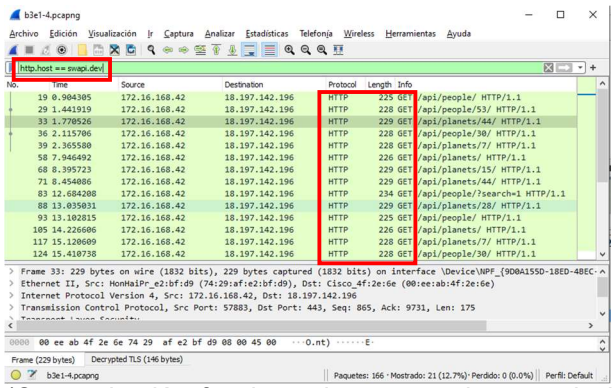
Práctica Bloque III

Alumno 1: Isidro Javier García Fernández

Titulación: Doble Grado de Matemáticas e Ingeniería Informática

PC de la práctica: 012

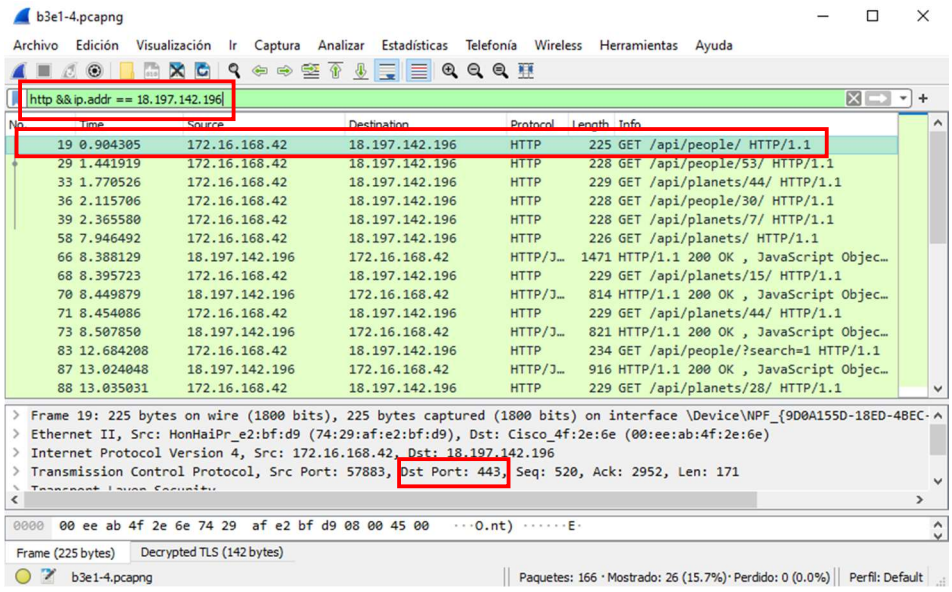


(Comprobación funcionamiento con el comando http.host == swapi.dev. Deben aparecer peticiones GET típicas de HTTP)

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.

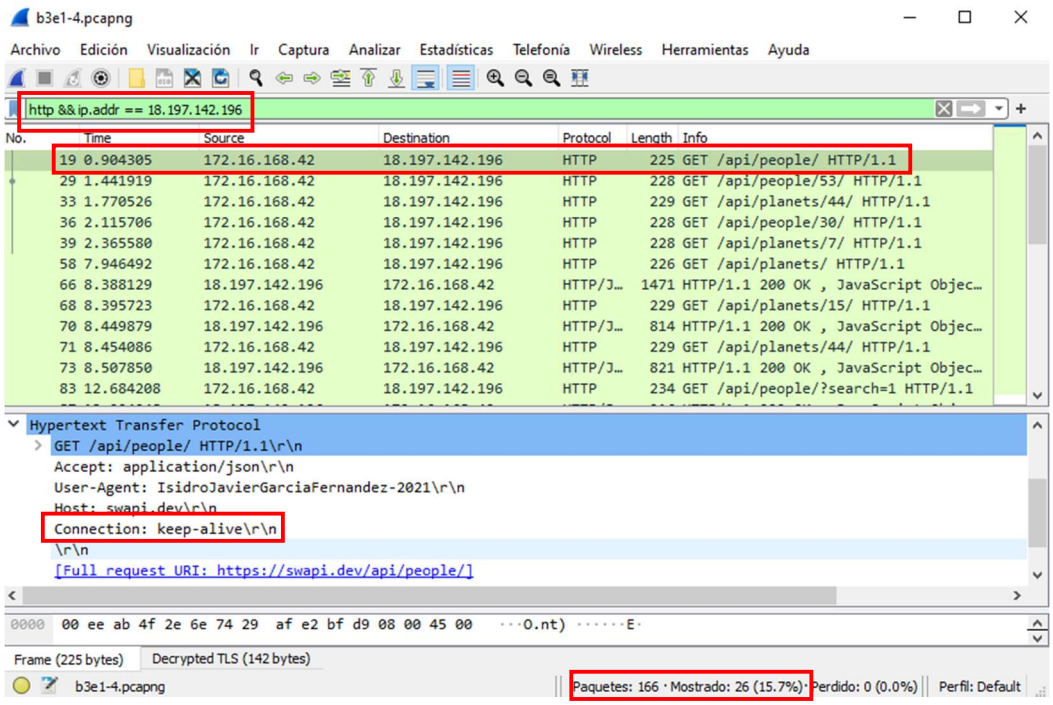
- Tramas analizadas:



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*

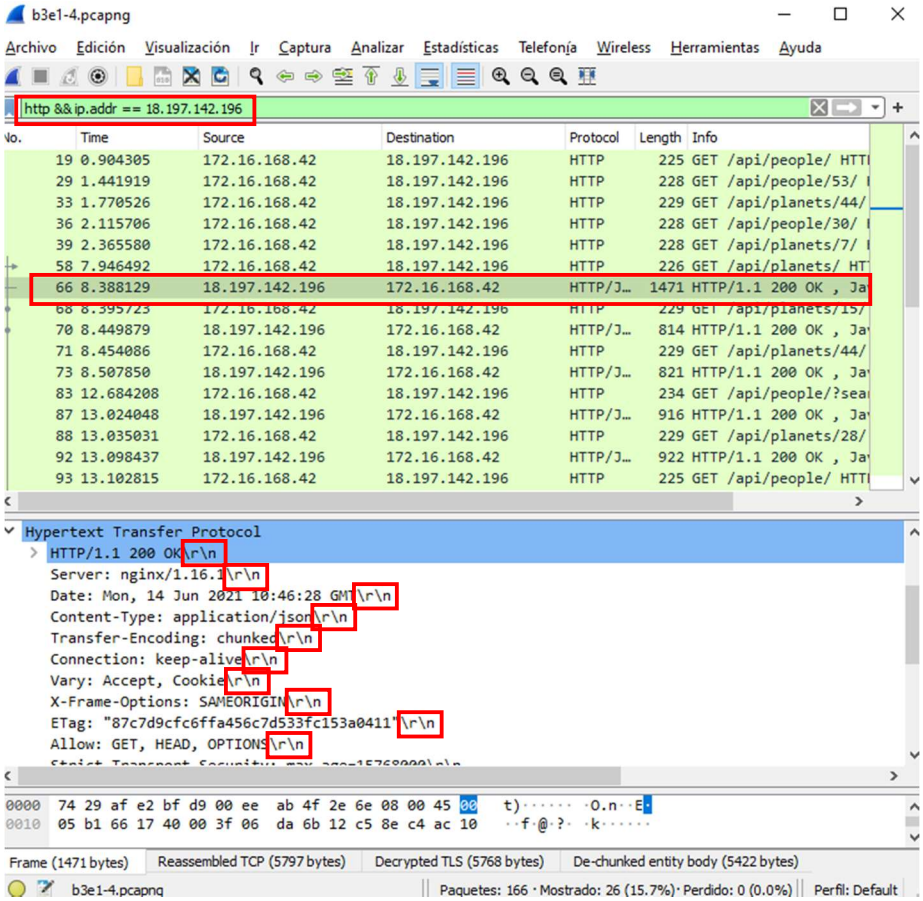
- Tramas analizadas:



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:



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

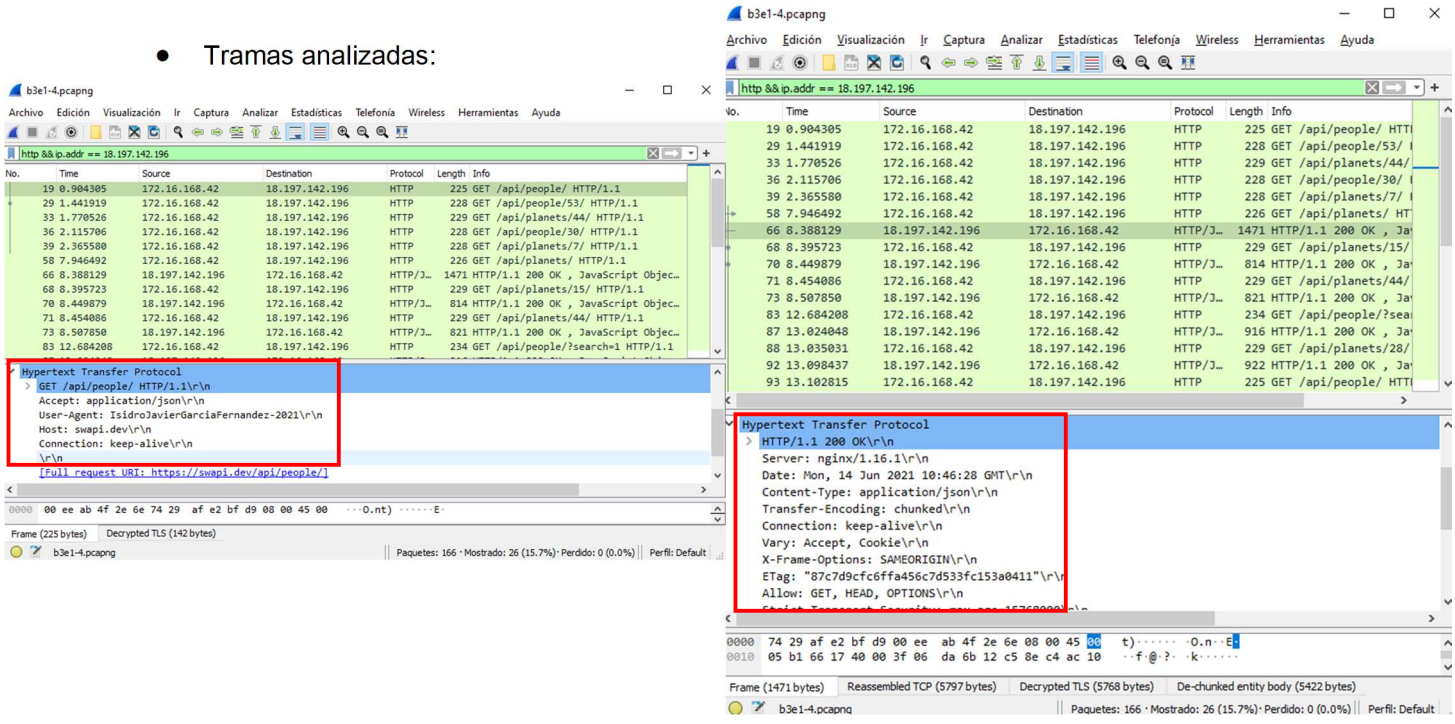
Petición:

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:



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)

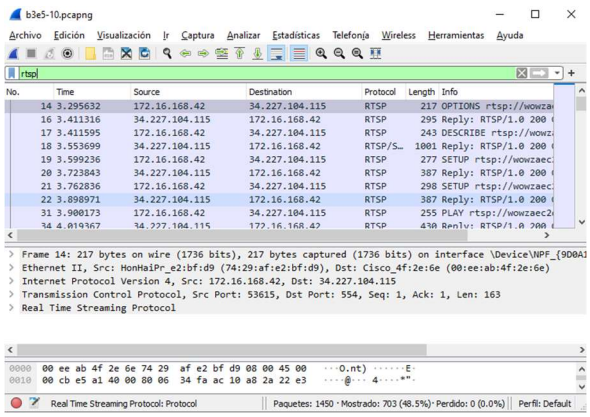
```
PLAY rtsp://wowzaec2demo.streamlock.net:554/vod/
mp4:BigBuckBunny_115k.mov/ RTSP/1.0
CSeq: 6
User-Agent: LibVLC/3.0.14 (LIVE555 Streaming Media v2016.11.28)
Session: 1280767087
Range: npt=0.000-
```

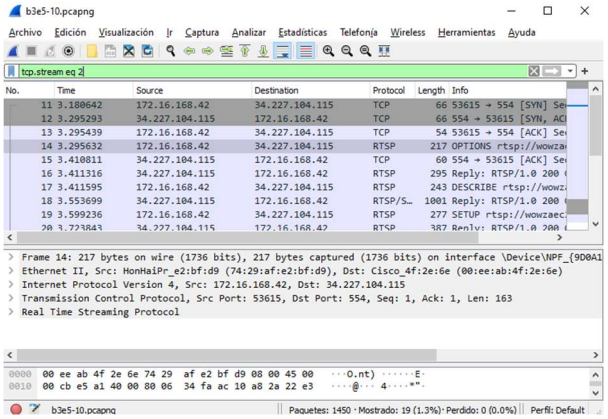
PLAY: comienza la transmisión de datos

```
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)

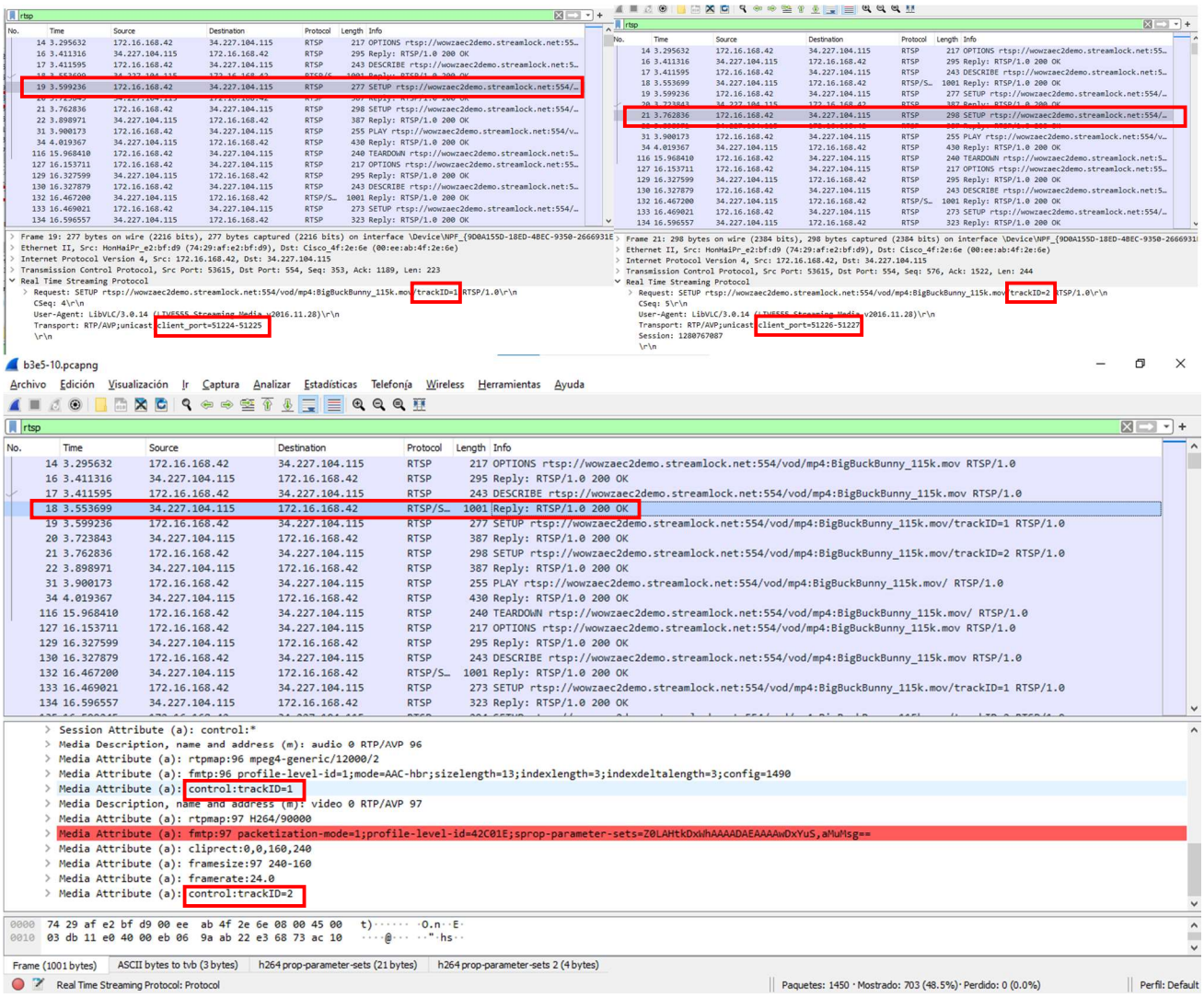
- Tramas analizadas:





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

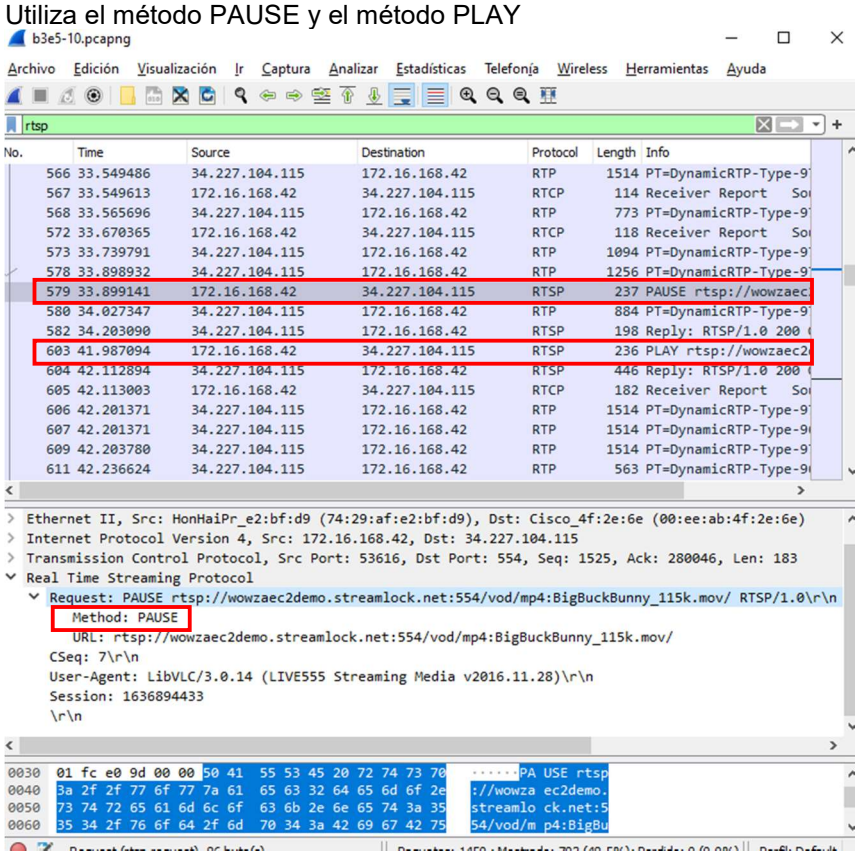
- Tramas analizadas:



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

- Tramas analizadas:

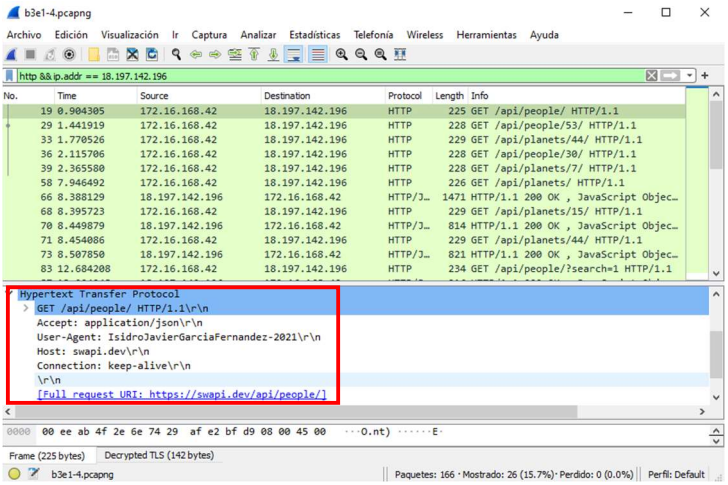
Utiliza el método PAUSE y el método PLAY

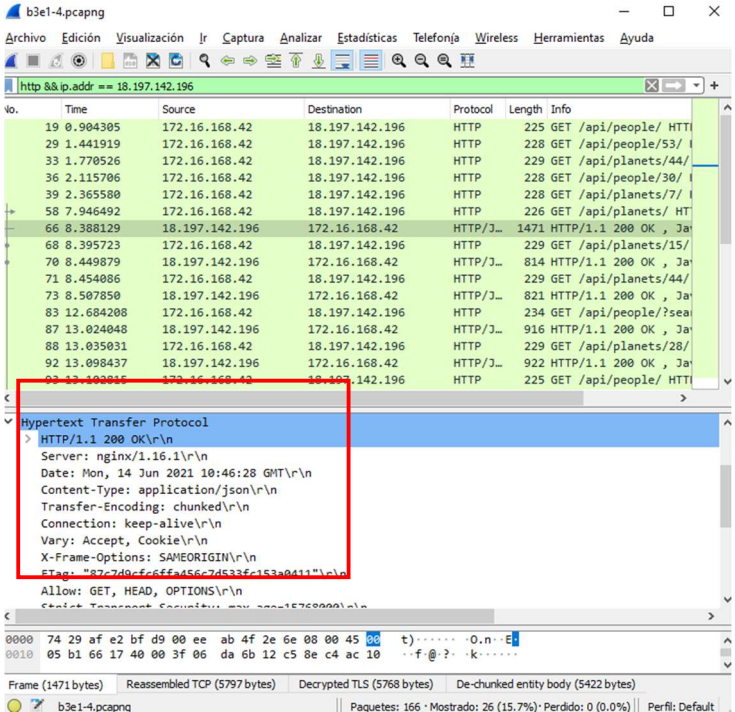


Ejercicio 8. Si observa los comandos y las respuestas son muy similares a las que usa HTTP. Indique dos cabeceras que use RTSP que también se usen en HTTP e indique (y explique) dos cabeceras de RTSP que no se usen en HTTP.

<u>Comunes a HTTP y RTSP:</u> User-Agent y Date
<u>No comunes a HTTP y RTSP:</u> CSeq y Cache-control
Cseq: número de secuencia Cache-control: si se utiliza la memoria caché

- Tramas analizadas:





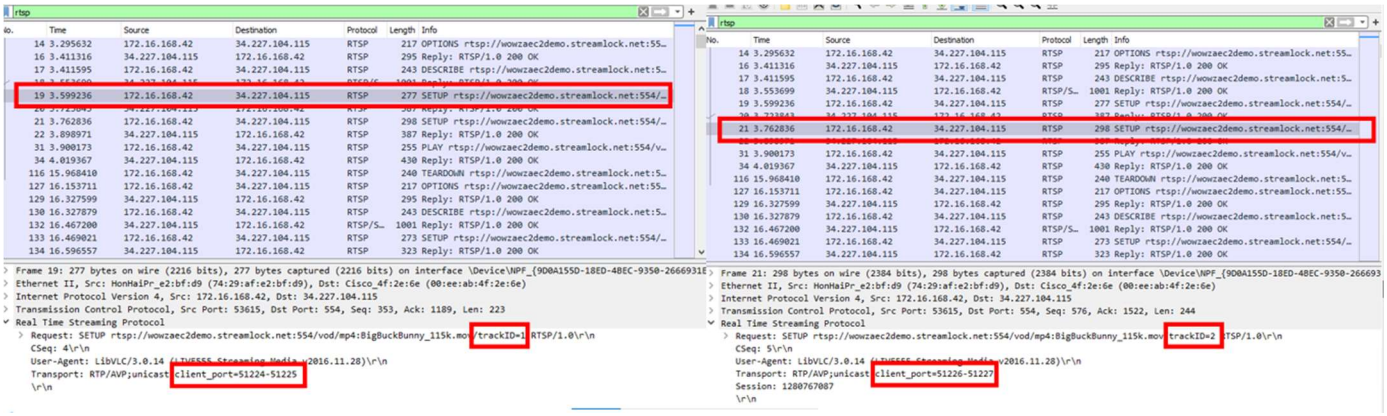
En HTTP **En RTSP**

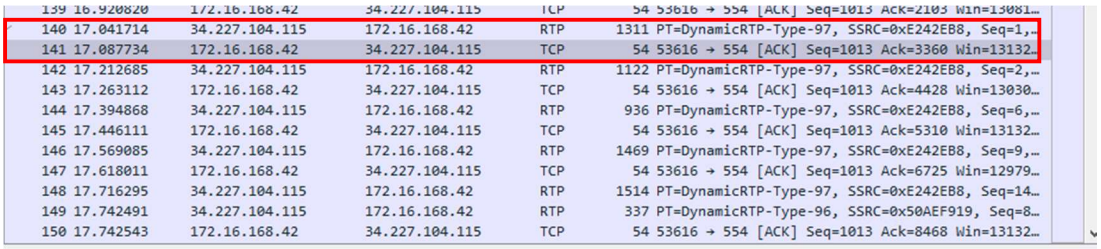
```
SETUP rtsp://wowzaec2demo.streamlock.net:554/vod/
mp4:BigBuckBunny_115k.mov/trackID=1 RTSP/1.0
CSeq: 4
User-Agent: LibVLC/3.0.14 (LIVE555 Streaming Media v2016.11.28)
Transport: RTP/AVP/TCP;unicast;interleaved=0-1

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
```

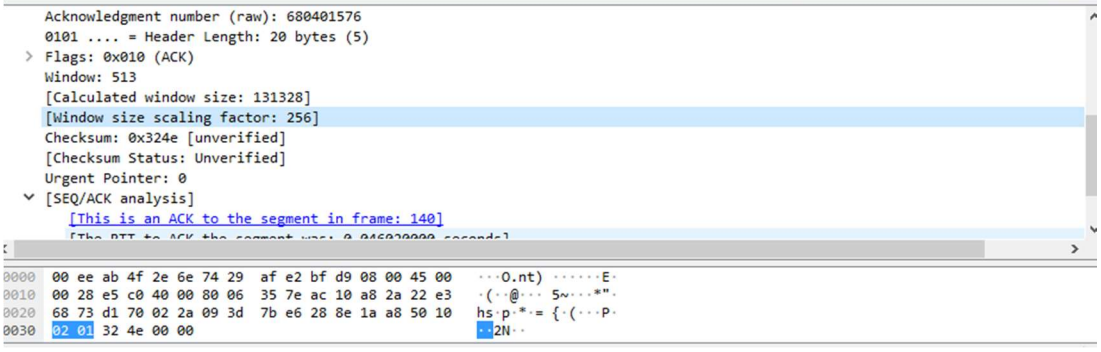
Ejercicio 9. Ahora filtre por el protocolo rtp que se utiliza para transmitir el recurso multimedia tal cual. ¿Cómo se decidieron los puertos a utilizar en estas comunicaciones RTP? ¿Se confirman de alguna forma cada uno de los envíos RTP?

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:







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.

Fraccion 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: