

PRÁCTICA 2

REDES DE LOS COMPUTADORES

JAIME HERNÁNDEZ

CUESTIÓN 1: PING

```
C:\>ping -n 1 -l 500 172.20.43.231

Haciendo ping a 172.20.43.231 con 500 bytes de datos:
Respuesta desde 172.20.43.231: bytes=500 tiempo=17ms TTL=255

Estadísticas de ping para 172.20.43.231:
    Paquetes: enviados = 1, recibidos = 1, perdidos = 0
      (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
      Mínimo = 17ms, Máximo = 17ms, Media = 17ms
```

a) ¿Que tipos de mensajes ICMP aparecen?

No.	Time	Source	Destination	Protocol	Length	Info
529	41.555000	172.20.43.212	172.20.43.231	ICMP	542	Echo (ping) request id=0x00000001
530	41.565151	172.20.43.231	172.20.43.212	ICMP	542	Echo (ping) reply id=0x00000001

b) Justificar la procedencia de cada dirección MAC e IP.

```
> Destination: ca:05:71:85:00:00 (ca:05:71:85:00:00)
> Source: Giga-Byt_07:ad:c1 (18:c0:4d:07:ad:c1)
```

c) Verificar que los tamaños de los datos y las cabeceras de los protocolos que aparecen en los paquetes ICMP (Ethernet, IP, ICMP) son los esperados

```
Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
  Code: 0
  Checksum: 0x2de3 [correct]
  [Checksum Status: Good]
  Identifier (BE): 1 (0x0001)
  Identifier (LE): 256 (0x0100)
  Sequence Number (BE): 85 (0x0055)
  Sequence Number (LE): 21760 (0x5500)
  [Response frame: 530]
```

CUESTIÓN 2: FRAGMENTACIÓN

C:\>ping -n 1 -l 2000 172.20.43.231

```
C:\>ping -n 1 -l 2000 172.20.43.231

Haciendo ping a 172.20.43.231 con 2000 bytes de datos:
Respuesta desde 172.20.43.231: bytes=2000 tiempo=4ms TTL=255

Estadísticas de ping para 172.20.43.231:
    Paquetes: enviados = 1, recibidos = 1, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 4ms, Máximo = 4ms, Media = 4ms
```

- a) Describir los paquetes IP generados por el PC del alumno asociados a la fragmentación del mensaje ICMP Echo Request

No.	Time	Source	Destination	Protocol	Length	Info
297	50.657581	172.20.43.212	172.20.43.231	ICMP	1514	Echo (ping) request id=0x0001, seq=86/22016, ttl=128 (reply in 299)
299	50.661798	172.20.43.231	172.20.43.212	ICMP	1514	Echo (ping) reply id=0x0001, seq=86/22016, ttl=255 (request in 297)

- b) Determinar el MTU de la máquina del alumno y de la máquina 172.20.43.231

C:\>ping -n 1 -l 1000 10.3.7.0

```
C:\>ping -n 1 -l 1000 10.3.7.0

Haciendo ping a 10.3.7.0 con 1000 bytes de datos:
Respuesta desde 10.3.7.0: bytes=1000 tiempo=264ms TTL=253

Estadísticas de ping para 10.3.7.0:
    Paquetes: enviados = 1, recibidos = 1, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 264ms, Máximo = 264ms, Media = 264ms
```

- a) Describir los paquetes IP generados por el PC del alumno asociados a la fragmentación del mensaje ICMP Echo Request.

16	4.433794	172.20.43.212	10.3.7.0	ICMP	1042	Echo (ping) request id=0x0001, seq=88/22528, ttl=128 (reply in 18)
18	4.697893	10.3.7.0	172.20.43.212	ICMP	610	Echo (ping) reply id=0x0001, seq=88/22528, ttl=253 (request in 16)

- b) Determinar el MTU del destino analizando el tamaño de los paquetes IP recibidos desde la dirección 10.3.7.0.

C:\>ping -n 1 -l 4500 172.20.43.232

```
C:\>ping -n 1 -l 4500 172.20.43.232

Haciendo ping a 172.20.43.232 con 4500 bytes de datos:
Respuesta desde 172.20.43.232: bytes=4500 tiempo=3ms TTL=64

Estadísticas de ping para 172.20.43.232:
    Paquetes: enviados = 1, recibidos = 1, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 3ms, Máximo = 3ms, Media = 3ms
```

a) Determina cuántos paquetes IP se generarán desde el PC del alumno y qué valores tendrán los campos OFFSET y bits DF y MF de la cabecera IP.

805	115.318880	172.20.43.212	172.20.43.232	ICMP	1514 Echo (ping) request	id=0x0001, seq=90/23040, ttl=128 (reply in 809)
809	115.312563	172.20.43.232	172.20.43.212	ICMP	1514 Echo (ping) reply	id=0x0001, seq=90/23040, ttl=64 (request in 805)

b) ¿Se corresponden con los paquetes IP capturados con el monitor de red ?

Podemos observar que es de esa manera:

114	14.664942	172.20.43.212	172.20.41.241	ICMP	74 Echo (ping) request	id=0x0001, seq=91/23296, ttl=128 (reply in 116)
115	14.675579	172.20.43.230	172.20.43.212	ICMP	70 Redirect	(Redirect for host)
116	14.918119	172.20.41.241	172.20.43.212	ICMP	74 Echo (ping) reply	id=0x0001, seq=91/23296, ttl=253 (request in 114)

Gateway Address: 172.20.43.231	
Internet Protocol Version 4, Src: 172.20.43.212, Dst: 172.20.41.241	
0100 = Version: 4	
.... 0101 = Header Length: 20 bytes (5)	
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)	
Total Length: 60	
Identification: 0x0686 (1670)	
> Flags: 0x00	
Fragment Offset: 0	
Time to Live: 127	
Protocol: ICMP (1)	
Header Checksum: 0x874d [validation disabled]	
[Header checksum status: Unverified]	
Source Address: 172.20.43.212	
Destination Address: 172.20.41.241	
> Internet Control Message Protocol	

CUESTIÓN 3: REDIRECT

```
C:\>ping -n 1 172.20.41.241

Haciendo ping a 172.20.41.241 con 32 bytes de datos:
Respuesta desde 172.20.41.241: bytes=32 tiempo=253ms TTL=253

Estadísticas de ping para 172.20.41.241:
    Paquetes: enviados = 1, recibidos = 1, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 253ms, Máximo = 253ms, Media = 253ms
```

- a) ¿Cuántos paquetes están involucrados?

Paquetes: 1465 · Mostrado: 3 (0.2%)

Paquetes: 1465 · Mostrado: 1465 (100.0%)

- b) ¿Qué estación envía el mensaje ICMP redirect?

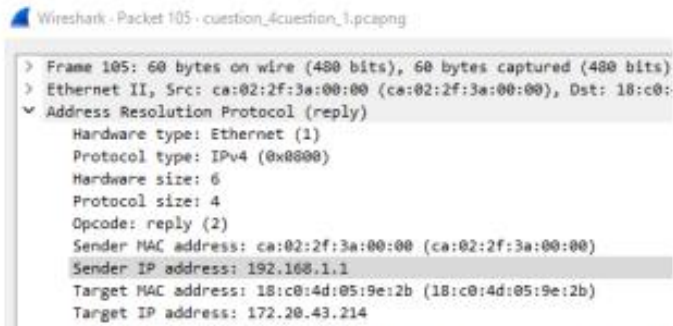
> Destination: ca:02:2f:3a:00:00 (ca:02:2f:3a:00:00)
> Source: Giga-Byt_07:ad:c1 (18:c0:4d:07:ad:c1)
Type: IPv4 (0x0800)

- c) ¿Qué datos complementarios transporta el mensaje ICMP redirect?

▼ Data (32 bytes)
Data: 6162636465666768696a6b6c6d6e6f70717273
[Length: 32]

CUESTIÓN 4 // Hecho la documentación en laoratio, he tomado captura de la documentación

b) ¿Qué router se emplea como puerta de enlace en el mensaje ICMP Echo Reply?

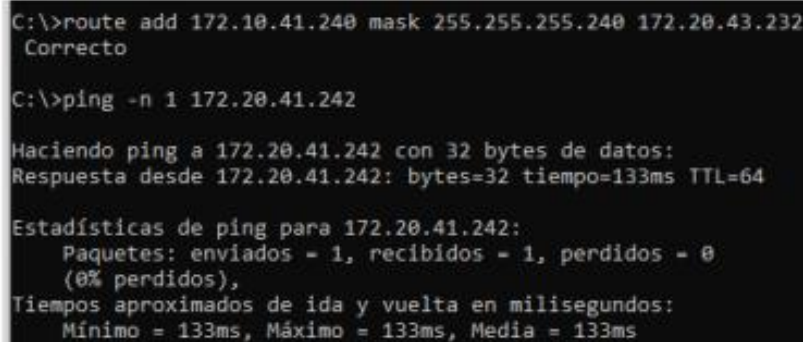


```
Wireshark - Packet 105 - cuestion_4cuestion_1.pcapng
> Frame 105: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
> Ethernet II, Src: ca:02:2f:3a:00:00 (ca:02:2f:3a:00:00), Dst: 18:c0:
  Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: ca:02:2f:3a:00:00 (ca:02:2f:3a:00:00)
    Sender IP address: 192.168.1.1
    Target MAC address: 18:c0:4d:05:9e:2b (18:c0:4d:05:9e:2b)
    Target IP address: 172.20.43.214
```

Cómo se puede ver en el Sender IP address la IP de la puerta de enlace es la misma que el del ECHO Request.

Con el comando route añade una ruta en la tabla de encaminamiento del PC del alumno para que los paquetes enviados a la red de destino 172.20.41.240/28 sean encaminados a través del router Linux 2. A continuación ejecutar el comando:

C:\>ping -n 1 172.20.41.242



```
C:\>route add 172.10.41.240 mask 255.255.255.240 172.20.43.232
Correcto

C:\>ping -n 1 172.20.41.242

Haciendo ping a 172.20.41.242 con 32 bytes de datos:
Respuesta desde 172.20.41.242: bytes=32 tiempo=133ms TTL=64

Estadísticas de ping para 172.20.41.242:
    Paquetes: enviados = 1, recibidos = 1, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 133ms, Máximo = 133ms, Media = 133ms
```

Determina:

a) ¿Qué router se emplea como puerta de enlace en el mensaje ICMP Echo Request? ¿Se corresponde con el especificado en la entrada añadida con el comando route?


```

  Destination: 00:54:10:f7:76:00 (00:54:10:f7:76:00)
    Address: 00:54:10:f7:76:00 (00:54:10:f7:76:00)
      .... ..0. .... = LG bit: Globally unique ad
      .... ..0. .... = IG bit: Individual address
  Source: 18:c0:4d:05:9e:2b (18:c0:4d:05:9e:2b)
    Address: 18:c0:4d:05:9e:2b (18:c0:4d:05:9e:2b)
      .... ..0. .... = LG bit: Globally unique ad
      .... ..0. .... = IG bit: Individual address
  Type: ARP (0x0806)
  Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: 18:c0:4d:05:9e:2b (18:c0:4d:05:9e:2b)
    Sender IP address: 172.20.43.214
    Target MAC address: 00:54:10:f7:76:00 (00:54:10:f7:76:00)
    Target IP address: 172.20.43.232

```

Como se puede ver la puerta de enlace es la que marcamos en el anterior comando en el terminal, pero eso no si antes preguntar a muchos antes, porque hemos puesto la máscara a 255.255.255.240.

147	15.679608	18:c0:4d:05:8f:28	Broadcast	ARP	60 Who has 169.254.27.46? Tell 0.0.0.0
158	16.679335	18:c0:4d:05:8f:28	Broadcast	ARP	60 Who has 169.254.27.46? Tell 0.0.0.0
178	17.679582	18:c0:4d:05:8f:28	Broadcast	ARP	60 Who has 169.254.27.46? Tell 0.0.0.0
195	18.679686	18:c0:4d:05:8f:28	Broadcast	ARP	60 Gratuitous ARP for 169.254.27.46 (Request)
273	28.668729	18:c0:4d:05:8f:28	Broadcast	ARP	60 Who has 172.20.43.200? Tell 0.0.0.0
274	28.753394	18:c0:4d:05:8f:28	Broadcast	ARP	60 Who has 172.20.43.195? Tell 172.20.43.200
276	29.005740	18:c0:4d:05:8f:28	Broadcast	ARP	60 Who has 172.20.43.195? Tell 172.20.43.200
303	29.673152	18:c0:4d:05:8f:28	Broadcast	ARP	60 Who has 172.20.43.200? Tell 0.0.0.0
328	30.669801	18:c0:4d:05:8f:28	Broadcast	ARP	60 Who has 172.20.43.200? Tell 0.0.0.0
338	31.669862	18:c0:4d:05:8f:28	Broadcast	ARP	60 Gratuitous ARP for 172.20.43.200 (Request)
371	32.195994	18:c0:4d:05:8f:28	Broadcast	ARP	60 Who has 172.20.43.224? Tell 172.20.43.200
376	32.344716	18:c0:4d:05:8f:28	Broadcast	ARP	60 Who has 172.20.43.220? Tell 172.20.43.200
415	33.094655	18:c0:4d:05:8f:28	Broadcast	ARP	60 Who has 172.20.43.221? Tell 172.20.43.200
416	33.100707	18:c0:4d:05:8f:28	Broadcast	ARP	60 Who has 172.20.43.211? Tell 172.20.43.200
417	33.116730	18:c0:4d:05:8f:28	Broadcast	ARP	60 Who has 172.20.43.203? Tell 172.20.43.200
442	33.704816	18:c0:4d:05:8f:28	Broadcast	ARP	60 Who has 172.20.43.219? Tell 172.20.43.200
450	33.924648	18:c0:4d:05:8f:28	Broadcast	ARP	60 Who has 172.20.43.216? Tell 172.20.43.200
451	33.946362	18:c0:4d:05:8f:28	Broadcast	ARP	60 Who has 172.20.43.215? Tell 172.20.43.200
455	34.164094	18:c0:4d:05:8f:28	Broadcast	ARP	60 Who has 172.20.43.228? Tell 172.20.43.200
568	38.974747	18:c0:4d:05:9e:2b	ca:02:2f:3a:00:00	ARP	42 Who has 172.20.43.230? Tell 172.20.43.214
569	38.983568	ca:02:2f:3a:00:00	18:c0:4d:05:9e:2b	ARP	60 172.20.43.230 is at ca:02:2f:3a:00:00
617	44.437532	00:54:10:f7:76:00	18:c0:4d:05:9e:2b	ARP	60 Who has 172.20.43.214? Tell 172.20.43.232
618	44.437553	18:c0:4d:05:9e:2b	00:54:10:f7:76:00	ARP	42 172.20.43.214 is at 18:c0:4d:05:9e:2b

b) ¿Qué router se emplea como puerta de enlace en el mensaje ICMP Echo Reply?

La puerta de enlace que se emplea es la misma que pusimos en el comando del terminal y que la puerta de enlace del ECHO Request.

CUESTIÓN 5

En base a los paquetes capturados, indicar:

a) Identificar las direcciones IP/MAC de los paquetes involucrados.

```
✓ Ethernet II, Src: Giga-Byt_07:ad:f9 (18:c0:4d:07:ad:f9), D
  > Destination: ca:02:2f:3a:00:00 (ca:02:2f:3a:00:00)
  > Source: Giga-Byt_07:ad:f9 (18:c0:4d:07:ad:f9)
  Type: IPv4 (0x0800) b)
```

¿Qué estación envía el mensaje ICMP Fragmentation Needed and Don't Fragment was Set (3/4)?

```
✓ Ethernet II, Src: Giga-Byt_07:ad:f9 (18:c0:4d:07:ad:f9), D
  > Destination: ca:02:2f:3a:00:00 (ca:02:2f:3a:00:00)
  > Source: Giga-Byt_07:ad:f9 (18:c0:4d:07:ad:f9)
  Type: IPv4 (0x0800) c)
```

Analiza la información de la cabecera ICMP del mensaje anterior ¿Cuál es el valor del MTU de la red que no puede transmitir el paquete ICMP Echo Request ?

MTU of next HOP == 600

Vuelve a ejecutar los siguientes comandos para eliminar información sobre el experimento anterior. C:\route delete 0.0.0.0

C:\ipconfig /release

C:\ipconfig /renew

C:\pracredes


```

C:\>route delete 0.0.0.0
Correcto

C:\>ipconfig /release

Configuración IP de Windows

Adaptador de Ethernet VirtualBox Host-Only Network:

    Sufijo DNS específico para la conexión. . . :
    Vínculo: dirección IPv6 local. . . : fe80::91c7:9b8a:2d70:bc77%17
    Dirección IPv4. . . . . : 192.168.56.1
    Máscara de subred . . . . . : 255.255.255.0
    Puerta de enlace predeterminada . . . . . :

Adaptador de Ethernet Ethernet:

    Sufijo DNS específico para la conexión. . . :
    Vínculo: dirección IPv6 local. . . : fe80::8090:fc18:cf1c:618e%11
    Puerta de enlace predeterminada . . . . . :

C:\>ipconfig /renew

Configuración IP de Windows

Adaptador de Ethernet VirtualBox Host-Only Network:

    Sufijo DNS específico para la conexión. . . :
    Vínculo: dirección IPv6 local. . . : fe80::91c7:9b8a:2d70:bc77%17
    Dirección IPv4. . . . . : 192.168.56.1
    Máscara de subred . . . . . : 255.255.255.0
    Puerta de enlace predeterminada . . . . . :

Adaptador de Ethernet Ethernet:

    Sufijo DNS específico para la conexión. . . :
    Vínculo: dirección IPv6 local. . . : fe80::8090:fc18:cf1c:618e%11
    Dirección IPv4. . . . . : 172.20.43.213
    Máscara de subred . . . . . : 255.255.255.192
    Puerta de enlace predeterminada . . . . . : 172.20.43.195

C:\>pracredes
Correcto
Correcto

C:\>ping -n 1 -l 1200 -f 10.3.7.0

Haciendo ping a 10.3.7.0 con 1200 bytes de datos:
Respuesta desde 10.4.2.5: Es necesario fragmentar el paquete pero se especificó DF.

Estadísticas de ping para 10.3.7.0:
    Paquetes: enviados = 1, recibidos = 1, perdidos = 0
    (0% perdidos),

C:\>route delete 0.0.0.0
Correcto

C:\>ipconfig /release

Configuración IP de Windows

```

Inicia el monitor de red y vuelve a ejecutar el mismo comando ping, pero sin la opción -f:

```
C:\>ping -n 1 -l 1200 10.3.7.0
```

En base a los paquetes capturados determina:

a) ¿El tamaño de los paquetes IP asociados al mensaje ICMP ECHO Reply tienen el tamaño adecuado al MTU que se ha informado en el mensaje ICMP Fragmentation Needed and Don't Fragment was Set (3/4) en el experimento anterior?

Total length == 596, < MTU == 600

b) ¿Cuál es la razón de que no coincidan? Recuerda que el campo Fragment Offset de la cabecera IP NO puede tener cualquier valor.

Es porque el mtu es el tamaño máximo de paquete que un enlace o una red específica puede transportar, sin embargo, el tamaño real es más pequeño que este valor

A continuación, se analizará el mensaje ICMP Destination Unreachable con el código 1: Host Unreachable. Con el comando route, establece una ruta para alcanzar la red 172.20.41.240/28 a través del router Linux 2. Para ello ejecuta el comando:

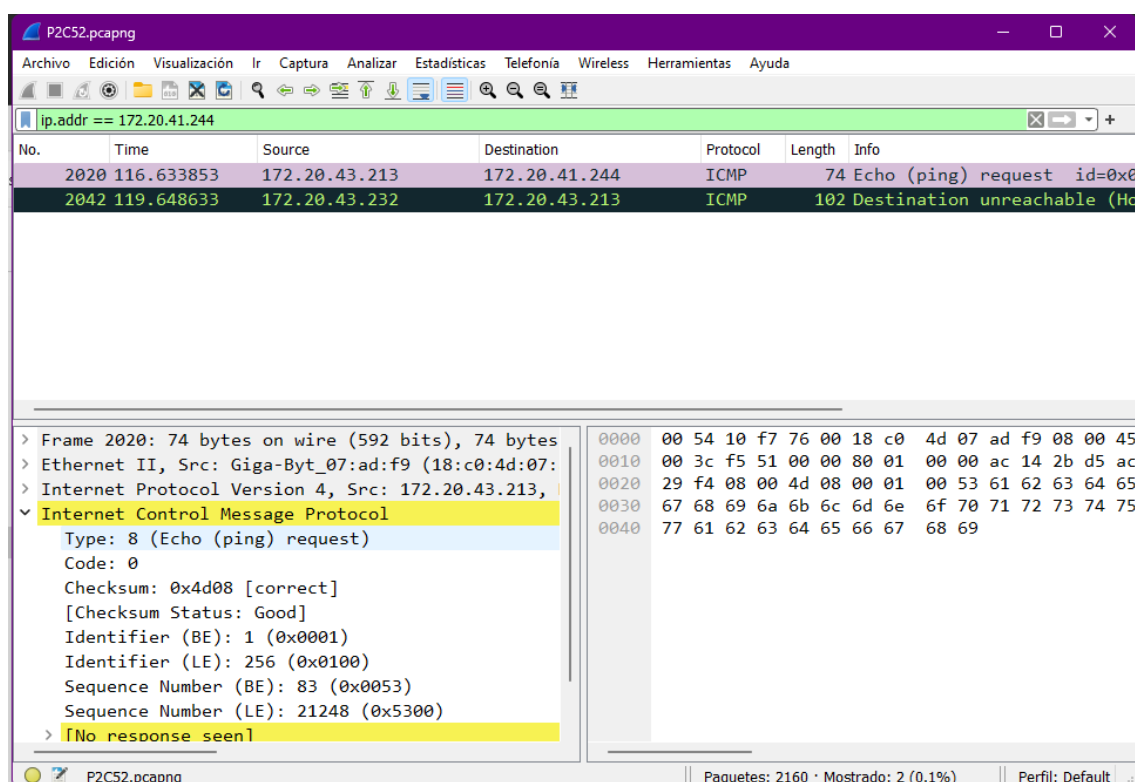
```
C:\>route add 172.20.41.240 mask 255.255.255.240 172.20.43.232
```

A continuación, ejecuta el comando:

```
C:\>ping -n 1 172.20.41.244
```

En base a los paquetes capturados, indicar:

a) Identificar las direcciones IP/MAC de los paquetes involucrados.



b) ¿Qué estación envía el mensaje ICMP Host Unreachable (3/1)?

```
Ethernet II, Src: Giga-Byt_07:ad:f9 (18:c0:4d:07:ad:f9)
  Destination: 00:54:10:f7:76:00 (00:54:10:f7:76:00)
  Source: Giga-Byt_07:ad:f9 (18:c0:4d:07:ad:f9)
  Type: IPv4 (0x0800)
```

Al finalizar este

ejercicio elimina la ruta añadida con el comando: C:\>route delete 172.20.41.240

CUESTIÓN 6

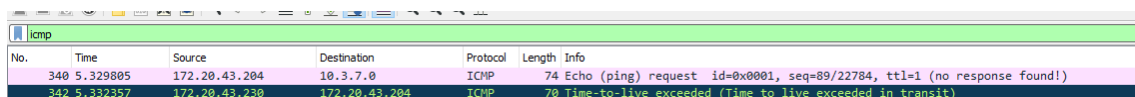
Detener la captura y determinar:

```
C:\>ping -i 1 -n 1 10.3.7.0

Haciendo ping a 10.3.7.0 con 32 bytes de datos:
Respuesta desde 172.20.43.230: TTL expirado en tránsito.

Estadísticas de ping para 10.3.7.0:
    Paquetes: enviados = 1, recibidos = 1, perdidos = 0
    (0% perdidos),
```

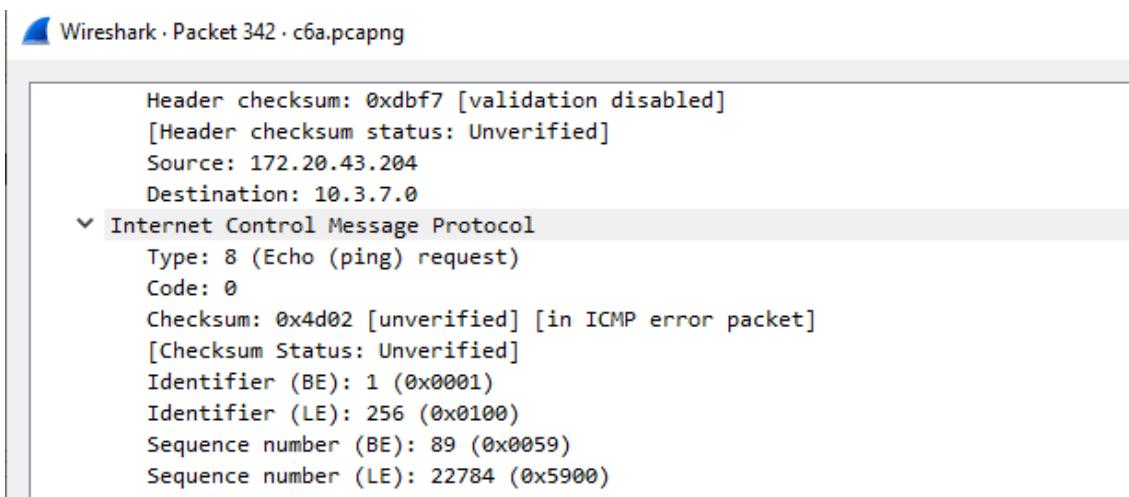
a) ¿Qué estación envía el mensaje ICMP *Time to Live exceeded in Transit*?



No.	Time	Source	Destination	Protocol	Length	Info
340	5.329805	172.20.43.204	10.3.7.0	ICMP	74	Echo (ping) request id=0x0001, seq=89/22784, ttl=1 (no response found!)
342	5.332357	172.20.43.230	172.20.43.204	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)

La estación "172.20.43.230" que en el laboratorio es la "R2".

b) ¿Qué paquete causó el error



Wireshark · Packet 342 · c6a.pcapng

Header checksum: 0xdbf7 [validation disabled]
[Header checksum status: Unverified]
Source: 172.20.43.204
Destination: 10.3.7.0

▼ Internet Control Message Protocol

Type: 8 (Echo (ping) request)
Code: 0
Checksum: 0x4d02 [unverified] [in ICMP error packet]
[Checksum Status: Unverified]
Identifier (BE): 1 (0x0001)
Identifier (LE): 256 (0x0100)
Sequence number (BE): 89 (0x0059)
Sequence number (LE): 22784 (0x5900)

Un echo ping request.

c) ¿Cuántos paquetes ICMP aparecen en el monitor de red?

Como se ve en la captura del apartado a, aparecen 2 paquetes icmp.

Iniciar de nuevo la captura y ejecutar a continuación el comando: **C:\> ping -i 2 -n 1 10.3.7.0**

Detener la captura y determinar:

```
C:\>ping -i 2 -n 1 10.3.7.0
```

```
Haciendo ping a 10.3.7.0 con 32 bytes de datos:  
Respuesta desde 10.4.2.5: TTL expirado en tránsito.
```

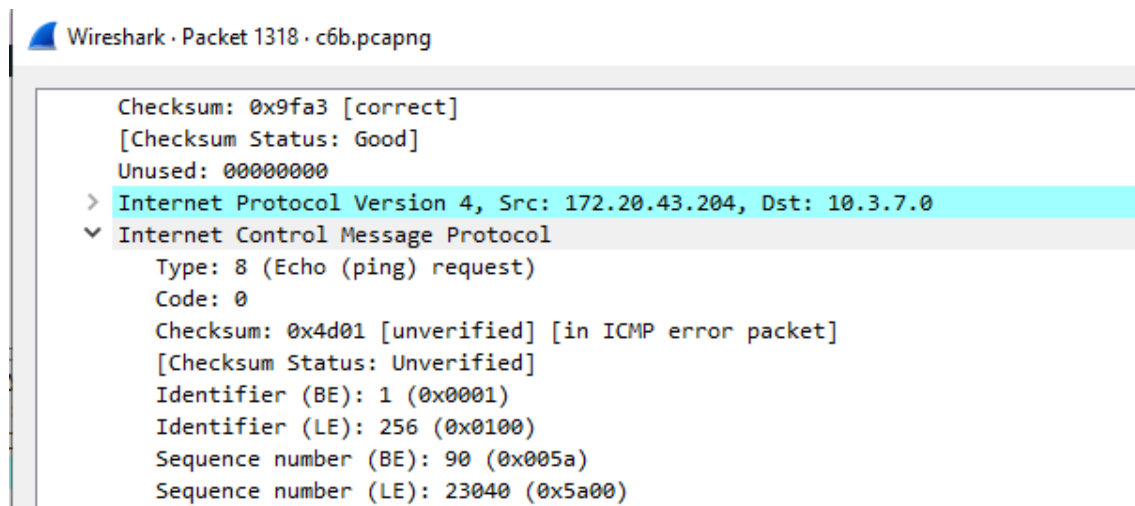
```
Estadísticas de ping para 10.3.7.0:  
Paquetes: enviados = 1, recibidos = 1, perdidos = 0  
(0% perdidos),
```

a) ¿Qué estación envía el mensaje ICMP *TTL exceeded*?

No.	Time	Source	Destination	Protocol	Length	Info
1316	16.041521	172.20.43.204	10.3.7.0	ICMP	74	Echo (ping) request id=0x0001, seq=90/23040, ttl=2 (no response found!)
1318	16.077349	10.4.2.5	172.20.43.204	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)

La estación “10.4.2.5” que en el laboratorio es la “R1”.

¿Qué paquete causó el error?



El Echo ping request.

c) ¿Cuántos paquetes ICMP aparecen en el monitor de red ?

2 como se ha visto anteriormente.

d) Justifica la diferencia con el caso anterior.

Me salen los mismos paquetes.

Emplear la aplicación **tracert** para determinar las rutas que siguen los paquetes IP dirigidos desde la red de los alumnos a diferentes destinos del laboratorio:

```
C:\tracert -d 10.3.7.0
```

```
C:\>tracert -d 10.3.7.0
```

```
Traza a 10.3.7.0 sobre caminos de 30 saltos como máximo.
```

1	7 ms	8 ms	9 ms	172.20.43.230
2	52 ms	50 ms	60 ms	10.4.2.5
3	253 ms	262 ms	252 ms	10.3.7.0

```
Traza completa.
```

```
C:\>tracert -d 172.20.41.241
```

```
C:\>tracert -d 172.20.41.241
```

```
Traza a 172.20.41.241 sobre caminos de 30 saltos como máximo.
```

1	2 ms	9 ms	9 ms	172.20.43.230
2	18 ms	22 ms	19 ms	10.4.2.2
3	259 ms	242 ms	241 ms	172.20.41.241

```
Traza completa.
```

```
C:\>tracert -d 10.4.2.5
```

```
C:\>tracert -d 10.4.2.5
```

```
Traza a 10.4.2.5 sobre caminos de 30 saltos como máximo.
```

1	3 ms	9 ms	9 ms	172.20.43.230
2	26 ms	39 ms	39 ms	10.4.2.5

```
Traza completa.
```

```
C:\>tracert -d 10.4.2.1
```

```
C:\>tracert -d 10.4.2.1
```

```
Traza a 10.4.2.1 sobre caminos de 30 saltos como máximo.
```

1	7 ms	9 ms	9 ms	172.20.43.230
2	41 ms	29 ms	50 ms	10.4.2.5
3	29 ms	29 ms	19 ms	10.4.2.1

```
Traza completa.
```

C:\tracert -d 10.4.2.2

```
C:\>tracert -d 10.4.2.2

Traza a 10.4.2.2 sobre caminos de 30 saltos como máximo.

 1      3 ms      9 ms      9 ms  172.20.43.230
 2     30 ms     29 ms     30 ms  10.4.2.2

Traza completa.
```

C:\tracert -d 172.20.41.242

```
C:\>tracert -d 172.20.41.242

Traza a 172.20.41.242 sobre caminos de 30 saltos como máximo.

 1      9 ms      9 ms      9 ms  172.20.43.230
 2     32 ms     29 ms     50 ms  10.4.2.5
 3    304 ms    242 ms    487 ms  10.3.7.0
 4    127 ms    140 ms    130 ms  172.20.41.242

Traza completa.
```

Determina las razones del comportamiento del encaminamiento al ejecutar el comando:

```

C:\>tracert -d 10.3.4.4

Traza a 10.3.4.4 sobre caminos de 30 saltos como máximo.

 1      2 ms      8 ms      9 ms  172.20.43.230
 2     30 ms     29 ms     29 ms  10.4.2.5
 3    313 ms    243 ms    242 ms  10.3.7.0
 4    244 ms    232 ms    232 ms  10.3.2.0
 5    476 ms    455 ms    454 ms  10.3.7.0
 6    443 ms    454 ms    465 ms  10.3.2.0
 7    676 ms    668 ms    689 ms  10.3.7.0
 8    661 ms    688 ms    667 ms  10.3.2.0
 9    876 ms    871 ms    881 ms  10.3.7.0
10    893 ms    892 ms    882 ms  10.3.2.0
11   1102 ms   1106 ms   1096 ms  10.3.7.0
12   1103 ms   1093 ms   1083 ms  10.3.2.0
13   1305 ms   1314 ms   1306 ms  10.3.7.0
14   1315 ms   1295 ms   1304 ms  10.3.2.0
15   1516 ms   1518 ms   1520 ms  10.3.7.0
16   1505 ms   1511 ms   1509 ms  10.3.2.0
17   1719 ms   1728 ms   1729 ms  10.3.7.0
18   1711 ms   1742 ms   1722 ms  10.3.2.0
19   1955 ms   1916 ms   1919 ms  10.3.7.0
20   1942 ms   1932 ms   1950 ms  10.3.2.0
21   2130 ms   2130 ms   2130 ms  10.3.7.0
22   2127 ms   2156 ms   2149 ms  10.3.2.0
23   2382 ms   2344 ms   2373 ms  10.3.7.0
24   2358 ms   2366 ms   2358 ms  10.3.2.0
25   2559 ms   2561 ms   2600 ms  10.3.7.0
26   2576 ms   2554 ms   2557 ms  10.3.2.0
27   2801 ms   2769 ms   2794 ms  10.3.7.0
28   2768 ms   2791 ms   2781 ms  10.3.2.0
29   3003 ms   3003 ms   2992 ms  10.3.7.0
30   3010 ms   2979 ms   2978 ms  10.3.2.0

Traza completa.

C:\>

```

Al ser una dirección que no coincide con ninguna puerta de enlace del aula de laboratorio como las demás anteriores, el paquete llega a la dirección "10.3.7.0" y dicha dirección piensa que la dirección final la tiene la "10.3.2.0", la cual piensa al contrario, lo que causa que cada vez que llegue a una de las dos, vuelva a la otra inmediatamente generando un

CUESTIÓN 7

```
Haciendo ping a 10.3.7.0 con 572 bytes de datos:
Respuesta desde 10.3.7.0: bytes=572 tiempo=263ms TTL=253

Estadísticas de ping para 10.3.7.0:
    Paquetes: enviados = 1, recibidos = 1, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 263ms, Máximo = 263ms, Media = 263ms

C:\>ping -n 1 -l 572 10.4.2.5

Haciendo ping a 10.4.2.5 con 572 bytes de datos:
Respuesta desde 10.4.2.5: bytes=572 tiempo=73ms TTL=254

Estadísticas de ping para 10.4.2.5:
    Paquetes: enviados = 1, recibidos = 1, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 73ms, Máximo = 73ms, Media = 73ms
```

Total Length: 600

Para hacer las gráficas y apuntar los datos donde hemos hecho los cálculos se ha usado excel como se muestra en la siguiente captura de pantalla:

