

**DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN**



**LABORATORIO #1: INTRODUCCIÓN A REDES DE DATOS**

**ISIS3204 – INFRAESTRUCTURA DE COMUNICACIONES**

**PROFESOR**

**CARLOS LOZANO**

**GRUPO 6**

**MARÍA LUCÍA BENAVIDES DOMÍNGUEZ – 202313423**

**DANIEL CAMILO QUIMBAY VELÁSQUEZ - 202313861**

**NIKOL RODRIGUEZ ORTIZ – 202317538**

**2025-20**

**TABLA DE CONTENIDOS**

# Laboratorio 2 – Análisis de Protocolos con Wireshark

## 1. Objetivo

Describir brevemente que el objetivo es analizar el tráfico de red para diferentes protocolos de capa de aplicación (DNS, HTTP/HTTPS, FTP, VoIP, RTMP) usando Wireshark, tal como exige el laboratorio.

## 2. Capturas y Análisis Wireshark

### 2.1 Prueba de Conectividad (Ping)

#### 2.1.1 Prueba para Ping\_DNS\_IP

| No. | Time       | Source       | Destination  | Protocol | Length | Destination           | Source Address        | Destination Address | Type | Info  |
|-----|------------|--------------|--------------|----------|--------|-----------------------|-----------------------|---------------------|------|---|
| 793 | 13.875480  | 10.65.19.143 | 10.65.19.50  | ICMP     | 74     | Where_a1:69:e9        | CloudNetwork_3a:32:c0 | 10.65.19.143        | 0    | Echo (ping) request 10-dm0001, seq=28/7368, ttl=228 (no response found) |
| 794 | 13.875489  | 10.65.19.143 | 10.65.19.50  | ICMP     | 74     | Where_a1:69:e9        | CloudNetwork_3a:32:c0 | 10.65.19.143        | 0    | Echo (ping) request 10-dm0001, seq=28/7368, ttl=228 (reply in 795)      |
| 795 | 13.8761015 | 10.65.19.50  | 10.65.19.143 | ICMP     | 74     | CloudNetwork_3a:32:c0 | CloudNetwork_3a:32:c0 | 10.65.19.143        | 0    | Echo (ping) reply 10-dm0001, seq=28/7368, ttl=464 (request in 794)      |
| 796 | 13.8769103 | 10.65.19.50  | 10.65.19.143 | ICMP     | 74     | CloudNetwork_3a:32:c0 | CloudNetwork_3a:32:c0 | 10.65.19.143        | 0    | Echo (ping) reply 10-dm0001, seq=28/7368, ttl=464 (request in 794)      |
| 797 | 14.0001040 | 10.65.19.143 | 10.65.19.50  | ICMP     | 74     | Where_a1:69:e9        | CloudNetwork_3a:32:c0 | 10.65.19.143        | 0    | Echo (ping) request 10-dm0001, seq=29/7424, ttl=228 (no response found) |
| 798 | 14.0001040 | 10.65.19.50  | 10.65.19.143 | ICMP     | 74     | CloudNetwork_3a:32:c0 | CloudNetwork_3a:32:c0 | 10.65.19.143        | 0    | Echo (ping) request 10-dm0001, seq=29/7424, ttl=228 (reply in 799)      |
| 799 | 14.0007003 | 10.65.19.143 | 10.65.19.50  | ICMP     | 74     | CloudNetwork_3a:32:c0 | CloudNetwork_3a:32:c0 | 10.65.19.143        | 0    | Echo (ping) reply 10-dm0001, seq=29/7424, ttl=464 (request in 798)      |
| 800 | 14.0007003 | 10.65.19.143 | 10.65.19.50  | ICMP     | 74     | CloudNetwork_3a:32:c0 | CloudNetwork_3a:32:c0 | 10.65.19.143        | 0    | Echo (ping) reply 10-dm0001, seq=29/7424, ttl=464 (request in 798)      |
| 801 | 15.1807269 | 10.65.19.143 | 10.65.19.50  | ICMP     | 74     | Where_a1:69:e9        | CloudNetwork_3a:32:c0 | 10.65.19.143        | 0    | Echo (ping) request 10-dm0001, seq=30/7580, ttl=228 (no response found) |
| 802 | 15.1807269 | 10.65.19.143 | 10.65.19.50  | ICMP     | 74     | Where_a1:69:e9        | CloudNetwork_3a:32:c0 | 10.65.19.143        | 0    | Echo (ping) request 10-dm0001, seq=30/7580, ttl=228 (reply in 803)      |
| 803 | 15.1807269 | 10.65.19.50  | 10.65.19.143 | ICMP     | 74     | CloudNetwork_3a:32:c0 | CloudNetwork_3a:32:c0 | 10.65.19.143        | 0    | Echo (ping) reply 10-dm0001, seq=30/7580, ttl=464 (request in 802)      |
| 804 | 15.1807269 | 10.65.19.50  | 10.65.19.143 | ICMP     | 74     | CloudNetwork_3a:32:c0 | CloudNetwork_3a:32:c0 | 10.65.19.143        | 0    | Echo (ping) reply 10-dm0001, seq=30/7580, ttl=464 (request in 802)      |
| 805 | 16.1223374 | 10.65.19.143 | 10.65.19.50  | ICMP     | 74     | Where_a1:69:e9        | CloudNetwork_3a:32:c0 | 10.65.19.143        | 0    | Echo (ping) request 10-dm0001, seq=31/7736, ttl=228 (no response found) |
| 806 | 16.1223380 | 10.65.19.143 | 10.65.19.50  | ICMP     | 74     | Where_a1:69:e9        | CloudNetwork_3a:32:c0 | 10.65.19.143        | 0    | Echo (ping) request 10-dm0001, seq=31/7736, ttl=228 (reply in 807)      |
| 807 | 16.1224440 | 10.65.19.50  | 10.65.19.143 | ICMP     | 74     | CloudNetwork_3a:32:c0 | CloudNetwork_3a:32:c0 | 10.65.19.143        | 0    | Echo (ping) reply 10-dm0001, seq=31/7736, ttl=464 (request in 806)      |
| 808 | 16.1224437 | 10.65.19.50  | 10.65.19.143 | ICMP     | 74     | CloudNetwork_3a:32:c0 | CloudNetwork_3a:32:c0 | 10.65.19.143        | 0    | Echo (ping) reply 10-dm0001, seq=31/7736, ttl=464 (request in 806)      |

Figura 2.1.1.1 Wireshark llamadas DNS filtrando por ICMP con source, destination e info

|  |   |
|--|---|
| <p>Ethernet II, Src: CloudNetwork_3a:32:c0 (10:6f:d9:3a:32:c0), Dst: Where_a1:69:e9 (00:0c:29:a1:69:e9)</p> <p>Destination: Where_a1:69:e9 (00:0c:29:a1:69:e9)</p> <p>... .. 0 ... .. 16 bit: Globally unique address (factory default)</p> <p>... .. 0 ... .. 20 bit: Individual address (unicast)</p> <p>Source: CloudNetwork_3a:32:c0 (10:6f:d9:3a:32:c0)</p> <p>... .. 0 ... .. 16 bit: Globally unique address (factory default)</p> <p>... .. 0 ... .. 20 bit: Individual address (unicast)</p> <p>Type: IPv4 (0x0008)</p> <p>[Stream index: 6]</p> <p>Internet Protocol Version 4, Src: 10.65.19.143, Dst: 10.65.19.50</p> <p>0x00 ... = Version: 4</p> <p>... .. 0x01 ... = Header Length: 20 bytes (5)</p> <p>Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)</p> <p>Total Length: 68</p> <p>Identification: 0x0000 (32864)</p> <p>0x00 ... = Flags: 0x0</p> <p>... .. 0x00000000 ... = Fragment Offset: 0</p> <p>Time to live: 128</p> <p>Protocol: ICMP (1)</p> <p>Header Checksum: 0x15a6 (validation disabled)</p> <p>[Header checksum status: Unverified]</p> <p>Source Address: 10.65.19.143</p> <p>Destination Address: 10.65.19.50</p> <p>[Stream index: 11]</p> <p>Internet Control Message Protocol</p> <p>Type: 8 (Echo (ping) request)</p> <p>Code: 0</p> <p>Checksum: 0x4d3f [correct]</p> | <p>Ethernet II, Src: CloudNetwork_3a:32:c0 (10:6f:d9:3a:32:c0), Dst: CloudNetwork_3a:32:c0 (10:6f:d9:3a:32:c0)</p> <p>Destination: CloudNetwork_3a:32:c0 (10:6f:d9:3a:32:c0)</p> <p>... .. 0 ... .. 16 bit: Globally unique address (factory default)</p> <p>... .. 0 ... .. 20 bit: Individual address (unicast)</p> <p>Source: CloudNetwork_3a:32:c0 (10:6f:d9:3a:32:c0)</p> <p>... .. 0 ... .. 16 bit: Globally unique address (factory default)</p> <p>... .. 0 ... .. 20 bit: Individual address (unicast)</p> <p>Type: IPv4 (0x0008)</p> <p>[Stream index: 9]</p> <p>Internet Protocol Version 4, Src: 10.65.19.50, Dst: 10.65.19.143</p> <p>0x00 ... = Version: 4</p> <p>... .. 0x01 ... = Header Length: 20 bytes (5)</p> <p>Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)</p> <p>Total Length: 68</p> <p>Identification: 0x0000 (32864)</p> <p>0x00 ... = Flags: 0x0</p> <p>... .. 0x00000000 ... = Fragment Offset: 0</p> <p>Time to live: 64</p> <p>Protocol: ICMP (1)</p> <p>Header Checksum: 0x0d6a (validation disabled)</p> <p>[Header checksum status: Unverified]</p> <p>Source Address: 10.65.19.50</p> <p>Destination Address: 10.65.19.143</p> <p>[Stream index: 13]</p> <p>Internet Control Message Protocol</p> <p>Type: 8 (Echo (ping) reply)</p> <p>Code: 0</p> <p>Checksum: 0x4d3f [correct]</p> <p>[Checksum status: Good]</p> <p>Identifier (ID): 1 (0x0001)</p> <p>Sequence Number (Seq): 28 (0x001c)</p> <p>Sequence Number (Seq): 28 (0x001c)</p> <p>Sequence Number (Seq): 28 (0x001c)</p> |
|--|---|

Figura 2.1.1.2 Imágenes de información de mensajes de request y response

En las imágenes se puede apreciar de manera clara la forma de las peticiones cuando se inspeccionan a profundidad y con la información de estas, llegamos a la siguiente tabla

| Campo        | Request           | Reply             |
|--------------|-------------------|-------------------|
| IP origen    | 10.65.19.143      | 10.65.19.50       |
| IP destino   | 10.65.19.50       | 10.65.19.143      |
| MAC cliente  | 10:6f:d9:3a:32:cb | 10:6f:d9:3a:32:cb |
| MAC servidor | 00:0c:29:a1:69:e9 | 10:6f:d9:3a:32:cb |

## 2.1.2 Prueba para Ping\_FTP\_IP

| No. | Time     | Source       | Destination  | Protocol | Length | Destination           | Source                | Source Address | Destination Address | Type | Info                |
|-----|----------|--------------|--------------|----------|--------|-----------------------|-----------------------|----------------|---------------------|------|---------------------|
| 1   | 0.000000 | 10.65.19.143 | 10.65.19.52  | ICMP     | 74     | AzureWaveTec_30:45:08 | CloudNetwork_3a:32:c0 | 10.65.19.143   | 10.65.19.52         | 0    | Echo (ping) request |
| 2   | 0.000000 | 10.65.19.143 | 10.65.19.52  | ICMP     | 74     | AzureWaveTec_30:45:08 | CloudNetwork_3a:32:c0 | 10.65.19.143   | 10.65.19.52         | 8    | Echo (ping) reply   |
| 3   | 0.184973 | 10.65.19.52  | 10.65.19.143 | ICMP     | 74     | CloudNetwork_3a:32:c0 | AzureWaveTec_30:45:08 | 10.65.19.52    | 10.65.19.143        | 0    | Echo (ping) request |
| 7   | 0.000000 | 10.65.19.143 | 10.65.19.52  | ICMP     | 74     | AzureWaveTec_30:45:08 | CloudNetwork_3a:32:c0 | 10.65.19.143   | 10.65.19.52         | 8    | Echo (ping) reply   |
| 8   | 0.000597 | 10.65.19.143 | 10.65.19.52  | ICMP     | 74     | AzureWaveTec_30:45:08 | CloudNetwork_3a:32:c0 | 10.65.19.143   | 10.65.19.52         | 0    | Echo (ping) request |
| 11  | 0.052769 | 10.65.19.52  | 10.65.19.143 | ICMP     | 74     | CloudNetwork_3a:32:c0 | AzureWaveTec_30:45:08 | 10.65.19.52    | 10.65.19.143        | 0    | Echo (ping) request |
| 12  | 0.027656 | 10.65.19.143 | 10.65.19.52  | ICMP     | 74     | AzureWaveTec_30:45:08 | CloudNetwork_3a:32:c0 | 10.65.19.143   | 10.65.19.52         | 8    | Echo (ping) reply   |
| 13  | 0.027662 | 10.65.19.143 | 10.65.19.52  | ICMP     | 74     | AzureWaveTec_30:45:08 | CloudNetwork_3a:32:c0 | 10.65.19.143   | 10.65.19.52         | 0    | Echo (ping) request |
| 14  | 0.046664 | 10.65.19.52  | 10.65.19.143 | ICMP     | 74     | CloudNetwork_3a:32:c0 | AzureWaveTec_30:45:08 | 10.65.19.52    | 10.65.19.143        | 0    | Echo (ping) request |
| 15  | 0.051338 | 10.65.19.143 | 10.65.19.52  | ICMP     | 74     | AzureWaveTec_30:45:08 | CloudNetwork_3a:32:c0 | 10.65.19.143   | 10.65.19.52         | 8    | Echo (ping) reply   |
| 16  | 0.051348 | 10.65.19.143 | 10.65.19.52  | ICMP     | 74     | AzureWaveTec_30:45:08 | CloudNetwork_3a:32:c0 | 10.65.19.143   | 10.65.19.52         | 0    | Echo (ping) request |
| 17  | 0.051927 | 10.65.19.52  | 10.65.19.143 | ICMP     | 74     | CloudNetwork_3a:32:c0 | AzureWaveTec_30:45:08 | 10.65.19.52    | 10.65.19.143        | 0    | Echo (ping) request |

Figura 2.1.2.1 Wireshark llamadas FTP filtrando por ICMP con source, destination e info

|   |   |
|---|---|
| Frame 21: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface Vmnic0 (vswan404-4464-463) > Ethernet II Src: CloudNetwork_3a:32:c0 (10:65:19:3a:32:c0), Dst: AzureWaveTec_30:45:08 (10:65:19:30:45:08) > Destination: AzureWaveTec_30:45:08 (10:65:19:30:45:08) > Source: CloudNetwork_3a:32:c0 (10:65:19:3a:32:c0) > Type: IP (60) (Stream index: 6) | Frame 31: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface Vmnic0 (vswan404-4464-463) > Ethernet II Src: AzureWaveTec_30:45:08 (10:65:19:30:45:08), Dst: CloudNetwork_3a:32:c0 (10:65:19:3a:32:c0) > Destination: CloudNetwork_3a:32:c0 (10:65:19:3a:32:c0) > Source: AzureWaveTec_30:45:08 (10:65:19:30:45:08) > Type: IP (60) (Stream index: 6) |
| Internet Protocol Version 4, Src: 10.65.19.143, Dst: 10.65.19.52  | Internet Protocol Version 4, Src: 10.65.19.52, Dst: 10.65.19.143  |
| ... .. Version: 4   | ... .. Version: 4   |
| ... .. Header Length: 20 bytes (5)  | ... .. Header Length: 20 bytes (5)  |
| Differentiated Services Field (DSCP): CS, ECN: Not-ECT  | Differentiated Services Field: none (DSCP: CS, ECN: Not-ECT)  |
| Total Length: 60  | Total Length: 60  |
| Identification: 84254 (7986)  | Identification: 84254 (7986)  |
| ... .. Flags: none  | ... .. Flags: none  |
| ... .. none none none > Fragment Offset: 0  | ... .. none none none > Fragment Offset: 0  |
| Time to Live: 310   | Time to Live: 64  |
| Protocol: ICMP (1)  | Protocol: ICMP (1)  |
| Header Checksum: 8e2228 (validation disabled)   | Header Checksum: 8e2228 (validation disabled)   |
| Source Address: 10.65.19.143  | Source Address: 10.65.19.52   |
| Destination Address: 10.65.19.52  | Destination Address: 10.65.19.143   |
| (Stream index: 6)   | (Stream index: 6)   |
|   | Internet control message protocol   |

Figura 2.1.1.2 Imágenes de información de mensajes de request y response

En las imágenes se puede apreciar de manera clara la forma de las peticiones cuando se inspeccionan a profundidad y con la información de estas, llegamos a la siguiente tabla

| Campo        | Request           | Reply             |
|--------------|-------------------|-------------------|
| IP origen    | 10.65.19.143      | 10.65.19.52       |
| IP destino   | 10.65.19.52       | 10.65.19.143      |
| MAC cliente  | 10:6f:d9:3a:32:cb | 1c:ce:51:93:45:08 |
| MAC servidor | 1c:ce:51:93:45:08 | 10:6f:d9:3a:32:cb |

## 2.2 Servicio DNS

### 2.2.1 Archivo Ping\_WEB\_IP

| No. | Time      | Source       | Destination   | Protocol | Length | Destination           | Source                | Source Address | Destination Address | Type                    | Info |
|-----|-----------|--------------|---------------|----------|--------|-----------------------|-----------------------|----------------|---------------------|-------------------------|------|
| 23  | 4.726671  | 10.65.19.58  | 93.189.91.139 | DNS      | 128    | 4a:fd:f8:20:a8:12     | CloudNetwork_3a:32:c0 | 10.65.19.58    | 93.189.91.139       | Standard query 8x237 A  |      |
| 24  | 4.728865  | 10.65.19.58  | 93.189.91.139 | DNS      | 128    | 4a:fd:f8:20:a8:12     | CloudNetwork_3a:32:c0 | 10.65.19.58    | 93.189.91.139       | Standard query 8x237 A  |      |
| 25  | 5.815132  | 10.65.19.139 | 10.65.19.58   | DNS      | 387    | CloudNetwork_3a:32:c0 | 4a:fd:f8:20:a8:12     | 93.189.91.139  | 10.65.19.58         | Standard query response |      |
| 71  | 13.140897 | 10.65.19.143 | 10.65.19.197  | DNS      | 75     | 4a:fd:f8:20:a8:12     | CloudNetwork_3a:32:c0 | 10.65.19.143   | 10.65.19.197        | Standard query 8x785 A  |      |
| 72  | 13.140897 | 10.65.19.143 | 10.65.19.197  | DNS      | 75     | 4a:fd:f8:20:a8:12     | CloudNetwork_3a:32:c0 | 10.65.19.143   | 10.65.19.197        | Standard query 8x785 A  |      |
| 73  | 13.222738 | 10.65.19.143 | 10.65.19.197  | DNS      | 75     | 4a:fd:f8:20:a8:12     | CloudNetwork_3a:32:c0 | 10.65.19.143   | 10.65.19.197        | Standard query 8x785 A  |      |
| 75  | 13.222766 | 10.65.19.143 | 10.65.19.197  | DNS      | 75     | 4a:fd:f8:20:a8:12     | CloudNetwork_3a:32:c0 | 10.65.19.143   | 10.65.19.197        | Standard query 8x785 A  |      |
| 76  | 13.446684 | 10.65.19.197 | 10.65.19.143  | DNS      | 91     | CloudNetwork_3a:32:c0 | 4a:fd:f8:20:a8:12     | 10.65.19.197   | 10.65.19.143        | Standard query response |      |
| 77  | 13.446684 | 10.65.19.197 | 10.65.19.143  | DNS      | 91     | CloudNetwork_3a:32:c0 | 4a:fd:f8:20:a8:12     | 10.65.19.197   | 10.65.19.143        | Standard query response |      |
| 139 | 14.680238 | 10.65.19.143 | 10.65.19.197  | DNS      | 96     | 4a:fd:f8:20:a8:12     | CloudNetwork_3a:32:c0 | 10.65.19.143   | 10.65.19.197        | Standard query 8x785 A  |      |
| 140 | 14.680247 | 10.65.19.143 | 10.65.19.197  | DNS      | 96     | 4a:fd:f8:20:a8:12     | CloudNetwork_3a:32:c0 | 10.65.19.143   | 10.65.19.197        | Standard query 8x785 A  |      |
| 144 | 14.761446 | 10.65.19.143 | 10.65.19.197  | DNS      | 96     | 4a:fd:f8:20:a8:12     | CloudNetwork_3a:32:c0 | 10.65.19.143   | 10.65.19.197        | Standard query 8x785 A  |      |
| 145 | 14.761446 | 10.65.19.143 | 10.65.19.197  | DNS      | 96     | 4a:fd:f8:20:a8:12     | CloudNetwork_3a:32:c0 | 10.65.19.143   | 10.65.19.197        | Standard query 8x785 A  |      |
| 146 | 14.907571 | 10.65.19.197 | 10.65.19.143  | DNS      | 112    | CloudNetwork_3a:32:c0 | 4a:fd:f8:20:a8:12     | 10.65.19.197   | 10.65.19.143        | Standard query response |      |
| 147 | 14.907571 | 10.65.19.197 | 10.65.19.143  | DNS      | 112    | CloudNetwork_3a:32:c0 | 4a:fd:f8:20:a8:12     | 10.65.19.197   | 10.65.19.143        | Standard query response |      |
| 201 | 15.188158 | 10.65.19.143 | 10.65.19.197  | DNS      | 75     | 4a:fd:f8:20:a8:12     | CloudNetwork_3a:32:c0 | 10.65.19.143   | 10.65.19.197        | Standard query 8x866 A  |      |
| 202 | 15.385173 | 10.65.19.143 | 10.65.19.197  | DNS      | 75     | 4a:fd:f8:20:a8:12     | CloudNetwork_3a:32:c0 | 10.65.19.143   | 10.65.19.197        | Standard query 8x866 A  |      |
| 205 | 15.464338 | 10.65.19.143 | 10.65.19.197  | DNS      | 75     | 4a:fd:f8:20:a8:12     | CloudNetwork_3a:32:c0 | 10.65.19.143   | 10.65.19.197        | Standard query 8x866 A  |      |
| 207 | 15.464338 | 10.65.19.143 | 10.65.19.197  | DNS      | 75     | 4a:fd:f8:20:a8:12     | CloudNetwork_3a:32:c0 | 10.65.19.143   | 10.65.19.197        | Standard query 8x866 A  |      |
| 271 | 15.537882 | 10.65.19.197 | 10.65.19.143  | DNS      | 91     | CloudNetwork_3a:32:c0 | 4a:fd:f8:20:a8:12     | 10.65.19.197   | 10.65.19.143        | Standard query response |      |
| 272 | 15.538564 | 10.65.19.197 | 10.65.19.143  | DNS      | 91     | CloudNetwork_3a:32:c0 | 4a:fd:f8:20:a8:12     | 10.65.19.197   | 10.65.19.143        | Standard query response |      |
| 301 | 15.654673 | 10.65.19.143 | 10.65.19.197  | DNS      | 79     | 4a:fd:f8:20:a8:12     | CloudNetwork_3a:32:c0 | 10.65.19.143   | 10.65.19.197        | Standard query 8x968 A  |      |
| 304 | 15.654891 | 10.65.19.143 | 10.65.19.197  | DNS      | 79     | 4a:fd:f8:20:a8:12     | CloudNetwork_3a:32:c0 | 10.65.19.143   | 10.65.19.197        | Standard query 8x968 A  |      |
| 308 | 15.724818 | 10.65.19.143 | 10.65.19.197  | DNS      | 79     | 4a:fd:f8:20:a8:12     | CloudNetwork_3a:32:c0 | 10.65.19.143   | 10.65.19.197        | Standard query 8x968 A  |      |
| 309 | 15.724896 | 10.65.19.143 | 10.65.19.197  | DNS      | 79     | 4a:fd:f8:20:a8:12     | CloudNetwork_3a:32:c0 | 10.65.19.143   | 10.65.19.197        | Standard query 8x968 A  |      |
| 320 | 15.793166 | 10.65.19.197 | 10.65.19.143  | DNS      | 86     | CloudNetwork_3a:32:c0 | 4a:fd:f8:20:a8:12     | 10.65.19.197   | 10.65.19.143        | Standard query response |      |

Figura 2.2.1.1 Wireshark llamadas DNS filtrando por ICMP con source, destination e info

Debido a la imagen se puede ver de manera clara las request y los replies que se hicieron con el protocolo DNS con su información más importante. Estos son todos los archivos capturados con DNS.

La información de la capa de aplicación que se puede obtener de este apartado es:

# 1. Las peticiones

| dns.flags.response == 0 |           |              |               |          |        |                   |                       |                |                     |
|-------------------------|-----------|--------------|---------------|----------|--------|-------------------|-----------------------|----------------|---------------------|
| No.                     | Time      | Source       | Destination   | Protocol | Length | Destination       | Source                | Source Address | Destination Address |
| 23                      | 4.728671  | 10.65.19.50  | 91.189.91.139 | DNS      | 128    | 4a:f5:e8:2b:a8:12 | CloudNetwork_3a:32:cb | 10.65.19.50    | 91.189.91.139       |
| 24                      | 4.728685  | 10.65.19.50  | 91.189.91.139 | DNS      | 128    | 4a:f5:e8:2b:a8:12 | CloudNetwork_3a:32:cb | 10.65.19.50    | 91.189.91.139       |
| 71                      | 13.148927 | 10.65.19.143 | 10.65.19.197  | DNS      | 75     | 4a:f5:e8:2b:a8:12 | CloudNetwork_3a:32:cb | 10.65.19.143   | 10.65.19.197        |
| 72                      | 13.148997 | 10.65.19.143 | 10.65.19.197  | DNS      | 75     | 4a:f5:e8:2b:a8:12 | CloudNetwork_3a:32:cb | 10.65.19.143   | 10.65.19.197        |
| 74                      | 13.223730 | 10.65.19.143 | 10.65.19.197  | DNS      | 75     | 4a:f5:e8:2b:a8:12 | CloudNetwork_3a:32:cb | 10.65.19.143   | 10.65.19.197        |
| 75                      | 13.223746 | 10.65.19.143 | 10.65.19.197  | DNS      | 75     | 4a:f5:e8:2b:a8:12 | CloudNetwork_3a:32:cb | 10.65.19.143   | 10.65.19.197        |
| 139                     | 14.689230 | 10.65.19.143 | 10.65.19.197  | DNS      | 96     | 4a:f5:e8:2b:a8:12 | CloudNetwork_3a:32:cb | 10.65.19.143   | 10.65.19.197        |
| 140                     | 14.689247 | 10.65.19.143 | 10.65.19.197  | DNS      | 96     | 4a:f5:e8:2b:a8:12 | CloudNetwork_3a:32:cb | 10.65.19.143   | 10.65.19.197        |
| 144                     | 14.761465 | 10.65.19.143 | 10.65.19.197  | DNS      | 96     | 4a:f5:e8:2b:a8:12 | CloudNetwork_3a:32:cb | 10.65.19.143   | 10.65.19.197        |
| 145                     | 14.761465 | 10.65.19.143 | 10.65.19.197  | DNS      | 96     | 4a:f5:e8:2b:a8:12 | CloudNetwork_3a:32:cb | 10.65.19.143   | 10.65.19.197        |
| 251                     | 15.385158 | 10.65.19.143 | 10.65.19.197  | DNS      | 75     | 4a:f5:e8:2b:a8:12 | CloudNetwork_3a:32:cb | 10.65.19.143   | 10.65.19.197        |
| 252                     | 15.385173 | 10.65.19.143 | 10.65.19.197  | DNS      | 75     | 4a:f5:e8:2b:a8:12 | CloudNetwork_3a:32:cb | 10.65.19.143   | 10.65.19.197        |
| 266                     | 15.456169 | 10.65.19.143 | 10.65.19.197  | DNS      | 75     | 4a:f5:e8:2b:a8:12 | CloudNetwork_3a:32:cb | 10.65.19.143   | 10.65.19.197        |
| 267                     | 15.456188 | 10.65.19.143 | 10.65.19.197  | DNS      | 75     | 4a:f5:e8:2b:a8:12 | CloudNetwork_3a:32:cb | 10.65.19.143   | 10.65.19.197        |
| 383                     | 15.656871 | 10.65.19.143 | 10.65.19.197  | DNS      | 78     | 4a:f5:e8:2b:a8:12 | CloudNetwork_3a:32:cb | 10.65.19.143   | 10.65.19.197        |
| 384                     | 15.656891 | 10.65.19.143 | 10.65.19.197  | DNS      | 78     | 4a:f5:e8:2b:a8:12 | CloudNetwork_3a:32:cb | 10.65.19.143   | 10.65.19.197        |
| 385                     | 15.724538 | 10.65.19.143 | 10.65.19.197  | DNS      | 78     | 4a:f5:e8:2b:a8:12 | CloudNetwork_3a:32:cb | 10.65.19.143   | 10.65.19.197        |
| 386                     | 15.724606 | 10.65.19.143 | 10.65.19.197  | DNS      | 78     | 4a:f5:e8:2b:a8:12 | CloudNetwork_3a:32:cb | 10.65.19.143   | 10.65.19.197        |

Figura 2.2.1.2 Wireshark filtrado para ver solo peticiones

| Campo        | Request           | Reply             |
|--------------|-------------------|-------------------|
| IP origen    | 10.65.19.143      | 10.65.19.50       |
| IP destino   | 10.65.19.50       | 10.65.19.143      |
| MAC cliente  | 10:6f:d9:3a:32:cb | 10:6f:d9:3a:32:cb |
| MAC servidor | 00:0c:29:a1:69:e9 | 10:6f:d9:3a:32:cb |

# 2. Responses

| No. | Time      | Source        | Destination  | Protocol | Length | Destination           | Source            | Source Address | Destination Address |
|-----|-----------|---------------|--------------|----------|--------|-----------------------|-------------------|----------------|---------------------|
| 25  | 5.015812  | 91.189.91.139 | 10.65.19.50  | DNS      | 387    | CloudNetwork_3a:32:cb | 4a:f5:e8:2b:a8:12 | 91.189.91.139  | 10.65.19.50         |
| 76  | 13.446684 | 10.65.19.197  | 10.65.19.143 | DNS      | 91     | CloudNetwork_3a:32:cb | 4a:f5:e8:2b:a8:12 | 10.65.19.197   | 10.65.19.143        |
| 77  | 13.446684 | 10.65.19.197  | 10.65.19.143 | DNS      | 91     | CloudNetwork_3a:32:cb | 4a:f5:e8:2b:a8:12 | 10.65.19.197   | 10.65.19.143        |
| 146 | 14.957571 | 10.65.19.197  | 10.65.19.143 | DNS      | 112    | CloudNetwork_3a:32:cb | 4a:f5:e8:2b:a8:12 | 10.65.19.197   | 10.65.19.143        |
| 147 | 14.957571 | 10.65.19.197  | 10.65.19.143 | DNS      | 112    | CloudNetwork_3a:32:cb | 4a:f5:e8:2b:a8:12 | 10.65.19.197   | 10.65.19.143        |
| 271 | 15.537988 | 10.65.19.197  | 10.65.19.143 | DNS      | 91     | CloudNetwork_3a:32:cb | 4a:f5:e8:2b:a8:12 | 10.65.19.197   | 10.65.19.143        |
| 272 | 15.538564 | 10.65.19.197  | 10.65.19.143 | DNS      | 91     | CloudNetwork_3a:32:cb | 4a:f5:e8:2b:a8:12 | 10.65.19.197   | 10.65.19.143        |
| 325 | 15.791166 | 10.65.19.197  | 10.65.19.143 | DNS      | 86     | CloudNetwork_3a:32:cb | 4a:f5:e8:2b:a8:12 | 10.65.19.197   | 10.65.19.143        |

Figura 2.2.1.3 Wireshark filtrado para ver solo respuestas

Dentro de las peticiones y respuestas, se pudo obtener información como la siguiente:

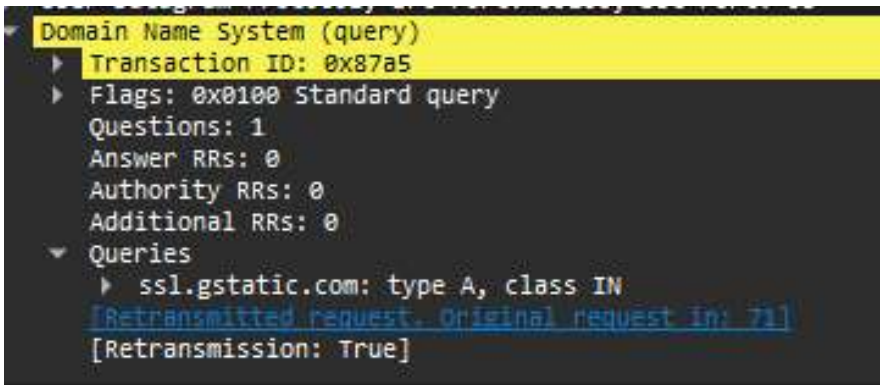


Figura 2.2.1.3 Información para capa de aplicación

Dentro de esta imagen, se puede observar la información de la capa de aplicación que aparece en los paquetes capturados que están relacionados con el servicio DNS. Tenemos un id para cada consulta/respuesta, el dominio solicitud (query name), el query type y Answers en donde podemos ver Tipo de registro, Dirección IP devuelta y TTL.

El protocolo de transporte identificado que se genera en las peticiones es UDP, ya que dns necesita ser rápido se usa este protocolo.

```

User Datagram Protocol, Src Port: 59106, Dst Port: 53
Source Port: 59106
Destination Port: 53
Length: 41
Checksum: 0x22a5 [unverified]
[Checksum Status: Unverified]
[Stream index: 3]
[Stream Packet Number: 3]
▶ [Timestamps]
UDP payload (33 bytes)

```

Figura

### 2.2.1.3 Información para capa de transporte y puerto

Así mismo en la imagen se puede ver que se hace uso del puerto 53, el cual siempre será el Destination Port. En la imagen para la petición escogida el puerto source fue el 59106, este se asigna como un numero aleatorio del cliente.

| <i>Campo</i> | <b>Request</b>    | <b>Reply</b>      |
|--------------|-------------------|-------------------|
| IP origen    | 10.65.19.139      | 10.65.19.50       |
| IP destino   | 10.65.19.50       | 10.65.19.139      |
| MAC cliente  | a:f5:e8:2b:a8:12  | 10:6f:d9:3a:32:cb |
| MAC servidor | 00:0c:29:a1:69:e9 | a:f5:e8:2b:a8:12  |

## 2.3 Servicio Web (HTTP/HTTPS)

### 2.3.1 HTTP

- Identifique la información de la capa de aplicación que aparecen en los paquetes capturados que estén relacionados con el servicio web:

```

▶ Frame 440: 599 bytes on wire (4792 bits), 599 bytes captured (4792 bits) on interface \Device\NPF_{CD18...
▶ Ethernet II, Src: CloudNetwork_29:e8:eb (ac:50:de:29:e8:eb), Dst: CloudNetwork_29:e8:eb (ac:50:de:29:e8:eb)
▶ Internet Protocol Version 4, Src: 10.87.74.214, Dst: 10.87.74.51
▶ Transmission Control Protocol, Src Port: 51341, Dst Port: 80, Seq: 1, Ack: 1, Len: 545
▼ Hypertext Transfer Protocol
  ▶ GET / HTTP/1.1\r\n
    Host: 10.87.74.51\r\n
    Connection: keep-alive\r\n
    Cache-Control: max-age=0\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/74.0.3729.169 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
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en\r\n
    If-None-Match: "341-63e784d8dfdb2-gzip"\r\n
    If-Modified-Since: Wed, 10 Sep 2025 20:31:18 GMT\r\n
    \r\n
    [Response in frame: 444]
    [Full request URI: http://10.87.74.51/]

```

Para la información de la capa de aplicación es el mismo protocolo HTTP, vemos información como el destino, el tipo de conexión, información del navegador, etc.

- Identifique el protocolo de la capa de transporte generado por las peticiones al servidor web.  
En la misma captura de pantalla podemos observar que el protocolo de la capa de transporte es TCP
- Identifique los puertos utilizados por el servicio web  
En la misma línea que nos indica el protocolo tcp podemos ver que el servicio web usó el puerto 80. Además, el cliente usó el 51341

### 2.3.2 HTTPS

- Identifique la información de la capa de aplicación que aparecen en los paquetes capturados que estén relacionados con el servicio web:

```

▶ Frame 1029: 458 bytes on wire (3664 bits), 458 bytes captured (3664 bits) on interface \Device\NPF_{CD1E
▶ Ethernet II, Src: 42:02:71:69:78:60 (42:02:71:69:78:60), Dst: CloudNetwork_29:e8:21 (ac:50:de:29:e8:21)
▶ Internet Protocol Version 4, Src: 34.117.13.33, Dst: 192.168.1.31
▶ Transmission Control Protocol, Src Port: 443, Dst Port: 63138, Seq: 1, Ack: 4604, Len: 404
▼ Transport Layer Security
  ▼ TLSv1.2 Record Layer: Application Data Protocol: Hypertext Transfer Protocol
    Content Type: Application Data (23)
    Version: TLS 1.2 (0x0303)
    Length: 399
    Encrypted Application Data [...]: f517709801bd330ac8ee0e33a503aa26c778864c74dd6dc9c4eb732122d30fbcfe
    [Application Data Protocol: Hypertext Transfer Protocol]

```

Para la información de la capa vemos el TLS, ya que al ser HTTPS el contenido del HTTP está cifrado. Podemos ver es la información encriptada del http.

- Identifique el protocolo de la capa de transporte generado por las peticiones al servidor web.  
En la misma captura de pantalla podemos observar que el protocolo de la capa de transporte es TCP
- Identifique los puertos utilizados por el servicio web  
En la misma línea que nos indica el protocolo TCP podemos ver que el servicio web usó el puerto 443. Además, el cliente usó el 63168
- Datos principales, puertos 80/443, etc.

## 2.4 Servicio FTP

### 3.4.1 Análisis de tráfico FTP – Descarga de archivo

| No. | Time     | Source       | Destination  | Protocol | Length | Info   |
|-----|----------|--------------|--------------|----------|--------|--|
| 41  | 6.941667 | 192.168.1.56 | 192.168.1.52 | FTP      | 62     | Request: TYPE I  |
| 43  | 6.950175 | 192.168.1.52 | 192.168.1.56 | FTP      | 73     | Response: 200 Type set to I  |
| 45  | 6.950367 | 192.168.1.56 | 192.168.1.52 | FTP      | 60     | Request: PASV  |
| 47  | 6.951396 | 192.168.1.52 | 192.168.1.56 | FTP      | 104    | Response: 227 Entering Passive Mode (192,168,1,52,192,59).   |
| 49  | 6.951540 | 192.168.1.56 | 192.168.1.52 | FTP      | 93     | Request: RETR archivo_prueba_ftp_descargar.pdf   |
| 57  | 6.955612 | 192.168.1.52 | 192.168.1.56 | FTP      | 146    | Response: 150 Opening BINARY mode data connection for archivo_prueba_ftp_descargar.pdf (53363 bytes) |
| 175 | 6.972517 | 192.168.1.52 | 192.168.1.56 | FTP      | 77     | Response: 226 Transfer complete  |

Ilustración 3.4.1 captura de pantalla del archivo FTP\_download.pcap

#### Capa de aplicación:

En la captura se observa que el cliente cambia el modo de transferencia a binario mediante el comando TYPE I, lo cual es confirmado por el servidor con el mensaje “200 Type set to I”. Luego, el cliente solicita el modo pasivo (PASV) y el servidor responde con “227 Entering Passive Mode (192,168,1,52,192,59)”, indicando la IP y el puerto que se usarán para la conexión de datos. Posteriormente, el cliente envía el comando RETR archivo\_prueba\_ftp\_descargar.pdf para descargar el archivo. El servidor confirma la apertura de la conexión con el mensaje “150 Opening BINARY mode data connection... (53363 bytes)” y, finalmente, cierra la operación con “226 Transfer complete”, confirmando que la descarga fue exitosa.

#### Capa de transporte:

El protocolo de transporte utilizado es **TCP**, ya que FTP requiere una conexión confiable para el envío y recepción de archivos.

#### Puertos utilizados:

El puerto **21/TCP** se emplea como canal de control para los comandos y respuestas. Además, en la transferencia de datos se abre un puerto dinámico en modo pasivo. Según la respuesta del servidor, el puerto calculado es **49211/TCP**, que se usó para enviar el archivo al cliente.

### 3.4.2 Análisis de tráfico FTP – Subida de archivo

| No. | Time      | Source       | Destination  | Protocol | Length | Info   |
|-----|-----------|--------------|--------------|----------|--------|--|
| 169 | 24.411521 | 192.168.1.52 | 192.168.1.56 | FTP      | 137    | Response: 220 ProFTPD Server (ftp.labrede16.com) [::ffff:192.168.1.52]             |
| 173 | 24.415565 | 192.168.1.52 | 192.168.1.56 | FTP      | 79     | Response: 500 AUTH not understood  |
| 175 | 24.415874 | 192.168.1.56 | 192.168.1.52 | FTP      | 64     | Request: AUTH SSL  |
| 177 | 24.416540 | 192.168.1.52 | 192.168.1.56 | FTP      | 79     | Response: 500 AUTH not understood  |
| 179 | 24.415102 | 192.168.1.56 | 192.168.1.52 | FTP      | 66     | Request: USER nikol  |
| 183 | 24.421169 | 192.168.1.52 | 192.168.1.56 | FTP      | 87     | Response: 331 Password required for nikol  |
| 183 | 24.421549 | 192.168.1.56 | 192.168.1.52 | FTP      | 69     | Request: PASS 1234abcd   |
| 191 | 24.580141 | 192.168.1.52 | 192.168.1.56 | FTP      | 80     | Response: 230 User nikol logged in   |
| 193 | 24.580393 | 192.168.1.56 | 192.168.1.52 | FTP      | 70     | Request: CWD file1234  |
| 197 | 24.589723 | 192.168.1.52 | 192.168.1.56 | FTP      | 62     | Response: 200 OK   |
| 199 | 24.590843 | 192.168.1.56 | 192.168.1.52 | FTP      | 68     | Request: QUIT  |
| 201 | 24.591027 | 192.168.1.52 | 192.168.1.56 | FTP      | 74     | Response: 200 QUIT set to on   |
| 203 | 24.594067 | 192.168.1.56 | 192.168.1.52 | FTP      | 68     | Request: CWD /Desktop  |
| 205 | 24.596677 | 192.168.1.52 | 192.168.1.56 | FTP      | 82     | Response: 250 CWD command successful   |
| 207 | 24.596080 | 192.168.1.56 | 192.168.1.52 | FTP      | 59     | Request: PWD   |
| 209 | 24.597740 | 192.168.1.52 | 192.168.1.56 | FTP      | 95     | Response: 257 "/Desktop" is the current directory                                  |
| 211 | 24.598136 | 192.168.1.56 | 192.168.1.52 | FTP      | 62     | Request: TYPE I  |
| 213 | 24.600342 | 192.168.1.52 | 192.168.1.56 | FTP      | 73     | Response: 200 Type set to I  |
| 215 | 24.600475 | 192.168.1.56 | 192.168.1.52 | FTP      | 104    | Response: 227 Entering Passive Mode (192,168,1,52,192,22).                         |
| 217 | 24.605672 | 192.168.1.52 | 192.168.1.56 | FTP      | 60     | Request: MLSD  |
| 219 | 24.608959 | 192.168.1.56 | 192.168.1.52 | FTP      | 104    | Response: 150 Opening BINARY mode data connection for MLSD                         |
| 227 | 24.604390 | 192.168.1.52 | 192.168.1.56 | FTP      | 77     | Response: 226 Transfer complete  |
| 237 | 24.608738 | 192.168.1.52 | 192.168.1.56 | FTP      | 60     | Request: PASV  |
| 243 | 24.626666 | 192.168.1.56 | 192.168.1.52 | FTP      | 104    | Response: 227 Entering Passive Mode (192,168,1,52,192,99).                         |
| 243 | 24.628217 | 192.168.1.52 | 192.168.1.56 | FTP      | 89     | Request: STOR archivo_prueba_ftp_subir.pdf   |
| 245 | 24.628817 | 192.168.1.56 | 192.168.1.52 | FTP      | 128    | Response: 150 Opening BINARY mode data connection for archivo_prueba_ftp_subir.pdf |
| 331 | 24.638462 | 192.168.1.52 | 192.168.1.56 | FTP      | 77     | Response: 226 Transfer complete  |
| 343 | 24.657967 | 192.168.1.52 | 192.168.1.56 | FTP      | 77     | Response: 226 Transfer complete  |

Ilustración 3.4.2 captura de pantalla del archivo FTP\_upload.pcap



### **Capa de aplicación:**

En este caso, el cliente inicia sesión con USER nikol y PASS 1234abcd, logrando autenticarse de manera exitosa con la respuesta “230 User nikol logged in”. Luego, cambia al directorio de trabajo con CWD /Desktop, consulta la ubicación actual con PWD, y establece el modo binario con TYPE I. Posteriormente, el cliente solicita el modo pasivo (PASV), y el servidor responde con “227 Entering Passive Mode (192,168,1,52,192,22)”. Con esta información, el cliente lista el directorio mediante MLSD, operación que es confirmada por el servidor con “150 Opening BINARY mode data connection for MLSD” y “226 Transfer complete”. Más adelante, se vuelve a activar el modo pasivo (PASV), esta vez con la respuesta “227 Entering Passive Mode (192,168,1,52,192,99)”. Con este nuevo canal de datos, el cliente inicia la subida del archivo con STOR archivo\_prueba\_ftp\_subir.pdf. El servidor abre la conexión con “150 Opening BINARY mode data connection for archivo\_prueba\_ftp\_subir.pdf” y confirma la finalización con “226 Transfer complete”.

### **Capa de transporte:**

El protocolo utilizado es **TCP**, garantizando una comunicación confiable entre cliente y servidor durante toda la transferencia.

### **Puertos utilizados:**

El puerto **21/TCP** se emplea para el canal de control. Para los datos, el servidor asigna puertos dinámicos en modo pasivo: en este caso, el puerto **49214/TCP** se utilizó para el listado de archivos y el puerto **49283/TCP** para la transferencia del archivo cargado.

## **2.5 Servicio VoIP**

**Archivo .pcap:** VoIP\_view.pcap

**Filtros usados:** sip, rtp, udp.port==5060

Dentro de VoIP se hace uso de 2 protocolos, SIP y RTP. Dentro de estos protocolos en capa de aplicación podemos ver lo siguiente en sus paquetes capturados

SIP:

| No. | Time      | Source      | Destination | Protocol | Length | Destination          | Source               | Source Address | Destination Address | Type            | Info           |
|-----|-----------|-------------|-------------|----------|--------|----------------------|----------------------|----------------|---------------------|-----------------|----------------|
| 21  | 1.199678  | 10.50.0.9   | 10.50.0.50  | SIP      | 893    | CloudNetwork_3a32:cb | 46:75:00:0f:5c:24    | 10.50.0.9      | 10.50.0.50          | Request: REGIS  | Status: 403 UN |
| 22  | 1.199930  | 10.50.0.50  | 10.50.0.9   | SIP      | 572    | 46:75:00:0f:5c:24    | CloudNetwork_3a32:cb | 10.50.0.50     | 10.50.0.9           | Request: 401 UN | Status: 401 UN |
| 23  | 1.199939  | 10.50.0.50  | 10.50.0.9   | SIP      | 572    | 46:75:00:0f:5c:24    | CloudNetwork_3a32:cb | 10.50.0.50     | 10.50.0.9           | Request: REGIS  | Status: 200 OK |
| 24  | 1.199786  | 10.50.0.9   | 10.50.0.50  | SIP      | 1835   | CloudNetwork_3a32:cb | 46:75:00:0f:5c:24    | 10.50.0.9      | 10.50.0.50          | Request: REGIS  | Status: 200 OK |
| 25  | 1.31182   | 10.50.0.50  | 10.50.0.9   | SIP      | 530    | 46:75:00:0f:5c:24    | CloudNetwork_3a32:cb | 10.50.0.50     | 10.50.0.9           | Request: 200 OK | Status: 200 OK |
| 26  | 1.312193  | 10.50.0.50  | 10.50.0.9   | SIP      | 530    | 46:75:00:0f:5c:24    | CloudNetwork_3a32:cb | 10.50.0.50     | 10.50.0.9           | Request: 200 OK | Status: 200 OK |
| 73  | 5.613847  | 10.50.0.9   | 10.50.0.50  | SIP/SDP  | 893    | CloudNetwork_3a32:cb | 46:75:00:0f:5c:24    | 10.50.0.9      | 10.50.0.50          | Request: INVITE | Status: 401 UN |
| 74  | 5.61525   | 10.50.0.50  | 10.50.0.9   | SIP      | 558    | 46:75:00:0f:5c:24    | CloudNetwork_3a32:cb | 10.50.0.50     | 10.50.0.9           | Request: 401 UN | Status: 401 UN |
| 75  | 5.615643  | 10.50.0.50  | 10.50.0.9   | SIP      | 558    | 46:75:00:0f:5c:24    | CloudNetwork_3a32:cb | 10.50.0.50     | 10.50.0.9           | Request: INVITE | Status: 200 OK |
| 76  | 5.621352  | 10.50.0.50  | 10.50.0.9   | SIP      | 382    | CloudNetwork_3a32:cb | 46:75:00:0f:5c:24    | 10.50.0.50     | 10.50.0.9           | Request: INVITE | Status: 100 TR |
| 80  | 5.123168  | 10.50.0.9   | 10.50.0.50  | SIP/SDP  | 1188   | CloudNetwork_3a32:cb | 46:75:00:0f:5c:24    | 10.50.0.9      | 10.50.0.50          | Request: INVITE | Status: 100 TR |
| 81  | 5.124414  | 10.50.0.50  | 10.50.0.9   | SIP      | 366    | 46:75:00:0f:5c:24    | CloudNetwork_3a32:cb | 10.50.0.50     | 10.50.0.9           | Request: INVITE | Status: 100 TR |
| 82  | 5.124424  | 10.50.0.50  | 10.50.0.9   | SIP      | 366    | 46:75:00:0f:5c:24    | CloudNetwork_3a32:cb | 10.50.0.50     | 10.50.0.9           | Request: INVITE | Status: 100 TR |
| 83  | 5.126358  | 10.50.0.50  | 10.50.0.239 | SIP/SDP  | 1005   | 12:86:23:1a:10:3d    | CloudNetwork_3a32:cb | 10.50.0.50     | 10.50.0.239         | Request: INVITE | Status: 100 TR |
| 84  | 5.126373  | 10.50.0.50  | 10.50.0.239 | SIP/SDP  | 1005   | 12:86:23:1a:10:3d    | CloudNetwork_3a32:cb | 10.50.0.50     | 10.50.0.239         | Request: INVITE | Status: 100 TR |
| 85  | 5.240339  | 10.50.0.239 | 10.50.0.50  | SIP      | 373    | CloudNetwork_3a32:cb | 12:86:23:1a:10:3d    | 10.50.0.239    | 10.50.0.50          | Status: 100 TR  | Status: 100 TR |
| 86  | 5.344386  | 10.50.0.239 | 10.50.0.50  | SIP      | 698    | CloudNetwork_3a32:cb | 12:86:23:1a:10:3d    | 10.50.0.239    | 10.50.0.50          | Status: 100 TR  | Status: 100 TR |
| 87  | 5.345729  | 10.50.0.50  | 10.50.0.9   | SIP      | 551    | 46:75:00:0f:5c:24    | CloudNetwork_3a32:cb | 10.50.0.50     | 10.50.0.9           | Status: 100 TR  | Status: 100 TR |
| 88  | 5.345739  | 10.50.0.50  | 10.50.0.9   | SIP      | 551    | 46:75:00:0f:5c:24    | CloudNetwork_3a32:cb | 10.50.0.50     | 10.50.0.9           | Status: 100 TR  | Status: 100 TR |
| 302 | 21.875757 | 10.50.0.9   | 10.50.0.50  | SIP      | 676    | CloudNetwork_3a32:cb | 46:75:00:0f:5c:24    | 10.50.0.9      | 10.50.0.50          | Request: CANCEL | Status: 200 OK |
| 304 | 21.876204 | 10.50.0.50  | 10.50.0.9   | SIP      | 403    | 46:75:00:0f:5c:24    | CloudNetwork_3a32:cb | 10.50.0.50     | 10.50.0.9           | Status: 200 OK  | Status: 200 OK |
| 305 | 21.877213 | 10.50.0.50  | 10.50.0.9   | SIP      | 403    | 46:75:00:0f:5c:24    | CloudNetwork_3a32:cb | 10.50.0.50     | 10.50.0.9           | Status: 407 Re  | Status: 407 Re |
| 306 | 21.877795 | 10.50.0.50  | 10.50.0.9   | SIP      | 530    | 46:75:00:0f:5c:24    | CloudNetwork_3a32:cb | 10.50.0.50     | 10.50.0.9           | Status: 407 Re  | Status: 407 Re |
| 307 | 21.877878 | 10.50.0.50  | 10.50.0.9   | SIP      | 530    | 46:75:00:0f:5c:24    | CloudNetwork_3a32:cb | 10.50.0.50     | 10.50.0.9           | Status: 407 Re  | Status: 407 Re |
| 308 | 21.880943 | 10.50.0.239 | 10.50.0.50  | SIP      | 520    | 12:86:23:1a:10:3d    | CloudNetwork_3a32:cb | 10.50.0.50     | 10.50.0.239         | Request: CANCEL | Status: 200 OK |
| 309 | 21.880955 | 10.50.0.50  | 10.50.0.239 | SIP      | 520    | 12:86:23:1a:10:3d    | CloudNetwork_3a32:cb | 10.50.0.50     | 10.50.0.239         | Request: CANCEL | Status: 200 OK |
| 309 | 21.893275 | 10.50.0.50  | 10.50.0.50  | SIP      | 303    | CloudNetwork_3a32:cb | 46:75:00:0f:5c:24    | 10.50.0.50     | 10.50.0.50          | Request: ACK    | Status: 200 OK |
| 309 | 21.893424 | 10.50.0.239 | 10.50.0.50  | SIP      | 453    | CloudNetwork_3a32:cb | 12:86:23:1a:10:3d    | 10.50.0.239    | 10.50.0.50          | Status: 200 OK  | Status: 200 OK |
| 309 | 21.897424 | 10.50.0.239 | 10.50.0.50  | SIP      | 438    | CloudNetwork_3a32:cb | 12:86:23:1a:10:3d    | 10.50.0.239    | 10.50.0.50          | Status: 407 Re  | Status: 407 Re |
| 309 | 21.897756 | 10.50.0.50  | 10.50.0.239 | SIP      | 502    | 12:86:23:1a:10:3d    | CloudNetwork_3a32:cb | 10.50.0.50     | 10.50.0.239         | Request: ACK    | Status: 200 OK |
| 309 | 21.897758 | 10.50.0.239 | 10.50.0.50  | SIP      | 502    | 12:86:23:1a:10:3d    | CloudNetwork_3a32:cb | 10.50.0.50     | 10.50.0.239         | Request: ACK    | Status: 200 OK |
| 403 | 30.496669 | 10.50.0.239 | 10.50.0.50  | SIP/SDP  | 902    | CloudNetwork_3a32:cb | 12:86:23:1a:10:3d    | 10.50.0.239    | 10.50.0.50          | Request: INVITE | Status: 200 OK |
| 404 | 30.497882 | 10.50.0.50  | 10.50.0.239 | SIP      | 562    | 12:86:23:1a:10:3d    | CloudNetwork_3a32:cb | 10.50.0.50     | 10.50.0.239         | Status: 401 UN  | Status: 401 UN |

Figura 2.5.1 Imagen de filtro de SIP en wireshark

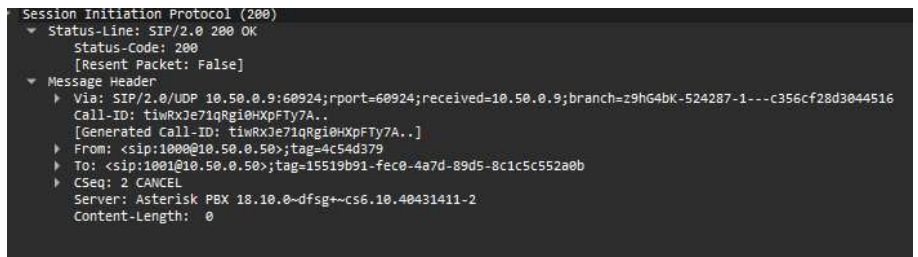


Figura 2.5.2 Imagen de capa de aplicación de paquetes con SIP

Dentro de esta imagen podemos ver el request line con OK, las cabeceras con un CALL-ID e información de los usuarios que se estaban conectando con el FROM y el TO. Se pueden ver tambien de manera clara la parte de VIA que tiene el puerto diferente información de envío.

RTP:

| No.  | Time       | Source      | Destination | Protocol | Length | Destination          | Source               | Source Address | Destination Address | Type                      | Info                      |
|------|------------|-------------|-------------|----------|--------|----------------------|----------------------|----------------|---------------------|---------------------------|---------------------------|
| 2995 | 132.450837 | 10.50.0.239 | 10.50.0.50  | RTP      | 55     | CloudNetwork_3a32:cb | 12:86:23:1a:10:3d    | 10.50.0.239    | 10.50.0.50          | PT=unassigned, SSRC=0     | PT=unassigned, SSRC=0     |
| 3106 | 132.493668 | 10.50.0.9   | 10.50.0.50  | RTP      | 55     | CloudNetwork_3a32:cb | 46:75:00:0f:5c:24    | 10.50.0.9      | 10.50.0.50          | PT=unassigned, SSRC=0     | PT=unassigned, SSRC=0     |
| 3224 | 144.405250 | 10.50.0.239 | 10.50.0.50  | RTP      | 244    | CloudNetwork_3a32:cb | 12:86:23:1a:10:3d    | 10.50.0.239    | 10.50.0.50          | 3 Destination unreachable | 3 Destination unreachable |
| 3225 | 144.405261 | 10.50.0.9   | 10.50.0.239 | SDP      | 242    | 12:86:23:1a:10:3d    | CloudNetwork_3a32:cb | 10.50.0.9      | 10.50.0.239         | 3 Destination unreachable | 3 Destination unreachable |
| 3274 | 144.405787 | 10.50.0.50  | 10.50.0.239 | SDP      | 242    | 12:86:23:1a:10:3d    | CloudNetwork_3a32:cb | 10.50.0.50     | 10.50.0.239         | 3 Destination unreachable | 3 Destination unreachable |

Figura 2.5.1 Imagen de filtro de RTP en wireshark

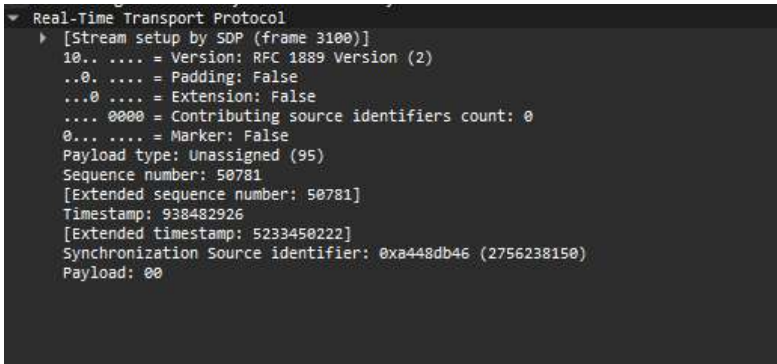


Figura 2.5.2

Imagen de capa de aplicación de paquetes con RTP

Dentro de esta imagen podemos ver información como el PayloadType que es unassigned, el sequence number, timestamp y tambien el SSRI que representa la información más importante de la capa de aplicación.

Por otro lado para la capa de transporte, podemos ver que dentro de la cabecera de una petición SIP y una petición RTP el protocolo es UDP ya que es un protocolo rapido de usar.

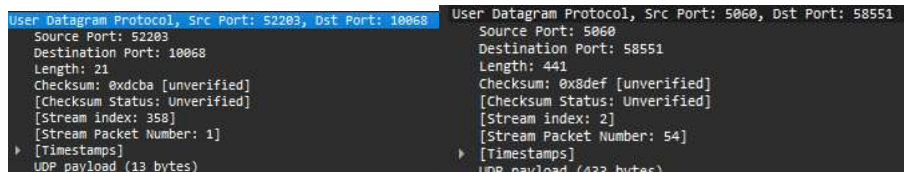


Figura 2.5.3 Imagen de capa de transporte en peticiones tanto RTP como SIP respectivamente

Dentro de las imagenes anteriores se puede ver que el puerto de SIP es 5060 y el puerto de RTP es 52203. El puerto SIP es definido, mientras que el puerto RTP esta en un rango dinámico dependiente del softphone.

## 2.6 Servicio RTMP

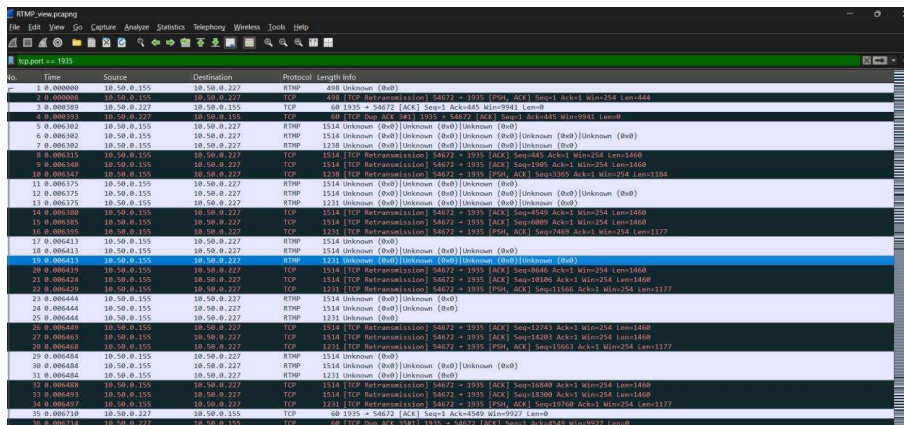


Ilustración 3.1 captura de pantalla del archivo RTMP\_view.pcap

### Capa de aplicación:

En la captura se observa tráfico identificado como **RTMP**. Sin embargo, no todo el tráfico aparece bajo ese filtro porque Wireshark no siempre reconoce automáticamente este protocolo. Por eso, en lugar de filtrar con `rtmp`, es más efectivo aplicar el filtro por **puerto**, en este caso `tcp.port == 1935`, que es el puerto estándar de RTMP. Una vez hecho esto, se puede distinguir el intercambio de paquetes relacionados con la transmisión en vivo.

En cuanto a la información que se aprecia, los paquetes contienen tramas propias de la transmisión de video en tiempo real, donde se observa una gran cantidad de mensajes “Unknown” o sin decodificar por parte de Wireshark. Esto ocurre porque RTMP encapsula datos multimedia (video y audio) que se envían de manera continua en bloques pequeños, por lo cual aparecen cientos o miles de paquetes durante la transmisión.

**Capa de transporte:**

El protocolo de transporte utilizado es **TCP**, ya que RTMP requiere un canal confiable para garantizar que los fragmentos de video y audio lleguen en orden y sin pérdidas al reproductor del cliente.

**Puertos utilizados:**

El puerto estándar utilizado para RTMP es el **1935/TCP**, que aparece claramente en la captura. Este es el canal por el cual se transporta la transmisión de video desde el cliente (OBS u otra aplicación emisora) hacia el servidor, y luego se redistribuye al receptor (por ejemplo, VLC).