54 54613 → 21 [ACK] Seq=15 Ack=122 Win=8071 Len=0
9 5.237792	192.168.56.1	192.168.56.2	FTP	71 Request: USER ort-grupo1
10 5.241824	192.168.56.2	192.168.56.1	FTP	93 Response: 331 Password required for ort-grupo1.
11 5.281759	192.168.56.1	192.168.56.2	TCP	54 54613 → 21 [ACK] Seq=32 Ack=161 Win=8032 Len=0
12 9.189183	192.168.56.1	192.168.56.2	FTP	71 Request: PASS ort-grupo1
13 9.152419	192.168.56.2	192.168.56.1	TCP	60 21 → 54613 [ACK] Seq=161 Ack=49 Win=64256 Len=0
14 9.205813	192.168.56.2	192.168.56.1	FTP	86 Response: 230 User ort-grupo1 logged in.
15 9.245695	192.168.56.1	192.168.56.2	TCP	54 54613 → 21 [ACK] Seq=49 Ack=193 Win=8000 Len=0

Otra prueba de esto.

42 37.970721	192.168.56.1	192.168.56.2	FTP	71 Request: USER ort-grupo1
43 37.972759	192.168.56.2	192.168.56.1	FTP	93 Response: 331 Password required for ort-grupo1.
44 38.013393	192.168.56.1	192.168.56.2	TCP	54 54602 → 21 [ACK] Seq=32 Ack=161 Win=8032 Len=0
45 42.723489	fe80::a00:27ff:fe06... ff02::2		ICMPv6	70 Router Solicitation from 08:00:27:06:b8:ff
46 50.178228	192.168.56.1	192.168.56.2	FTP	93 Request: PASS esta contrasenia no esta cifrada

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

```
230 User ort-grupo1 logged in.  
ftp> pwd  
257 "/home/ort-grupo1" is current directory.  
ftp> cd ..  
250 CWD command successful.  
ftp> cd publico  
250 CWD command successful.  
ftp> ls  
200 PORT command successful.  
150 Opening ASCII mode data connection for 'file list'.  
texto.txt  
putty.exe  
226 Transfer complete.  
ftp: 25 bytes recibidos en 0.00segundos 25000.00a KB/s.  
ftp>
```

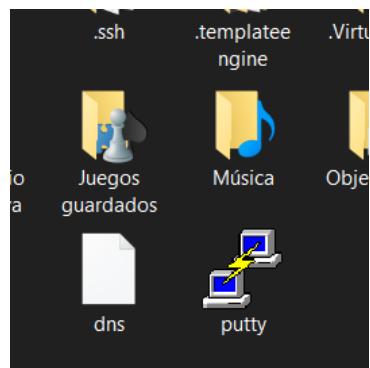
Se observa los archivos *texto.txt* y *putty.exe*

5. 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.

Primero, ponemos en modo binario la transferencia de archivos para asegurar que no hay corrupciones en la misma. Luego, seguimos con los comandos de la imagen.

```
ftp> binary  
200 Type set to I.  
ftp> get putty.exe  
200 PORT command successful.  
150 Opening BINARY mode data connection for 'putty.exe' (483328 bytes).  
226 Transfer complete.  
ftp: 483328 bytes recibidos en 0.03segundos 19333.12a KB/s.  
ftp>
```

Se descargó en el directorio que estábamos parados en nuestra pc al momento de iniciar la conexión.

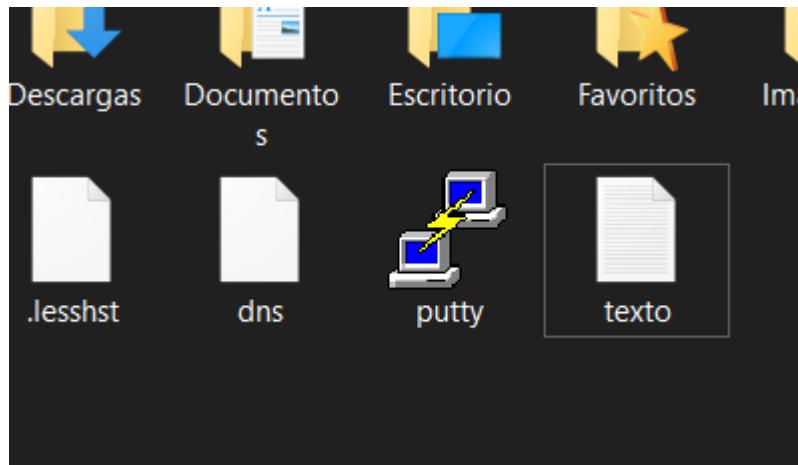


6. ¿Qué secuencia de comandos utilizaría si deseara copiar a su equipo todo el contenido del directorio actual indicando que no se desea recibir confirmación para cada archivo a transferir?

```
ftp> binary  
200 Type set to I.  
ftp> prompt  
Modo interactivo Desactivado .  
ftp> mget *  
200 Type set to I.  
200 PORT command successful.  
150 Opening BINARY mode data connection for 'putty.exe' (483328 bytes).  
226 Transfer complete.  
ftp: 483328 bytes recibidos en 0.87segundos 553.64a KB/s.  
200 PORT command successful.  
150 Opening BINARY mode data connection for 'texto.txt' (447 bytes).  
226 Transfer complete.  
ftp: 447 bytes recibidos en 0.00segundos 447000.00a KB/s.  
ftp>
```

Podemos usar el comando “prompt” que deshabilita el modo interactivo, lo que hace que no se necesite la confirmación para cada archivo. A continuación, ejecutamos “mget *” para descargar todos los archivos del directorio actual.

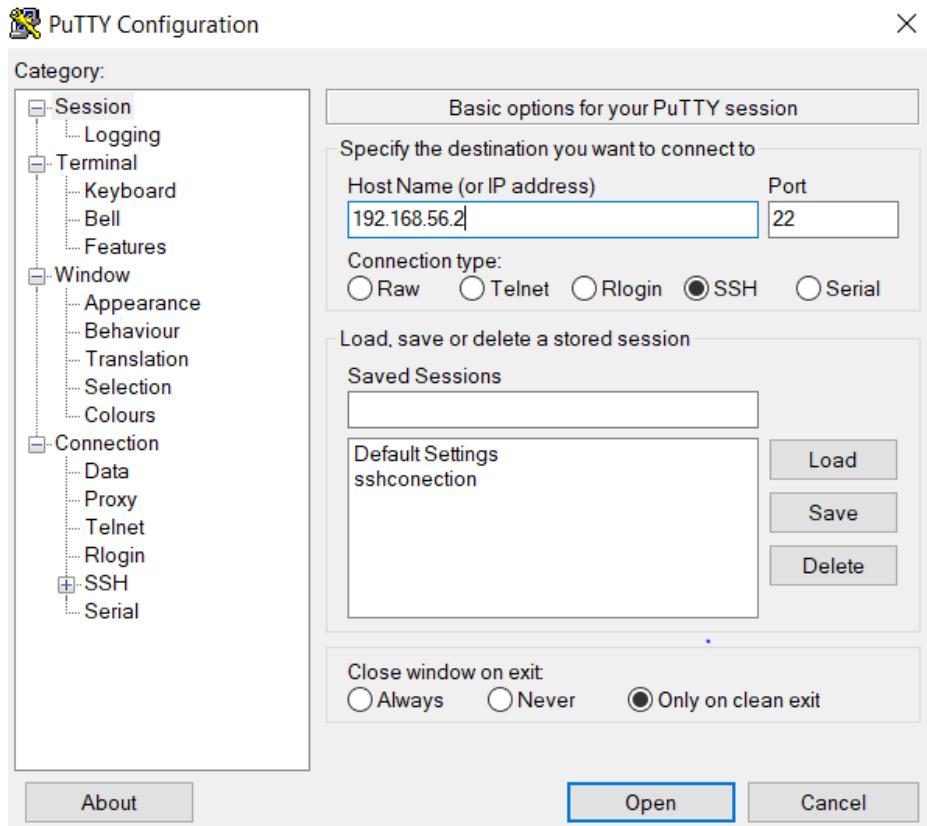
Verificamos que se hayan descargado “putty” y “texto” que son los archivos que están en el directorio público (del cual acabamos de descargar todo su contenido):



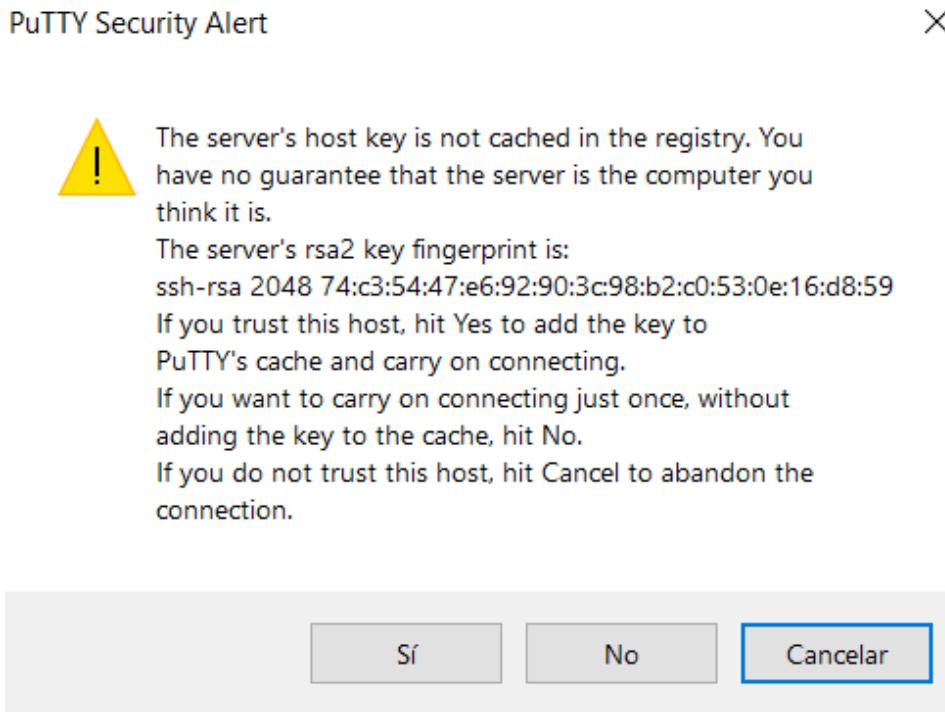
Si entramos a cada uno, podemos ver que la fecha de modificación de ambos fue recién, así que se acaban de descargar.

Parte 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.



Si es la primera vez que nos conectamos a un servidor, PuTTY mostrará una advertencia sobre la clave del servidor no estar en el caché del registro. Esto es normal para una primera conexión.



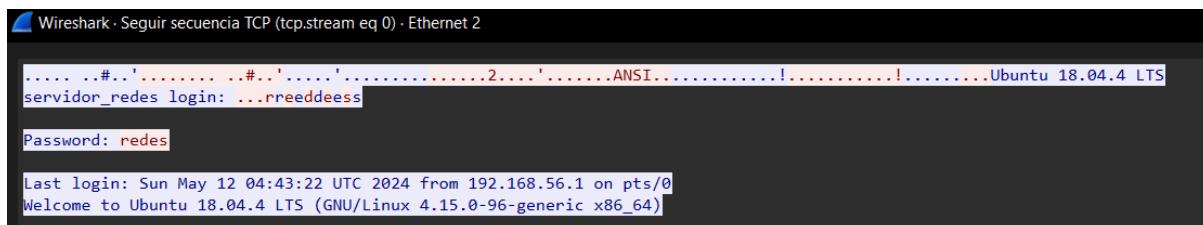
Nos logueamos y accedemos:

```
login as: redes
redes@192.168.56.2's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-96-generic x86_64)
```

2. ¿Qué diferencias existen entre usar SSH y Telnet? Demuestre comparando capturas de tráfico para ambos casos.

Si seguimos el TCP Stream en la conexión y el login al servidor por Telnet, podremos ver que tanto nuestro usuario y contraseña son visibles a través de Wireshark. Se envían los mensajes como texto plano a través de este protocolo. Esto hace que cualquier intermediario de una conversación Telnet pueda ver nuestro usuario y contraseña si sigue la secuencia TCP.

En la captura en rojo se ven los mensajes enviados por nosotros, y en azul los enviados por el servidor.



Ahora hagamos lo mismo, pero conectándonos por SSH.

Como vemos, la situación es muy diferente. Desde el principio, todo el tráfico que se puede ver está cifrado, como se evidencia por los datos que aparecen como una secuencia de caracteres y símbolos sin sentido aparente. Esto incluye el intercambio de claves, la autenticación y cualquier comando transmitido posteriormente. No se puede distinguir

información comprensible simplemente mirando los paquetes capturados, lo que evidencia el alto nivel de seguridad ofrecido por SSH gracias a su cifrado fuerte.

Entonces, la principal diferencia entre SSH y Telnet es la seguridad, ya que SSH proporciona seguridad de los datos transferidos porque se envían cifrados mientras que Telnet los envía en texto plano, lo que hace que cualquier persona con acceso a la red pueda interceptar y leer los datos, incluidas las contraseñas.

DNS

1. Realice una consulta DNS por un registro A usando el comando nslookup. Elija un sitio que no haya sido utilizado recientemente tratando de comenzar el intercambio de tráfico contra algún root servers. Indique el sitio, el comando utilizado y la respuesta.

Antes de hacer la consulta reiniciamos el servicio del DNS, para borrar la caché utilizando el siguiente comando:

```
redes@servidor_redes:~$ sudo service bind9 restart
```

Seteamos la ip de la máquina virtual para usar su DNS:

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

Luego ejecutamos nslookup para obtener las direcciones ip de los servidores de reddit:

```
> nslookup reddit.com  
Servidor: reddit.com  
Addresses: 2a04:4e42:400::396  
          2a04:4e42::396  
          2a04:4e42:200::396  
          2a04:4e42:600::396  
          151.101.1.140  
          151.101.65.140  
          151.101.129.140  
          151.101.193.140
```

Ejecutamos set type=ns para que nos devuelva el nombre de los servers:

```
> set type=ns
> .
Servidor: [192.168.56.2]
Address: 192.168.56.2

Respuesta no autoritativa:
(root) nameserver = b.root-servers.net
(root) nameserver = a.root-servers.net
(root) nameserver = m.root-servers.net
(root) nameserver = h.root-servers.net
(root) nameserver = e.root-servers.net
(root) nameserver = g.root-servers.net
(root) nameserver = f.root-servers.net
(root) nameserver = d.root-servers.net
(root) nameserver = j.root-servers.net
(root) nameserver = l.root-servers.net
(root) nameserver = i.root-servers.net
(root) nameserver = c.root-servers.net
(root) nameserver = k.root-servers.net
> set type=a
> b.root-servers.net
Servidor: [192.168.56.2]
Address: 192.168.56.2

DNS request timed out.
    timeout was 2 seconds.
Respuesta no autoritativa:
Nombre: b.root-servers.net
Address: 170.247.170.2
```

2. Identifique el root-server que responde indicando: nombre, IP y ubicación geográfica.

Usamos el siguiente filtro en wireshark, porque el puerto 53 es el que se utiliza para recibir y responder solicitudes DNS:

udp.port==53 || tcp.port==53

Se supone que inicialmente se envían por UDP para garantizar velocidad, pero en caso de que haya inconvenientes se hará por TCP.

Luego con los mensajes que nos quedamos en el wireshark, buscamos alguno en el cual el *Destination* corresponda a la ip de un root server. En la captura debajo se puede ver señalado en negro dicho mensaje, donde podemos ver que fuimos contra el server de ip 192.112.36.4, y en la lista de root servers, vemos que corresponde al root server del *US Department of Defense (NIC)*.

udp.port==53 tcp.port==53						
No.	Time	Source	Destination	Protocol	Length	Info
36477	49.232924	fe80::f0a3:5aff:fe9... fe80::7d8c:4ef8:9b1...	DNS	277	Standard query response 0x7e97 AAAA teams.microsoft.com CNAME teams.office.	
36483	49.253149	fe80::7d8c:4ef8:9b1... fe80::f0a3:5aff:fe9...	DNS	106	Standard query 0x20dc A config.teams.microsoft.com	
36484	49.253217	fe80::7d8c:4ef8:9b1... fe80::f0a3:5aff:fe9...	DNS	106	Standard query 0x726e AAAA config.teams.microsoft.com	
36486	49.260570	fe80::f0a3:5aff:fe9... fe80::7d8c:4ef8:9b1...	DNS	268	Standard query response 0xc597 HTTPS teams.microsoft.com CNAME teams.office	
36508	49.293394	fe80::f0a3:5aff:fe9... fe80::7d8c:4ef8:9b1...	DNS	264	Standard query response 0x20dc A config.teams.microsoft.com CNAME config.te	
36509	49.293394	fe80::f0a3:5aff:fe9... fe80::7d8c:4ef8:9b1...	DNS	276	Standard query response 0x726e AAAA config.teams.microsoft.com CNAME config	
37487	50.616162	172.20.10.8	192.112.36.4	DNS	82	Standard query 0xa0a2 NS <Root> OPT
37488	50.616321	172.20.10.8	192.112.36.4	DNS	93	Standard query 0xdc71 A reddit.com OPT
37615	50.814035	192.112.36.4	172.20.10.8	DNS	109	Standard query response 0xdc71 A reddit.com OPT
37616	50.814035	192.112.36.4	172.20.10.8	DNS	98	Standard query response 0xa0a2 NS <Root> OPT
37620	50.814833	172.20.10.8	192.112.36.4	TCP	66	61989 → 53 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
37620	50.814986	172.20.10.8	192.112.36.4	TCP	66	61990 → 53 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
37652	50.613138	192.168.56.1	192.168.56.2	DNS	70	Standard query 0x0003 A reddit.com
37730	51.024521	192.112.36.4	172.20.10.8	TCP	66	53 → 61989 [SYN, ACK] Seq=0 Ack=1 Win=14000 Len=0 MSS=1400 WS=1 SACK_PERM
37737	51.024521	192.112.36.4	172.20.10.8	TCP	66	53 → 61990 [SYN, ACK] Seq=0 Ack=1 Win=14000 Len=0 MSS=1400 WS=1 SACK_PERM
37738	51.024615	172.20.10.8	192.112.36.4	TCP	54	61989 → 53 [ACK] Seq=1 Ack=1 Win=131584 Len=0
37739	51.024629	172.20.10.8	192.112.36.4	TCP	54	61990 → 53 [ACK] Seq=1 Ack=1 Win=131584 Len=0
37740	51.025515	172.20.10.8	192.112.36.4	DNS	112	Standard query 0xb448 NS <Root> OPT
37747	51.025940	172.20.10.8	192.112.36.4	DNS	123	Standard query 0x8d0a A reddit.com OPT
37853	51.290470	192.112.36.4	172.20.10.8	TCP	54	53 → 61989 [ACK] Seq=1 Ack=0 Win=14069 Len=0
37854	51.290490	192.112.36.4	172.20.10.8	DNS	1254	Standard query response 0x8d0a A reddit.com NS e.gtld-servers.net NS c.gtld
37855	51.290518	192.112.36.4	172.20.10.8	TCP	54	53 → 61990 [ACK] Seq=1 Ack=59 Win=14058 Len=0

List of Root Servers

HOSTNAME	IP ADDRESSES	OPERATOR
a.root-servers.net	198.41.0.4, 2001:503:ba3e::2:30	Verisign, Inc.
b.root-servers.net	170.247.170.2, 2801:1b8:10::b	University of Southern California, Information Sciences Institute
c.root-servers.net	192.33.4.12, 2001:500:2::c	Cogent Communications
d.root-servers.net	199.7.91.13, 2001:500:2d::d	University of Maryland
e.root-servers.net	192.203.230.10, 2001:500:a8::e	NASA (Ames Research Center)
f.root-servers.net	192.5.5.241, 2001:500:2f::f	Internet Systems Consortium, Inc.
g.root-servers.net	192.112.36.4, 2001:500:12::d0d	US Department of Defense (NIC)
h.root-servers.net	198.97.190.53, 2001:500:1::53	US Army (Research Lab)
i.root-servers.net	192.36.148.17, 2001:7fe::53	Netnod
j.root-servers.net	192.58.128.30, 2001:503:c27::2:30	Verisign, Inc.
k.root-servers.net	193.0.14.129, 2001:7fd::1	RIPE NCC
l.root-servers.net	199.7.83.42, 2001:500:9f::42	ICANN
m.root-servers.net	202.12.27.33, 2001:dc3::35	WIDE Project

3. Detalle 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.

udp.port==53 tcp.port==53) && dns.qry.name == "reddit.com"						
No.	Time	Source	Destination	Protocol	Length	Info
37652	50.613138	192.168.56.1	192.168.56.2	DNS	70	Standard query 0x0003 A reddit.com
37488	50.616321	172.20.10.8	192.112.36.4	DNS	93	Standard query 0xdc71 A reddit.com OPT
37615	50.814035	192.112.36.4	172.20.10.8	DNS	109	Standard query response 0xdc71 A reddit.com OPT
37747	51.025940	172.20.10.8	192.112.36.4	DNS	123	Standard query 0x8d0a A reddit.com OPT
37854	51.290490	192.112.36.4	172.20.10.8	DNS	1254	Standard query response 0x8d0a A reddit.com NS e.gtld-servers.net NS c.gtld
37882	51.295235	172.20.10.8	192.55.83.30	DNS	93	Standard query 0x32cc A reddit.com OPT
37952	51.439146	192.55.83.30	172.20.10.8	DNS	492	Standard query response 0x32cc A reddit.com NS ns-557.awsdns-05.net NS ns-378.awsdns-47.
38032	51.590317	172.20.10.8	192.55.83.30	DNS	107	Standard query 0x0058 A reddit.com OPT
38162	51.750076	192.55.83.30	172.20.10.8	DNS	605	Standard query response 0x0058 A reddit.com NS ns-557.awsdns-05.net NS ns-378.awsdns-47.
38163	51.751568	172.20.10.8	205.251.193.122	DNS	93	Standard query 0x57a1 A reddit.com OPT
38196	51.790342	205.251.193.122	172.20.10.8	DNS	282	Standard query response 0x57a1 A reddit.com A 151.101.65.140 A 151.101.1.140 A 151.101.1
38256	51.791184	192.168.56.2	192.168.56.1	DNS	271	Standard query response 0x0003 A reddit.com A 151.101.1.140 A 151.101.65.140 A 151.101.1
38257	51.791575	192.168.56.1	192.168.56.2	DNS	70	Standard query 0x0004 AAAA reddit.com
38199	51.792068	172.20.10.8	205.251.193.122	DNS	93	Standard query 0x415e AAAA reddit.com OPT
38235	51.854026	205.251.193.122	172.20.10.8	DNS	330	Standard query response 0x415e AAAA reddit.com AAAA 2a04:4e42:400::396 AAAA 2a04:4e42:20
38258	51.857790	192.168.56.2	192.168.56.1	DNS	319	Standard query response 0x0004 AAAA reddit.com AAAA 2a04:4e42:400::396 AAAA 2a04:4e42::3

Agregamos el filtro: dnsqry.name == "reddit.com"

Cuando vemos la captura de wireshark con este filtro, podemos ver que comienza con la tarjeta de red de nuestra pc, dirección IPv4, que es 192.168.56.1. Entonces, nuestra PC le pregunta al servidor de redes (cuya IPv4 es 192.168.56.2), que hemos configurado como DNS donde está reddit.com. Como no tiene nada en caché, comienza a hacer las consultas recursivas.

Nuestro servidor de redes, sale por nuestro router, de dirección 172.20.10.7 a Internet a preguntarle a algún root server. Decidió ir al root server 192.112.36.4. Cómo podemos identificar por los colores de wireshark (rosado para TCP y azul para UDP), primero se intentó hacer por UDP, pero al parecer hubo algún inconveniente como puede ser una respuesta truncada, así que se envió por TCP.

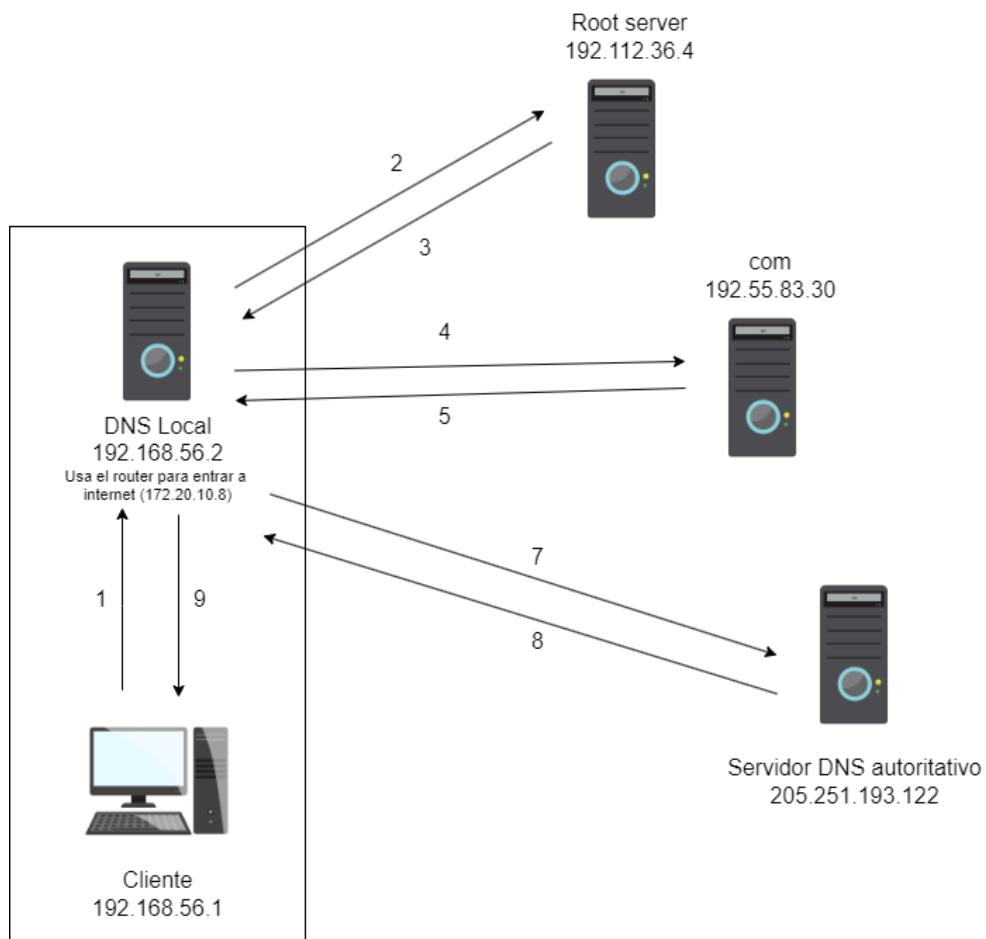
Cuando la respuesta llegó correctamente a nuestro router, y luego a nuestro servidor de redes, siguió con la recursión con la información otorgada de a qué servidor DNS debería ir a seguir su consulta. Seguramente la dirección 192.55.83.30 es de un DNS de nivel superior como por ejemplo '.com'

Las respuestas que siguen, son posiblemente respuestas de servidores de nombres autoritativos para el dominio de reddit.com o partes de él.

Finalmente, vemos que hay una respuesta de servidores de redes (192.168.56.2) a nuestra PC(192.168.56.1). Como comentario, la transferencia de paquetes que hay luego de esta respuesta es para consultar por la IPv6. Esto se deduce por que los registros que tienen A hacen referencia a IPv4 y los que tienen AAAA son IPv6. Además vemos que al final de nuevo hay una respuesta de 192.168.56.2 a 192.168.56.1

Además de todo lo mencionado, al comienzo de la comunicación hubo una solicitud de conexión TCP, que no se ve en esta captura pero sí se puede ver en una de las capturas del punto 2.

A continuación se puede ver un diagrama de lo recién explicado.



También usamos el comando `cmd tracert 192.112.36.4` para ver la traza.

```
Traza a la dirección G.ROOT-SERVERS.NET [192.112.36.4]
sobre un máximo de 30 saltos:

 1    <1 ms      <1 ms      <1 ms  192.168.1.1 [192.168.1.1]
 2      5 ms      6 ms      3 ms  tia4brasi1.antel.net.uy [200.40.78.196]
 3      4 ms      2 ms      4 ms  cbb4tia1-be125-605.antel.net.uy [200.40.78.16]
 4      6 ms      8 ms     14 ms  cbb4mlg1-be30-2001.antel.net.uy [179.31.59.229]
 5      6 ms      6 ms      6 ms  brpa51.pdp-be29-2001.antel.net.uy [179.31.59.224]
 6     20 ms     16 ms     16 ms  crton2.eze-ae1003.antel.net.uy [179.31.62.106]
 7     15 ms     14 ms     15 ms  84.16.6.137
 8     18 ms     19 ms     18 ms  84.16.6.136
 9     40 ms     40 ms     42 ms  176.52.249.39
10      *   159 ms      *   176.52.249.37
11      *      *      * Tiempo de espera agotado para esta solicitud.
12   130 ms   130 ms   131 ms  ae2.3604.ear4.Miami2.level3.net [4.69.207.41]
13   132 ms   130 ms   134 ms  ge-2-1-4.chi11.ip.tiscali.net [4.68.110.146]
14   152 ms   151 ms   147 ms  atx3-edge-01.inet.qwest.net [67.14.120.142]
15      *      *      * Tiempo de espera agotado para esta solicitud.
16      *      *      * Tiempo de espera agotado para esta solicitud.
17      *      *      * Tiempo de espera agotado para esta solicitud.
18      *      *      * Tiempo de espera agotado para esta solicitud.
19      *      *      * Tiempo de espera agotado para esta solicitud.
20      *      *      * Tiempo de espera agotado para esta solicitud.
21      *      *      * Tiempo de espera agotado para esta solicitud.
22      *      *      * Tiempo de espera agotado para esta solicitud.
23      *      *      * Tiempo de espera agotado para esta solicitud.
24      *      *      * Tiempo de espera agotado para esta solicitud.
25      *      *      * Tiempo de espera agotado para esta solicitud.
26      *      *      * Tiempo de espera agotado para esta solicitud.
27      *      *      * Tiempo de espera agotado para esta solicitud.
28      *      *      * Tiempo de espera agotado para esta solicitud.
29      *      *      * Tiempo de espera agotado para esta solicitud.
30      *      *      * Tiempo de espera agotado para esta solicitud.
```

4. 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. Indique al menos 2 (dos) formas de darse cuenta.

(udp.port==53 tcp.port==53) && dns.qry.name == "reddit.com"						
No.	Time	Source	Destination	Protocol	Length	Info
1796	1.804491	192.168.56.1	192.168.56.2	DNS	70	Standard query 0x0015 A reddit.com
1797	1.805161	192.168.56.2	192.168.56.1	DNS	403	Standard query response 0x0015 A reddit.com A 151.101.65.140 A 151.101
1798	1.805412	192.168.56.1	192.168.56.2	DNS	70	Standard query 0x0016 AAAA reddit.com
1799	1.805769	192.168.56.2	192.168.56.1	DNS	451	Standard query response 0x0016 AAAA reddit.com AAAA 2a04:4e42:600::396

El servidor contesta de caché. Se distingue de dos formas distintas, comparando con el caso anterior.

1 - Hay muchos menos intercambios de paquetes DNS que en la consulta anterior (la consulta es más rápida).

2 - En ningún momento vemos que haya una IPv4 distinta de la de nuestra máquina y nuestro servidor de redes. O sea, en ningún momento sale a Internet a preguntar por la ip del dominio, ya está guardada en el servidor de redes.

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

```
> nslookup www.lab.ort.edu.uy
Servidor: www.lab.ort.edu.uy
Address: 192.168.56.2
```

La dirección coincide con la de la máquina virtual. O sea que es un nombre de dominio para el servidor de redes.

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

En este caso tenemos que setear el tipo de registro en PTR y para consultar por la IP debemos escribirla al revés y agregarle .in-addr.arp al final

```
> set querytype=ptr
> server 192.168.56.2
Servidor predeterminado: 192.168.56.2
Address: 192.168.56.2

> 2.168.192.in-addr.arpa
Servidor: 192.168.56.2
Address: 192.168.56.2

2.168.192.in-addr.arpa      name = www.lab.ort.edu.uy
2.168.192.in-addr.arpa      name = mail.lab.ort.edu.uy
2.168.192.in-addr.arpa      name = dns.lab.ort.edu.uy
56.168.192.in-addr.arpa nameserver = dns.lab.ort.edu.uy
dns.lab.ort.edu.uy         internet address = 192.168.56.2
>
```

7. ¿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.

Usamos el tipo de registro mx que se corresponde con el nombre de la máquina u otro dominio que recibe correo para este dominio.

```
C:\Users\feder>nslookup
Servidor predeterminado: UnKnown
Address: fe80::f0a3:5aff:fe9b:be64

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

> set type=mx
> lab.ort.edu.uy
Servidor: [192.168.56.2]
Address: 192.168.56.2

lab.ort.edu.uy MX preference = 0, mail exchanger = mail.lab.ort.edu.uy
lab.ort.edu.uy nameserver = dns.lab.ort.edu.uy
mail.lab.ort.edu.uy internet address = 192.168.56.2
dns.lab.ort.edu.uy internet address = 192.168.56.2
```

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

Seteamos el tipo en ns (name server) y luego consultamos por el dominio com.uy. Estos son los servidores responsables de gestionar las consultas DNS para el dominio uy.

```
> set type=ns
> com.uy
Servidor: [192.168.56.2]
Address: 192.168.56.2
```

Respuesta no autoritativa:

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

9. 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?

Para realizar la consulta debemos hacer las siguientes configuraciones:

- set no recurse (para que no sea recursiva)
- set type=a (para consultar por el registro A)

La respuesta que obtenemos del root server son las direcciones de los Top Level Domain donde encontrar facebook.com. Esto significa que, como la consulta no es recursiva, el root server nos manda a buscar en esos dominios, en lugar de hacerlo él. Por lo tanto no obtenemos las direcciones IP de los servidores de facebook.

```
> set norecurse
> set type=a
> facebook.com
Servidor: [192.168.56.2]
Address: 192.168.56.2
```

Nombre: facebook.com

Served by:

- c.gtld-servers.net

com

- g.gtld-servers.net

com

- a.gtld-servers.net

com

- j.gtld-servers.net

com

- i.gtld-servers.net

com

- d.gtld-servers.net

com

- f.gtld-servers.net

com

- m.gtld-servers.net

com

- l.gtld-servers.net

com

- e.gtld-servers.net

com

10. 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 podemos obtener la respuesta. Esto es debido a que el set recurse le dice a nuestro dns que busque recursivamente de nuevo, como lo estaba haciendo por defecto.

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

Respuesta no autoritativa:
Nombre: facebook.com
Address: 31.13.94.35

> |
```

11. 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. ¿Cuál es la funcionalidad de esto? ¿Es realmente efectivo? ¿Existen soluciones con mejores resultados?, indicar.

En ambos casos obtuvimos las mismas direcciones IP pero con la diferencia de que están en distinto orden. Esto se debe al mecanismo de Round Robin, que se utiliza para balancear la carga de tráfico en los servidores. Cuando dos clientes distintos o la cantidad que sea preguntan por yahoo.com, se le otorga al primero en consultar cierta lista de direcciones ip de los servidores en orden. Cuando el segundo pregunta, verá como primera en la lista de IPs la que al original le aparecía como segunda. Así que cada uno irá a un servidor distinto, balanceando la carga.

Ventajas:

- Es una técnica muy simple para implementar balanceo de carga entre los servidores.
- No requiere hardware o software de balanceo de carga.

Desventajas:

- No tiene en cuenta la carga del servidor.
- No gestiona si alguno de los servidores falla.

Una mejor solución sería balanceadores de carga que tengan algún tipo de hardware o software, de los cuales hay muchos tipos pero claramente se pierde la ventaja de la simplicidad por tener más seguridad.

Algunas de las mejores soluciones que existen incluyen el balanceo de carga según las capacidades del servidor, el tráfico, o la geolocalización, entre otras.

A continuación se muestra captura de la consulta y la respuesta obtenida.

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

Respuesta no autoritativa:

```
Nombre: yahoo.com
Addresses: 98.137.11.164
           74.6.143.25
           74.6.143.26
           98.137.11.163
           74.6.231.20
           74.6.231.21
```

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

Respuesta no autoritativa:

```
Nombre: yahoo.com
Addresses: 74.6.231.20
           74.6.231.21
           98.137.11.164
           74.6.143.26
           98.137.11.163
           74.6.143.25
```

TCP/HTTP

Parte 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.

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.

Time	Source	Destination	Protocol	Length	Info
1 0.000000	192.168.56.1	192.168.56.2	TCP	66	52539 → 80 [SYN] Seq=2282586890 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
2 0.000499	192.168.56.2	192.168.56.1	TCP	66	80 → 52539 [SYN, ACK] Seq=1120116454 Ack=2282586891 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
3 0.000546	192.168.56.1	192.168.56.2	TCP	54	52539 → 80 [ACK] Seq=2282586891 Ack=1120116455 Win=2102272 Len=0
4 0.000720	192.168.56.1	192.168.56.2	HTTP	448	GET / HTTP/1.1
5 0.001117	192.168.56.2	192.168.56.1	TCP	60	80 → 52539 [ACK] Seq=1120116455 Ack=2282587285 Win=64128 Len=0
6 0.001726	192.168.56.2	192.168.56.1	HTTP	719	HTTP/1.1 200 OK (text/html)
7 0.033728	192.168.56.1	192.168.56.2	HTTP	428	GET /logoot.gif HTTP/1.1
8 0.034662	192.168.56.2	192.168.56.1	TCP	1514	80 → 52539 [ACK] Seq=1120117120 Ack=2282587659 Win=64128 Len=1460 [TCP segment of a reassembled PDU]
9 0.035212	192.168.56.2	192.168.56.1	HTTP	1469	HTTP/1.1 200 OK (GIF89a)
10 0.035288	192.168.56.1	192.168.56.2	TCP	54	52539 → 80 [ACK] Seq=2282587659 Ack=1120119995 Win=2102272 Len=0
11 2.795608	192.168.56.1	192.168.56.2	TCP	54	52539 → 80 [FIN, ACK] Seq=2282587659 Ack=1120119995 Win=2102272 Len=0
12 2.796433	192.168.56.2	192.168.56.1	TCP	60	80 → 52539 [FIN, ACK] Seq=1120119995 Ack=2282587660 Win=64128 Len=0
13 2.796471	192.168.56.1	192.168.56.2	TCP	54	52539 → 80 [ACK] Seq=2282587660 Ack=1120119996 Win=2102272 Len=0

2. Identifique el establecimiento de conexión. Describa de la misma: quién la inicia (¿siempre se cumple?, justifique), 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.

Time	Source	Destination	Protocol	Length	Info
1 0.000000	192.168.56.1	192.168.56.2	TCP	66	52539 → 80 [SYN] Seq=2282586890 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
2 0.000499	192.168.56.2	192.168.56.1	TCP	66	80 → 52539 [SYN, ACK] Seq=1120116454 Ack=2282586891 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
3 0.000546	192.168.56.1	192.168.56.2	TCP	54	52539 → 80 [ACK] Seq=2282586891 Ack=1120116455 Win=2102272 Len=0

Siempre la inicia el cliente. Esto se debe a que el servidor es el que está escuchando y el cliente es el que se está conectando. En este contexto no se podría dar el caso contrario ya que el cliente no está escuchando los puertos. Sin embargo, hay casos especiales en los que sí.

SEQ:

El número de secuencia inicial del host es 2282586890.

El número de secuencia inicial del servidor de redes es 1120116454.

ACK:

El ACK no está en el primero porque es el que está iniciando, no tiene que reconocer a nadie.

El ACK en el segundo es 2282586891, porque si bien el ACK se calcula como SEQ + LEN del mensaje anterior (y LEN es 0), como el mensaje trae la flag SYN se incrementa en 1.

El ACK en el tercero es 1120116455, por la misma razón que el segundo.

Largo del segmento:

En este caso el largo de todos los paquetes TCP enviados es cero porque no se está mandando nada, solo se está estableciendo la conexión.

Banderas:

El primero, que es la solicitud de conexión por parte del host al servidor de redes tiene SYN, que es la bandera para iniciar la conexión.

El segundo, la confirmación de aceptar la conexión por parte del servidor es SYN y ACK (significa que acepta la conexión).

El tercero, es la confirmación del cliente de que se va a conectar y lleva la bandera ACK.

3. *Identifique la finalización de la conexión, describa: quién la inicia (¿siempre se cumple?, justifique, la secuencia de segmentos intercambiados indicando: los números de SEQ y ACK, como así también banderas activas y largo de segmentos.*

11 2.795608	192.168.56.1	192.168.56.2	TCP	54 52539 → 80 [FIN, ACK] Seq=2282587659 Ack=1120119995 Win=2102272 Len=0
12 2.796433	192.168.56.1	192.168.56.1	TCP	60 80 → 52539 [FIN, ACK] Seq=1120119995 Ack=2282587660 Win=64128 Len=0
13 2.796471	192.168.56.1	192.168.56.2	TCP	54 52539 → 80 [ACK] Seq=2282587660 Ack=1120119996 Win=2102272 Len=0

En este caso nuestro PC que es el cliente inicia el cierre de la conexión. Esto se da porque nosotros cerramos el navegador, entonces fuimos quienes produjeron el cierre.

Cabe recalcar, que hubo problemas para capturar esto por el hecho de que en los navegadores modernos por más que uno cierre la pestaña, se produce el cierre solo del lado del servidor, el cliente la deja abierta para utilizarla en caso de que se necesite. Nuestra solución a esto fue usar el modo incógnito, que cierra la conexión por completo.

La secuencia de cierre muestra que se produce un cierre de 3 vías. El largo de todos los segmentos es 0, porque no se están enviando datos, solo se está cerrando la conexión.

El primer segmento del cierre lleva las banderas FIN y ACK, lleva el FIN porque está solicitando el cierre y el ACK porque hay una conexión establecida, al igual que la respuesta del servidor, donde le acepta el cierre.

SEQ y ACK en el cierre de conexión:

- El número de secuencia inicial del cliente es 2282587659 y su ACK es 1120119995.
- El número de secuencia inicial del servidor de redes es 1120119995 y su ACK es 2282587660. Aumenta en uno en relación al número de secuencia anterior porque la bandera de FIN cuenta como 1 byte.
- El número de secuencia final del cliente es 2282587660 y su ACK es 11201119996. El número de secuencia es igual que el ACK anterior, y el ACK aumenta en uno nuevamente por la bandera de FIN.

ACK:

- El ACK no está en el primero porque es el que está iniciando, no tiene que reconocer a nadie.
- El ACK en el segundo es 2282586891, porque si bien el ACK se calcula como SEQ + LEN del mensaje anterior (y LEN es 0), como el mensaje trae la flag SYN se incrementa en 1.
- El ACK en el tercero es 1120116455, por la misma razón que el segundo.

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.

```

Frame 4: 448 bytes on wire (3584 bits), 448 bytes captured (3584 bits) on interface \Device\NPF_{44349464-5D7B-41A7-AF7B-8E81F7116DEC
Ethernet II, Src: 0a:00:27:00:00:06 (0a:00:27:00:00:06), Dst: PCSSystemtecn_29:de:6c (08:00:27:29:de:6c)
Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.2
Transmission Control Protocol, Src Port: 52539, Dst Port: 80, Seq: 2282586891, Ack: 1120116455, Len: 394
Hypertext Transfer Protocol
  GET / HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
    Request Method: GET
    Request URI: /
    Request Version: HTTP/1.1
    Host: 192.168.56.2\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.0.0 Safari/537.36\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8\r\n
    Sec-GPC: 1\r\n
    Accept-Language: es-ES,es\r\n
    Accept-Encoding: gzip, deflate\r\n
\r\n
[Full request URI: http://192.168.56.2/]
[HTTP request 1/2]
[Response in frame: 6]
[Next request in frame: 7]

```

El encabezado que le brinda información al servidor acerca del navegador web cliente es el User-Agent, que en nuestro caso tiene el valor *"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.0.0 Safari/537.36"*

Significado:

- *Mozilla/5.0*: es un navegador que sigue las pautas de compatibilidad de Mozilla.
- *(Windows NT 10.0; Win64; x64)*: indica que el sistema operativo del cliente es Windows 10 en un arquitectura de 64 bits (Win64) en un procesador de 64 bits (x64).
- *AppleWebKit/537.36*: es el motor de renderizado del navegador web.
- *(KHTML, like Gecko)*: indica que el navegador es compatible con el motor de renderizado Gecko.
- *Chrome/124.0.0.0 Safari/537.36*: esto significa que el navegador se está identificando como Chrome y como Safari.

El fin de este encabezado es que los servidores web puedan identificar y adaptar la respuesta de la solicitud HTTP según las necesidades del cliente. Por ejemplo, es útil conocer qué tipo de navegador está realizando la solicitud por temas de optimización de contenido, o de compatibilidad con el navegador del cliente. También para poder redireccionar al usuario a versiones específicas de páginas web según su dispositivo o navegador, por ejemplo, si se detecta que es un navegador en un dispositivo móvil, redirigir a una versión adaptada para móvil del sitio.

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

```

▼ Hypertext Transfer Protocol
  ▼ HTTP/1.1 200 OK\r\n
    ▶ [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
    Response Version: HTTP/1.1
    Status Code: 200
    [Status Code Description: OK]
    Response Phrase: OK
    Vary: Accept-Encoding\r\n
    Content-Encoding: gzip\r\n
    Last-Modified: Thu, 08 Sep 2011 18:35:16 GMT\r\n
    ETag: "1287407511"\r\n
    Content-Type: text/html\r\n
    Accept-Ranges: bytes\r\n
    Content-Length: 403\r\n
    Date: Thu, 09 May 2024 21:15:12 GMT\r\n
    Server: lighttpd/1.4.45\r\n
  \r\n
  [HTTP response 1/2]
  [Time since request: 0.001006000 seconds]
  [Request in frame: 4]

```

El encabezado que brinda información acerca del servidor web es “Server”.

En este caso, como se puede ver en la captura, el servidor web usa el software Lighttpd en la versión 1.4.45.

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.

Time	Source	Destination	Protocol	Length	Info
1 0.000000	192.168.56.1	192.168.56.2	TCP	66	52539 → 80 [SYN] Seq=2282586890 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
2 0.000499	192.168.56.2	192.168.56.1	TCP	66	80 → 52539 [SYN, ACK] Seq=1120116454 Ack=2282586891 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
3 0.000546	192.168.56.1	192.168.56.2	TCP	54	52539 → 80 [ACK] Seq=2282586891 Ack=1120116455 Win=2102272 Len=0
4 0.000720	192.168.56.1	192.168.56.2	HTTP	448	GET / HTTP/1.1
5 0.001117	192.168.56.2	192.168.56.1	TCP	60	80 → 52539 [ACK] Seq=1120116455 Ack=2282587285 Win=64128 Len=0
6 0.001726	192.168.56.2	192.168.56.1	HTTP	719	HTTP/1.1 200 OK (text/html)
7 0.033728	192.168.56.1	192.168.56.2	HTTP	428	GET /logoort.gif HTTP/1.1
8 0.034662	192.168.56.2	192.168.56.1	TCP	1514	80 → 52539 [ACK] Seq=1120117120 Ack=2282587659 Win=64128 Len=1460 [TCP segment of a reassembled PDU]
9 0.035212	192.168.56.2	192.168.56.1	HTTP	1469	HTTP/1.1 200 OK (GIF89a)
10 0.035288	192.168.56.1	192.168.56.2	TCP	54	52539 → 80 [ACK] Seq=2282587659 Ack=1120119995 Win=2102272 Len=0
11 2.795608	192.168.56.1	192.168.56.2	TCP	54	52539 → 80 [FIN, ACK] Seq=2282587659 Ack=1120119995 Win=2102272 Len=0
12 2.796433	192.168.56.2	192.168.56.1	TCP	60	80 → 52539 [FIN, ACK] Seq=1120119995 Ack=2282587660 Win=64128 Len=0
13 2.796471	192.168.56.1	192.168.56.2	TCP	54	52539 → 80 [ACK] Seq=2282587660 Ack=1120119996 Win=2102272 Len=0

Los números de secuencia iniciales se generan aleatoriamente, con las condiciones de que no sean 0 y que no estén siendo utilizados por otros procesos, preferiblemente que estén alejados de los que están siendo usados por los otros procesos, para no generar confusiones.

En el establecimiento de conexión (mensaje del No. 1 al No. 3) comienzan a incrementarse estos números por las banderas, y también el valor corresponde al valor del ACK del mensaje anterior.

Para entender el valor de estos números es necesario entender cómo aumenta el valor del ACK. El ACK lleva un registro de cuál es el número de secuencia siguiente que el receptor está esperando de la parte contraria. El criterio es el siguiente:

- El valor del ACK corresponde a la suma de los valores de SEQ + LEN del segmento anterior.
- Si el segmento anterior trae alguna de las flags SYN, FIN o RST, se aumenta el valor del ACK en 1, aunque el LEN sea 0, o sea ACK = SEQ + 1.

El SEQ, es igual al ACK del último segmento enviado por la parte contraria.

Ya se habló en la parte 2) de porque se aumentan los números de secuencia en el inicio de conexión (es por las banderas, que cuentan como 1 byte cada una).

Luego del establecimiento de conexión, se empiezan a intercambiar segmentos TCP y el número de secuencia aumenta por el envío de datos, que va en el LEN. Podemos ver en la

captura, que según los length de los mensajes TCP, el SEQ del siguiente aumenta. Sin embargo, si analizamos esta parte:

3 0.000546	192.168.56.1	192.168.56.2	TCP	54 52539 + 80 [ACK] Seq=2282586891 Ack=1120116455 Win=2102272 Len=0
4 0.000720	192.168.56.1	192.168.56.2	HTTP	448 GET / HTTP/1.1
5 0.001117	192.168.56.2	192.168.56.1	TCP	60 80 + 52539 [ACK] Seq=1120116455 Ack=2282587285 Win=64128 Len=0
6 0.001726	192.168.56.2	192.168.56.1	HTTP	719 HTTP/1.1 200 OK (text/html)

Sería común pensar que hubo un error, porque el ACK ya no es igual al del último segmento TCP que aparece. Sin embargo, tenemos que recordar que el HTTP GET que vemos en la captura está corriendo sobre TCP.

Transmission Control Protocol, Src Port: 52539, Dst Port: 80, Seq: 2282586891, Ack: 1120116455, Len: 394

Y ahí podemos ver, que el largo del segmento TCP que envía el cliente al servidor es 394, que es la diferencia que hay entre 2282586891 y 2282587285. Esto nos molesta ya que queremos ver los números de secuencia, y Wireshark cuando hay un HTTP por encima de un TCP, prioriza mostrarnos la información del HTTP. Si desactivamos el análisis de HTTP, nos quedara una captura que se asemeja mucho a los ejercicios hechos en clase.

Time	Source	Destination	Protocol	Length Info
1 0.000000	192.168.56.1	192.168.56.2	TCP	66 52539 + 80 [SYN] Seq=2282586890 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
2 0.000499	192.168.56.2	192.168.56.1	TCP	66 80 + 52539 [SYN, ACK] Seq=1120116454 Ack=2282586891 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
3 0.000546	192.168.56.1	192.168.56.2	TCP	54 52539 + 80 [ACK] Seq=2282586891 Ack=1120116455 Win=2102272 Len=0
4 0.000720	192.168.56.1	192.168.56.2	TCP	448 52539 + 80 [PSH, ACK] Seq=2282586891 Ack=1120116455 Win=2102272 Len=394
5 0.001117	192.168.56.2	192.168.56.1	TCP	60 80 + 52539 [ACK] Seq=1120116455 Ack=2282587285 Win=64128 Len=0
6 0.001726	192.168.56.2	192.168.56.1	TCP	719 80 + 52539 [PSH, ACK] Seq=1120116455 Ack=2282587285 Win=64128 Len=665
7 0.033728	192.168.56.1	192.168.56.2	TCP	428 52539 + 80 [PSH, ACK] Seq=2282587285 Ack=1120117120 Win=2101504 Len=374
8 0.034662	192.168.56.2	192.168.56.1	TCP	1514 80 + 52539 [ACK] Seq=1120117120 Ack=2282587659 Win=64128 Len=1460
9 0.035212	192.168.56.2	192.168.56.1	TCP	1469 80 + 52539 [PSH, ACK] Seq=1120118586 Ack=2282587659 Win=64128 Len=1415
10 0.035288	192.168.56.1	192.168.56.2	TCP	54 52539 + 80 [ACK] Seq=1120119995 Ack=2282587659 Win=2102272 Len=0
11 2.795608	192.168.56.1	192.168.56.2	TCP	54 52539 + 80 [FIN, ACK] Seq=2282587659 Ack=1120119995 Win=2102272 Len=0
12 2.796433	192.168.56.2	192.168.56.1	TCP	60 80 + 52539 [FIN, ACK] Seq=1120119995 Ack=2282587660 Win=64128 Len=0
13 2.796471	192.168.56.1	192.168.56.2	TCP	54 52539 + 80 [ACK] Seq=2282587660 Ack=1120119996 Win=2102272 Len=0

En esta captura, es muy fácil analizar qué regla mencionada anteriormente sobre el ACK y el SEQ se cumple. Además podemos observar un ACK acumulativo (línea 10) en la parte que se envía la imagen.

Luego, en el fin de conexión, también se habló anteriormente porque sus números de secuencia aumentan.

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

Sí, se puede, una forma posible (hay varias) es tomando los números de secuencia iniciales de cada parte, y haciendo la diferencia con el último número de secuencia de la misma parte, sin contar las banderas de FIN y SYN, o sea sin contar el Handshake TCP y la finalización de la conexión. Como se dió un intercambio normal (sin retransmisiones ni pérdidas) podemos decir que este fue el TOTAL de bytes enviados de cada parte.

SEQfinal - SEQinicial

Cliente: 2282587659 - 2282586891 = 768 bytes enviados al servidor

Servidor: 1120119995 - 1120116455 = 3540 bytes enviados al cliente

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

tcp.flags.push == 1					
No.	Time	Source	Destination	Protocol	Length Info
4 0.000720	192.168.56.1	192.168.56.2	HTTP	448 GET / HTTP/1.1	
6 0.001726	192.168.56.2	192.168.56.1	HTTP	719 HTTP/1.1 200 OK (text/html)	
7 0.033728	192.168.56.1	192.168.56.2	HTTP	428 GET /logoort.gif HTTP/1.1	
9 0.035212	192.168.56.2	192.168.56.1	HTTP	1469 HTTP/1.1 200 OK (GIF89a)	

El TCP que tienen por detrás:

tcp.flags.push == 1						
No.	Time	Source	Destination	Protocol	Length	Info
4	0.000720	192.168.56.1	192.168.56.2	TCP	448	52539 → 80 [PSH, ACK] Seq=2282586891 Ack=11201116455 Win=2102272 Len=394
6	0.001726	192.168.56.2	192.168.56.1	TCP	719	80 → 52539 [PSH, ACK] Seq=1120116455 Ack=2282587285 Win=64128 Len=665
7	0.033728	192.168.56.1	192.168.56.2	TCP	428	52539 → 80 [PSH, ACK] Seq=2282587285 Ack=1120117120 Win=2101504 Len=374
9	0.035212	192.168.56.2	192.168.56.1	TCP	1469	80 → 52539 [PSH, ACK] Seq=1120118580 Ack=2282587659 Win=64128 Len=1415

Usamos el filtro `tcp.flags.push == 1` para buscar la bandera PSH en TCP y la encontramos solamente en los mensajes HTTP.

En los detalles del mensaje vemos lo siguiente:

```
Flags: 0x018 (PSH, ACK)
 000. .... .... = Reserved: Not set
 ...0 .... .... = Accurate ECN: Not set
 .... 0.... .... = Congestion Window Reduced: Not set
 .... .0.. .... = ECN-Echo: Not set
 .... ..0. .... = Urgent: Not set
 .... ...1 .... = Acknowledgment: Set
 .... .... 1... = Push: Set
 .... .... .0.. = Reset: Not set
 .... .... ..0. = Syn: Not set
 .... .... ...0 = Fin: Not set
 [TCP Flags: .....AP....]
```

La bandera PSH (Push) en TCP se utiliza para indicar que los datos enviados deben ser entregados inmediatamente al destino, sin que se acumulen en el búfer. El servidor cuando recibe este paquete lo envía de inmediato a la capa de aplicación, incluso si se están esperando más datos. Esto es útil en aplicaciones de mensajería en tiempo real o videoconferencia. Pero en nuestro caso se utiliza para indicar que ya se envió el último segmento de datos (ya que el mensaje HTTP podría estar compuesto de varios segmentos TCP) y que ya se debe procesar el mensaje. En la primera fila (No. 4), significa que ya se envió toda la solicitud (a pesar de que es solamente un segmento), en la segunda (No.6), la respuesta de esta. Luego en el No.7 es la request de la imagen del servidor. La No.9 indica que ya llegó toda la imagen y tiene la respuesta HTTP. Como comentario el No. 8 que no tiene el push, está enviando la primera parte de la imagen, y el No. 9 envía la parte restante y como ya envió todo hace el PSH.

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

Cuando una parte de la comunicación TCP desea enviar solamente un reconocimiento (ACK) y no datos, el número de secuencia que debe enviar será el mismo que el último número de secuencia que esa parte envió y que ha sido confirmado por la otra parte. Este número de secuencia se mantiene constante ya que no se están enviando nuevos datos.

10. Capture nuevamente e intente acceder ahora a la dirección del servidor pero en el puerto 443. (<http://192.168.56.2:443>). ¿Logró conectarse? ¿Por qué sucede esto? ¿Qué bandera se utiliza para señalizar esto?

No, no logramos conectarnos.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.56.1	192.168.56.2	TCP	66	59105 → 443 [SYN] Seq=3604148805 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
2	0.000456	192.168.56.2	192.168.56.1	TCP	66	443 → 59105 [RST, ACK] Seq=0 Ack=3604148806 Win=0 Len=0
3	0.252231	192.168.56.1	192.168.56.2	TCP	66	59106 → 443 [SYN] Seq=3005128179 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
4	0.252615	192.168.56.2	192.168.56.1	TCP	66	443 → 59106 [RST, ACK] Seq=0 Ack=3005128180 Win=0 Len=0
5	0.500943	192.168.56.1	192.168.56.2	TCP	66	[TCP Port numbers reused] 59105 → 443 [SYN] Seq=3604148805 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
6	0.501330	192.168.56.1	192.168.56.2	TCP	66	443 → 59105 [RST, ACK] Seq=0 Ack=3604148806 Win=0 Len=0
7	0.752806	192.168.56.1	192.168.56.2	TCP	66	[TCP Port numbers reused] 59106 → 443 [SYN] Seq=3005128179 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
8	0.753319	192.168.56.2	192.168.56.1	TCP	66	443 → 59106 [RST, ACK] Seq=0 Ack=3005128180 Win=0 Len=0
9	1.002344	192.168.56.1	192.168.56.2	TCP	66	[TCP Port numbers reused] 59105 → 443 [SYN] Seq=3604148805 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
10	1.002732	192.168.56.2	192.168.56.1	TCP	66	443 → 59105 [RST, ACK] Seq=0 Ack=3604148806 Win=0 Len=0
11	1.254557	192.168.56.1	192.168.56.2	TCP	66	[TCP Port numbers reused] 59106 → 443 [SYN] Seq=3005128179 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
12	1.254911	192.168.56.2	192.168.56.1	TCP	66	443 → 59106 [RST, ACK] Seq=0 Ack=3005128180 Win=0 Len=0
13	1.503108	192.168.56.1	192.168.56.2	TCP	66	[TCP Port numbers reused] 59105 → 443 [SYN] Seq=3604148805 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
14	1.503483	192.168.56.2	192.168.56.1	TCP	66	443 → 59105 [RST, ACK] Seq=0 Ack=3604148806 Win=0 Len=0
15	1.755211	192.168.56.1	192.168.56.2	TCP	66	[TCP Port numbers reused] 59106 → 443 [SYN] Seq=3005128179 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
16	1.755581	192.168.56.2	192.168.56.1	TCP	66	443 → 59106 [RST, ACK] Seq=0 Ack=3005128180 Win=0 Len=0
17	2.004201	192.168.56.1	192.168.56.2	TCP	66	[TCP Port numbers reused] 59105 → 443 [SYN] Seq=3604148805 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
18	2.004632	192.168.56.2	192.168.56.1	TCP	66	443 → 59105 [RST, ACK] Seq=0 Ack=3604148806 Win=0 Len=0
19	2.256270	192.168.56.1	192.168.56.2	TCP	66	[TCP Port numbers reused] 59106 → 443 [SYN] Seq=3005128179 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
20	2.256818	192.168.56.2	192.168.56.1	TCP	66	443 → 59106 [RST, ACK] Seq=0 Ack=3005128180 Win=0 Len=0
21	3.040333	192.168.56.1	192.168.56.2	TCP	66	59110 → 443 [SYN] Seq=1523897152 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
22	3.040828	192.168.56.2	192.168.56.1	TCP	66	443 → 59110 [RST, ACK] Seq=0 Ack=1523897153 Win=0 Len=0
23	3.291300	192.168.56.1	192.168.56.2	TCP	66	59111 → 443 [SYN] Seq=2123245192 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
24	3.291665	192.168.56.2	192.168.56.1	TCP	66	443 → 59111 [RST, ACK] Seq=0 Ack=2123245193 Win=0 Len=0
25	3.541321	192.168.56.1	192.168.56.2	TCP	66	[TCP Port numbers reused] 59110 → 443 [SYN] Seq=1523897152 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
26	3.541687	192.168.56.2	192.168.56.1	TCP	66	443 → 59110 [RST, ACK] Seq=0 Ack=1523897153 Win=0 Len=0
27	3.792358	192.168.56.1	192.168.56.2	TCP	66	[TCP Port numbers reused] 59111 → 443 [SYN] Seq=2123245192 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
28	3.792714	192.168.56.2	192.168.56.1	TCP	66	443 → 59111 [RST, ACK] Seq=0 Ack=2123245193 Win=0 Len=0
29	4.042240	192.168.56.1	192.168.56.2	TCP	66	[TCP Port numbers reused] 59110 → 443 [SYN] Seq=1523897152 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM

Esto sucede porque 443 es el puerto en el que opera HTTPS y nosotros específicamente estamos intentando acceder mediante HTTP.

La bandera que se utiliza para señalizar el rechazo de la conexión es RST (Reset). En la captura de tráfico, después de cada intento de conexión (indicado por el flag SYN), el servidor responde con un segmento TCP que tiene el flag RST activado. Esto indica que el servidor está cerrando la conexión inmediatamente y que no acepta nuevas conexiones en ese puerto.

Como comentario, si vamos a las reglas de coloreado de Wireshark, vemos que los rojos indican TCP RST, y los negros BAD TCP

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 respuesta HTTP.

No.	Time	Source	Destination	Protocol	Length	Info
4	0.000671	192.168.56.1	192.168.56.2	HTTP	482	GET / HTTP/1.1
7	0.002118	192.168.56.2	192.168.56.1	HTTP	719	HTTP/1.1 200 OK (text/html)
10	0.017090	192.168.56.1	192.168.56.2	HTTP	423	GET /logoort.gif HTTP/1.1
12	0.018752	192.168.56.2	192.168.56.1	HTTP	1469	HTTP/1.1 200 OK (GIF89a)

Analicemos el mensaje HTTP seleccionado, enviado por el servidor al cliente (el envío del HTML de la página).

▼ Hypertext Transfer Protocol
► HTTP/1.1 200 OK\r\n
Vary: Accept-Encoding\r\n
Content-Encoding: gzip\r\n
Last-Modified: Thu, 08 Sep 2011 18:35:16 GMT\r\n

Como vemos hay un encabezado Last-Modified, qué nos dice que la última modificación de la página fue el 8 de septiembre de 2011 y el código de respuesta es 200, que significa éxito.

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.

http						
No.	Time	Source	Destination	Protocol	Length	Info
4	0.000629	192.168.56.1	192.168.56.2	HTTP	587	GET / HTTP/1.1
7	0.001692	192.168.56.2	192.168.56.1	HTTP	259	HTTP/1.1 304 Not Modified

La respuesta 304 Not Modified indica que el contenido solicitado no ha sido modificado desde la última vez que fue accedido o solicitado por el cliente. En consecuencia, no se envía nuevamente el contenido completo al cliente, lo cual es un mecanismo eficiente para ahorrar ancho de banda y reducir la carga en el servidor.

Este mensaje solo se da, porque nuestro navegador guardó en caché la página y su última fecha de modificación (2011). Entonces, en esta nueva solicitud HTTP se envía un encabezado If-Modified-Since con la última fecha guardada en nuestra PC. También utiliza el If-None-Match, que se fija si el Entity Tag sigue siendo el mismo que en la última solicitud.

Esto indica que el cliente solo quiere descargar una actualización de la página si esta cambia. Como no cambió, el servidor le responde un 304 Not modified, y como vemos los encabezados tienen un EntityTag y un Last-Modification que coinciden con los enviados en la solicitud (en otras palabras, todo está al día). A continuación las imágenes que muestran esto:

Si entramos a la solicitud HTTP, veremos los encabezados nombrados:

```
Accept-Language: es-419,es;q=0.9,pt;q=0.8,pt-br;q=0.7
If-None-Match: "1287407511"\r\n
If-Modified-Since: Thu, 08 Sep 2011 18:35:16 GMT\r\n
\r\n
```

Y en la respuesta HTTP, coinciden como se dijo anteriormente:

```
Hypertext Transfer Protocol
  HTTP/1.1 304 Not Modified\r\n
  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
```

Parte 2 - Análisis de las conexiones

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

Al ejecutar el comando:

Conexiones activas			
Proto	Dirección local	Dirección remota	Estado
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	0.0.0.0:5040	0.0.0.0:0	LISTENING
TCP	0.0.0.0:7680	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	0.0.0.0:49669	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49677	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49703	0.0.0.0:0	LISTENING
TCP	0.0.0.0:57621	0.0.0.0:0	LISTENING
TCP	0.0.0.0:58919	0.0.0.0:0	LISTENING
TCP	0.0.0.0:62885	0.0.0.0:0	LISTENING
TCP	127.0.0.1:5500	0.0.0.0:0	LISTENING
TCP	127.0.0.1:6463	0.0.0.0:0	LISTENING
TCP	127.0.0.1:49706	0.0.0.0:0	LISTENING
TCP	127.0.0.1:49707	0.0.0.0:0	LISTENING
TCP	127.0.0.1:49708	0.0.0.0:0	LISTENING
TCP	192.168.1.4:139	0.0.0.0:0	LISTENING
TCP	192.168.1.4:2030	0.0.0.0:0	LISTENING
TCP	192.168.1.4:58135	52.226.139.121:443	ESTABLISHED
TCP	192.168.1.4:58185	200.40.28.75:443	CLOSE_WAIT
TCP	192.168.1.4:58186	192.16.49.85:80	CLOSE_WAIT
TCP	192.168.1.4:58426	52.112.38.151:443	ESTABLISHED
TCP	192.168.1.4:58428	25.106.224.26:443	ESTABLISHED

La herramienta **netstat** proporciona información sobre las conexiones de red, el estado de las conexiones y estadísticas de la interfaz de red. El **-n** indica a netstat que muestre las direcciones IP y los números de puerto en forma numérica en lugar de intentar determinar los nombres de dominio y nombres de servicios a partir de esos números.

El **-a** le indica que muestra todas las conexiones y puertos de escucha, incluyendo tanto conexiones TCP como UDP.

- ¿Qué significan los estados "ESTABLISHED" y "LISTENING" que observa?

"ESTABLISHED" indica que una conexión ha sido exitosamente establecida entre dos dispositivos en la red. En este estado, ambos extremos de la conexión han completado el proceso de handshake de TCP, que es necesario para iniciar una conexión TCP.

El estado "LISTENING" indica que un host está esperando (escuchando) conexiones entrantes en un puerto específico. En otras palabras, no hay una conexión activa en ese

momento, pero el host está listo y disponible para aceptar una nueva conexión en ese puerto. Es crucial para aplicaciones que necesitan aceptar conexiones entrantes.

3. *Describa además que significa el estado “TIME-WAIT”.*

Es un estado en el protocolo TCP que ocurre después de que una conexión ha sido cerrada por una parte y se ha enviado el último ACK para confirmar la recepción del paquete FIN de la otra parte. En este estado, una conexión es técnicamente cerrada, pero el socket permanece abierto por un período de tiempo determinado. Sirve para asegurar que cualquier paquete duplicado que llegue tarde (que fue enviado por la otra parte antes de recibir el aviso de cierre de conexión) sea identificado y descartado. Esto evita confusiones en la apertura de nuevas conexiones que podrían usar el mismo par de puerto/IP.

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

Escribimos en consola: telnet 192.168.56.2 23 para acceder a la consola del servidor y la diferencia que encontramos con la salida anterior es que ahora aparece una conexión establecida con el servidor en el puerto 23.

TCP	192.168.56.1:55217	192.168.56.2:23	ESTABLISHED
-----	--------------------	-----------------	-------------

Parte 3 - Throughput de una conexión TCP

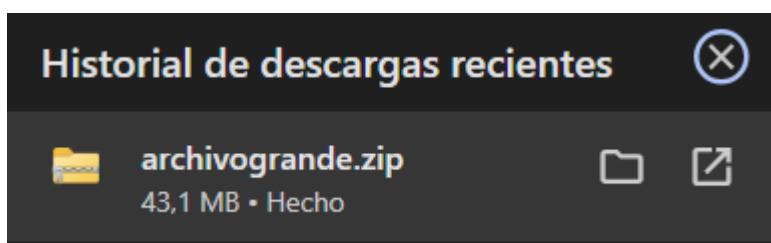
- Se comenzará estudiando la transferencia sobre un enlace con las siguientes características:
 - Ancho de banda: 10 Mbps (Mega-bits por segundo)
 - Retardo: 50 ms (mili-segundos)

Para obtener esto, ingrese a la VM, con el usuario y contraseña "redes", y ejecute la siguiente línea en la terminal de línea de comandos de la VM Linux:

`./enlace1.sh`

```
redes@servidor_redes:~$ ./enlace1.sh
[sudo] password for redes:
RTNETLINK answers: No such file or directory
redes@servidor_redes:~$
```

- Descargue el archivo `archivogrande.zip` del servidor mediante HTTP, usando:
<http://192.168.56.2/archivogrande.zip>



Vemos que se descargó correctamente.

- 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.
 - El tiempo transcurrido.
 - Con los datos anteriores, calcule el throughput en Mbps y comparelo con el configurado como límite, utilizando la aplicación tc.

Inicio:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.56.1	192.168.56.2	TCP	66	53499 → 80 [SYN] Seq=3083274003 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
2	0.051891	192.168.56.2	192.168.56.1	TCP	66	80 → 53499 [SYN, ACK] Seq=2189617763 Ack=3083274004 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
3	0.051989	192.168.56.1	192.168.56.2	TCP	54	53499 → 80 [ACK] Seq=3083274004 Ack=2189617764 Win=262656 Len=0

Fin:

46437	43.202129	192.168.56.2	192.168.56.1	TCP	60	80 → 53499 [FIN, ACK] Seq=2234804154 Ack=3083274415 Win=64128 Len=0
46438	43.202160	192.168.56.1	192.168.56.2	TCP	54	53499 → 80 [ACK] Seq=3083274415 Ack=2234804155 Win=262656 Len=0

(Luego del handshake y antes del fin de la conexión)

Número de secuencia inicial de cliente: 3083274004

Número de secuencia inicial del servidor: 2189617764

Número de secuencia final de cliente: 3083274415

Número de secuencia final del servidor: 2234804154

- a) Para calcular los bytes enviados restamos los números de secuencia final con los iniciales de cada parte, como hicimos en la parte 1.7 de HTTP/TCP, sin contar el aumento de las banderas de FIN y SYN.

Bytes enviados por el cliente: $3083274415 - 3083274004 = 411$ Bytes

Bytes enviados por el servidor: $2234804154 - 2189617763 - 1 = 45.186.390$ Bytes = 45,186390 MBytes

- b) El tiempo transcurrido lo vemos en la columna "Time" y fueron 43,202160 segundos.

- c) El throughput en Mbps se calcula como cantidad de bits enviados dividido el tiempo que tardaron en llegar.

Primero convertimos bytes a bits: 45,186390 MBytes = 361,49112 Mbits.

Y ahora hacemos la división: 361,49112 Mbits / 43,202160 segundos = 8,367431628 Mbps

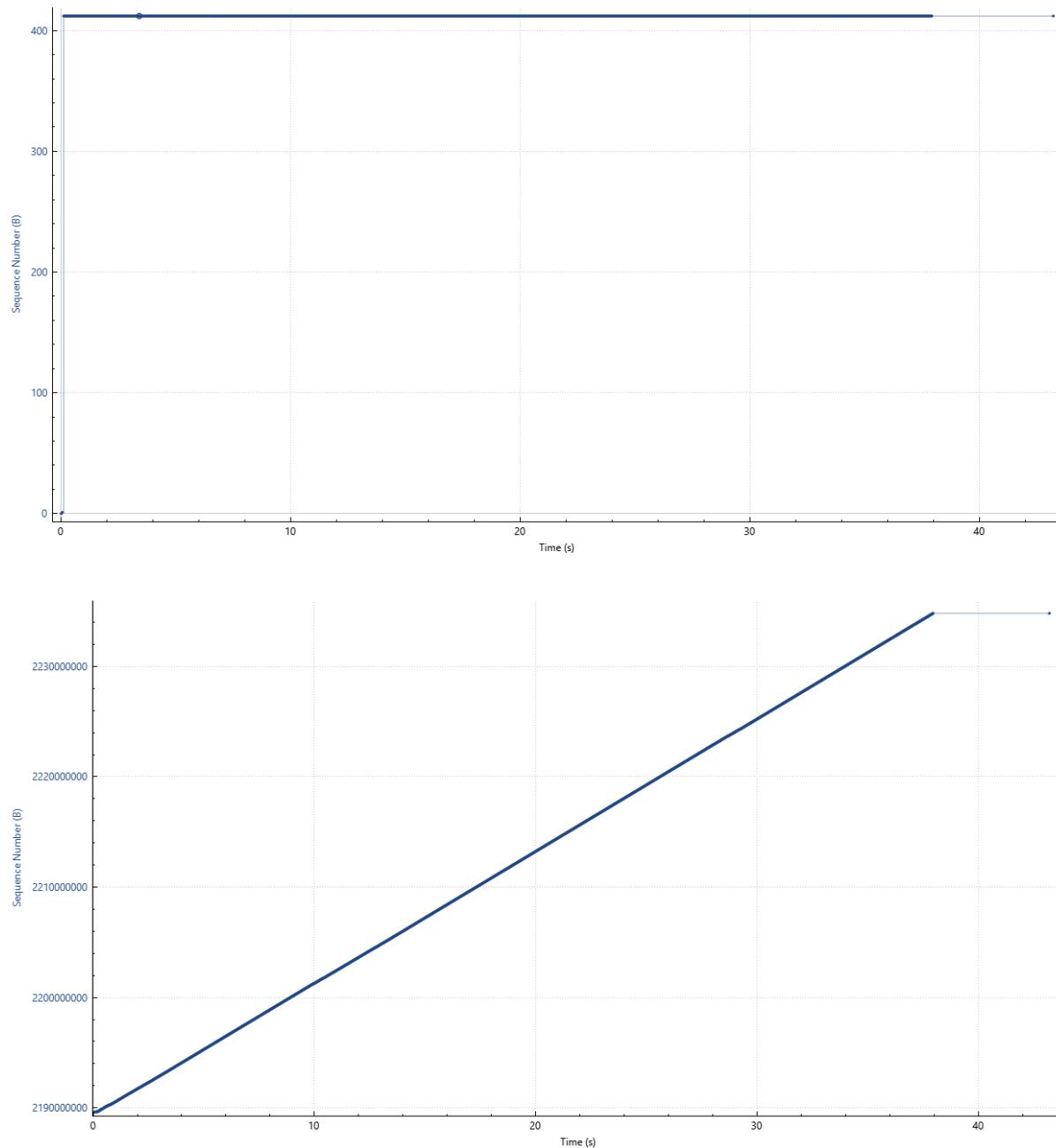
Es un poco menor de lo establecido con el script enlace1.sh, esto es normal ya que el ancho de banda de 10Mbps es el teórico, no el práctico, y se estableció inicialmente un retardo de 50 ms. Además, cuando tomamos el tiempo final, tenemos que tener en cuenta que hay un tiempo de 6 segundos (del segundo 37,943904 al 43,202160) aproximadamente entre que el servidor termina de enviar la última parte del archivo y se da el ACK del FIN.

Si hiciéramos la cuenta solo contando hasta el fin de envío del archivo, la división daría 361,49112 Mbits / 37,943904 segundos = 9,526988051 Mbps, lo cual es más cercano a 10Mbps.

46436 37.943904	192.168.56.1	192.168.56.2	TCP	54 53499 → 80 [ACK] Seq=3083274415 Ack=2234804154 Win=262656 Len=0
46437 43.202129	192.168.56.2	192.168.56.1	TCP	60 80 → 53499 [FIN, ACK] Seq=2234804154 Ack=3083274415 Win=64128 Len=0
46438 43.202160	192.168.56.1	192.168.56.2	TCP	54 53499 → 80 [ACK] Seq=3083274415 Ack=2234804155 Win=262656 Len=0

Esta imagen ilustra el retraso mencionando entre el fin del envío del archivo y la solicitud de finalizar la conexión.

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.



Nos paramos en el último paquete y vemos el número de secuencia final y también confirmamos el tiempo (43,2 s) y los bytes enviados por el cliente (411) y por el servidor (45 MB).

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

Es asimétrica, en el caso de una descarga de archivo desde una página web, es común que solo el servidor envíe un FIN para iniciar el cierre de la conexión, dado que su rol principal es transmitir datos y una vez finalizada esta tarea, no tiene razón para recibir más datos del cliente.

6. Se pasará ahora a utilizar un enlace con las siguientes características:

- Ancho de banda: 10 Mbps
- Retardo: 50 ms
- Tasa de pérdida de paquetes: 0.5 %

Para obtener esto, ejecute la siguiente línea en la terminal de línea de comandos de la VM Linux1:

`./enlace2.sh`

Repita ahora las pruebas de los puntos 2 y 3. ¿Qué cambios observa?

¿Por qué ocurren los mismos?

Inicio:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.56.1	192.168.56.2	TCP	66	57374 → 80 [SYN] Seq=1687780520 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
2	0.052387	192.168.56.2	192.168.56.1	TCP	66	80 → 57374 [SYN, ACK] Seq=2343233773 Ack=1687780521 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
3	0.052509	192.168.56.1	192.168.56.2	TCP	54	57374 → 80 [ACK] Seq=1687780521 Ack=2343233774 Win=262656 Len=0

Fin:

47824	82.393166	192.168.56.2	192.168.56.1	TCP	60	80 → 57374 [FIN, ACK] Seq=2388420164 Ack=1687780932 Win=64128 Len=0
-	47825	82.393274	192.168.56.1	TCP	54	57374 → 80 [ACK] Seq=1687780932 Ack=2388420165 Win=262656 Len=0

(Luego del handshake y antes del fin de la conexión)

Número de secuencia inicial de cliente: 1687780521

Número de secuencia inicial del servidor: 2343233774

Número de secuencia final de cliente: 1687780932

Número de secuencia final del servidor: 2388420164

- a) Para calcular los bytes enviados restamos los números de secuencia final con los iniciales de cada parte, como hicimos en la parte 1.7 de HTTP/TCP, sin contar el aumento de las banderas de FIN y SYN.

Bytes enviados por el cliente: 1687780932 - 1687780521 = 411 Bytes

Bytes enviados por el servidor: 2388420164 - 2343233773 = 45.186.390 Bytes = 45,186390 MBytes

- b) El tiempo transcurrido lo vemos en la columna “Time” y fueron 82,393274 segundos

- c) El throughput en Mbps se calcula como cantidad de bits enviados dividido el tiempo que tardaron en llegar.

Primero convertimos bytes a bits: 45,186390 MBytes = 361,49112 Mbits.

Y ahora hacemos la división: 361,49112 Mbits / 82,393274 segundos = 4,387386 Mbps

¿Qué cambios observa?

¿Por qué ocurren los mismos?

Como vemos, los bytes transmitidos y recibidos efectivamente (sin contar retransmisiones) por cada parte son los mismos, pero ahora el throughput es mucho menor. Esto se debe a la pérdida de paquetes. Cada vez que un paquete se pierde, TCP necesita retransmitir ese paquete. Esto no solo incluye el tiempo de retransmisión, sino también el tiempo adicional de espera debido al tiempo de ida y vuelta (RTT) antes de que la pérdida sea detectada y el paquete retransmitido. Además, TCP ajusta su ventana de congestión y disminuye la tasa

de envío de datos en respuesta a las pérdidas detectadas, lo que también contribuye a una disminución en la velocidad de transmisión de datos.

Segunda parte

6. Asignación de direccionamiento, configuración del router e interfaces

6.1 - Topología

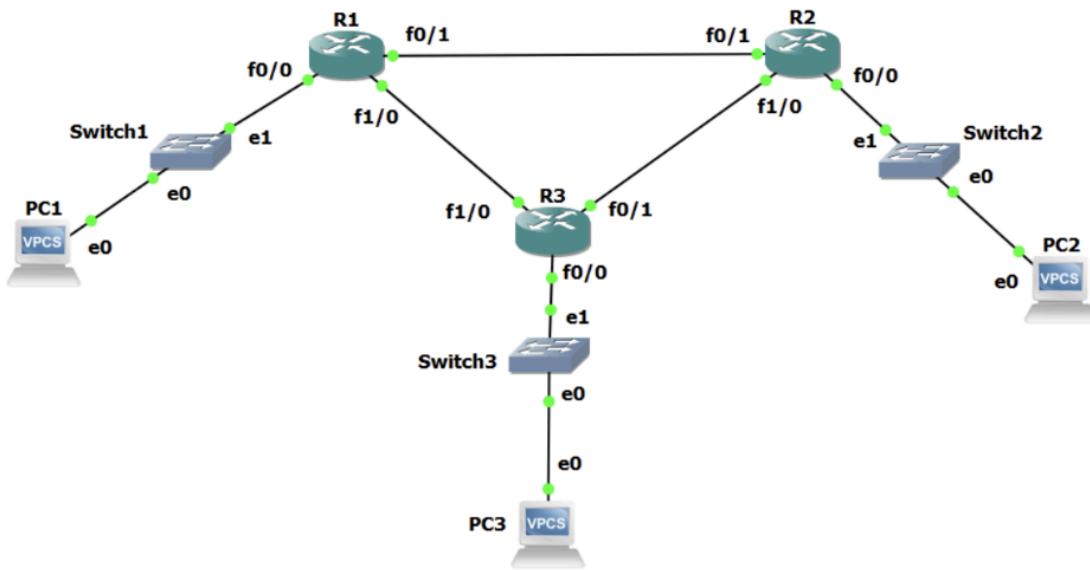
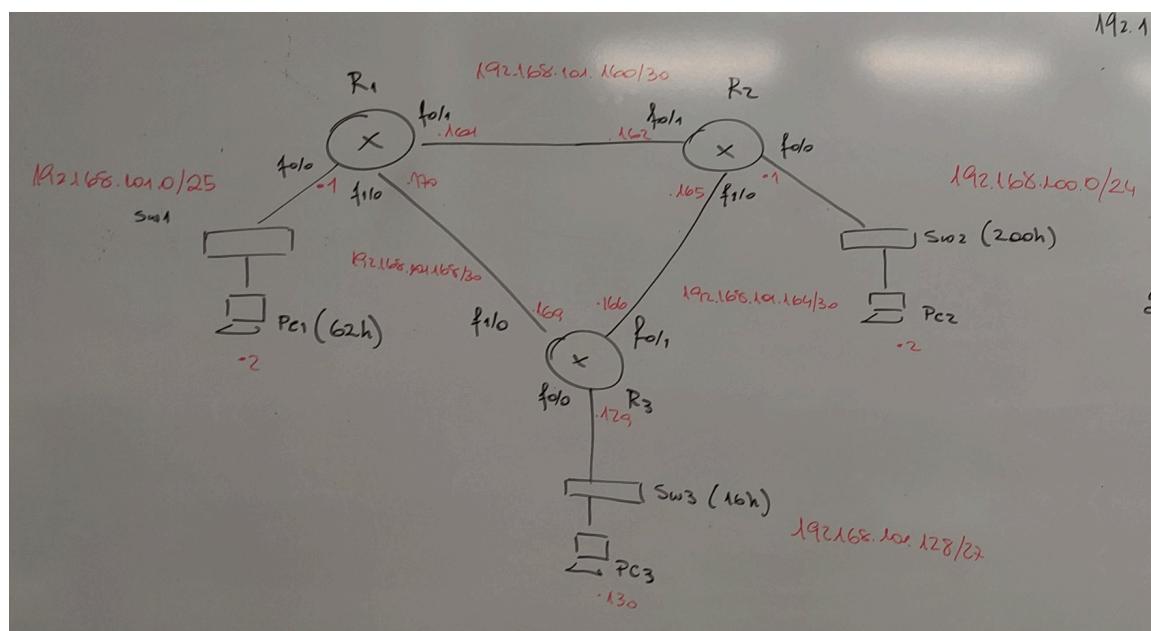


Figura 1: Topología a utilizar



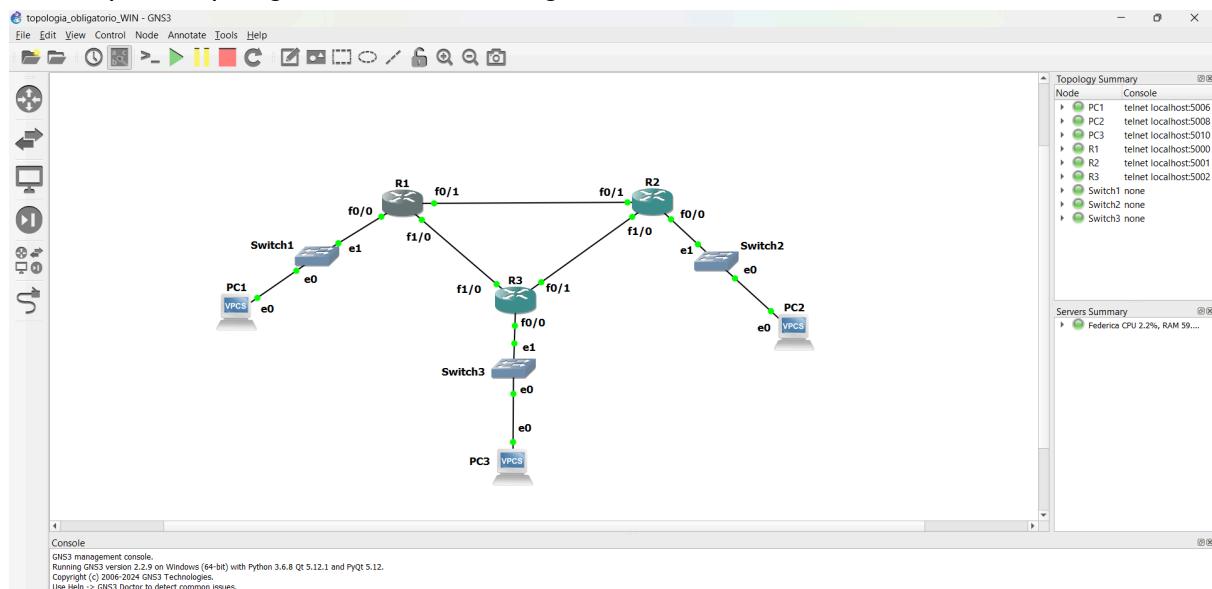
6.2 - Asignación de direcciones IP

1. En base a la topología y a las directivas, complete las siguientes tablas de asignación de direcciones

Enlace	Subred (X.X.X.X/M)	IP Router	IP Router
R1 - R2	192.168.101.160/30	.161	.162
R2 - R3	192.168.101.164/30	.165	.166
R3 - R1	192.168.101.168/30	.169	.170

Enlace	Subred (X.X.X.X/M)	IP Router	IP D. Gateway	IP Broadcast
SW1	192.168.101.0/25	.1	.1	.127
SW2	192.168.100.0/24	.1	.1	.255
SW3	192.168.101.128/27	.129	.129	.159

2. En GNS3 importe el proyecto portable suministrado por el docente de teórico, verá que la topología coincide con la Figura 1.



Se observa que la topología coincide con la de la figura 1.

3. Comience la simulación y despliegue las consolas de los tres (3) routers.

The screenshot shows three separate terminal windows, each representing a Cisco router (R1, R2, and R3). The windows are arranged horizontally. Each window displays a command-line interface with configuration text and copyright information.

R1 Window:

```
Connected to Dynamips VM "R2" (ID 2, type c3745) - Console port
Press ENTER to get the prompt.
.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134-1706
```

R2 Window:

```
Cisco IOS Software, 3700 Software (C3745-ADVIPSERVICESK9-M), Version 12.4(15)T6,
RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2008 by Cisco Systems, Inc.
Compiled Mon 07-Jul-08 12:57 by prod_rel_team
Image text-base: 0x60008930, data-base: 0x634A0000

BIST FAILED...
This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.
```

R3 Window:

```
Connected to Dynamips VM "R2" (ID 2, type c3745) - Console port
Press ENTER to get the prompt.
```

Captura de la ventana con las consolas de los 3 routers.

6.3 - Configuración de interfaces Ethernet

1. *Basándose en la guía de comandos del Anexo, configure las interfaces hacia el switch y hacia los demás routers en cada router. Detalle los comandos utilizados.*

Router 1

Comandos para configurar la interfaz del **R1 al R2**:

- conf t
- int f0/1
- ip address 192.168.101.161 255.255.255.252
- no shutdown

Comandos para configurar la interfaz del **R1 al R3**:

- conf t
- int f1/0
- ip address 192.168.101.170 255.255.255.252
- no shutdown

Comandos para configurar la interfaz del **R1 al SW1**:

- conf t
- int f0/0
- ip address 192.168.101.1 255.255.255.128
- no shutdown

Router 2

Comandos para configurar la interfaz del **R2 al R1**:

- conf t
- int f0/1
- ip address 192.168.101.162 255.255.255.252
- no shutdown

Comandos para configurar la interfaz del **R2 al R3**:

- conf t
- int f1/0
- ip address 192.168.101.165 255.255.255.252
- no shutdown

Comandos para configurar la interfaz del **R2 al SW2**:

- conf t
- int f0/0
- ip address 192.168.100.1 255.255.255.0
- no shutdown

Router 3

Comandos para configurar la interfaz del **R3 al R1**:

- conf t
- int f1/0
- ip address 192.168.101.169 255.255.255.252

- no shutdown

Comandos para configurar la interfaz del **R3 al R2**:

- conf t
- int f0/1
- ip address 192.168.101.166 255.255.255.252
- no shutdown

Comandos para configurar el enlace **R3-SW3**:

- conf t
- int f0/0
- ip address 192.168.101.129 255.255.255.224
- no shutdown

2. Verifique el estado actual de las interfaces. Detalle los comandos utilizados y los resultados obtenidos. Si las interfaces no se encuentran operativas, detalle el porqué y las acciones que deber realizar para que queden operativas.

Usamos el comando *show ip interface brief* para obtener un resumen de las interfaces que configuramos para cada router. Para cada enlace muestra la ip del otro router y detalla si está operativo o no. Si quisieramos ver mas detalles de cada configuración podríamos hacer *show ip interface f0/1* (o cualquier interfaz).

Router 1

```
R1#show ip int brief
Interface          IP-Address      OK? Method Status      Protocol
FastEthernet0/0    192.168.101.1   YES manual up        up
FastEthernet0/1    192.168.101.161 YES manual up        up
FastEthernet1/0    192.168.101.170 YES manual up        up
FastEthernet2/0    unassigned      YES unset administratively down down
```

Router 2

```
R2#show ip int brief
Interface          IP-Address      OK? Method Status      Protocol
FastEthernet0/0    162.168.100.1   YES manual up        up
FastEthernet0/1    192.168.101.162 YES manual up        up
FastEthernet1/0    192.168.101.165 YES manual up        up
FastEthernet2/0    unassigned      YES unset administratively down down
```

Router 3

```
R3#show ip int brief
Interface          IP-Address      OK? Method Status      Protocol
FastEthernet0/0    192.168.101.129 YES manual up        up
FastEthernet0/1    192.168.101.166 YES manual up        up
FastEthernet1/0    192.168.101.169 YES manual up        up
FastEthernet2/0    unassigned      YES unset administratively down down
```

Están todas operativas porque utilizamos el comando *no shutdown* para levantarlas.

3. ¿Cómo vería todas las interfaces que tiene conectadas cada uno de los routers e información sobre cada una de ellas a modo de resumen? Detalle la salida obtenida.

```
R1#show ip interface brief
Interface          IP-Address      OK? Method Status      Protocol
FastEthernet0/0    192.168.101.1   YES manual up       up
FastEthernet0/1    192.168.101.162 YES manual up       up
FastEthernet1/0    192.168.101.169 YES manual up       up
FastEthernet2/0    unassigned      YES unset administratively down down
```

Con el mismo comando que el utilizado en el punto anterior. Como vemos, en todos los routers ya quedaron las interfaces operativas, ya que cada una tiene la IP que habíamos planeado en la topología inicialmente, y además están todas encendidas. También podemos ver que dice que las hemos configurado manualmente.

Parte 4 - Prueba de conectividad

1. Desde la consola de R2, pruebe la conectividad realizando ping a las seis (6) direcciones IP del resto de los routers. ¿Logra tener éxito en todos los casos? ¿Qué falta para que el router logre llegar a todas las direcciones IP? Detalle los comandos y el resultado obtenido.

R1:

```
R2#ping 192.168.101.161

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.101.161, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 28/41/76 ms
```

```
R2#ping 192.168.101.170

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.101.170, timeout is 2 seconds:
..... 
Success rate is 0 percent (0/5)
```

```
R2#ping 192.168.101.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.101.1, timeout is 2 seconds:
..... 
Success rate is 0 percent (0/5)
```

R3:

```
R2#ping 192.168.101.166

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.101.166, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 28/35/52 ms
```

```
R2#ping 192.168.101.169

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.101.169, timeout is 2 seconds:
..... 
Success rate is 0 percent (0/5)
```

```
R2#ping 192.168.101.129

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.101.129, timeout is 2 seconds:
..... 
Success rate is 0 percent (0/5)
```

Solamente llegamos a las interfaces que configuramos que están conectadas directamente. Para llegar a todas las IP faltaría configurar la ruta hacia ellas.

2. *Guarde la configuración de cada uno de los routers. Detalle el comando utilizado.*

NOTA: Cada vez que lo considere necesario durante la práctica puede repetir esta acción para no perder los avances.

Utilizamos el comando `wr` para guardar las configuraciones.

7. Ruteo estático

1. Posicionado en el router 2, genere solamente una ruta estática para alcanzar el enlace p2p, entre R1 y R3. Liste la configuración aplicada. Ahora, realice un ping a las interfaces de ese enlace, ¿logra tener éxito? ¿si, no?, justifique que es lo que sucede.

Para configurar la ruta estática utilizamos los siguientes comandos:

- conf t
- ip route 192.168.101.168 255.255.255.252 192.168.101.161

Realizamos ping a las interfaces .170 y .169 y solamente tenemos éxito con la .170.

El problema con la .169 es que ahora sabemos llegar pero no volver, ya que desde R3 no tenemos configurado el enlace entre R1 y R2. Esto se debe a que los datagramas tienen siempre un origen y destino fijos. Al salir de la 192.168.101.162(R1), con destino 192.168.169(R3), el R2 sabe que debe enviarlo a R1 para llegar. Luego R1 sabe que lo debe mandar a R3(192.168.101.169). Al llegar a este router, intenta iniciar el ping de vuelta, pero el problema es que no sabe a dónde debe redirigir los datagramas para llegar a la IP 192.168.101.162(R1).

A la interfaz .170 sí sabemos llegar y volver ya que el R2 ahora sabe que para llegar a las IPs de esa red debe redirigir al R1. Luego, el R1 cuando lo recibe, sabe como mandarlo de vuelta, ya que está por enlace directo

```
R2#ping 192.168.101.170
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.101.170, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/31/40 ms
```

```
R2#ping 192.168.101.169
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.101.169, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

2. Complete en cada router las rutas estáticas que le permitan llegar a todas las subredes, que hasta el momento no son alcanzables. Detalle los comandos utilizados.

R1:

- conf t
- ip route 192.168.101.164 255.255.255.252 192.168.101.169
- ip route 192.168.100.0 255.255.255.0 192.168.101.162
- ip route 192.168.101.128 255.255.255.224 192.168.101.169

R2:

- conf t
- ip route 192.168.101.0 255.255.255.128 192.168.101.161
- ip route 192.168.101.128 255.255.255.224 192.168.101.166

R3:

- conf t
- ip route 192.168.101.160 255.255.255.252 192.168.101.165
- ip route 192.168.101.0 255.255.255.128 192.168.101.170
- ip route 192.168.100.0 255.255.255.0 192.168.101.165

Verificamos con el comando ping que ahora sí alcanzamos todas las subredes.

3. Verifique el estado de la tabla de ruteo de los routers. Detalle los comandos utilizados y las salidas obtenidas.

```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

S   192.168.100.0/24 [1/0] via 192.168.101.162
    192.168.101.0/24 is variably subnetted, 5 subnets, 3 masks
C     192.168.101.0/25 is directly connected, FastEthernet0/0
S     192.168.101.128/27 [1/0] via 192.168.101.169
C     192.168.101.168/30 is directly connected, FastEthernet1/0
S     192.168.101.164/30 [1/0] via 192.168.101.169
C     192.168.101.160/30 is directly connected, FastEthernet0/1
```

```
R2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C   192.168.100.0/24 is directly connected, FastEthernet0/0
    192.168.101.0/24 is variably subnetted, 5 subnets, 3 masks
S     192.168.101.0/25 [1/0] via 192.168.101.161
S     192.168.101.128/27 [1/0] via 192.168.101.166
S     192.168.101.168/30 [1/0] via 192.168.101.161
C     192.168.101.164/30 is directly connected, FastEthernet1/0
C     192.168.101.160/30 is directly connected, FastEthernet0/1
```

```
R3#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

S   192.168.100.0/24 [1/0] via 192.168.101.165
    192.168.101.0/24 is variably subnetted, 5 subnets, 3 masks
S     192.168.101.0/25 [1/0] via 192.168.101.170
C     192.168.101.128/27 is directly connected, FastEthernet0/0
C     192.168.101.168/30 is directly connected, FastEthernet1/0
C     192.168.101.164/30 is directly connected, FastEthernet0/1
S   192.168.101.160/30 [1/0] via 192.168.101.165
```

4. Desde la consola de R2, pruebe la conectividad realizando ping a las seis (6) direcciones IP del resto de los routers. ¿Logra tener éxito en todos los casos?

R1:

```
R2#ping 192.168.101.161

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.101.161, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/28/32 ms
```

```
R2#ping 192.168.101.170
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.101.170, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/31/32 ms
```

```
R2#ping 192.168.101.1
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.101.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/33/48 ms
```

R3:

```
R2#ping 192.168.101.166

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.101.166, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/28/36 ms
```

```
R2#ping 192.168.101.169
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.101.169, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/44/52 ms

R2#ping 192.168.101.129
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.101.129, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/28/32 ms
```

Tuvimos éxito en todos los casos, ya que todos los routers saben a cual redirigir un datagrama según la red destino a la que pertenece.

5. *Si quisiera optimizar la tabla de rutas, ¿tendría alguna forma de implementarla?. ¿Qué cambiaría?*

En este caso, hallamos que una summarización es complicada porque las subredes utilizadas no permiten una summarización efectiva. La distribución de las subredes entre los routers y las conexiones punto a punto requieren rutas específicas para cada subred, lo que complica la posibilidad de sumarizar.

Por lo tanto, optamos por la implementación de una ruta por default en R1 que apunte a R2. Esto simplifica la tabla de rutas en R1, que solo necesita conocer las rutas específicas a las redes que no pasan por R2. Esto nos permite eliminar de su tabla de rutas la red R2-R3 y la red LAN de R2.

```
C      192.168.101.0/25 is directly connected, FastEthernet0/0
S      192.168.101.128/27 [1/0] via 192.168.101.169
C      192.168.101.168/30 is directly connected, FastEthernet1/0
C      192.168.101.160/30 is directly connected, FastEthernet0/1
S*    0.0.0.0/0 [1/0] via 192.168.101.162
```

A su vez, sólo se configura en este router, ya que agregarlo a los demás podría generar loops en los envíos de datagramas. Por lo que ahora nuestra tabla de ruteo tiene 5 entradas en vez de 6.

8. Ruteo dinámico

8.1 - Protocolo RIP

1. *En esta tarea se configura el ruteo dinámico en cada router utilizando para ello el protocolo RIP. Utilice la guía de comandos del Anexo y detalle los comandos que ingresó.*

R1:

- enable
- conf t
- router rip
- version 2
- network 192.168.101.0
- network 192.168.101.160
- network 192.168.101.168
- passive-interface f0/0

R2:

- enable
- conf t
- router rip
- version 2
- network 192.168.100.0
- network 192.168.101.160
- network 192.168.101.164
- passive-interface f0/0

R3:

- enable
- conf t
- router rip
- version 2
- network 192.168.101.128
- network 192.168.101.164
- network 192.168.101.168
- passive-interface f0/0

2. *¿Puede ver las rutas configuradas vía RIP en la tabla de ruteo? ¿Y las rutas estáticas? ¿Por qué? Detalle el comando utilizado para visualizar las rutas y la salida obtenida en cada uno de los routers.*

Las rutas configuradas vía RIP no las vemos en la tabla de ruteo. Esto se debe a que todas las rutas que acabamos de configurar vía RIP ya las habíamos configurado de manera estática o directa y estas últimas dos tienen una distancia administrativa menor que RIP, por lo tanto son de mayor prioridad. Específicamente las directas tienen distancia 0, las estáticas distancia 1 y RIP tiene distancia 120.

A modo de ejemplo, se muestra la salida del comando `show ip route` al ejecutarlo en el router 1.

```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

S    192.168.100.0/24 [1/0] via 192.168.101.162
      192.168.101.0/24 is variably subnetted, 5 subnets, 3 masks
C      192.168.101.0/25 is directly connected, FastEthernet0/0
S      192.168.101.128/27 [1/0] via 192.168.101.169
C      192.168.101.168/30 is directly connected, FastEthernet1/0
S      192.168.101.164/30 [1/0] via 192.168.101.169
C      192.168.101.160/30 is directly connected, FastEthernet0/1
```

Se puede ver que la salida es igual a la del punto 3 de *Ruteo estático*. Los códigos a la izquierda indican cómo se aprendió la ruta, donde S significa que es una ruta estática configurada manualmente y C indica una ruta directamente conectada, si hubiera alguna ruta aprendida vía RIP se mostraría con una R.

Las capturas de los otros routers no se incluyen en esta parte por ser iguales a las del punto 3 de *Ruteo estático*.

3. *¿Cómo haría para que se utilizarán únicamente las rutas dinámicas para encaminar los paquetes? Detalle los comandos utilizados.*

Debemos eliminar las rutas estáticas

R1:

- conf t
- no ip route 192.168.101.164 255.255.255.252 192.168.101.169
- no ip route 192.168.100.0 255.255.255.0 192.168.101.162
- no ip route 192.168.101.128 255.255.255.224 192.168.101.169

R2:

- conf t
- no ip route 192.168.101.0 255.255.255.128 192.168.101.161
- no ip route 192.168.101.128 255.255.255.224 192.168.101.166
- no ip route 192.168.101.168 255.255.255.252 192.168.101.161

R3:

- conf t
- no ip route 192.168.101.160 255.255.255.252 192.168.101.165
- no ip route 192.168.101.0 255.255.255.128 192.168.101.170
- no ip route 192.168.100.0 255.255.255.0 192.168.101.165

4. ¿Cuál es la distancia administrativa y la métrica en cada ruta aprendida por RIP?
 ¿Dónde y con qué comando se puede observar esto?

La distancia administrativa en RIP es fija y es 120 y la métrica es la cantidad de saltos hasta el destino.

Podemos ver esto utilizando el comando `show ip route`

```
R2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.100.0/24 is directly connected, FastEthernet0/0
     192.168.101.0/24 is variably subnetted, 5 subnets, 3 masks
R      192.168.101.0/25
          [120/1] via 192.168.101.161, 00:00:26, FastEthernet0/1
R      192.168.101.128/27
          [120/1] via 192.168.101.166, 00:00:21, FastEthernet1/0
R      192.168.101.168/30
          [120/1] via 192.168.101.166, 00:00:21, FastEthernet1/0
          [120/1] via 192.168.101.161, 00:00:26, FastEthernet0/1
C      192.168.101.164/30 is directly connected, FastEthernet1/0
C      192.168.101.160/30 is directly connected, FastEthernet0/1
```

Junto a la rutas aprendidas vía RIP el [120/1] significa que la DA es 120 y la cantidad de saltos es 1.

5. Desde la consola de R2, pruebe la conectividad realizando ping a las seis (6) direcciones IP del resto de los routers. ¿Logra tener éxito en todos los casos?

R1:

```
R2#ping 192.168.101.161
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.101.161, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/27/32 ms
R2#ping 192.168.101.170
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.101.170, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/30/36 ms
R2#ping 192.168.101.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.101.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/27/32 ms
```

R3:

```
R2#ping 192.168.101.166
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.101.166, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/28/32 ms
R2#ping 192.168.101.169
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.101.169, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/45/60 ms
R2#ping 192.168.101.129
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.101.129, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/28/36 ms
```

Llegamos a todas las IP.

6. *Baje la interfaz entre R1 - R3, utilizando el comando shutdown dentro de la misma. ¿Qué comportamiento observa? ¿El protocolo reacciona al cambio? Describa los comandos que utilizó para responder y las salidas obtenidas. Vuelva a levantar la interfaz luego de realizado el ejercicio.*

Bajamos la interfaz desde el router 1 utilizando los siguientes comandos:

- conf t
- int f1/0 (para seleccionar la interfaz con el router 3)
- shutdown

Esperamos un poco para revisar las tablas de enrutamiento después de dar de baja una interfaz en RIP debido a la naturaleza periódica y los mecanismos de temporización del protocolo RIP. RIP envía actualizaciones de enrutamiento cada 30 segundos, y cuando una interfaz se da de baja, el router afectado marca la ruta correspondiente como inalcanzable (con una métrica de 16) en su tabla de enrutamiento. Esta información se propaga a los routers vecinos en la próxima actualización periódica. Además, RIP utiliza un temporizador de espera (holddown timer) para evitar inestabilidades y asegurar la coherencia de la red, lo que añade un retraso adicional. Por lo tanto, para asegurarnos de que todos los routers en la red hayan recibido y procesado la actualización, es prudente esperar un tiempo antes de revisar las tablas de enrutamiento para confirmar que la caída de la interfaz ha sido correctamente propagada y registrada en toda la red.

Luego, para ver si hubo cambios ejecutamos *show ip route* en todos los routers para ver sus tablas de enrutamiento.

En R1 podemos ver que aumentaron a 2 la cantidad de saltos necesarios para llegar a la 192.168.101.128/27. Esto es porque al bajar la interfaz, ahora R1 no puede llegar directamente a R3, lo que hace que tenga los datagramas de intercambio tengan que hacer el camino R1-R2-R3, aumentando el número de saltos a dos y redirigiendo todos los paquetes que iban al R3 directamente, al R2. También aumentó a dos saltos los necesarios para llegar a la red que estaba entre R1 y R3 192.168.101.168/30, ya que ahora tiene que

pasar por R2 y luego por R3 para llegar a la misma. Esto lo podemos deducir porque esta haciendo más saltos de lo normal.

```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

R    192.168.100.0/24 [120/1] via 192.168.101.162, 00:00:21, FastEthernet0/1
     192.168.101.0/24 is variably subnetted, 5 subnets, 3 masks
C    192.168.101.0/25 is directly connected, FastEthernet0/0
R    192.168.101.128/27
     [120/2] via 192.168.101.162, 00:00:21, FastEthernet0/1
R    192.168.101.168/30
     [120/2] via 192.168.101.162, 00:00:21, FastEthernet0/1
R    192.168.101.164/30
     [120/1] via 192.168.101.162, 00:00:21, FastEthernet0/1
C    192.168.101.160/30 is directly connected, FastEthernet0/1
```

Luego, en el R2 podemos ver que ahora ya no tiene dos caminos de igual costo para llegar a la red 192.168.101.168/30 (la red entre R1 y R3) ya que al dar de baja la interfaz de R1 a R3, la única forma llegar a esta red es que R2 vaya a R3.

```
R2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.100.0/24 is directly connected, FastEthernet0/0
     192.168.101.0/24 is variably subnetted, 5 subnets, 3 masks
R    192.168.101.0/25
     [120/1] via 192.168.101.161, 00:00:21, FastEthernet0/1
R    192.168.101.128/27
     [120/1] via 192.168.101.166, 00:00:26, FastEthernet1/0
R    192.168.101.168/30
     [120/1] via 192.168.101.166, 00:00:26, FastEthernet1/0
C    192.168.101.164/30 is directly connected, FastEthernet1/0
C    192.168.101.160/30 is directly connected, FastEthernet0/1
```

A su vez, el R3 ahora para llegar a la lan de R1 necesita hacer dos saltos, debido a que no puede ir directamente a R1. Tiene que pasar primero por R2 y luego llegar a R1. (Camino R3-R2-R1)

```
R3#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

R    192.168.100.0/24 [120/1] via 192.168.101.165, 00:00:18, FastEthernet0/1
      192.168.101.0/24 is variably subnetted, 5 subnets, 3 masks
R        192.168.101.0/25
          [120/2] via 192.168.101.165, 00:00:18, FastEthernet0/1
C        192.168.101.128/27 is directly connected, FastEthernet0/0
C        192.168.101.168/30 is directly connected, FastEthernet1/0
C        192.168.101.164/30 is directly connected, FastEthernet0/1
R        192.168.101.160/30
          [120/1] via 192.168.101.165, 00:00:18, FastEthernet0/1
```

8.2 - Protocolo OSPF

- Configure OSPF en todos los routers. Asegúrese de que el protocolo quede activo en todas las interfaces. Detalle los comandos utilizados.

Para configurar OSPF, usamos los siguientes comandos. Para cada router, se tiene que indicar las redes a las que tiene interfaces directamente conectadas:

R1:

```
configure terminal
router ospf 10
network 192.168.101.160 0.0.0.3 area 0
network 192.168.101.168 0.0.0.3 area 0
network 192.168.101.0 0.0.0.127 area 0
```

R2:

```
configure terminal
router ospf 10
network 192.168.101.160 0.0.0.3 area 0
network 192.168.101.164 0.0.0.3 area 0
network 192.168.100.0 0.0.0.255 area 0
```

R3:

```
configure terminal
router ospf 10
network 192.168.101.164 0.0.0.3 area 0
network 192.168.101.168 0.0.0.3 area 0
network 192.168.101.128 0.0.0.31 area 0
```

Aseguremonos que quedó activo:

```
R1#show ip route ospf
0    192.168.100.0/24 [110/20] via 192.168.101.162, 00:02:06, FastEthernet0/1
      192.168.101.0/24 is variably subnetted, 5 subnets, 3 masks
0        192.168.101.128/27
          [110/11] via 192.168.101.169, 00:00:32, FastEthernet1/0
0        192.168.101.164/30
          [110/11] via 192.168.101.169, 00:01:06, FastEthernet1/0
          [110/11] via 192.168.101.162, 00:03:06, FastEthernet0/1
R1#
```

```
R2#show ip route ospf
      192.168.101.0/24 is variably subnetted, 5 subnets, 3 masks
0        192.168.101.0/25
          [110/12] via 192.168.101.166, 00:01:11, FastEthernet1/0
0        192.168.101.128/27
          [110/11] via 192.168.101.166, 00:00:37, FastEthernet1/0
0        192.168.101.168/30
          [110/2] via 192.168.101.166, 00:01:11, FastEthernet1/0
R2#
```

```
R3#show ip route ospf
0    192.168.100.0/24 [110/20] via 192.168.101.165, 00:01:19, FastEthernet0/1
      192.168.101.0/24 is variably subnetted, 5 subnets, 3 masks
0        192.168.101.0/25
          [110/11] via 192.168.101.170, 00:01:00, FastEthernet1/0
0        192.168.101.160/30
          [110/11] via 192.168.101.170, 00:01:00, FastEthernet1/0
R3#
```

El comando show ip route ospf nos muestra las rutas que el router ha aprendido por OSPF. Vemos que tienen el prefijo 0 (OSPF) y además, que en los 3 routers se muestra que han aprendido por OSPF como llegar a las 3 redes que no tienen directamente conectadas.

2. ¿Qué rutas aprendió el router? ¿Por qué visualiza estas rutas? ¿Qué ocurrió con las rutas aprendidas por RIP? ¿Por qué?

En el punto anterior ya mostramos las rutas que aprendió.

El router visualiza estas rutas porque las ha aprendido a través del protocolo OSPF. OSPF es un protocolo de enrutamiento dinámico que permite a los routers intercambiar información sobre la topología de la red y aprender sobre las rutas disponibles. Las rutas visualizadas son aquellas anunciadas por los routers vecinos a través de sus interfaces OSPF. Las demás no se visualizan porque son las directamente conectadas, que no son las que se han aprendido por OSPF, sino las que nosotros le indicamos a OSPF que utilice para trabajar.

¿Qué ocurrió con las rutas por RIP?

Hagamos show ip route para ver como quedó la tabla de rutas, del R1 por ejemplo:

```
Gateway of last resort is not set

0    192.168.100.0/24 [110/20] via 192.168.101.162, 00:15:17, FastEthernet0/1
     192.168.101.0/24 is variably subnetted, 5 subnets, 3 masks
C      192.168.101.0/25 is directly connected, FastEthernet0/0
0      192.168.101.128/27
          [110/11] via 192.168.101.169, 00:13:43, FastEthernet1/0
C      192.168.101.168/30 is directly connected, FastEthernet1/0
0      192.168.101.164/30
          [110/11] via 192.168.101.169, 00:14:18, FastEthernet1/0
          [110/11] via 192.168.101.162, 00:16:18, FastEthernet0/1
C      192.168.101.160/30 is directly connected, FastEthernet0/1
R1#
```

Cuando se cambia la configuración de un router de RIP a OSPF, las rutas aprendidas por RIP pueden desaparecer de la tabla de enrutamiento activa. Esto ocurre porque OSPF tiene una distancia administrativa (AD) más baja que RIP, lo que hace que el router prefiera las rutas OSPF sobre las rutas RIP. Concretamente, RIP como mencionamos anteriormente tiene una DA de 120, mientras que OSPF tiene 110.

La distancia administrativa es una métrica utilizada por los routers para seleccionar la mejor ruta cuando existen múltiples rutas hacia un destino desde diferentes protocolos de enrutamiento. Una distancia administrativa más baja indica una ruta más confiable. Como OSPF tiene una AD más baja (110) en comparación con RIP (120), el router prefiere las rutas aprendidas a través de OSPF y descarta las rutas RIP cuando ambas están disponibles para el mismo destino.

3. Desde la consola de R2, pruebe la conectividad realizando ping a las seis (6) direcciones IP del resto de los routers. ¿Logra tener éxito en todos los casos?

Ping a las direcciones de R1:

```
R2#ping 192.168.101.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.101.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/42/60 ms
R2#ping 192.168.101.170

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.101.170, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/52/64 ms
R2#ping 192.168.101.161

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.101.161, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/28/32 ms
```

Ping a las direcciones de R3:

```
R2#ping 192.168.101.166
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.101.166, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/25/32 ms
R2#ping 192.168.101.169

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.101.169, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/28/32 ms
R2#ping 192.168.101.129

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.101.129, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/27/32 ms
R2#
```

Como vemos, tenemos éxito llegando a todas.

4. ¿Cuál es la distancia administrativa ahora? ¿Es la misma para todas las rutas?

Como dijimos anteriormente, la distancia administrativa ahora es de 110 para las redes que no tienen interfaces conectadas directamente a cada router. Ya mostramos la de R2, pero mostremos la de R1 también para ilustrar el ejemplo:

```
0    192.168.100.0/24 [110/20] via 192.168.101.162, 00:25:30, FastEthernet0/1
     192.168.101.0/24 is variably subnetted, 5 subnets, 3 masks
C      192.168.101.0/25 is directly connected, FastEthernet0/0
0      192.168.101.128/27
          [110/11] via 192.168.101.169, 00:23:56, FastEthernet1/0
C      192.168.101.168/30 is directly connected, FastEthernet1/0
0      192.168.101.164/30
          [110/11] via 192.168.101.169, 00:24:30, FastEthernet1/0
          [110/11] via 192.168.101.162, 00:26:30, FastEthernet0/1
C      192.168.101.160/30 is directly connected, FastEthernet0/1
```

Las redes que se determinó su ruteo por OSPF, tienen 110 de DA como se dijo antes, pero las 3 directamente conectadas(C), siguen con DA 0, ya que no deben ser calculadas por OSPF porque tienen enlace directo.

5. *¿Y cuál es la métrica en cada ruta aprendida? Compare con la métrica que tenían las rutas aprendidas por RIP. ¿Hay diferencia? ¿Por qué? ¿Son comparables las métricas?*

La métrica en OSPF se basa en el costo, que se calcula utilizando el ancho de banda de las interfaces. La fórmula para calcular el costo en OSPF es: (10 a la 8)/Ancho de banda en bits por segundo.

Como se observa en la imagen anterior (show ip route ospf de R1), el costo de los enlaces para llegar a las redes es de 11 en todos los casos, excepto para llegar a la LAN de R2, que tiene un costo de 20. Esto puede deberse a que la interfaz hacia la LAN de R2 tiene un ancho de banda menor en comparación con las otras interfaces, resultando en un costo mayor.

Las métricas de OSPF y RIP no son directamente comparables porque se basan en diferentes criterios. OSPF utiliza el ancho de banda de los enlaces para calcular el costo, mientras que RIP utiliza el número de saltos (hops). Debido a estos diferentes enfoques, OSPF suele ser más preciso y eficiente en redes con enlaces de diferentes capacidades, mientras que RIP es más simple y adecuado para redes pequeñas y homogéneas.

6. *Pruebe cambiar ahora el parámetro bandwidth del enlace R1 - R3. ¿Qué ocurre con la métrica de las rutas? ¿Qué ruta siguen ahora los paquetes? ¿Por qué?*

Veamos cómo está la métrica de las rutas actualmente:

```

0  192.168.100.0/24 [110/20] via 192.168.101.162, 00:25:30, FastEthernet0/1
    192.168.101.0/24 is variably subnetted, 5 subnets, 3 masks
C    192.168.101.0/25 is directly connected, FastEthernet0/0
0    192.168.101.128/27
        [110/11] via 192.168.101.169, 00:23:56, FastEthernet1/0
C    192.168.101.168/30 is directly connected, FastEthernet1/0
0    192.168.101.164/30
        [110/11] via 192.168.101.169, 00:24:30, FastEthernet1/0
        [110/11] via 192.168.101.162, 00:26:30, FastEthernet0/1
C    192.168.101.160/30 is directly connected, FastEthernet0/1

```

Ahora, cambiemos el ancho de banda entre R1 y R3, para esto en R1 hagamos:

```

configure terminal
interface f1/0
bandwidth 10000

```

Ahora hagamos un show ip route en R1:

```

Gateway of last resort is not set

O    192.168.100.0/24 [110/20] via 192.168.101.162, 00:50:52, FastEthernet0/1
      192.168.101.0/24 is variably subnetted, 5 subnets, 3 masks
C      192.168.101.0/25 is directly connected, FastEthernet0/0
O      192.168.101.128/27
          [110/20] via 192.168.101.169, 00:00:35, FastEthernet1/0
C      192.168.101.168/30 is directly connected, FastEthernet1/0
O      192.168.101.164/30
          [110/11] via 192.168.101.162, 00:51:52, FastEthernet0/1
C      192.168.101.160/30 is directly connected, FastEthernet0/1
R1#■

```

Como vemos, ahora la métrica para la red 192.168.101.128/27 pasó de 11 a 20. Esto indica que el enlace tiene un menor ancho de banda que antes. Sin embargo, el router sigue prefiriendo llegar a esta red por este camino ya que está a solo un router de distancia y el costo de llegar por estos enlaces sigue siendo el menor.

Antes del cambio, los paquetes que se dirigían a la red 192.168.101.164/30 tenían dos caminos posibles con el mismo costo, por lo que se alternaban entre ellos. Después del cambio, los paquetes que se dirigen a 192.168.101.164/30 ahora prefieren la ruta a través de 192.168.101.162 (FastEthernet0/1) en lugar de 192.168.101.169 (FastEthernet1/0). Esto ocurre porque la métrica de la ruta a través de 192.168.101.169 aumentó a 20 debido al cambio en el ancho de banda mencionado anteriormente.

O sea que, el cambio en el ancho de banda de la interfaz FastEthernet1/0 afecta las métricas de las rutas aprendidas por OSPF y hace que el protocolo ajuste las rutas preferidas en función de estos cambios de costo.

Estos cambios no hubieran aparecido si con RIP cambiamos el ancho de banda, ya que RIP como se dijo no lo tiene en cuenta.

7. *Investigue que sucede con el protocolo RIP, sigue funcionando o dejó de hacerlo al aprovisionar OSPF. Demuestre su respuesta.*

```

R1#show ip route rip
R1#show running-config | section rip
router rip
  version 2
  passive-interface FastEthernet0/0
  network 192.168.101.0
  ■

```

Al investigar el estado del protocolo RIP después de introducir ruteo con OSPF, encontramos que RIP sigue configurado en el router. Esto se puede observar en la salida del comando `show running-config | section rip`, que muestra la configuración de RIP, incluyendo la versión y las redes configuradas. Sin embargo, al ejecutar el comando `show ip route rip`, no se muestran rutas aprendidas por RIP. La ausencia de rutas RIP en la tabla de enrutamiento sugiere que las rutas obtenidas por RIP no están siendo utilizadas, debido a que OSPF ya está funcionando y tiene menor DA.

Pero esto no quiere decir que RIP haya dejado de funcionar, sino que simplemente, sus rutas no serán las preferidas para el enrutamiento de los paquetes en la red.

¿Por qué razón es bueno que a pesar de usar OSPF, RIP siga corriendo por detrás?

La razón más válida es para generar una redundancia o sea que si OSPF falla o se detiene por alguna razón, RIP será la alternativa y ya está funcionando, por lo que no se generarán errores en la red.

9. Protocolo ARP

- En el escenario del punto anterior, configure en la PC2 alguna de las direcciones IP disponibles para su red de área local y el router R2 como gateway por defecto.
Análogamente, configure en la PC3 alguna de las direcciones IP disponibles para su red de área local y el router R3 como gateway por defecto. Detalle los comandos utilizados en cada PC.

Configuramos el PC2 primero y luego verificamos con el comando show ip:

```
PC2> ip 192.168.100.2 255.255.255.0 192.168.100.1
Checking for duplicate address...
PC1 : 192.168.100.2 255.255.255.0 gateway 192.168.100.1

PC2> show ip

NAME      : PC2[1]
IP/MASK   : 192.168.100.2/24
GATEWAY   : 192.168.100.1
DNS       :
MAC       : 00:50:79:66:68:01
LPORT     : 10038
RHOST:PORT: 127.0.0.1:10039
MTU:      : 1500
```

Ahora el PC3:

```
PC3> ip 192.168.101.130 255.255.255.224 192.168.101.129
Checking for duplicate address...
PC1 : 192.168.101.130 255.255.255.224 gateway 192.168.101.129

PC3> show ip

NAME      : PC3[1]
IP/MASK   : 192.168.101.130/27
GATEWAY   : 192.168.101.129
DNS       :
MAC       : 00:50:79:66:68:02
LPORT     : 10040
RHOST:PORT: 127.0.0.1:10041
MTU:      : 1500
```

- Realice una prueba de conectividad mediante ping desde la PC2 hacia su gateway por defecto, el router R2. Detalle la respuesta obtenida y el estado de la tabla ARP en el router R2.

```
PC2> ping 192.168.100.1
84 bytes from 192.168.100.1 icmp_seq=1 ttl=255 time=15.726 ms
84 bytes from 192.168.100.1 icmp_seq=2 ttl=255 time=15.255 ms
84 bytes from 192.168.100.1 icmp_seq=3 ttl=255 time=15.166 ms
84 bytes from 192.168.100.1 icmp_seq=4 ttl=255 time=15.382 ms
84 bytes from 192.168.100.1 icmp_seq=5 ttl=255 time=15.304 ms
```

El ping es exitoso.

Ahora veamos la tabla ARP en R2.

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	192.168.100.1	-	c402.5b58.0000	ARPA	FastEthernet0/0
Internet	192.168.100.2	1	0050.7966.6801	ARPA	FastEthernet0/0
Internet	192.168.101.161	105	c401.34fc.0001	ARPA	FastEthernet0/1
Internet	192.168.101.162	-	c402.5b58.0001	ARPA	FastEthernet0/1
Internet	192.168.101.165	-	c402.5b58.0010	ARPA	FastEthernet1/0
Internet	192.168.101.166	105	c403.6444.0001	ARPA	FastEthernet1/0

La tabla muestra que la dirección IP del PC2 está correctamente asociada con su dirección MAC, indicando que la resolución ARP funciona correctamente y la conectividad entre el PC2 y R2 está establecida. La tabla ARP del router guardó la dirección MAC del PC2 debido a que antes de poder mandar un ping, el PC2 necesita saber quién (qué dirección MAC) tiene la IP del default gateway en la topología.

Entonces, envía el mensaje de ARP request en broadcast, lo que hace que el R2 la reciba, y por tanto guarde en su tabla ARP que el PC2 con la IP que envía el datagrama tiene esa MAC. Luego, envía la respuesta ARP lo que hace que el PC2 guarde la MAC del R2, la razón por la cual inició esta comunicación ARP.

3. Realice una prueba de conectividad mediante ping desde la PC2 hacia la PC3.

Detalle la respuesta obtenida y el estado de la tabla ARP en el router R2.

```
PC2> ping 192.168.101.130
84 bytes from 192.168.101.130 icmp_seq=1 ttl=62 time=61.028 ms
84 bytes from 192.168.101.130 icmp_seq=2 ttl=62 time=60.451 ms
84 bytes from 192.168.101.130 icmp_seq=3 ttl=62 time=60.049 ms
84 bytes from 192.168.101.130 icmp_seq=4 ttl=62 time=60.723 ms
84 bytes from 192.168.101.130 icmp_seq=5 ttl=62 time=61.374 ms
```

El ping es correcto.

Tabla ARP de R2:

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	192.168.100.1	-	c402.5b58.0000	ARPA	FastEthernet0/0
Internet	192.168.100.2	1	0050.7966.6801	ARPA	FastEthernet0/0
Internet	192.168.101.161	110	c401.34fc.0001	ARPA	FastEthernet0/1
Internet	192.168.101.162	-	c402.5b58.0001	ARPA	FastEthernet0/1
Internet	192.168.101.165	-	c402.5b58.0010	ARPA	FastEthernet1/0
Internet	192.168.101.166	110	c403.6444.0001	ARPA	FastEthernet1/0

Como vemos, esta acción no actualiza la tabla ARP de R2. La razón de esto es que como el PC3 está en otra red y no está conectado directamente a alguna interfaz de R2, el router no necesita saber la dirección MAC del PC3. Como la dirección MAC cambia entre nodos, al router no le interesa saber la MAC de un nodo(en este caso PC3) que no esté en los dominios de broadcast a los que el mismo pertenece. Solo necesita saber la MAC del siguiente salto que hará el ping, que es ir a R3.

Algo a aclarar, es que las MAC de las interfaces que pertenecen a R2 ya están en su tabla siempre, por eso no tienen un "age". Luego las interfaces directamente conectadas de otros routers que son la 192.168.101.161 y la 192.168.101.166, ya estaban registradas ya que los pasos que hemos hecho antes para configurar las interfaces y tablas de enrutamiento seguramente generaron actualizaciones en la tabla ARP.

4. ¿Encuentra una asociación MAC Address - IP para la PC3 en la tabla ARP del router R2? ¿Por qué?

No se encuentra por lo mencionado en el punto anterior, la tabla ARP de R2 no necesita tener un registro de la MAC del PC3 ya que no es un nodo que esté conectado directamente por enlace a él.

5. *Bajo el supuesto de que PC2 conoce la MAC de PC3, qué direccionamiento de capa dos (2), origen y destino, tendría un ping hacia PC3 desde PC2 en todo el trayecto. Justifique cada respuesta.*

Ya que se completaron los registros de las tablas ARP necesarios para este intercambio en los puntos anteriores:

IDA DEL PING:

Primero el PC2 empaqueta el ping en un datagrama, con la dirección ip origen del PC2 y la dirección IP destino del PC3. Como sabe que esta IP está en otra red, se la envía al DG, por lo que busca en su tabla ARP la asociación IP del DG con MAC del DG, y empaqueta el datagrama en una trama con origen la MAC del PC2 y destino la MAC que acaba de encontrar en su tabla, o sea la del DG.

A continuación, el R2 recibe la trama, verifica que la dirección MAC destino sea la suya, y lo es, por lo que la desencapsula para obtener el datagrama para analizar la dirección IP destino de la misma. Al ver que se trata del PC3, se fija en su tabla de ruteo y decide enviarlo al R3, por la interfaz que tiene directamente conectada. Busca en sus tablas de ARP (tiene una para cada interfaz) que dirección MAC tiene la IP de la interfaz de R3 conectada a R2. Por lo tanto, vuelve a encapsular el mismo datagrama en una nueva trama, esta vez con la dirección MAC origen de la interfaz FastEthernet1/0 de R2, y destino la dirección MAC de la interfaz FastEthernet0/1 de R3.

R3 recibe la trama, se fija que la dirección MAC destino sea la suya, lo es, por lo cual lo desencapsula, lee el datagrama, ve la dirección IP del PC3, se fija en su tabla de enrutamiento y en base a esto decide enviar el datagrama ya al PC3 porque está directamente conectado a él. Para hacerlo, se fija en su tabla ARP que interfaz pertenece a esa red(al igual que en el router anterior) que es la FastEthernet0/0 , se fija también en la tabla ARP la dirección MAC de la IP a la que finalmente quiere llegar, crea del datagrama con estos origen y destino respectivamente, y lo envía.

VUELTA DEL PING:

El PC3 luego de validar el destino y desencapsular, procesa el ping y decide devolverlo, por lo que crea un nuevo datagrama donde el origen es la IP del PC3 y el destino es la IP del PC2. Ya vimos en la ida del PING, que este origen y destino no cambia en todo el recorrido. Entonces se fija que este PC3 no está en su red, por lo que lo envía al DG(R3). Para hacer esto, busca la IP del DG en su tabla ARP, para encontrar qué MAC tiene asociada. Una vez la encuentra, encapsula el datagrama en la trama con origen la MAC del PC3 y destino la MAC del DG.

A continuación, el R3 recibe la trama, verifica que la dirección MAC destino sea la suya, y lo es, por lo que la desencapsula para obtener el datagrama para analizar la dirección IP

destino de la misma. Al ver que se trata del PC2, se fija en su tabla de ruteo y decide enviarlo al R2, por la interfaz que tiene directamente conectada. Busca en sus tablas de ARP (tiene una para cada interfaz) que dirección MAC tiene la IP de la interfaz de R2 conectada a R3. Por lo tanto, vuelve a encapsular el mismo datagrama en una nueva trama, esta vez con la dirección MAC origen de la interfaz FastEthernet0/1 de R3, y destino la dirección MAC de la interfaz FastEthernet1/0 de R2.

R2 recibe la trama, se fija que la dirección MAC destino sea la suya, lo es, por lo cual lo desencapsula, lee el datagrama, ve la dirección IP del PC2, se fija en su tabla de enrutamiento y en base a esto decide enviar el datagrama ya al PC2 porque está directamente conectado a él. Para hacerlo, se fija en su tabla ARP que interfaz pertenece a esa red(al igual que en el router anterior) que es la FastEthernet0/0 , se fija también en la tabla ARP la dirección MAC de la IP a la que finalmente quiere llegar, crea del datagrama con estos origen y destino respectivamente, y lo envía.

El PC2 recibe la trama y se fija que las dirección destino sea la suya, la desencapsula, lee el datagrama, ve que es el destino por lo cual da por finalizado el PING.

Como conclusión, podemos ver que agregar esta dirección MAC del PC3 a la tabla ARP del PC2 no cambia nada, ya que la primera trama se envía al DG debido a que el lugar del próximo salto de una trama se empieza procesando desde la capa de red, y al ser una IP de otra red, nunca buscará al PC3 en la su tabla de ARP para crear la trama.