Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza

dar-pr-1

Diseño y administración de redes

Autor 1:	Toral Pallás, Héctor - 798095
Autor 2:	Lahoz Bernad, Fernando - 800989
Autor 3:	Martínez Lahoz, Sergio - 801621
Grado:	Ingeniería Informática
Curso:	2023-2024

9 de octubre de 2023

Índice

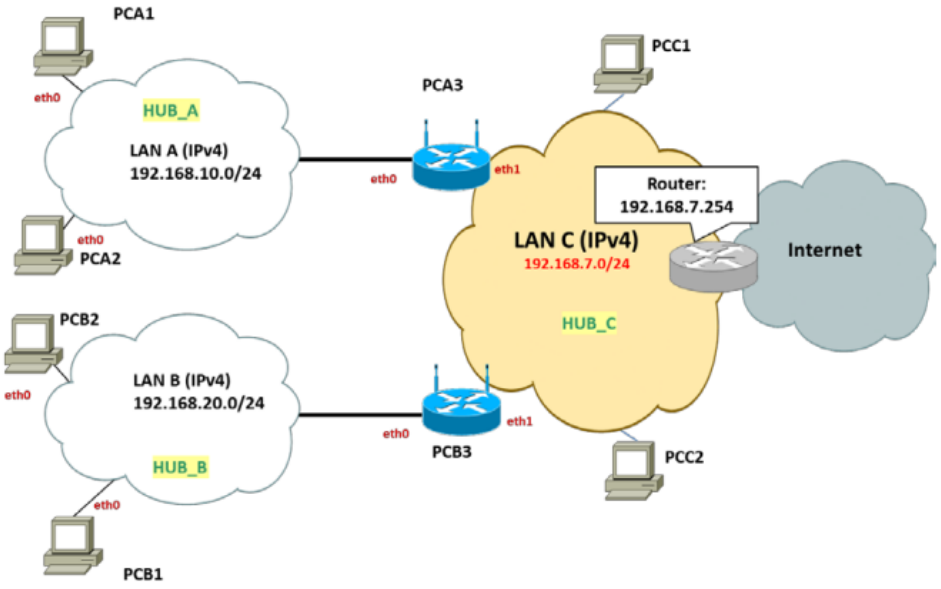

Figura 1: Escenario de interconexión de redes.

Máquina	**SO**	**eth0**	**eth1**
PCA1	centos	192.168.10.1/24	—
PCA2	tinycore	192.168.10.2/24	—
PCA3	centos	192.168.10.3/24	192.168.7.10/24
PCB1	centos	192.168.20.1/24	—
PCB2	tinycore	192.168.20.2/24	—
PCB3	centos	192.168.20.3/24	192.168.7.20/24
PCC1	centos	192.168.7.1/24	—
PCC2	tinycore	192.168.7.2/24	—
Router	?	192.168.7.3/24	?

01

Pregunta 1

Cuando el escenario esté correctamente configurado, verificarlo en PCA1 y PCA2 mediante las herramientas traceroute (en la máquina tinycore) y ping –R (en Centos) (Anexo I.2.B, las equivalentes para windows serían tracert y ping –r 9, Anexo I.2.B) y comprobándolo mediante las capturas de wireshark correspondientes a los tres saltos de la comunicación. Indica sobre el ejemplo concreto de comunicación desde PCA1 y PCA2 hacia PCB1, qué información proporciona cada una de estas herramientas y relaciónala con lo capturado.

Comando ping -r

La herramienta ping -r ha sido empleada para comprobar la conexión entre PCA1 (origen) y PCB1(destino). Al ejecutarla muestra las siguientes direcciones IP:

1. 192.168.10.1 (eth0 PCA1)
2. 192.168.7.10 (eth1 PCA3)
3. 192.168.20.3 (eth0 PCB3)
4. 192.168.20.1 (eth0 PCB1)
5. 192.168.20.1 (eth0 PCB1)
6. 192.168.7.20 (eth1 PCB3)
7. 192.168.10.3 (eth0 PCA3)
8. 192.168.10.1 (eth0 PCA1)

En la columna derecha aparecen las interfaces que reenvían el paquete ICMP Request, y en la columna izquierda aparecen las que reenvían el ICMP Reply.

Para comprobar como se van añadiendo las direcciones a la cabecera IP del mensaje podemos observar las capturas de wireshark de cada red. En todas aparece un único paquete ICMP Request y Reply. En la captura de la red A el paquete 1 es el Request original. En las opciones de su cabecera IP aparece la opción Record Route, que vemos que es de tipo 7, y dentro hay espacio reservado para 9 direcciones, de las cuales sólo ha sido registrada la de origen.

El siguiente salto se observa en el paquete 7 de la captura de la red C. Si comprobamos la cabecera IP podemos ver que PCA3 ha agregado la dirección de su interfaz eth1 (192.168.7.10), justamente el que está conectado a esta red. El último reenvío es el paquete 4 de la red B, que incluye la dirección de la interfaz eth0 (192.168.20.3) de PCB3.

La respuesta sigue el camino contrario. En el paquete 5 de la red B se puede ver que PCB1 ha añadido su dirección dos veces: una por la recepción del Request y otra por el envío del Reply. El último paquete se encuentra en la red A (paquete 2 de la captura P1_lanA), donde quedan registradas las direcciones hasta la dirección de salida de PCA3. La dirección de PCA1 se registra al recibirlo.

Comando traceroute

El comando traceroute ha sido ejecutado desde PCA2 destino PCB1. Al ejecutarlo muestra las siguientes direcciones IP:

1. 192.168.10.2 (eth0 PCA2)
2. 192.168.10.3 (eth0 PCA3)
3. 192.168.7.20 (eth1 PCB3)
4. 192.168.20.1 (eth0 PCB1)

Al observarlas se aprecia que son las direcciones de destino de cada uno de los nodos que recorre hasta llegar a PCB1, a excepción de la dirección de la máquina origen.

Si comprobamos la captura de wireshark de la red LAN A podemos comprobar que los paquetes 7 al 24 corresponden a la ejecución de este comando. Lo primero que se aprecia es que no envía mensajes ICMP request, como lo haría ping, sino que esta herramienta opta por enviar datagramas UDP. El campo TTL de sus cabeceras IP va aumentando con cada dirección descubierta, pero se puede ver que esta implementación de traceroute envía 3 mensajes con el mismo tiempo de vida antes de hacer el incremento. La traza también explica por qué esta herramienta muestra únicamente las direcciones desde el punto de vista del origen: cada datagrama UDP provoca un mensaje ICMP de tipo 11 (time to live exceeded) y este es enviado desde la misma interfaz por el que recibe el datagrama.

Los paquetes número 7, 9 y 11 tienen como objetivo averiguar la primera interfaz de destino, y por eso son enviados con TTL 1. Como el destino es de una red externa, PCA2 envía el mensaje indirectamente a través de la interfaz eth0 de PCA3 (192.168.10.3). Sus correspondientes respuestas (8, 10 y 12) son enviadas desde esta misma interfaz.

Los siguientes 3 paquetes UDP (13, 15 y 17) son enviados con TTL 2 (uno más que el anterior). Cuando llegan a PCA3 su tiempo de vida se reduce a 1, y son reenviados a través de PCB3 por la red C. En la captura de la red LAN C se pueden ver los paquetes reenviados: 19, 21 y 23 respectivamente. Es en esta red en la que se originan las respuestas de control por tiempo de vida excedido, enviadas desde la interfaz eth1 de PCB3 (192.168.7.20), la misma por la que llegó el mensaje UDP. De esta forma PCA2 puede conocer la siguiente dirección de salto.

Los últimos 3 paquetes de la red LAN A (19, 21 y 23 de P1_lanA) son enviados con TTL 3, se reenvían por la red LAN C con tiempo de vida 2 (25, 27 y 29 de P1_lanC) y llegan a PCB1 (destino) a través de la red LAN B (13, 15 y 17 de P1_lanB). Al llegar al destino son rechazados: la máquina no tiene ese puerto abierto por motivos de seguridad. Los mensajes ICMP producidos son enviados desde la dirección IP que se pretendía alcanzar, por lo que el programa termina su ejecución.

02

Pregunta 2

Mientras Se va realizando la configuración del encaminamiento dinámico verifica en las capturas adecuadas, que aparecen los paquetes RIP y cuál es su contenido. También comprueba en los router los valores de las tablas de encaminamiento (telnet localhost ripd ≫ tiempo de vida de las rutas y métricas) relacionándolo con la aparición de los paquetes RIP. Cuando el escenario esté correctamente configurado, verificarlo en PCA1 (Centos) mediante la herramienta ping -R y explica los resultados. No hace falta captura de wireshark. Después, parar el servicio ripd en uno de los router y comprobar que transcurridos 180 segundos la línea de la tabla de encaminamiento del otro router (telnet localhost ripd ≫ tiempo de vida de las rutas y métricas) pasa a tener métrica 16 y que transcurridos otros 120 segundos, esta línea se borra. Para comprobarlo, hay que entregar una captura de pantalla con los parámetros de la tabla de encaminamiento.

Se ha configurado el protocolo de enrutamiento dinámico RIP versión 2 en los routers PCA3 y PCB3. En el archivo denominado “DAR2324_Pr1_g1_P2.pcapng”, se pueden observar ejemplos de un mensaje de Request en la trama 7 y un mensaje de Response en la trama 11 de este protocolo.

En el caso del mensaje Request, se puede apreciar cómo el host PCA3 realiza una petición broadcast que se encuentra encapsulada en una trama Ethernet. En esta solicitud, se evidencia que la métrica enviada es 16, lo que indica que nuestro router no posee información sobre la red que se está explorando.

```
    7 13.372828     192.168.10.3          224.0.0.9             RIPv2       66 Request
> Frame 7: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface -, id 0
> Ethernet II, Src: 0c:b1:38:19:00:01 (0c:b1:38:19:00:01), Dst: IPv4mcast_09 (01:00:5e:00:00:09)
> Internet Protocol Version 4, Src: 192.168.10.3, Dst: 224.0.0.9
> User Datagram Protocol, Src Port: 520, Dst Port: 520
v Routing Information Protocol
    Command: Request (1)
    Version: RIPv2 (2)
  v Address not specified, Metric: 16
      Address Family: Unspecified (0)
      Route Tag: 0
      Netmask: 0.0.0.0
      Next Hop: 0.0.0.0
      Metric: 16
```

Figura 2: RIPv2 Request example

En el mensaje Response siguiente, el paquete contiene una entrada en la tabla de enrutamiento que proporciona información acerca de la red C (192.168.7.0/24) mediante el campo métrica (Metric) a 1. Esto denota que el router que emite esta respuesta posee información sobre dicha red confirmando la existencia de una ruta válida y de bajo costo hacia dicha red.

```
   11 14.348228     192.168.10.3          224.0.0.9             RIPv2       66 Response
> Frame 11: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface -, id 0
> Ethernet II, Src: 0c:b1:38:19:00:01 (0c:b1:38:19:00:01), Dst: IPv4mcast_09 (01:00:5e:00:00:09)
> Internet Protocol Version 4, Src: 192.168.10.3, Dst: 224.0.0.9
> User Datagram Protocol, Src Port: 520, Dst Port: 520
v Routing Information Protocol
    Command: Response (2)
    Version: RIPv2 (2)
  v IP Address: 192.168.7.0, Metric: 1
      Address Family: IP (2)
      Route Tag: 0
      IP Address: 192.168.7.0
      Netmask: 255.255.255.0
      Next Hop: 0.0.0.0
      Metric: 1
```

Figura 3: RIPv2 Response example

Una vez finalizada la configuración del enrutamiento, se llevó a cabo una prueba con la herramienta “ping -R 192.168.20.1” desde la máquina PCA1 hacia la PCB1. Como resultado de esta acción, se obtuvieron las siguientes direcciones:

IP	Hostname	Interfaz	traceroute PCA1 a PCB1	traceroute PCB1 a PCA1
192.168.10.1	PCA1	eth0	-	si (3)
192.168.7.10	PCA3	eth1	-	si (2)
192.168.20.3	PCB3	eth0	-	si (1)
192.168.20.1	PCB1	eth0	-	si (0)
192.168.20.1	PCB1	eth0	si (3)	-
192.168.7.20	PCB3	eth1	si (2)	-
192.168.10.3	PCA3	eth0	si (1)	-
192.168.10.1	PCA1	eth0	si (0)	-

Es importante destacar que cuando el mensaje Request llega a PCB1, este recibe todas las direcciones de manera similar a si hubiera ejecutado el comando “traceroute”. Posteriormente, cuando la máquina PCA1 recibe la Reply, obtiene tanto las direcciones de la interfaz de entrada como las de salida.

Finalmente, se procedió a detener el servicio ripdz se observó cómo cambiaba el estado de la tabla de encaminamiento en la máquina PCA1, lo que resultó en la eliminación de la entrada correspondiente a la red B, que se había vuelto inalcanzable. A continuación, se presentan capturas que ilustran estos cambios.

```
rip> show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface

     Network            Next Hop         Metric From            Tag Time
C(i) 192.168.7.0/24     0.0.0.0               1 self              0
C(i) 192.168.10.0/24    0.0.0.0               1 self              0
R(n) 192.168.20.0/24    192.168.7.20          2 192.168.7.20      0 02:44
```

Figura 4: RIPv2 tabla de encaminamiento estado inicial.

```
     Network            Next Hop         Metric From            Tag Time
C(i) 192.168.7.0/24     0.0.0.0               1 self              0
C(i) 192.168.10.0/24    0.0.0.0               1 self              0
R(n) 192.168.20.0/24    192.168.7.20          2 192.168.7.20      0 00:00
rip> show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface

     Network            Next Hop         Metric From            Tag Time
C(i) 192.168.7.0/24     0.0.0.0               1 self              0
C(i) 192.168.10.0/24    0.0.0.0               1 self              0
R(n) 192.168.20.0/24    192.168.7.20         16 192.168.7.20      0 01:57
```

Figura 5: RIPv2 tabla de encaminamiento metric 16.

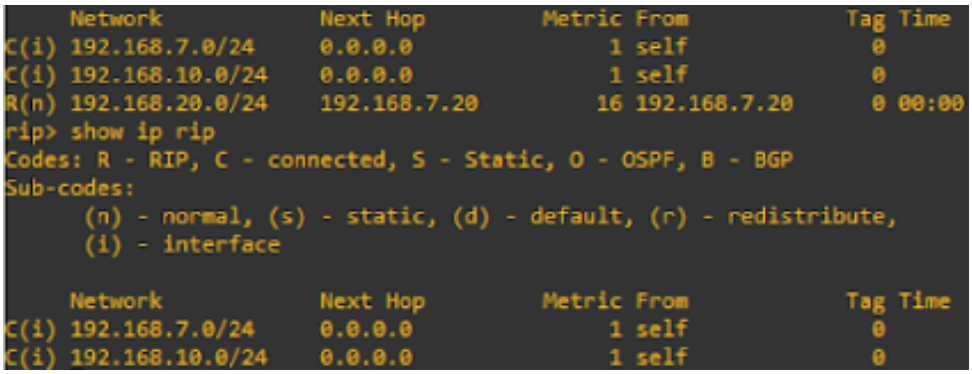

Figura 6: RIPv2 tabla de encaminamiento actualizada.

03

Pregunta 3

Cuando el escenario esté correctamente configurado, verificarlo mediante una conexión a los equipos internos de las redes A y B, utilizando la herramienta ping –R para Linux (Anexo I.2.B) y explica los resultados. No hace falta captura de wireshark. Vamos a aprovechar para comprobar si hay paquetes ICMP redirect en la red y justifica su presencia. Ahora sí será necesaria la captura de wireshark.

Para la puesta en marcha del siguiente escenario, se ha realizado una limpieza previa a la configuración del encaminamiento dinámico en LAN_C. Esto incluye la eliminación de direcciones manuales y rutas estáticas previamente definidas, seguido de una configuración manual de direccionamiento a través del servicio zebra. Además, se ha establecido manualmente que PCA3 sea el router por defecto para PCC1.

Para la comprobación del correcto funcionamiento se ha realizado una serie de tramas ICMP mediante la utilidad Ping.

Conexión con LAN A

En primer lugar se ha realizado dicha prueba comprobando si hay conectividad con alguna de las máquinas en la Lan A. Para ello se ha realizado “ping 192.168.10.1” desde la maquina PCC1

No.	Time	Source	Destination	Protocol	Length	Info
13	31.406748	192.168.7.1	192.168.10.1	ICMP	138	Echo (ping) request id=0x5606, seq=1/256, ttl=64 (reply in 14)
14	31.409747	192.168.10.1	192.168.7.1	ICMP	138	Echo (ping) reply id=0x5606, seq=1/256, ttl=63 (request in 13)

Figura 7: PCC1: ping 192.168.10.1

Conexión con LAN B

A conitnuación se procede a realiza la misma prueba con otra de las máquinas situada en la lan B. Para ello se ha realizado “ping 192.168.20.2” desde la maquina PCC1

No.	Time	Source	Destination	Protocol	Length	Info
111	250.149470	192.168.7.1	192.168.20.2	ICMP	138	Echo (ping) request id=0x5806, seq=2/512, ttl=64 (no response found!)
112	250.152468	192.168.7.10	192.168.7.1	ICMP	206	Redirect (Redirect for host)
113	250.153468	192.168.7.1	192.168.20.2	ICMP	138	Echo (ping) request id=0x5806, seq=2/512, ttl=63 (reply in 114)
114	250.168442	192.168.20.2	192.168.7.1	ICMP	138	Echo (ping) reply id=0x5806, seq=2/512, ttl=63 (request in 113)

Figura 8: PCC1: ping 192.168.20.2

En la captura de la Figura 8, se observa que PCC1 inicia un ping hacia PCB2, pero no recibe respuesta. En su lugar, el router emite un mensaje de redirección (redirect) hacia PCC1, indicando la ruta alternativa que PCC1 debe seguir. Posteriormente, se registran un mensaje de solicitud (request) y una respuesta (reply), ~~siguiendo la ruta sugerida previamente por el router por defecto,~~ PCA3. Cabe destacar que en el resto de tramas presentes en la captura se puede ver como PCC1 no aprende del mensaje de redirección y realiza siempre el mismo procedimiento.

04

Pregunta 4

Cuando el escenario esté correctamente configurado, verificarlo mediante una conexión a los equipos internos de las redes A y B, utilizando la herramienta ping -R para Linux (Anexo I.2.B) y explica los resultados. No hace falta captura de wireshark. Comprueba que no aparecen ahora ICMP redirect y justifica el porqué.

Para preparar el escenario se ha generardo de forma dinámica la tabla de enrutamiento de los hosts en la red C mediante el Protocolo de Información de Enrutamiento (RIP). Los procedimientos que se llevarán a cabo son esencialmente idénticos a los realizados en PCA3 o PCB3. La diferencia principal radica en que, en este caso, no es necesario especificar ningún router por defecto ya que los va a descubrir automáticamente.

Al igual que en la cuestión anterior, se procede a realizar una serie de pruebas con la herramienta Ping para comprobar que hay conectividad hacia las otras redes.

Conexión con LAN A

No.	Time	Source	Destination	Protocol	Length	Info
33	76.454701	192.168.7.1	192.168.10.2	ICMP	138	Echo (ping) request id=0x3207, seq=1/256, ttl=64 (reply in 34)
34	76.461683	192.168.10.2	192.168.7.1	ICMP	138	Echo (ping) reply id=0x3207, seq=1/256, ttl=63 (request in 33)

Figura 9: PCC1: ping 192.168.10.2

Como se puede notar en la captura, no se evidencia ningún cambio con respecto a la pregunta anterior, lo que significa que el acceso a la red A sigue inalterado.

Conexión con LAN B

No.	Time	Source	Destination	Protocol	Length	Info
53	94.317263	192.168.7.1	192.168.20.1	ICMP	138	Echo (ping) request id=0x3307, seq=1/256, ttl=64 (reply in 54)
54	94.320262	192.168.20.1	192.168.7.1	ICMP	138	Echo (ping) reply id=0x3307, seq=1/256, ttl=63 (request in 53)

Figura 10: PCC1: ping 192.168.20.1

En este escenario particular, podemos destacar una diferencia con respecto a la pregunta anterior: ya no se observa el mensaje de redireccionamiento (redirect). Esto se debe a que, gracias al protocolo RIP, PCC1 ha logrado aprender las rutas hacia las otras redes.

```
     Network          Next Hop        Metric From          Tag Time
C(i) 192.168.7.0/24   0.0.0.0              1 self           0
R(n) 192.168.10.0/24  192.168.7.10         2 192.168.7.10   0 02:31
R(n) 192.168.20.0/24  192.168.7.20         2 192.168.7.20   0 02:42
```

Figura 11: PCC1: Tabla encaminamiento RIP.

05

Pregunta 5 - 9

Pregunta 5

Realizamos un ping desde PCA1 a PCB1 (de centos a centos).

Indicar los distintos casos de entrega directa e indirecta observados especificando su relación con el tráfico ARP capturado: Pon un ejemplo de un paquete IP encaminado mediante entrega directa especificando la IP destino y la MAC destino. Pon un ejemplo de un paquete IP encaminado mediante entrega indirecta especificando la IP destino y la MAC destino.

Las entregas directas son aquellas en las que tanto dirección IP de la máquina que envía como la de destino están dentro de una misma red. En este caso, los dos únicos casos de encaminamiento directo son la entrega del paquete ICMP Request de PCB3 a PCB1, dentro de la red B, y la entrega del paquete ICMP Reply de PCA3 a PCA1, dentro de la red A.

Las entregas indirectas son las que se realizan a un nodo intermedio porque la dirección de destino se encuentra en otra red. El paquete se envía a un encaminador. Ese el caso de la entrega del paquete Request de PCA1 a PCA3, el cual es su encaminador por defecto, o de PCA3 a PCB3, que es el encaminador aprendido con RIP para acceder a la red B. En la entrega del mensaje Reply también hay encaminamiento indirecto: en la entrega de PCB1 a PCB3 y en la de PCB3 a PCA3 (camino opuesto).

El caso de entrega indirecta dentro de la red A, PCA1 a PCA3, lo podemos comprobar observando los mensajes ARP: PCA1 pregunta por la dirección MAC de la IP 192.168.10.3 (paquete 3 de lan_A), la cual corresponde a PCA3. Una vez resuelta, podemos comprobar que la dirección de respuesta (0c:b1:38:19:00:00) coincide con la dirección MAC destino dentro de la trama Ethernet que encapsula el paquete ICMP (paquete 5 de lan_A), por lo que realmente no está enviando el paquete a PCB1 (192.168.20.1), sino lo está reenviando a través de PCA3, y se trata de una entrega indirecta.

En el caso de entrega directa dentro de la red B, PCB3 a PCB1, pasa lo contrario. El mensaje ARP pregunta por la MAC de PCB1 (paquete 5 de lan_B), y esta aparece como destino de la trama Ethernet (paquete 7 de lan_B), luego el mensaje está siendo enviado a la dirección IP destino, y se trata de entrega directa.

Pregunta 6

¿A qué se deben las preguntas ARP durante la transmisión del primer ICMP Echo Request?

Antes de realizar el experimento se habían vaciado las tablas ARP de todos los nodos, por lo que ninguno tiene información de qué dirección MAC corresponde a cada IP. Para realizar el envío en el nivel de enlace es necesaria esta dirección, por lo que en cada reenvío del paquete Request es necesario realizar una pregunta y respuesta ARP.

Pregunta 7

¿Por qué no hay preguntas ARP durante la transmisión del primer ICMP Echo Reply?

Porque el camino de ida es simétrico al de vuelta. Cada reenvío del ICMP Echo Request provocaba una pregunta ARP, pero la dirección obtenida no se utiliza únicamente para ese envío concreto, sino que es guardada en las tablas ARP del nodo. Cuando PCA1 preguntaba por la dirección de PCA3 lo hacía incluyendo sus direcciones MAC e IP dentro del cuerpo del mensaje (en wireshark son el campo "Sender MAC/IP Address"). PCA3 aprovecha esta información para rellenar su tabla ARP, de modo que no es necesario realizar la pregunta opuesta al enviar el ICMP Echo Reply.

Pregunta 8

¿Qué ocurre con los siguientes paquetes ICMP?

Al igual que con el paquete Reply, las tablas ARP contienen la información del nivel de enlace para todos los nodos involucrados. Tendría que pasar mucho tiempo, del orden de minutos, entre dos paquetes para que las tablas volviesen a estar vacías, pero el intervalo de tiempo entre dos ICMP Echo Reply es de un segundo.

Los únicos mensajes ARP que aparecen no son preguntas broadcast para descubrir una dirección MAC específica, sino unicast para comprobar que la asociación IP-MAC sigue siendo correcta.

Pregunta 9

Sin dejar de capturar, parar el anterior ping e inmediatamente después iniciar un ping desde un host de LAN C a PCA1. ¿Aparecen mensajes ARP en LAN interna y en LAN externa? ¿Por qué?

El ping ha sido realizado desde la máquina PCC1 (Centos). En la red C (LAN interna) aparece una pregunta ARP por parte de PCC1 para saber qué MAC tiene PCA3 (pregunta por 192.168.7.10), pese a que PCA3 ya había anunciado su MAC dentro de esta red. El motivo puede ser que las máquinas de una misma red no rellenan la entrada de la máquina que pregunta con broadcast a no ser que la IP preguntada coincida con la IP propia, seguramente por no aumentar el tamaño de las tablas ARP. Cuando una máquina pregunta por la MAC correspondiente a una IP propia es casi seguro que vaya a haber envíos en ambas direcciones (la gran mayoría de protocolos utilizan comunicación en ambos sentidos).

En la red A (LAN externa) no aparece ninguna pregunta ARP, ya que en el anterior ping ya han interactuado PCA3 y PCA1, y no ha habido tiempo suficiente para vaciar sus tablas ARP.

06

Pregunta 10

Indicar el número y tamaño de los paquetes capturados en las redes LAN A, LAN C y LAN B. Para ello, rellena la tabla adjunta y justifica teóricamente los tamaños indicados.

- MTU PCA3(eth0)=800
- MTU PCA3(eth1)=1100
- MTU PCB3(eth0)=700
- MTU PCB3(eth1)=1300

	Nº paquete	Tamaño	Campos cabecera IP				
			ID	Flags		Offset	
				DF	MF	Original	Wireshark
LAN A	3	$1428 = (1400 + 8_{icmp} + 20_{ip})$	54873	0	0	0	0
LAN C	21	$1100 = (1072 + 8_{icmp} + 20_{ip})$	54873	0	1	0	0
	22	$348 = (328 + 20_{ip})$	54873	0	0	$135 = 1080/8$	1080
LAN B	3	$700 = (672 + 8_{icmp} + 20_{ip})$	54873	0	1	0	0
	4	$420 = (400 + 20_{ip})$	54873	0	1	$85 = 680/8$	680
	5	$348 = (328 + 20_{ip})$	54873	0	0	$135 = 1080/8$	1080
LAN B	6	$1428 = (1400 + 8_{icmp} + 20_{ip})$	27452	0	0	0	0
LAN C	23	$1300 = (1272 + 8_{icmp} + 20_{ip})$	27452	0	1	0	0
	24	$148 = (128 + 20_{ip})$	27452	0	0	$160 = 1280/8$	1280
LAN A	4	$796 = (768 + 8_{icmp} + 20_{ip})$	27452	0	1	0	0
	5	$524 = (504 + 20_{ip})$	27452	0	1	97= 776/8	776
	6	$148 = (128 + 20_{ip})$	27452	0	0	$160 = 1280/8$	1280

El tamaño de los paquetes IP que aparece en wireshark tiene en cuenta tanto la cabecera ICMP (8 bytes) como la cabecera IP (20 bytes). Cuando un paquete es fragmentado, el offset se calcula como la suma de de los tamaños de datos de los fragmentos anteriores:

$$Offset_i = \sum_{j<i} Data_j$$

El paquete Request dentro de la red A no se fragmenta, ya que PCA1 no tiene definido un valor MTU. Al ser reenviado por PCA3, a través de la red C, sale por la interfaz eth1. Como el MTU está configurado a 1100 el primer fragmento del paquete tiene esta longitud, pero como este tamaño incluye los 20 bytes de cabecera IP, todavía quedan por enviar 328 bytes (1400 + 8 - 1080). La fragmentación es internet y no intranet, por lo que PCB3 no reensambla el paquete original; en su lugar reenvía ambos fragmentos. La interfaz por la que los envía es eth0, que tiene un MTU de 700, inferior a los 1100 bytes del primer fragmento, es por ello que tiene que volver a dividir en dos fragmentos: el primero de tamaño 680 bytes (con la cabera IP suma 700) y el segundo de 400 bytes (1080 - 680). Junto al otro fragmento, que se ve inalterado, son tres los que aparecen en la red B.

Al paquete Reply le ocurre algo parecido: al ser reenviado por la red C debe fragmentar los 1408 bytes de ICMP en 1280 (cabecera ICMP incluida) y 148; pero PCA3 no los reensambla y su interfaz eth0 tiene un MTU de 800, por lo que tiene que dividir el fragmento de 1280 bytes en otros dos. En lugar de hacerlo en uno 780 y otro de 500 (780 + 500 = 1280), lo que hace es dividirlo en 776 y 504, ya que 780 no es divisible por 8 y, por lo tanto, no es un offset válido.

07

Pregunta 11

¿Cuántas cabeceras ICMP Echo Request aparecen en LAN A, B y C?

Tres: una en cada LAN, dado que sólo el primer fragmento de cada paquete IP contiene la cabecera ICMP.

08

Pregunta 12

Mide el retardo existente entre PCA1 y PCB1, utilizando para ello las dos instrucciones que se muestran a continuación:

Listing 1: caso 1

```
[root@localhost ~]# ping 192.168.20.1 -c 25 -s 600
PING 192.168.20.1 (192.168.20.1) 600(628) bytes of data.
608 bytes from 192.168.20.1: icmp_seq=1 ttl=62 time=6.78 ms
608 bytes from 192.168.20.1: icmp_seq=2 ttl=62 time=16.9 ms
608 bytes from 192.168.20.1: icmp_seq=3 ttl=62 time=13.6 ms
608 bytes from 192.168.20.1: icmp_seq=4 ttl=62 time=16.5 ms
608 bytes from 192.168.20.1: icmp_seq=5 ttl=62 time=5.28 ms
608 bytes from 192.168.20.1: icmp_seq=6 ttl=62 time=20.7 ms
608 bytes from 192.168.20.1: icmp_seq=7 ttl=62 time=8.48 ms
608 bytes from 192.168.20.1: icmp_seq=8 ttl=62 time=17.9 ms
608 bytes from 192.168.20.1: icmp_seq=9 ttl=62 time=21.0 ms
608 bytes from 192.168.20.1: icmp_seq=10 ttl=62 time=11.7 ms
608 bytes from 192.168.20.1: icmp_seq=11 ttl=62 time=18.3 ms
608 bytes from 192.168.20.1: icmp_seq=12 ttl=62 time=19.5 ms
608 bytes from 192.168.20.1: icmp_seq=13 ttl=62 time=7.56 ms
608 bytes from 192.168.20.1: icmp_seq=14 ttl=62 time=6.16 ms
608 bytes from 192.168.20.1: icmp_seq=15 ttl=62 time=12.4 ms
608 bytes from 192.168.20.1: icmp_seq=16 ttl=62 time=21.2 ms
608 bytes from 192.168.20.1: icmp_seq=17 ttl=62 time=12.1 ms
608 bytes from 192.168.20.1: icmp_seq=18 ttl=62 time=18.7 ms
608 bytes from 192.168.20.1: icmp_seq=19 ttl=62 time=10.3 ms
608 bytes from 192.168.20.1: icmp_seq=20 ttl=62 time=20.8 ms
608 bytes from 192.168.20.1: icmp_seq=21 ttl=62 time=5.58 ms
608 bytes from 192.168.20.1: icmp_seq=22 ttl=62 time=15.4 ms
608 bytes from 192.168.20.1: icmp_seq=23 ttl=62 time=9.21 ms
608 bytes from 192.168.20.1: icmp_seq=24 ttl=62 time=16.1 ms
608 bytes from 192.168.20.1: icmp_seq=25 ttl=62 time=9.09 ms

--- 192.168.20.1 ping statistics ---
25 packets transmitted, 25 received, 0% packet loss, time 24083ms
rtt min/avg/max/mdev = 5.284/13.694/21.237/5.317 ms
```

Listing 2: caso 2

```
[root@localhost ~]# ping 192.168.20.1 -c 25 -s 1300
PING 192.168.20.1 (192.168.20.1) 1300(1328) bytes of data.
1308 bytes from 192.168.20.1: icmp_seq=1 ttl=62 time=11.9 ms
1308 bytes from 192.168.20.1: icmp_seq=2 ttl=62 time=46.7 ms
1308 bytes from 192.168.20.1: icmp_seq=3 ttl=62 time=33.9 ms
1308 bytes from 192.168.20.1: icmp_seq=4 ttl=62 time=8.67 ms
1308 bytes from 192.168.20.1: icmp_seq=5 ttl=62 time=11.0 ms
1308 bytes from 192.168.20.1: icmp_seq=6 ttl=62 time=13.6 ms
1308 bytes from 192.168.20.1: icmp_seq=7 ttl=62 time=10.3 ms
1308 bytes from 192.168.20.1: icmp_seq=8 ttl=62 time=9.94 ms
1308 bytes from 192.168.20.1: icmp_seq=9 ttl=62 time=22.2 ms
1308 bytes from 192.168.20.1: icmp_seq=10 ttl=62 time=12.4 ms
1308 bytes from 192.168.20.1: icmp_seq=11 ttl=62 time=22.5 ms
1308 bytes from 192.168.20.1: icmp_seq=12 ttl=62 time=8.00 ms
1308 bytes from 192.168.20.1: icmp_seq=13 ttl=62 time=28.3 ms
1308 bytes from 192.168.20.1: icmp_seq=14 ttl=62 time=20.9 ms
1308 bytes from 192.168.20.1: icmp_seq=15 ttl=62 time=27.8 ms
1308 bytes from 192.168.20.1: icmp_seq=16 ttl=62 time=10.1 ms
1308 bytes from 192.168.20.1: icmp_seq=17 ttl=62 time=12.4 ms
1308 bytes from 192.168.20.1: icmp_seq=18 ttl=62 time=25.5 ms
1308 bytes from 192.168.20.1: icmp_seq=19 ttl=62 time=28.3 ms
1308 bytes from 192.168.20.1: icmp_seq=20 ttl=62 time=22.3 ms
1308 bytes from 192.168.20.1: icmp_seq=21 ttl=62 time=11.2 ms
1308 bytes from 192.168.20.1: icmp_seq=22 ttl=62 time=11.8 ms
1308 bytes from 192.168.20.1: icmp_seq=23 ttl=62 time=23.4 ms
1308 bytes from 192.168.20.1: icmp_seq=24 ttl=62 time=18.8 ms
1308 bytes from 192.168.20.1: icmp_seq=25 ttl=62 time=31.7 ms

--- 192.168.20.1 ping statistics ---
25 packets transmitted, 25 received, 0% packet loss, time 24109ms
rtt min/avg/max/mdev = 8.004/19.393/46.744/9.603 ms
```

Como las latencias obtenidas con ping varían mucho de un envío a otro, se decidió enviar 25 paquetes por medición y comparar los estadísticos del tiempo de ida y vuelta que muestra la herramienta.

En el caso 1 no hay fragmentación, ya que todos los MTU configurados anteriormente superan los 600 bytes. En el caso 2 la fragmentación es similar al ejercicio 10: mismo número de fragmentos por red, aunque variando tamaños. Se puede ver que en el caso con fragmentación el tiempo de ida y vuelta es mayor para todos los valores del resumen:

- **Máxima latencia:** 21.237 ms $<$ 46.744 ms
- **Mínima latencia:** 5.284 ms $<$ 8.004 ms
- **Latencia media:** 13.694 ms $<$ 19.393 ms