



# Evaluación del Rendimiento del Protocolo IP en una Red de Datos

Bryan Esteven Ariza Palma

Sebastian Herrán

Universidad Sergio Arboleda

Carrera de Ciencias de la Computación e Inteligencia Artificial y Ingeniería Electrónica

## **Resumen**

Este documento presenta la evaluación del rendimiento del protocolo IP en una red de datos mediante el análisis de paquetes capturados con Wireshark. El análisis se centra en la latencia, la fragmentación de paquetes, la eficiencia del enrutamiento y la pérdida de paquetes, proporcionando una visión general de los problemas que podrían afectar el rendimiento de la red.

## **1. Introducción:**

Se presenta un análisis detallado del rendimiento del protocolo IP en una red simulada, basado en capturas de paquetes con Wireshark. Aunque el análisis se llevó a cabo en una red doméstica, se utilizó una configuración que simula el comportamiento de una red corporativa, empleando varios dispositivos para generar tráfico similar al que se encontraría en dicho entorno. De esta manera, se pretende identificar posibles problemas de rendimiento que podrían afectar la calidad del servicio en redes más grandes y complejas.

El análisis incluye la evaluación de métricas clave como la latencia, la fragmentación de paquetes, la eficiencia del enrutamiento y la pérdida de paquetes. A través de esta simulación, se busca identificar áreas de mejora en el rendimiento de la red y sugerir posibles soluciones.

## **2. Objetivos**

El principal objetivo de este trabajo es evaluar el rendimiento del protocolo IP mediante el análisis de paquetes capturados. Específicamente, se busca:

- Medir la latencia en la red.
- Identificar y analizar la fragmentación de paquetes.
- Evaluar la eficiencia de las rutas de enrutamiento.
- Determinar la tasa de pérdida de paquetes y su impacto en la calidad de la red.

## **3. Metodología**

El análisis del rendimiento del protocolo IP se realizó siguiendo un enfoque estructurado, utilizando Wireshark como herramienta principal para la captura y análisis de paquetes en la red. El proceso consistió en varias etapas, detalladas a continuación:

- Captura del tráfico: Wireshark se utilizó para capturar el tráfico de red durante un período de 30 minutos en momentos de alta demanda, replicando el comportamiento de una red corporativa en plena actividad. La captura de paquetes se realizó en el punto de acceso de la red Wi-Fi, lo que permitió obtener tanto el tráfico interno como el externo. Durante la captura, se utilizaron filtros específicos para concentrarse en el tráfico IP, asegurando que los datos recogidos fueran relevantes para el análisis. Asimismo, se diversificó varios filtros para tener en cuenta que hay variedad de paquetes en la transmisión de datos a través de Wireshark.

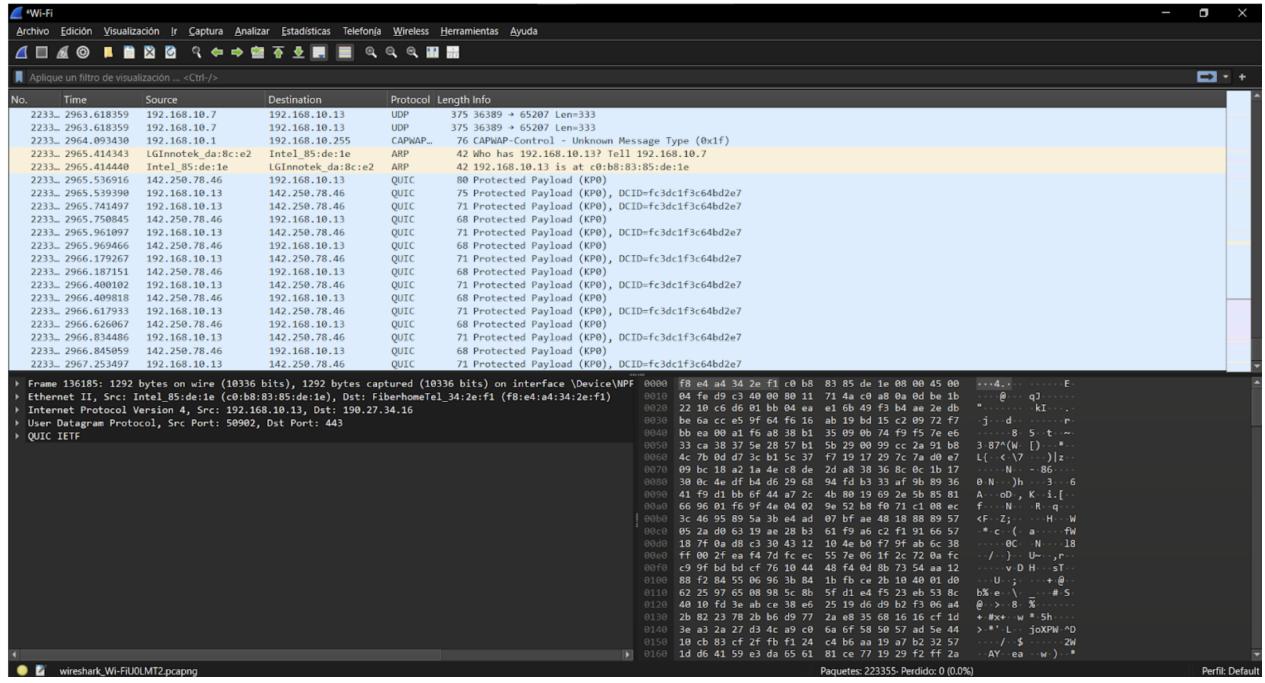


Figura 1: Screenshot de Tráfico general

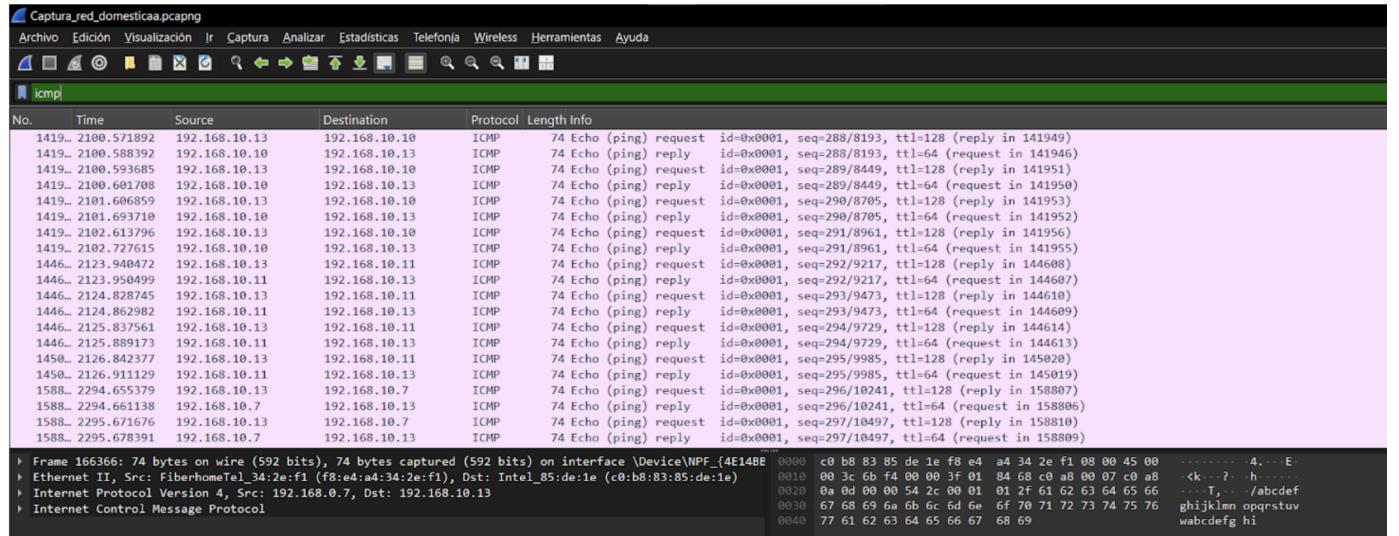


Figura 2: Tráfico ICMP.

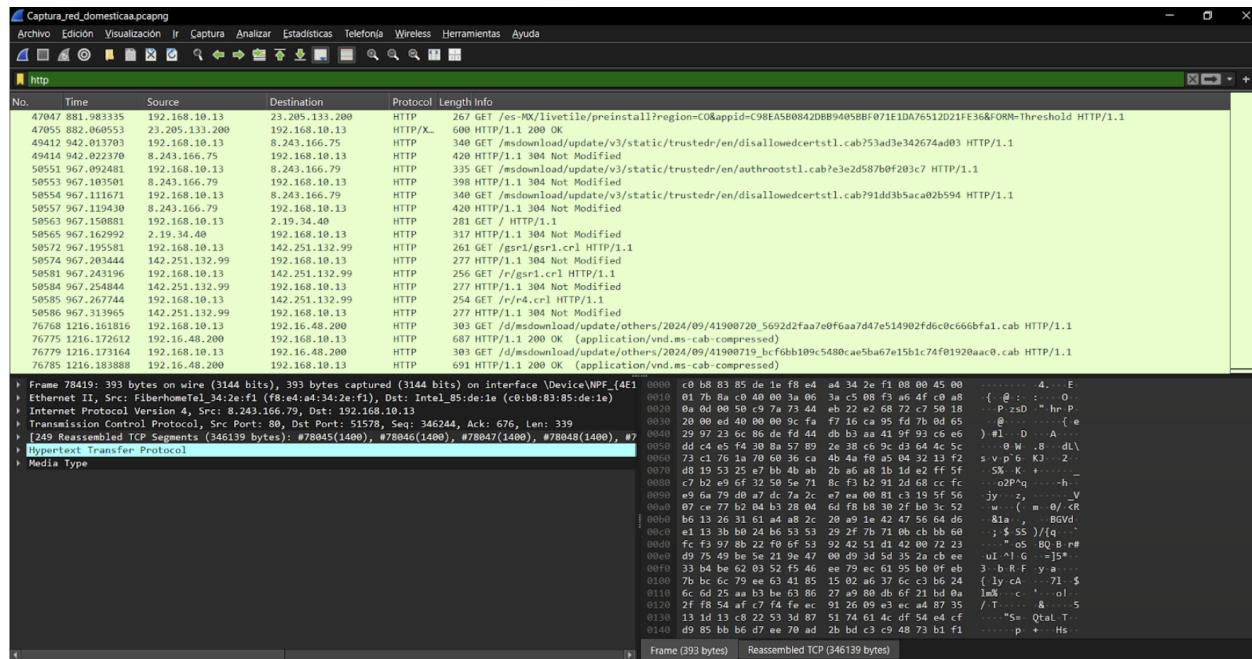


Figura 3: Aplicando filtro para paquetes HTTP

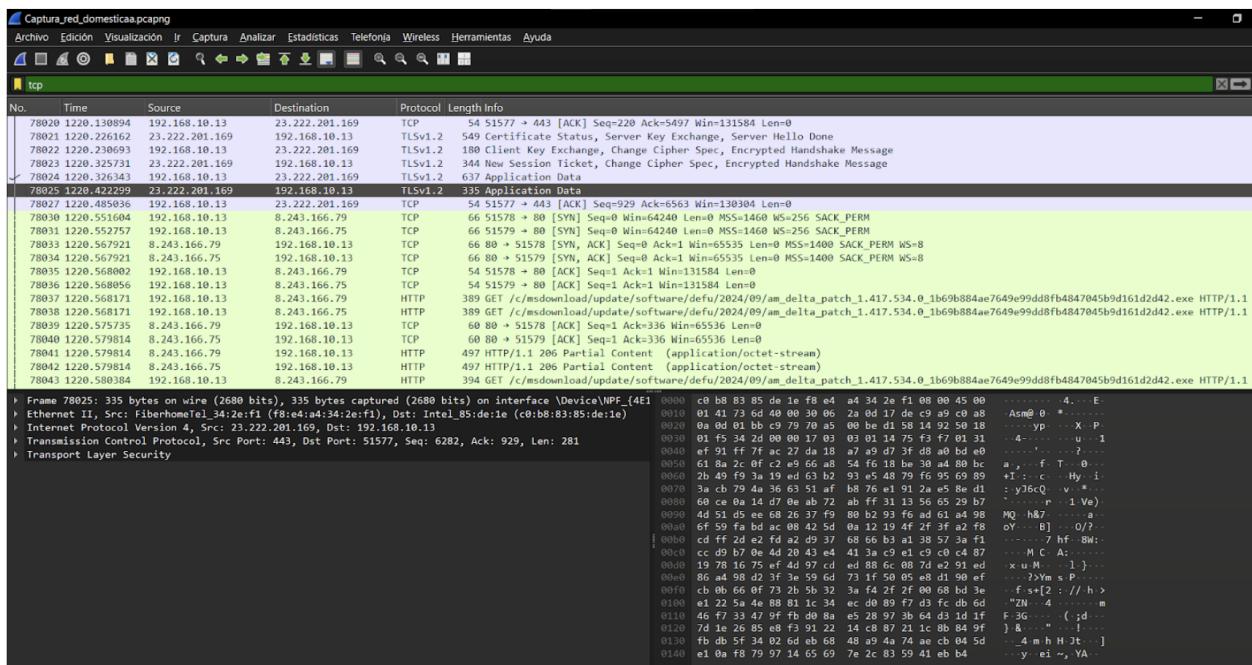


Figura 4: Aplicando filtro para paquetes TCP

### 3.1. Captura de Tráfico Interno

En las siguientes imágenes se puede observar el tráfico interno y externo de la red, capturado utilizando filtros correspondientes y así obtener todas las combinaciones de IP's dentro de la captura, aplicando un filtro general

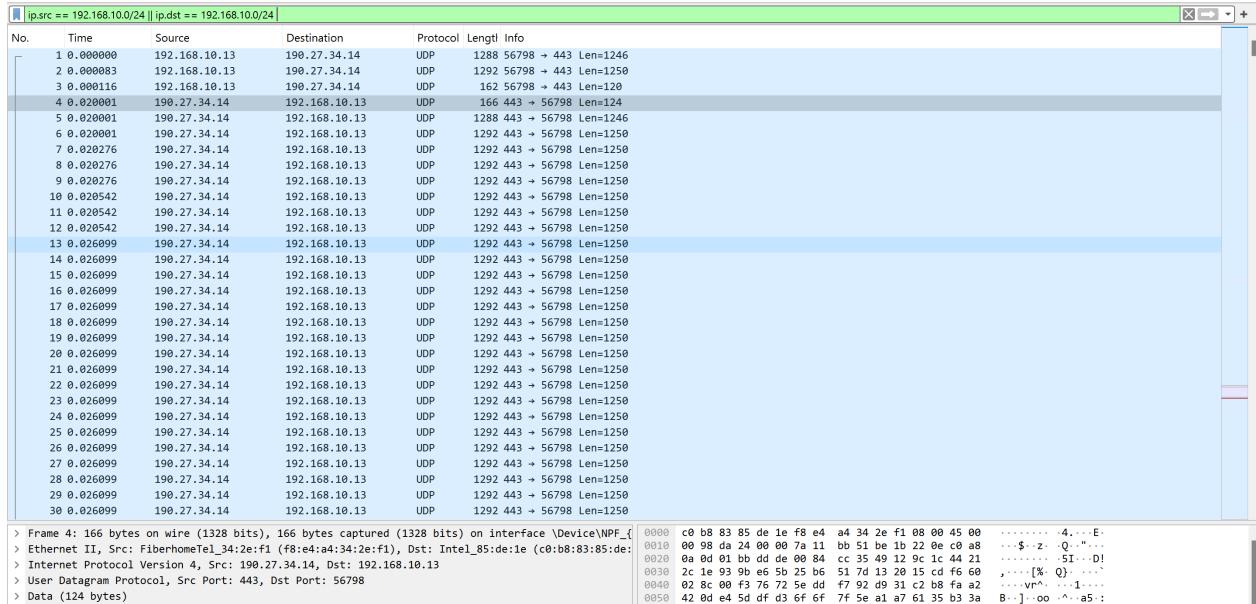


Figura 5: Captura de tráfico interno en Wireshark

### 3.2. Captura de Tráfico Externo

A continuación se presenta una captura del tráfico externo, con paquetes dirigidos hacia IPs públicas:

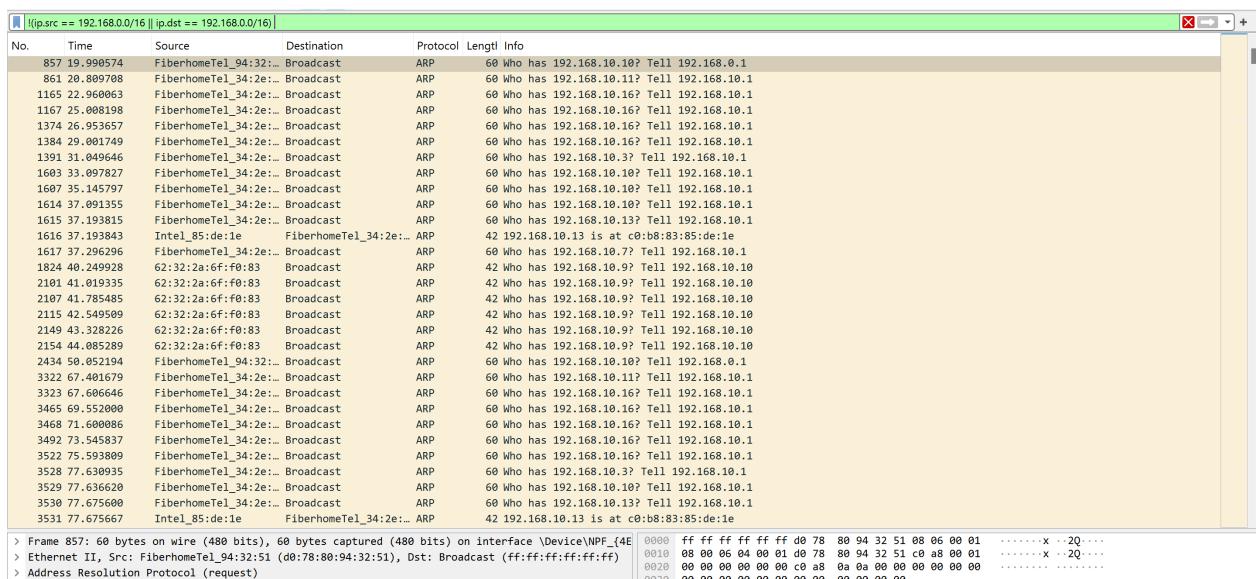


Figura 6: Captura de tráfico externo en Wireshark

Se verifica exitosamente los filtros, y dando a conocer los paquetes que se gestionaron en el tráfico interno y externo.

## 4. Análisis de Latencia

### 4.1. Importancia del Análisis de Latencia en la Evaluación del Rendimiento del Protocolo IP

El análisis de latencia, es medir el tiempo de ida y vuelta (RTT) de los paquetes, es de necesidad para evaluar el rendimiento de la red en varias formas:

- **Evaluación de la Experiencia del Usuario:** Un RTT bajo mejora la experiencia del usuario en aplicaciones sensibles al tiempo como videoconferencias y juegos en línea.
- **Detección de Problemas de Red:** Latencias altas pueden indicar problemas en la red, como congestión, fallos en el hardware, o problemas de enrutamiento.
- **Optimización de la Red:** Permite a los administradores de red ajustar configuraciones de enruteamiento para mejorar la eficiencia y estabilidad.
- **Comparación de Rendimiento:** Facilita la comparación entre diferentes nodos o redes, ayudando en la toma de decisiones sobre mejoras y ajustes.

Se procede a analizar la latencia por medio de los paquetes ICMP (Internet Control Message Protocol) para comprobar la conexión entre dispositivos y su latencia en ms.

The image shows two separate command-line windows from a Windows operating system. Both windows have a title bar 'Símbolo del sistema' and a status bar at the bottom indicating the path 'C:\Users\sebas>' followed by the command entered.

**Top Window (ping 192.168.10.10):**

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19045.4780]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\sebas>ping 192.168.10.10

Haciendo ping a 192.168.10.10 con 32 bytes de datos:
Respuesta desde 192.168.10.10: bytes=32 tiempo=1039ms TTL=64
Respuesta desde 192.168.10.10: bytes=32 tiempo=8ms TTL=64
Respuesta desde 192.168.10.10: bytes=32 tiempo=87ms TTL=64
Respuesta desde 192.168.10.10: bytes=32 tiempo=114ms TTL=64

Estadísticas de ping para 192.168.10.10:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
              (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 8ms, Máximo = 1039ms, Media = 312ms
```

**Bottom Window (ping 192.168.10.11):**

```
C:\Users\sebas>ping 192.168.10.11

Haciendo ping a 192.168.10.11 con 32 bytes de datos:
Respuesta desde 192.168.10.11: bytes=32 tiempo=130ms TTL=64
Respuesta desde 192.168.10.11: bytes=32 tiempo=34ms TTL=64
Respuesta desde 192.168.10.11: bytes=32 tiempo=51ms TTL=64
Respuesta desde 192.168.10.11: bytes=32 tiempo=69ms TTL=64

Estadísticas de ping para 192.168.10.11:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
              (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 34ms, Máximo = 130ms, Media = 71ms
```

Figura 7: Al expandir las pruebas, se observó una conexión estable a 192.168.10.11, mientras que 192.168.10.10 presentó fluctuaciones en la latencia

```
C:\Users\sebas>ping 192.168.10.7

Haciendo ping a 192.168.10.7 con 32 bytes de datos:
Respuesta desde 192.168.10.7: bytes=32 tiempo=5ms TTL=64
Respuesta desde 192.168.10.7: bytes=32 tiempo=6ms TTL=64
Respuesta desde 192.168.10.7: bytes=32 tiempo=13ms TTL=64
Respuesta desde 192.168.10.7: bytes=32 tiempo=5ms TTL=64

Estadísticas de ping para 192.168.10.7:
  Paquetes: enviados = 4, recibidos = 4, perdidos = 0
              (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
  Mínimo = 5ms, Máximo = 13ms, Media = 7ms

C:\Users\sebas>ping 192.168.0.7

Haciendo ping a 192.168.0.7 con 32 bytes de datos:
Respuesta desde 192.168.0.7: bytes=32 tiempo=80ms TTL=63
Respuesta desde 192.168.0.7: bytes=32 tiempo=93ms TTL=63
Respuesta desde 192.168.0.7: bytes=32 tiempo=106ms TTL=63
Respuesta desde 192.168.0.7: bytes=32 tiempo=121ms TTL=63

Estadísticas de ping para 192.168.0.7:
  Paquetes: enviados = 4, recibidos = 4, perdidos = 0
              (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
  Mínimo = 80ms, Máximo = 121ms, Media = 100ms

C:\Users\sebas>
```

Figura 8: Las pruebas de ping indicaron una conexión estable a ambas direcciones, siendo la primera notablemente más rápida.

En este caso, el RTT se puede calcular mediante el análisis de paquetes ICMP (protocolo de ping).

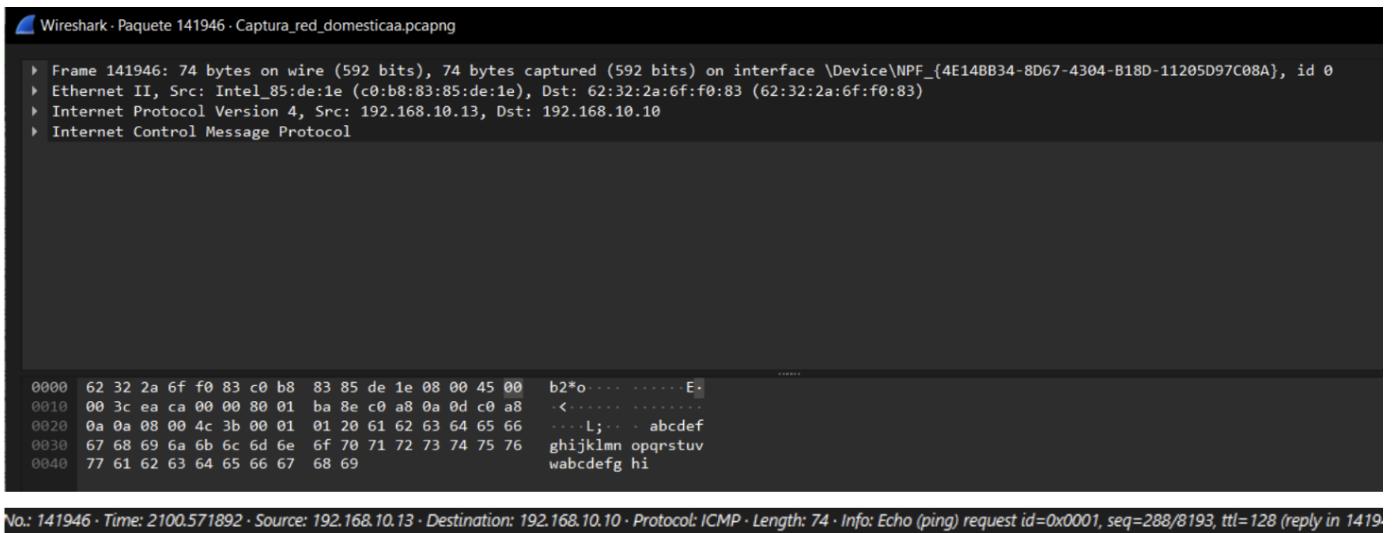


Figura 9: Echo Request: Es como un "ping" que se envía a un dispositivo para verificar si está activo y disponible.

```

Frame 141949: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{4E14BB34-8D67-4304-B18D-11205D97C08A}, id 0
Ethernet II, Src: 62:32:2a:6f:f0:83 (62:32:2a:6f:f0:83), Dst: Intel_85:de:1e (c0:b8:83:85:de:1e)
Internet Protocol Version 4, Src: 192.168.10.10, Dst: 192.168.10.13
Internet Control Message Protocol

0000  c0 b8 83 85 de 1e 62 32  2a 6f f0 83 08 00 45 00  .... b2 *o ..E.
0010  00 3c 03 db 00 00 40 01  e1 7e c0 a8 0a 0a c0 a8  <....@. ~....
0020  0a 0d 00 00 54 3b 00 01  01 20 61 62 63 64 65 66  ...T;.. abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76  ghiJKLMNOPQRSTUV
0040  77 61 62 63 64 65 66 67  68 69  wabcdefg hi

```

No.: 141949 · Time: 2100.588392 · Source: 192.168.10.10 · Destination: 192.168.10.13 · Protocol: ICMP · Length: 74 · Info: Echo (ping) reply id=0x0001, seq=288/8193, ttl=64 (request in 141946)

Figura 10: Echo Reply: Es la respuesta al Echo Request, confirmando que el dispositivo ha recibido el paquete y está funcionando.

No.	Time
→ 141946	2100.571892
← 141949	2100.588392

Figura 11: A continuación se muestra dos paquetes ICMP, el primero son mensajes del ICMP echo request y el segundo echo reply.

#### 4.2. RTT Calculado

Para el nodo **Computador a Celular 1**, se obtuvieron los siguientes RTT:

- RTT = 0.016500 segundos (16.5 ms)

Para los otros 3 pares de paquetes enviados a través del comando ping, se calcularon los siguientes RTT:

- RTT = 2100.601708000 - 2100.593685000 = 0.008023000 segundos (8.023 ms)
- RTT = 2101.693710000 - 2101.606859000 = 0.086851000 segundos (86.851 ms)
- RTT = 2102.727615000 - 2102.613796000 = 0.113819000 segundos (113.819 ms)

Para el nodo **Celular 2**, los RTT calculados fueron:

- RTT = 2123.950499000 - 2123.940472000 = 0.010027000 segundos (10.027 ms)

- RTT = 2124.862982000 - 2124.828745000 = 0.034237000 segundos (34.237 ms)
- RTT = 2125.889173000 - 2125.837561000 = 0.051612000 segundos (51.612 ms)
- RTT = 2126.911129000 - 2126.842377000 = 0.068752000 segundos (68.752 ms)

Para el nodo **TV**, se calcularon los siguientes RTT:

- RTT = 2294.661138000 - 2294.655379000 = 0.005759000 segundos (5.759 ms)
- RTT = 2295.678391000 - 2295.671676000 = 0.006715000 segundos (6.715 ms)
- RTT = 2296.697972000 - 2296.684841000 = 0.013131000 segundos (13.131 ms)
- RTT = 2297.700989000 - 2297.696032000 = 0.004957000 segundos (4.957 ms)

Para los nodos analizados, los RTT medios son los siguientes:

- **Celular 1:** 56.548 ms
- **Celular 2:** 41.182 ms
- **TV:** 7.640 ms

- Se puede identificar claramente que el TV tiene una latencia significativamente menor a la de los otros celulares, esto debido a que el TV está ubicado más cerca al router que los celulares, los cuales están conectados a través de repetidores.
- Se muestra la siguiente gráfica la cual equivale a un gráfico de caja, estas se usan para visualizar mejor la distribución de los datos y también la mediana de ellos, el rango de los valores para cada RTT etc.

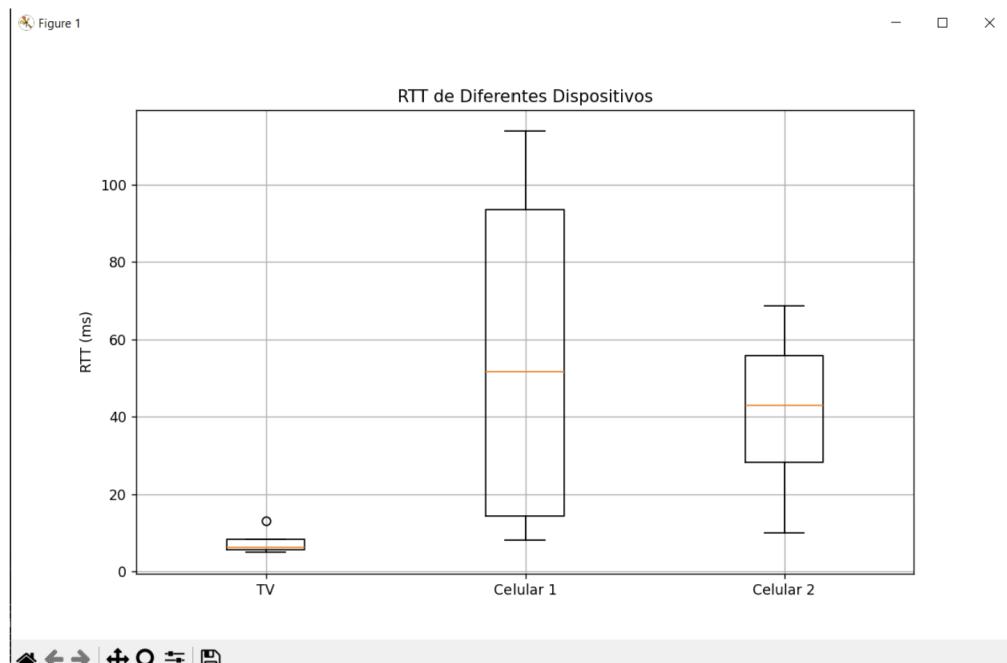


Figura 12: Interpretación de RTT en diferentes dispositivos.

## 5. Fragmentación

La fragmentación ocurre cuando los paquetes de datos son demasiado grandes para ser transmitidos en una sola unidad de red. Los paquetes se dividen en fragmentos más pequeños para su envío. Al llegar al destino, estos fragmentos se ensamblan nuevamente para reconstruir el paquete original.

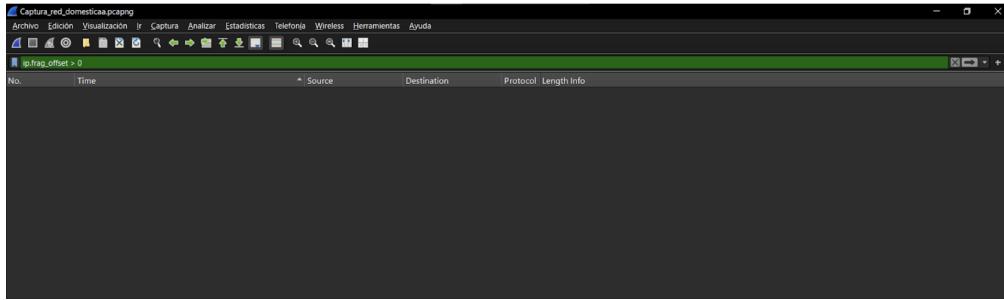


Figura 13: Verificación de paquetes fragmentados

- En este caso, al ser una red doméstica, y dado que no había la posibilidad de acceder a una red corporativa, y también considerando que, aunque la captura duró aproximadamente 1 hora, no se encontró ninguna fragmentación. Esto puede haber ocurrido también porque en ninguno de los nodos se excedió el MTU (Maximum Transmission Unit). Se cree que esto ocurrió debido a que no se estaba realizando ninguna acción pesada en la red; es decir, tal vez con una descarga de una aplicación o algo parecido se hubiera podido presentar una fragmentación. Sin embargo, como se pudo ver en la imagen anterior, al aplicar el filtro `ip.frag.offset > 0`, utilizado para filtrar alguna fragmentación, este no encontró ningún tipo de fragmentación y, por lo tanto, no hubo ningún paquete que tuviera necesidad de fragmentarse.
- En la siguiente imagen se pueden observar los paquetes con el mayor tamaño de toda la captura. Normalmente, la fragmentación de paquetes ocurre cuando el tamaño de un paquete excede el valor del MTU (Maximum Transmission Unit) de la red, que en muchas redes es de 1500 bytes. Dado que no se encontraron paquetes con un tamaño superior a 1500 bytes, no hay evidencia de fragmentación en los datos capturados. A continuación, se muestra la configuración del MTU en el PC:

No.	Time	Source	Destination	Protocol	Length	Info
223267	2953.335220	192.168.10.13	52.182.143.213	TCP	1454	51671 → 443 [ACK] Seq=8218 Ack=6360 Win=261376 Len=1400 [TCP PDU reassembled in 2...]
223266	2953.335220	192.168.10.13	52.182.143.213	TCP	1454	51671 → 443 [ACK] Seq=6818 Ack=6360 Win=261376 Len=1400 [TCP PDU reassembled in 2...]
223265	2953.335220	192.168.10.13	52.182.143.213	TCP	1454	51671 → 443 [ACK] Seq=5418 Ack=6360 Win=261376 Len=1400 [TCP PDU reassembled in 2...]
223264	2953.335220	192.168.10.13	52.182.143.213	TCP	1454	51671 → 443 [ACK] Seq=4018 Ack=6360 Win=261376 Len=1400 [TCP PDU reassembled in 2...]
223263	2953.335220	192.168.10.13	52.182.143.213	TCP	1454	51671 → 443 [ACK] Seq=2618 Ack=6360 Win=261376 Len=1400 [TCP PDU reassembled in 2...]
223262	2953.335220	192.168.10.13	52.182.143.213	TCP	1454	51671 → 443 [ACK] Seq=1218 Ack=6360 Win=261376 Len=1400 [TCP PDU reassembled in 2...]
223251	2953.228071	52.182.143.213	192.168.10.13	TCP	1454	443 → 51671 [ACK] Seq=4201 Ack=200 Win=4194048 Len=1400 [TCP PDU reassembled in 2...]
223250	2953.228071	52.182.143.213	192.168.10.13	TCP	1454	443 → 51671 [ACK] Seq=2801 Ack=200 Win=4194048 Len=1400 [TCP PDU reassembled in 2...]
223249	2953.228071	52.182.143.213	192.168.10.13	TCP	1454	443 → 51671 [ACK] Seq=1401 Ack=200 Win=4194048 Len=1400 [TCP PDU reassembled in 2...]
223248	2953.228071	52.182.143.213	192.168.10.13	TCP	1454	443 → 51671 [ACK] Seq=1 Ack=200 Win=4194048 Len=1400 [TCP PDU reassembled in 223...]
212291	2797.693637	216.239.32.3	192.168.10.13	TCP	1454	443 → 51670 [ACK] Seq=2801 Ack=1824 Win=71168 Len=1400 [TCP PDU reassembled in 21...]
212298	2797.693637	216.239.32.3	192.168.10.13	TCP	1454	443 → 51670 [PSH, ACK] Seq=1401 Ack=1824 Win=71168 Len=1400 [TCP PDU reassembled ...]
212289	2797.693637	216.239.32.3	192.168.10.13	TLSv1.3	1454	Server Hello, Change Cipher Spec
212276	2797.630282	192.168.10.13	216.239.32.3	TCP	1454	51670 → 443 [ACK] Seq=1 Ack=1 Win=131584 Len=1400 [TCP PDU reassembled in 212277]
210636	2782.415443	34.80.89.126	192.168.10.13	TCP	1454	443 → 51669 [ACK] Seq=2801 Ack=1824 Win=64128 Len=1400 [TCP PDU reassembled in 21...]
210635	2782.415443	34.80.89.126	192.168.10.13	TCP	1454	443 → 51669 [PSH, ACK] Seq=1401 Ack=1824 Win=64128 Len=1400 [TCP PDU reassembled ...]
210634	2782.413097	34.80.89.126	192.168.10.13	TLSv1.3	1454	Server Hello, Change Cipher Spec, Application Data
210630	2782.181875	192.168.10.13	34.80.89.126	TCP	1454	51669 → 443 [ACK] Seq=1 Ack=1 Win=131584 Len=1400 [TCP PDU reassembled in 210631]
209676	2775.558136	34.37.6.135	192.168.10.13	TCP	1454	443 → 51668 [ACK] Seq=2801 Ack=1761 Win=64128 Len=1400 [TCP PDU reassembled in 20...]
209675	2775.558136	34.37.6.135	192.168.10.13	TCP	1454	443 → 51668 [PSH, ACK] Seq=1401 Ack=1761 Win=64128 Len=1400 [TCP PDU reassembled ...]

Figura 14: Verificación de paquetes fragmentados

Se puede evidenciar que después de 1500 ya se tiene que fragmentar el paquete, dado que 1472 equivale

al MTU 1500, por que en el comando se debe aplicar con 28 bytes menos

C:\Users\sebas>netsh interface ipv4 show subinterfaces						
MTU	MediaSenseState	Bytes ent.	Bytes sal.	Interfaz		
4294967295		1	0	3215385	Loopback Pseudo-Interface 1	
1500		5	0	0	Conexión de red Bluetooth	
1500		1	44925294	1830392	Wi-Fi	
1500		2	11792929432	165526306	Ethernet	
1500		5	0	0	Conexión de área local* 9	
1500		5	0	0	Conexión de área local* 10	
1500		1	6620378		VirtualBox Host-Only Network	

```
C:\Users\sebas>ping -f -l 1472 192.168.10.1

Haciendo ping a 192.168.10.1 con 1472 bytes de datos:
Respuesta desde 192.168.10.1: bytes=1472 tiempo=12ms TTL=64
Respuesta desde 192.168.10.1: bytes=1472 tiempo=33ms TTL=64
Respuesta desde 192.168.10.1: bytes=1472 tiempo=137ms TTL=64
Respuesta desde 192.168.10.1: bytes=1472 tiempo=11ms TTL=64

Estadísticas de ping para 192.168.10.1:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
                (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 11ms, Máximo = 137ms, Media = 48ms

C:\Users\sebas>ping -f -l 1500 192.168.10.1

Haciendo ping a 192.168.10.1 con 1500 bytes de datos:
Es necesario fragmentar el paquete pero se especificó DF.
Es necesario fragmentar el paquete pero se especificó DF.
Es necesario fragmentar el paquete pero se especificó DF.
Es necesario fragmentar el paquete pero se especificó DF.

Estadísticas de ping para 192.168.10.1:
    Paquetes: enviados = 4, recibidos = 0, perdidos = 4
                (100% perdidos),
```

Figura 15: Evaulando MTU

## 6. Análisis de Enrutamiento

El análisis de enrutamiento sirve para evaluar la eficiencia y efectividad de las rutas que toman los paquetes a través de la red. Ayuda a identificar si los paquetes siguen rutas óptimas y si hay rutas inesperadas o subóptimas que podrían afectar negativamente el rendimiento y la velocidad de la red. En resumen, asegura que los paquetes viajen de la manera más eficiente posible desde el origen hasta el destino.

```

...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 128
Protocol: ICMP (1)
Header Checksum: 0x90a6 [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.168.10.13
Destination Address: 192.168.10.11
[Stream index: 175]
▼ Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
  Code: 0
  Checksum: 0x4c35 [correct]
  [Checksum Status: Good]
  Identifier (BE): 1 (0x0001)
  Identifier (LE): 256 (0x0100)

0000 44 59 e3 85 9d 3d c0 b8 83 85 de 1e 08 00 45 00 DY ...=... E.
0010 00 3c 14 b2 00 00 80 01 90 a6 c0 a8 0a 0d c0 a8 <..... .
0020 0a 0b 08 00 4c 35 00 01 01 26 61 62 63 64 65 66 ...L5... &abcdef
0030 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 ghijklmn opqrstuv
0040 77 61 62 63 64 65 66 67 68 69 wabcdefg hi

```

Figura 16: Detalles de paquetes Echo Request

- El paquete ICMP de tipo `echo request` mostrado en la imagen anterior tiene un TTL (Time to Live) inicial de 128, que es un valor estándar en Windows.
- Se analizará ahora el paquete ICMP de tipo `echo reply`.

Comparando el TTL del `echo request` con el TTL del `echo reply`, se puede inferir si el paquete pasó por otros routers.

- La diferencia entre los valores de TTL de ambos paquetes indica el número de saltos (hops) que el paquete ha realizado a través de la red.

```

0000 00.. = Differentiated Services Codepoint: Default (0)
.... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
Total Length: 60
Identification: 0x75ac (30124)
▶ 000. .... = Flags: 0x0
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 64
Protocol: ICMP (1)
Header Checksum: 0x6fac [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.168.10.11
Destination Address: 192.168.10.13
[Stream index: 175]
▼ Internet Control Message Protocol
  Type: 0 (Echo (ping) reply)

0000 c0 b8 83 85 de 1e 44 59 e3 85 9d 3d 08 00 45 00 ..... DY ...=.. E.
0010 00 3c 75 ac 00 00 40 01 6f ac c0 a8 0a 0b c0 a8 <u...@ o.... .
0020 0a 0d 00 00 54 35 00 01 01 26 61 62 63 64 65 66 ...T5... &abcdef
0030 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 ghijklmn opqrstuv
0040 77 61 62 63 64 65 66 67 68 69 wabcdefg hi

```

Figura 17: Detalles de paquetes Echo Reply

- Por lo que podemos deducir el número de saltos de la siguiente forma: Número de Saltos = TTL Inicial - TTL Final =  $128 - 64 = 64$
  - En el número de saltos se puede observar que disminuye en gran cantidad, esto puede ser debido al proveedor de servidores que se está conectando aunque también se sugiere que esto se deba a la configuración del TTL de cada dispositivo, pues en el dispositivo del echo request es un pc y en el dispositivo del echo reply es un celular. El valor de 64 TTL puede ser un valor normal para un dispositivo Android en este caso Huawei, por lo que se puede interpretar que estos saltos sean esperados.
  - Sin embargo, en el dispositivo TV (nodo lejano al PC) se puede observar que el TTL es de 63. De acuerdo a esto se puede interpretar que se está saltando 1 vez antes de llegar al TV. Esto tiene bastante sentido puesto a que el PC donde se hizo el ICMP está lejos del TV y de igual forma tienen un repetidor entre ambos, que puede equivaler a ese salto que fue percibido en el TTL. A continuación se muestra el TTL del TV:

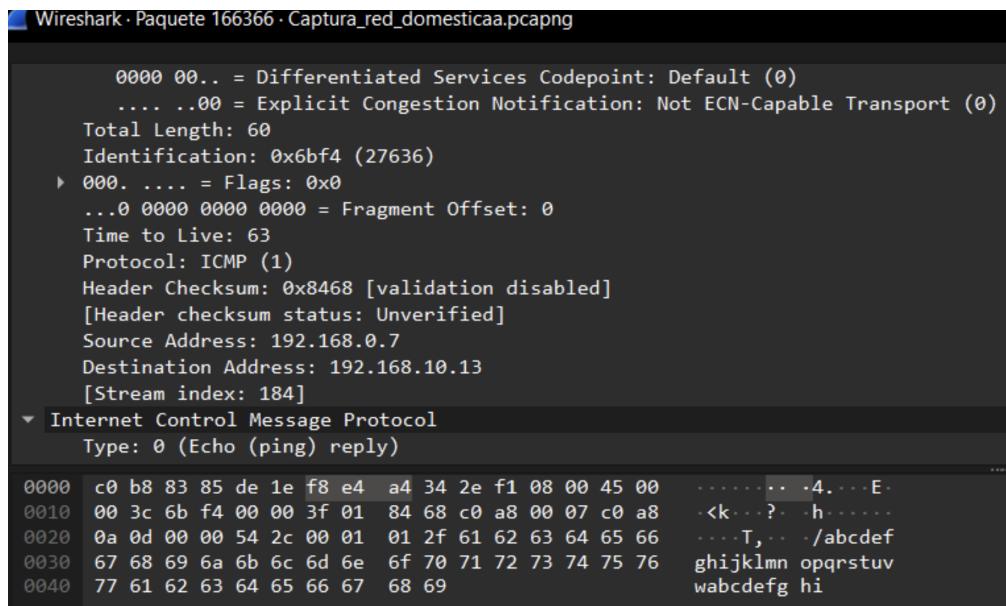


Figura 18: Paquete observación TTL TV

## 7. Análisis de Pérdida de Paquetes

La pérdida de paquetes es un fenómeno en redes de datos donde algunos paquetes de información enviados desde el origen no llegan a su destino. Esto puede ser causado por diversas razones, incluyendo congestión en la red, errores en el hardware, problemas en la configuración de la red o interferencias en el medio de transmisión. El análisis de pérdida de paquetes es crucial para evaluar el rendimiento y la calidad del servicio en una red.

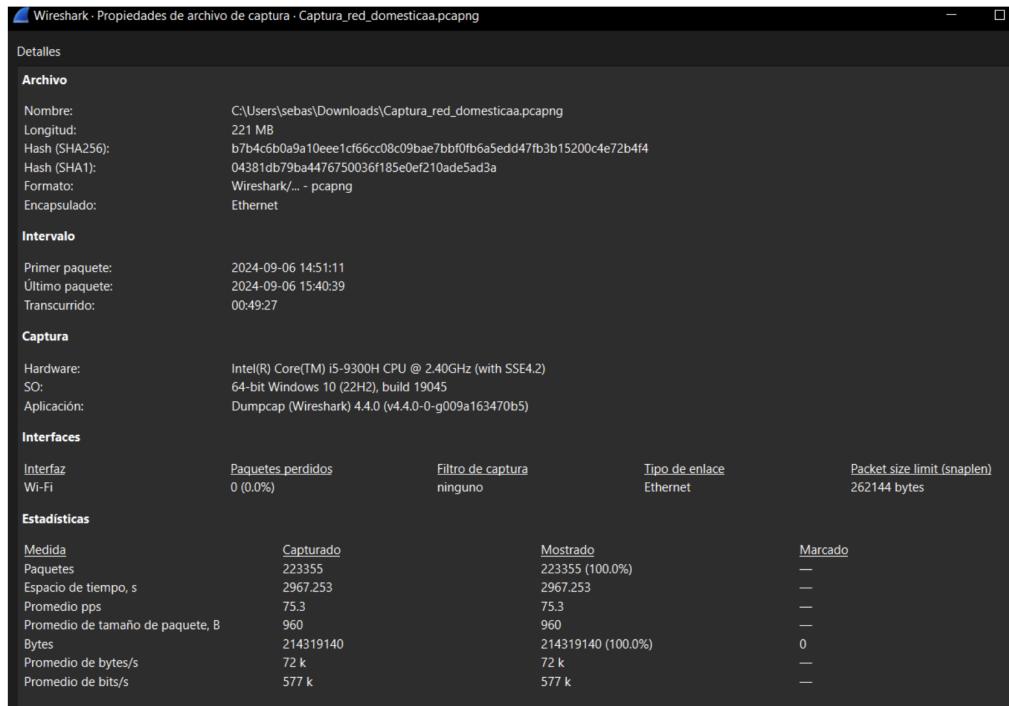


Figura 19: Detalles de perdida paquete

- En la imagen anterior, se puede evidenciar la pérdida de paquetes de captura, el tiempo de la captura y otros datos importantes. En este caso la captura fue exitosa, puesto a que no hubo ninguna pérdida de paquetes de ningún tipo, por lo tanto podemos inferir que la red estaba funcionando correctamente sin ningún cambio drástico que ocasionara la pérdida de paquetes entre nodos.

## 8. Conclusiones

En esta evaluación del rendimiento del protocolo IP, se abordaron varios temas para entender cómo funciona en una red de datos. A continuación, se resumen las principales conclusiones:

- **Latencia:** - La latencia medida en la red muestra variaciones dependiendo de la distancia y el tipo de dispositivo. Los dispositivos más cercanos al router, como el TV, tienen una latencia más baja (7.640 ms en promedio), mientras que los dispositivos más lejanos, como los celulares, presentan latencias mayores (entre 41.182 ms y 56.548 ms). Esto indica que la distancia y el número de repetidores afectan la velocidad de respuesta.
- **Fragmentación de Paquetes:** - Como la red no es excesivamente masiva, tampoco hay tanta probabilidad de que ocurra fragmentación, pues los dispositivos no estaban requiriendo de una transferencia de paquetes de gran longitud o que fuera más grande que el MTU para hacer una fragmentación de paquetes.
- **Enrutamiento:** - Se pudo interpretar el enrutamiento con la captura de wireshark por medio de los TTL (Time to live) que se analizaron en los paquetes ICMP, calculando la diferencia entre el TTL del echo request y el TTL del echo reply, se intentó obtener el MTU de configuración del router para saber con exactitud cual era pero no fue posible puesto a que al entrar en una página web para verificar el MTU no se pudo acceder. De igual forma se pudo analizar por medio de cmd el MTU del router y se interpretó que era de 1500 bytes, después de ese valor se tienen que fragmentar los paquetes.

- **Pérdida de Paquetes:** - La captura no mostró una pérdida significativa de paquetes, lo que indica que la red está operando de manera estable en términos de transmisión de datos. La pérdida de paquetes es un indicador importante del rendimiento de la red, y la ausencia de pérdidas sugiere que la calidad de la red es buena en las condiciones evaluadas.