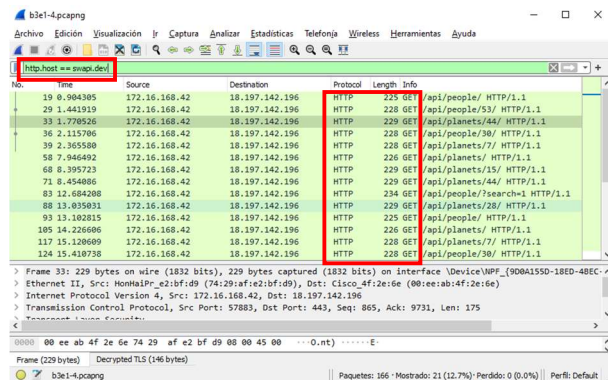


Práctica Bloque III

Alumno 1: **Isidro Javier García Fernández**

Titulación: Doble Grado de Matemáticas e Ingeniería Informática

PC de la práctica: **012**



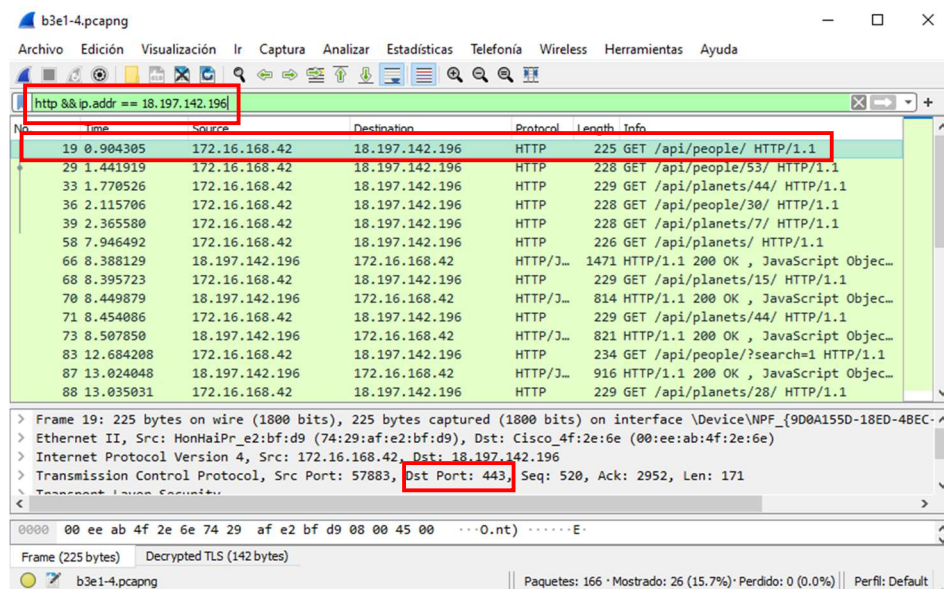
(Comprobación funcionamiento con el comando `http.host == swapi.dev`. Deben aparecer peticiones GET típicas de HTTP)

Ejercicio 1. ¿Cuál es el puerto utilizado por el servidor? ¿Es el normal de HTTP (80)? ¿Por qué?

Puerto servidor: 443

No. Suele ser utilizado por HTTPS pero no estamos con HTTPS, estamos con HTTP.

- Tramas analizadas:



Ejercicio 2. Observe el número de conexiones realizadas. ¿Cuántas hace? ¿Usa una conexión permanente (en la misma conexión hace varias peticiones) o no permanente (solo realiza una por conexión)? En caso de ser permanente, ¿qué cabecera de la petición indica que queremos que sea permanente?

Se realizan 26 conexiones.

Se utiliza una conexión permanente, ya que en la misma conexión hace varias peticiones.

Cabecera: *Connection: keep-alive\r\n*

- Tramas analizadas:

The screenshot shows the Wireshark interface with a packet capture named 'b3e1-4.pcapng'. The filter bar is set to 'http && ip.addr == 18.197.142.196'. The packet list shows 26 HTTP GET requests to various endpoints on 18.197.142.196. The packet details pane is expanded to show the 'Hypertext Transfer Protocol' section of a selected packet, highlighting the 'Connection: keep-alive\r\n' header. The status bar at the bottom indicates 166 packets captured, with 26 (15.7%) displayed in a red box.

No.	Time	Source	Destination	Protocol	Length	Info
19	0.904305	172.16.168.42	18.197.142.196	HTTP	225	GET /api/people/ HTTP/1.1
29	1.441919	172.16.168.42	18.197.142.196	HTTP	228	GET /api/people/53/ HTTP/1.1
33	1.770526	172.16.168.42	18.197.142.196	HTTP	229	GET /api/planets/44/ HTTP/1.1
36	2.115706	172.16.168.42	18.197.142.196	HTTP	228	GET /api/people/30/ HTTP/1.1
39	2.365580	172.16.168.42	18.197.142.196	HTTP	228	GET /api/planets/7/ HTTP/1.1
58	7.946492	172.16.168.42	18.197.142.196	HTTP	226	GET /api/planets/ HTTP/1.1
66	8.388129	18.197.142.196	172.16.168.42	HTTP/1.1	1471	HTTP/1.1 200 OK, JavaScript Objec...
68	8.395723	172.16.168.42	18.197.142.196	HTTP	229	GET /api/planets/15/ HTTP/1.1
70	8.449879	18.197.142.196	172.16.168.42	HTTP/1.1	814	HTTP/1.1 200 OK, JavaScript Objec...
71	8.454086	172.16.168.42	18.197.142.196	HTTP	229	GET /api/planets/44/ HTTP/1.1
73	8.507850	18.197.142.196	172.16.168.42	HTTP/1.1	821	HTTP/1.1 200 OK, JavaScript Objec...
83	12.684208	172.16.168.42	18.197.142.196	HTTP	234	GET /api/people/?search=1 HTTP/1.1

Hypertext Transfer Protocol

- GET /api/people/ HTTP/1.1\r\n
- Accept: application/json\r\n
- User-Agent: IsidroJavierGarciaFernandez-2021\r\n
- Host: swapi.dev\r\n
- Connection: keep-alive\r\n**
- \r\n
- [Full request URI: https://swapi.dev/api/people/]

Frame (225 bytes) | Decrypted TLS (142 bytes)

b3e1-4.pcapng | Paquetes: 166 · Mostrado: 26 (15.7%) · Perdido: 0 (0.0%) | Perfil: Default

Ejercicio 3. Observe una respuesta, ¿cómo se identifica dónde acaban las cabeceras HTTP y empieza el recurso?

Acaban utilizando los caracteres: `\r\n`

- Tramas analizadas:

Wireshark packet capture showing an HTTP 200 OK response. The packet list shows a GET request to /api/planets/157 and a corresponding 200 OK response. The packet details pane shows the response structure with headers and body. The packet bytes pane shows the raw data.

No.	Time	Source	Destination	Protocol	Length	Info
19	0.904305	172.16.168.42	18.197.142.196	HTTP	225	GET /api/people/ HTTP
29	1.441919	172.16.168.42	18.197.142.196	HTTP	228	GET /api/people/53/ HTTP
33	1.770526	172.16.168.42	18.197.142.196	HTTP	229	GET /api/planets/44/ HTTP
36	2.115706	172.16.168.42	18.197.142.196	HTTP	228	GET /api/people/30/ HTTP
39	2.365580	172.16.168.42	18.197.142.196	HTTP	228	GET /api/planets/7/ HTTP
58	7.946492	172.16.168.42	18.197.142.196	HTTP	226	GET /api/planets/ HTTP
66	8.388129	18.197.142.196	172.16.168.42	HTTP/J...	1471	HTTP/1.1 200 OK , Ja
68	8.395725	172.16.168.42	18.197.142.196	HTTP	229	GET /api/planets/157/ HTTP
70	8.449879	18.197.142.196	172.16.168.42	HTTP/J...	814	HTTP/1.1 200 OK , Ja
71	8.454086	172.16.168.42	18.197.142.196	HTTP	229	GET /api/planets/44/ HTTP
73	8.507850	18.197.142.196	172.16.168.42	HTTP/J...	821	HTTP/1.1 200 OK , Ja
83	12.684208	172.16.168.42	18.197.142.196	HTTP	234	GET /api/people/?sea HTTP
87	13.024048	18.197.142.196	172.16.168.42	HTTP/J...	916	HTTP/1.1 200 OK , Ja
88	13.035031	172.16.168.42	18.197.142.196	HTTP	229	GET /api/planets/28/ HTTP
92	13.098437	18.197.142.196	172.16.168.42	HTTP/J...	922	HTTP/1.1 200 OK , Ja
93	13.102815	172.16.168.42	18.197.142.196	HTTP	225	GET /api/people/ HTTP

Hypertext Transfer Protocol

> HTTP/1.1 200 OK\r\n

Server: nginx/1.16.1\r\n

Date: Mon, 14 Jun 2021 10:46:28 GMT\r\n

Content-Type: application/javascript\r\n

Transfer-Encoding: chunked\r\n

Connection: keep-alive\r\n

Vary: Accept, Cookie\r\n

X-Frame-Options: SAMEORIGIN\r\n

ETag: "87c7d9cfc6ffa456c7d533fc153a0411"\r\n

Allow: GET, HEAD, OPTIONS\r\n

Strict-Transport-Security: max-age=15768000\r\n

0000 74 29 af e2 bf d9 00 ee ab 4f 2e 6e 08 00 45 00 t).....0.n..E

0010 05 b1 66 17 40 00 3f 06 da 6b 12 c5 8e c4 ac 10 ..f.@.?..k.....

Frame (1471 bytes) | Reassembled TCP (5797 bytes) | Decrypted TLS (5768 bytes) | De-chunked entity body (5422 bytes)

b3e1-4.pcapng | Paquetes: 166 · Mostrado: 26 (15.7%) · Perdido: 0 (0.0%) | Perfil: Default

Ejercicio 4. Describa el significado de las cabeceras de una petición y una respuesta (sin incluir las que empiecen por x-).

Petición:

GET/api/people/HTTP/1.1 : método usado, dirección, versión HTTP usada
 Accept: application/json : formato de la información
 User-Agent: IsidroJavierGarciaFernandez-2021 : nombre de la aplicación
 Host: swapi.dev : nombre del host
 Connection: keep-alive : tipo de conexión (permanente)

Respuesta:

HTTP/1.1 200 OK : versión HTTP usada, estado de la conexión (en este caso "200 OK")
 Server: nginx/1.16.1 : versión del servidor
 Date: Mon, 14 Jun 2021: 10:45:28 GMT : fecha y hora de la respuesta a la petición
 Content-Type: application/json : formato del contenido de la respuesta
 Transfer-Encoding: chunked : respuesta transferida (enviada) mediante distintos chunks
 Connection: keep-alive : tipo de conexión (permanente)
 Vary: Accept, Cookie : variantes de la respuesta
 ETag: "87c7d9cfc6ffa456c7d533fc153a0411" : ID (identificador) de respuesta
 Allow: GET, HEAD, OPTIONS : métodos que se pueden utilizar (en el caso de la petición, GET)
 Strict-Transport-Security: max-age=15768000: TTL (tiempo de vida) del mensaje de respuesta)

Tramas analizadas:

The left screenshot shows a list of captured packets in Wireshark. The selected packet (No. 19) is an HTTP GET request to /api/people/. The details pane shows the Hypertext Transfer Protocol section with the following headers:

```

GET /api/people/ HTTP/1.1
Host: swapi.dev
User-Agent: IsidroJavierGarciaFernandez-2021
Accept: application/json
Connection: keep-alive

```

The right screenshot shows the details of packet 19, which is an HTTP GET request. The details pane is expanded to show the Hypertext Transfer Protocol section, which contains the request headers and the full request URI:

```

HTTP/1.1 200 OK
Server: nginx/1.16.1
Date: Mon, 14 Jun 2021 10:46:28 GMT
Content-Type: application/json
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
ETag: "87c7d9cfc6ffa456c7d533fc153a0411"
Allow: GET, HEAD, OPTIONS
Strict-Transport-Security: max-age=15768000

```


Ejercicio 5. Filtre por el protocolo rtsp y use la opción **Follow TCP Stream** de Wireshark para observar el diálogo completo que han mantenido el cliente de correo y el servidor. Explique brevemente (una línea) el significado de cada comando enviado por el cliente (si algún comando se repite solo debe explicarlo una vez).

```
OPTIONS rtsp://wowzaec2demo.streamlock.net:554/vod/
mp4:BigBuckBunny_115k.mov RTSP/1.0
CSeq: 2
User-Agent: LibVLC/3.0.14 (LIVE555 Streaming Media v2016.11.28)
```

OPTIONS: opciones para la comunicación

```
DESCRIBE rtsp://wowzaec2demo.streamlock.net:554/vod/
mp4:BigBuckBunny_115k.mov RTSP/1.0
CSeq: 3
User-Agent: LibVLC/3.0.14 (LIVE555 Streaming Media v2016.11.28)
Accept: application/sdp
```

DESCRIBE: descripción de la conexión

```
SETUP rtsp://wowzaec2demo.streamlock.net:554/vod/
mp4:BigBuckBunny_115k.mov/trackID=1 RTSP/1.0
CSeq: 4
User-Agent: LibVLC/3.0.14 (LIVE555 Streaming Media v2016.11.28)
Transport: RTP/AVP;unicast;client_port=51224-51225
```

SETUP: estado inicial para la conexión y transmisión de datos

```
SETUP rtsp://wowzaec2demo.streamlock.net:554/vod/
mp4:BigBuckBunny_115k.mov/trackID=2 RTSP/1.0
CSeq: 5
User-Agent: LibVLC/3.0.14 (LIVE555 Streaming Media v2016.11.28)
Transport: RTP/AVP;unicast;client_port=51226-51227
Session: 1280767087
```

(repetido)

```
PLAY rtsp://wowzaec2demo.streamlock.net:554/vod/
mp4:BigBuckBunny_115k.mov/ RTSP/1.0
CSeq: 6
User-Agent: LibVLC/3.0.14 (LIVE555 Streaming Media v2016.11.28)
Session: 1280767087
Range: npt=0.000-
```

PLAY: comienza la transmisión de datos

```
TEARDOWN rtsp://wowzaec2demo.streamlock.net:554/vod/
mp4:BigBuckBunny_115k.mov/ RTSP/1.0
CSeq: 7
User-Agent: LibVLC/3.0.14 (LIVE555 Streaming Media v2016.11.28)
Session: 1280767087
```

TEARDOWN: se interrumpe la conexión (finaliza la conexión)

• Tramas analizadas:

Wireshark capture of RTSP traffic. The packet list shows frames 14 through 34. Frame 14 is selected, showing details for Ethernet II, Internet Protocol Version 4, and Real Time Streaming Protocol. The packet bytes pane shows the raw data in hexadecimal and ASCII.

Wireshark capture of RTSP traffic. The packet list shows frames 11 through 20. Frame 11 is selected, showing details for Ethernet II, Internet Protocol Version 4, and Real Time Streaming Protocol. The packet bytes pane shows the raw data in hexadecimal and ASCII.

Ejercicio 6. ¿Por qué se hacen dos comandos SETUP? ¿Cómo sabía que debía hacer dos comandos de ese estilo?

Se hacen dos comandos SETUP, uno para cada flujo (trackID=1, trackID=2). Se indica al servidor por qué puertos se recibe el audio y el vídeo.

Porque en la respuesta del método DESCRIBE se especifica que hayan dos comandos SETUP

- Tramas analizadas:

The image shows a Wireshark capture of RTSP traffic between a client and a server. The capture is filtered for 'rtsp'. The packet list shows several packets, with two specific packets highlighted in red:

- Packet 19: RTSP SETUP request for trackID=1.
- Packet 21: RTSP SETUP request for trackID=2.

The packet details for these two packets are shown below:

Packet 19: RTSP SETUP request for trackID=1

```

Request: SETUP rtsp://wowzaec2demo.streamlock.net:554/vod/mp4:BigBuckBunny_115k.mov trackID=1 RTSP/1.0\r\n
CSeq: 4\r\n
User-Agent: LibVLC/3.0.14 (Ubuntu 12.04 LTS)\r\n
Transport: RTP/AVP;unicast client_port=51226-51227\r\n

```

Packet 21: RTSP SETUP request for trackID=2

```

Request: SETUP rtsp://wowzaec2demo.streamlock.net:554/vod/mp4:BigBuckBunny_115k.mov trackID=2 RTSP/1.0\r\n
CSeq: 5\r\n
User-Agent: LibVLC/3.0.14 (Ubuntu 12.04 LTS)\r\n
Transport: RTP/AVP;unicast client_port=51228-51229\r\n

```

The packet details for these two packets are also shown below:

Packet 19: RTSP SETUP request for trackID=1

```

> Session Attribute (a): control:*
> Media Description, name and address (m): audio 0 RTP/AVP 96
> Media Attribute (a): rtpmap:96 mpeg4-generic/128000/2
> Media Attribute (a): fmtp:96 profile-level-id=1;mode=AAC-hbr;size-length=13;index-length=3;index-delta-length=3;config=1490
> Media Attribute (a): control:trackID=1
> Media Description, name and address (m): video 0 RTP/AVP 97
> Media Attribute (a): rtpmap:97 H264/90000
> Media Attribute (a): fmtp:97 packetization-mode=1;profile-level-id=42C01E;sprop-parameter-sets=Z0LAHtkDxiHAAAADAEEAAAwDxyuS,aNultsg=
> Media Attribute (a): cliprect:0,0,160,240
> Media Attribute (a): framesize:97 240-160
> Media Attribute (a): framerate:24.0
> Media Attribute (a): control:trackID=2

```

Packet 21: RTSP SETUP request for trackID=2

```

> Session Attribute (a): control:*
> Media Description, name and address (m): audio 0 RTP/AVP 96
> Media Attribute (a): rtpmap:96 mpeg4-generic/128000/2
> Media Attribute (a): fmtp:96 profile-level-id=1;mode=AAC-hbr;size-length=13;index-length=3;index-delta-length=3;config=1490
> Media Attribute (a): control:trackID=1
> Media Description, name and address (m): video 0 RTP/AVP 97
> Media Attribute (a): rtpmap:97 H264/90000
> Media Attribute (a): fmtp:97 packetization-mode=1;profile-level-id=42C01E;sprop-parameter-sets=Z0LAHtkDxiHAAAADAEEAAAwDxyuS,aNultsg=
> Media Attribute (a): cliprect:0,0,160,240
> Media Attribute (a): framesize:97 240-160
> Media Attribute (a): framerate:24.0
> Media Attribute (a): control:trackID=2

```


Ejercicio 7. ¿Qué comandos ha provocado adelantar la reproducción del vídeo? ¿Cómo indica por donde debe seguir la reproducción tras el cambio?

- Tramas analizadas:

Utiliza el método PAUSE y el método PLAY

The image shows a Wireshark capture of an RTSP session. The packet list pane highlights two packets: packet 579 (RTSP PAUSE) and packet 603 (RTSP PLAY). The packet details pane for packet 603 shows the RTSP method 'PLAY' and the URL 'rtsp://wowzaec2demo.streamlock.net:554/vod/mp4:BigBuckBunny_115k.mov/'. The packet bytes pane shows the raw data of the RTSP message.

Ejercicio 8. Si observa los comandos y las respuestas son muy similares a las que usa HTTP. Indique dos cabeceras que use RTSP que también se usen en HTTP e indique (y explique) dos cabeceras de RTSP que no se usen en HTTP.

Comunes a HTTP y RTSP: User-Agent y Date

No comunes a HTTP y RTSP: CSeq y Cache-control

Cseq: número de secuencia Cache-control: si se utiliza la memoria caché

- Tramas analizadas:

The image shows a Wireshark capture of an HTTP session. The packet list pane highlights a GET request (packet 19). The packet details pane shows the HTTP method 'GET' and the URL 'https://swapi.dev/api/people/'. The packet bytes pane shows the raw data of the HTTP message.

The image shows a Wireshark capture of an HTTP session. The packet list pane highlights a GET response (packet 225). The packet details pane shows the HTTP status '200 OK' and the content type 'application/json'. The packet bytes pane shows the raw data of the HTTP message.

En HTTP

En RTSP

```

SETUP rtsp://wowzaec2demo.streamlock.net:554/vod/
mp4:BigBuckBunny_115k.mov/trackID=1 RTSP/1.0
CSeq: 4
User-Agent: LibVLC/3.0.14 (LIVE555 Streaming Media v2016.11.28)
Transport: RTP/AVP/TCP;unicast;interleaved=0-1

```

```

RTSP/1.0 200 OK
CSeq: 4
Server: Wowza Streaming Engine 4.8.10 build20210217143515
Cache-Control: no-cache
Expires: Mon, 14 Jun 2021 11:55:51 UTC
Transport: RTP/AVP/TCP;unicast;interleaved=0-1
Date: Mon, 14 Jun 2021 11:55:51 UTC
Session: 1636894433;timeout=60

```

Ejercicio 9. Ahora filtre por el protocolo rtp que se utiliza para transmitir el recurso multimedia tal cual. ¿Cómo se decidieron los puertos a utilizar en estas comunicaciones RTP? ¿Se confirman de alguna forma cada uno de los envíos RTP?

Los puertos son los mismos que en el ejercicio 6.

No se confirman porque no se ven mensajes de confirmación entre las tramas de tipo RTP.

- Tramas analizadas:

139	1b.920820	172.16.168.42	34.227.104.115	TCP	54 53616 → 554 [ACK] Seq=1013 Ack=2103 Win=13081
140	17.041714	34.227.104.115	172.16.168.42	RTP	1311 PT=DynamicRTP-Type-97, SSRC=0xE242E88, Seq=1...
141	17.087734	172.16.168.42	34.227.104.115	TCP	54 53616 → 554 [ACK] Seq=1013 Ack=3360 Win=13132
142	17.212685	34.227.104.115	172.16.168.42	RTP	1122 PT=DynamicRTP-Type-97, SSRC=0xE242E88, Seq=2...
143	17.263112	172.16.168.42	34.227.104.115	TCP	54 53616 → 554 [ACK] Seq=1013 Ack=4428 Win=13030
144	17.394868	34.227.104.115	172.16.168.42	RTP	936 PT=DynamicRTP-Type-97, SSRC=0xE242E88, Seq=6...
145	17.446111	172.16.168.42	34.227.104.115	TCP	54 53616 → 554 [ACK] Seq=1013 Ack=5310 Win=13132
146	17.569085	34.227.104.115	172.16.168.42	RTP	1469 PT=DynamicRTP-Type-97, SSRC=0xE242E88, Seq=9...
147	17.618011	172.16.168.42	34.227.104.115	TCP	54 53616 → 554 [ACK] Seq=1013 Ack=6725 Win=12979
148	17.716295	34.227.104.115	172.16.168.42	RTP	1514 PT=DynamicRTP-Type-97, SSRC=0xE242E88, Seq=14...
149	17.742491	34.227.104.115	172.16.168.42	RTP	337 PT=DynamicRTP-Type-96, SSRC=0x50AEF919, Seq=8...
150	17.742543	172.16.168.42	34.227.104.115	TCP	54 53616 → 554 [ACK] Seq=1013 Ack=8468 Win=13132

Acknowledgment number (raw): 680401576

0101 ... = Header Length: 20 bytes (5)

Flags: 0x010 (ACK)

Window: 513

[Calculated window size: 131328]

[Window size scaling factor: 256]

Checksum: 0x324e [unverified]

[Checksum Status: Unverified]

Urgent Pointer: 0

[SEQ/ACK analysis]

[This is an ACK to the segment in frame: 140]

[The RTT to ACK the segment was 0.04600000 seconds]

```

0000 00 ee ab 4f 2e 6e 74 29 af e2 bf d9 08 00 45 00 ...O.nt) .....E
0010 00 28 e5 c0 40 00 00 06 35 7e ac 10 a8 2a 22 e3 ...(.@... 5w...*..
0020 68 73 d1 70 02 2a 09 3d 7b e6 28 8e 1a a8 50 10 hs.p*:= {.(...P
0030 02 01 32 4e 00 00

```


Ejercicio 10. Finalmente filtre por el protocolo `rtcp` usado para controlar el estado de la conexión. Observe alguna trama que sea *Receiver Report*. Despliegue esa cabecera y marque (y explique) dos valores reportados que nos aporten información para poder ajustar la reproducción de acuerdo a las características de la comunicación.

Fraction lost: tasa de pérdida de paquetes. Se halla dividiendo el nº de paquetes perdidos entre el nº de paquetes enviados (esperados a recibir)

Cumulative number of packets lost: Número de paquetes RTP que se han perdido desde el inicio de la conexión. (nº paquetes enviados – nº paquetes recibidos)

- Tramas analizadas:

The screenshot shows the Wireshark interface with the file `b3e5-10.pcapng` open. The packet list on the left shows several RTCP Receiver Report packets. Packet 273 is selected and highlighted in red. The packet details pane on the right shows the expanded structure of this packet:

- 10.. = Version: RFC 1889 Version (2)
- ..0. = Padding: False
- ...0 0001 = Reception report count: 1
- Packet type: Receiver Report (201)
- Length: 7 (32 bytes)
- Sender SSRC: 0x427ba720 (111539968)
- Source 1
 - Identifier: 0x0e242eb8 (237252280)
 - SSRC contents
 - Fraction lost: 0 / 256
 - Cumulative number of packets lost: -1

The bottom status bar indicates: Real-time Transport Control Protocol: Protocol | Paquetes: 1450 · Mostrado: 32 (2.2%) | Perfil: Default