PRACTICAS DE LABORATORIO

PRACTICA BLOQUE 3

REDES Y SISTEMAS DISTRIBUIDOS
CURSO 2021/22

LUIS PATIÑO CAMACHO

UNIVERSIDAD DE MÁLAGA

TAREA 1: Creación de un cliente básico de SWAPI en Java
CLICK VIDEO YT - LUIS PATIÑO

TAREA 2: Análisis de tráfico HTTP/1.1 generado por la aplicación

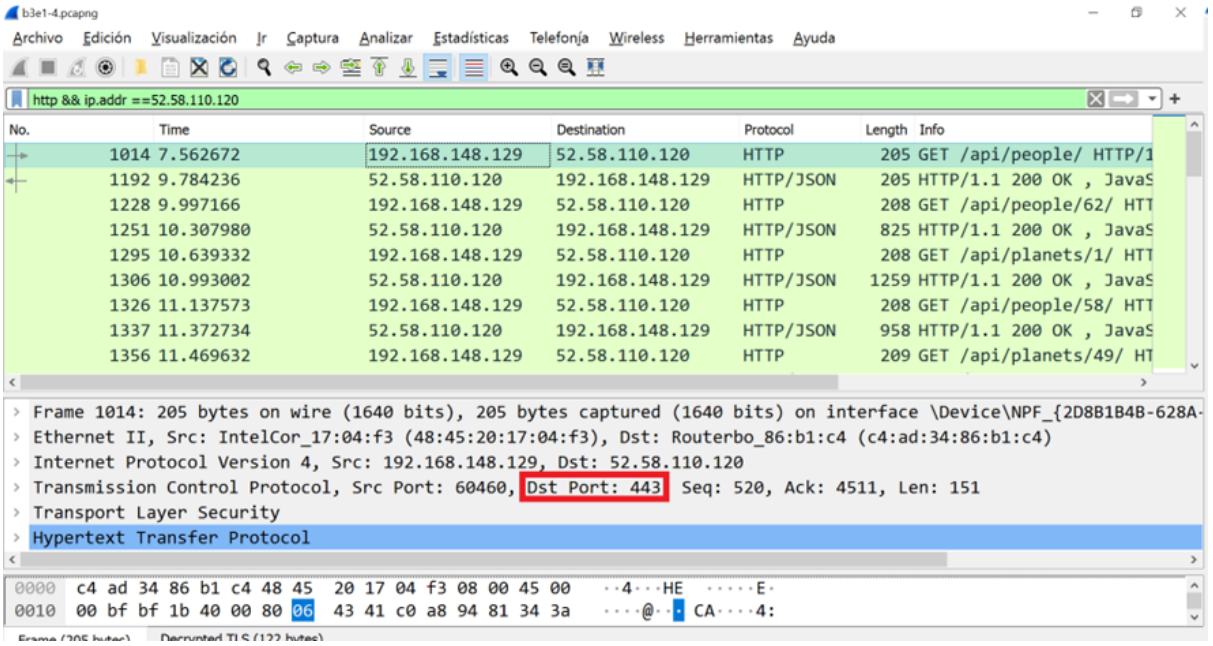
EJERCICIO 1

- ¿Cuál es el puerto utilizado por el servidor?

El puerto utilizado por el servidor es el 443.

- ¿Es el normal de HTTP (80)? ¿Por qué?

No. Porque no estamos utilizando el protocolo http si no, https la versión segura del mismo cuyo puerto normal es el 443.



Trama usada: 1014

EJERCICIO 2

- Observe el número de conexiones realizadas. ¿Cuántas hace?

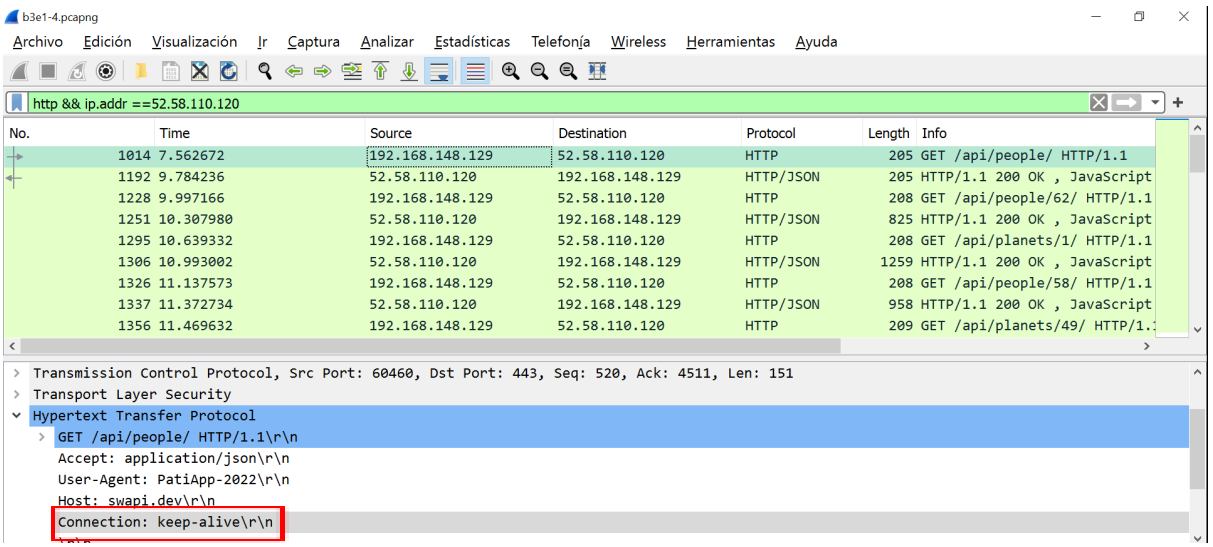
Realiza tan solo una conexión.

- ¿Usa una conexión permanente (en la misma conexión hace varias peticiones) o no permanente (sólo realiza una por conexión)?

Realiza una conexión permanente, en la misma conexión hace varias peticiones.

- En caso de ser permanente, ¿qué cabecera de la petición indica que queremos que sea permanente?

La cabecera connection que contiene el valor keep-alive.



EJERCICIO 3

- Describa el significado de las cabeceras de una petición y una respuesta (sin incluir las X-*).

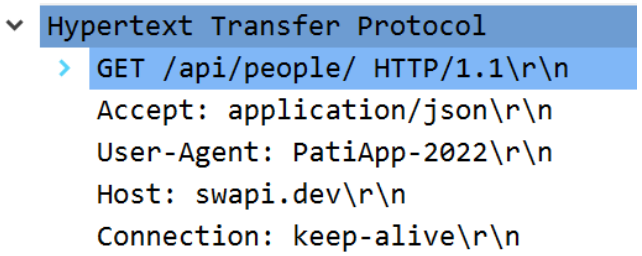
  Cabecera petición: (trama 1014)

- La cabecera de pedido **Accept** anuncia que tipo de contenido el cliente puede procesar, expresado como un tipo MIME.

- La solicitud de cabecera del **Agente de Usuario** contiene una cadena característica que permite identificar el protocolo de red que ayuda a descubrir el tipo de aplicación, sistema operativo, proveedor del software o la versión del software de la petición del agente de usuario.

- El encabezado de solicitud **Host** especifica el nombre de dominio del servidor (para hosting virtual), y (opcionalmente) el número de puerto TCP en el que el servidor está escuchando.

- El encabezado **Connection** controla si la conexión de red permanece abierta después de que finaliza la transacción actual. Si el valor enviado es keep-alive, la conexión es persistente y no cerrada, lo que permite realizar solicitudes posteriores al mismo servidor.



Cabecera de respuesta: (trama 1192)

- La cabecera **Server** contiene la información acerca del software usado por el servidor original encargado de la solicitud.

- La cabecera **date** contiene la fecha y la hora en la que se creó el mensaje.

- **Content-Type** dice al cliente que tipo de contenido será retornado.

- El encabezado **Transfer-Encoding** especifica la forma de codificación utilizada para transferir de forma segura el cuerpo del payload al usuario.

- El encabezado **Connection** controla si la conexión de red permanece abierta después de que finaliza la transacción actual.

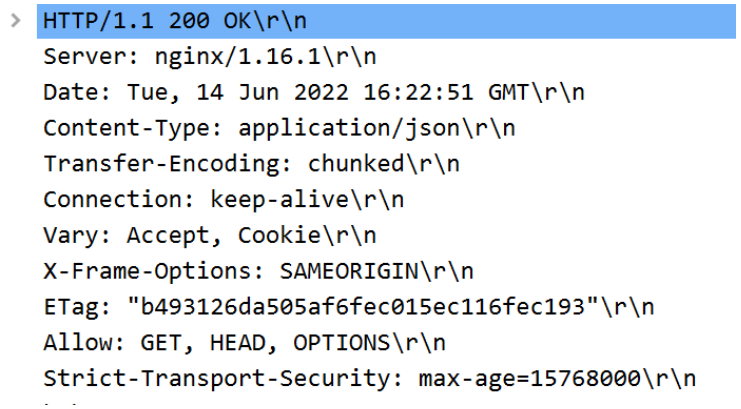
- El encabezado **Vary** describe las partes del mensaje de solicitud además del método y la URL que influyeron en el contenido de la respuesta en la que se produce.

- El encabezado de respuesta HTTP **X-Frame-Options** puede ser usado para indicar si debería permitirsele a un navegador renderizar una página en un <frame>, <iframe>, <embed> u <object>. Las páginas web pueden usarlo para evitar ataques de click-jacking, asegurándose de que su contenido no es embebido en otros sitios.

- El encabezado de respuesta de HTTP **ETag** es un identificador para una versión específica de un recurso. Permite a la memoria caché ser más eficiente, y ahorrar ancho de banda, en tanto que un servidor web no necesita enviar una respuesta completa si el contenido no ha cambiado. Por otro lado, si el contenido cambió, los etags son útiles para ayudar a prevenir actualizaciones simultáneas de un recurso de sobre-escribirlo por otro ("colisiones en el aire").

- La cabecera **Allow** enumera el conjunto de métodos admitidos por un recurso.

- **Strict-Transport-Security** es una característica de seguridad que permite a un sitio web indicar a los navegadores que sólo se debe comunicar con HTTPS en lugar de usar HTTP.

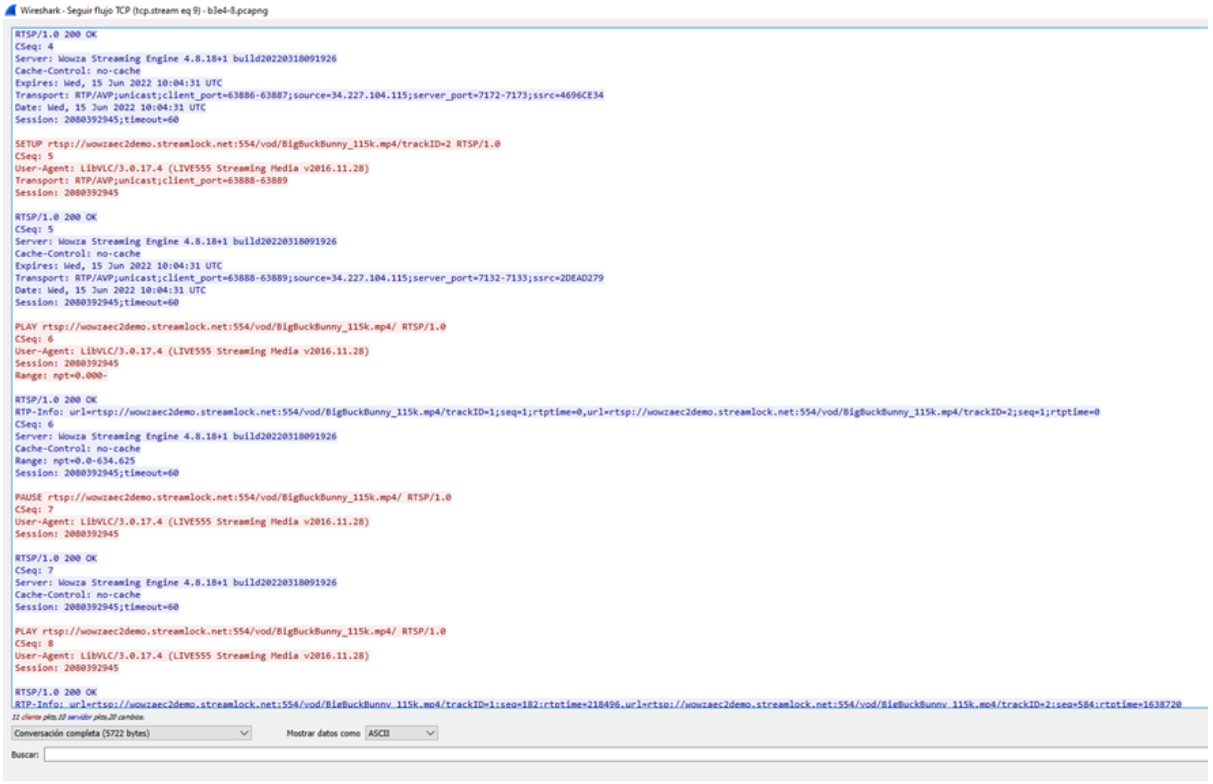


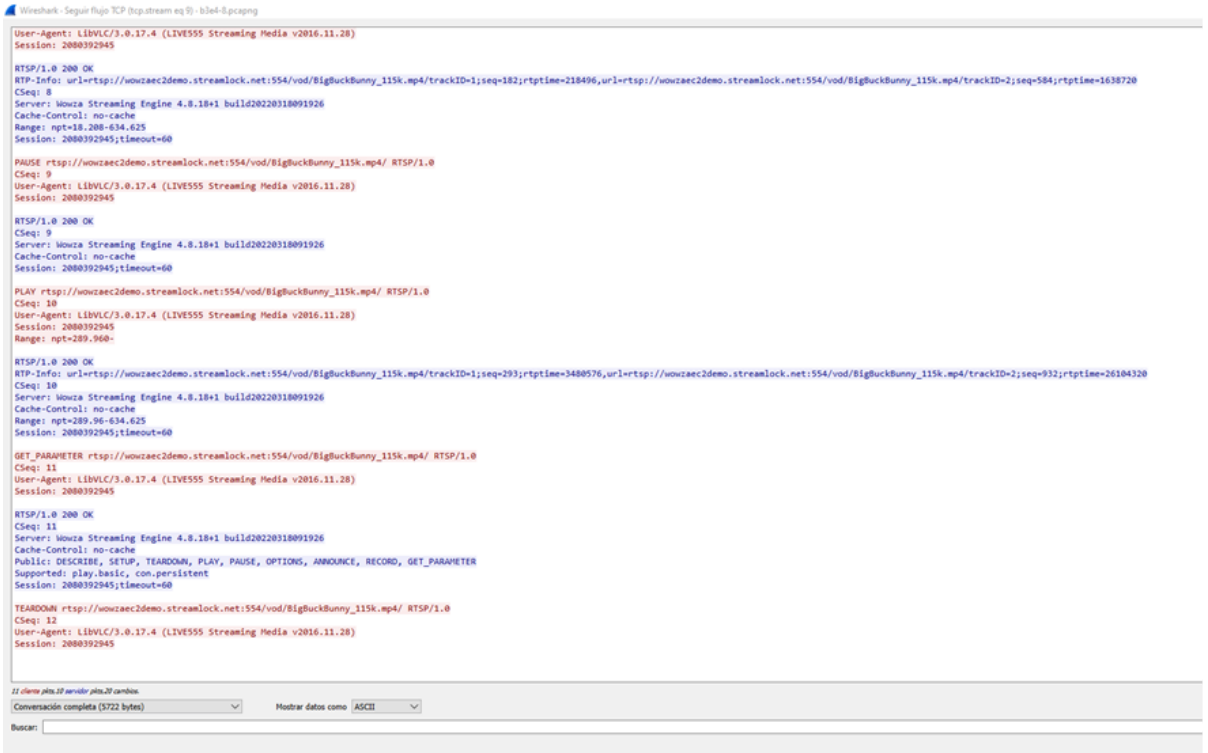
TAREA 3: Análisis de tráfico multimedia (RTSP)

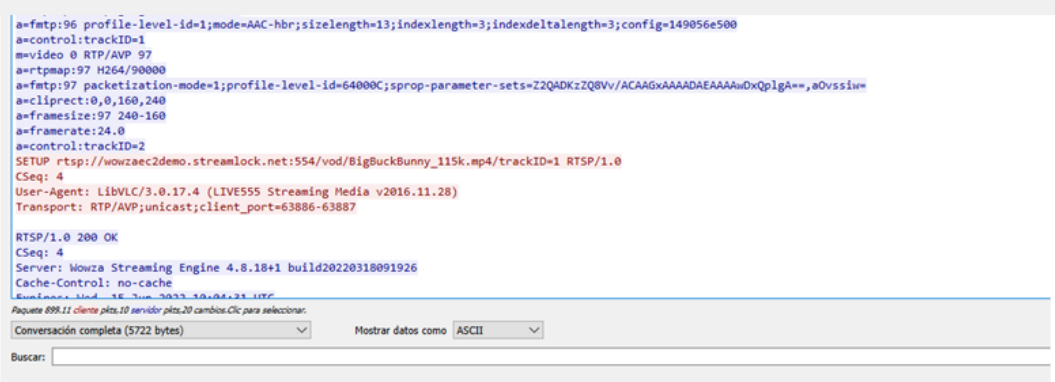
EJERCICIO 4

Filtre por el protocolo rtsp y use la opción Follow TCP Stream de Wireshark para observar el diálogo completo que han mantenido el cliente 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).
Comandos enviados por el cliente:

- OPTIONS: petición enviada por el cliente al servidor cuando la conexión se ha establecido correctamente.
- DESCRIBE: petición enviada por el cliente al servidor para obtener una descripción de la presentación
- SETUP: especifica los protocolos aceptados para el transporte de los datos.
- PLAY: provoca que el servidor comience a enviar datos.
- PAUSE: detiene temporalmente uno o todos los flujos.
- GET_PARAMETER: consigue el valor de un parámetro de una presentación o flujo.
- TEARDOWN: el cliente solicita al servidor que detenga el envío del flujo especificado y libere todos los recursos asociados a él.


Wireshark · Seguir flujo TCP (tcp.stream eq 9) · b3e4-8.pcapng
RTSP/1.0 200 OK
CSeq: 4
Server: Wowza Streaming Engine 4.8.18+1 build20220318091926
Cache-Control: no-cache
Expires: Wed, 15 Jun 2022 10:04:31 UTC
Transport: RTP/AVP;unicast;client_port=63886-63887;source=34.227.104.115;server_port=7172-7173;ssrc=4696CE34
Date: Wed, 15 Jun 2022 10:04:31 UTC
Session: 2080392945;timeout=60
SETUP rtsp://wowzaec2demo.streamlock.net:554/vod/BigBuckBunny_115k.mp4/trackID=2 RTSP/1.0
CSeq: 5
User-Agent: LibVLC/3.0.17.4 (LIVE555 Streaming Media v2016.11.28)
Transport: RTP/AVP;unicast;client_port=63888-63889
Session: 2080392945
RTSP/1.0 200 OK
CSeq: 5
Server: Wowza Streaming Engine 4.8.18+1 build20220318091926
Cache-Control: no-cache
Expires: Wed, 15 Jun 2022 10:04:31 UTC
Transport: RTP/AVP;unicast;client_port=63888-63889;source=34.227.104.115;server_port=7132-7133;ssrc=2DEAD279
Date: Wed, 15 Jun 2022 10:04:31 UTC
Session: 2080392945;timeout=60
PLAY rtsp://wowzaec2demo.streamlock.net:554/vod/BigBuckBunny_115k.mp4/ RTSP/1.0
CSeq: 6
User-Agent: LibVLC/3.0.17.4 (LIVE555 Streaming Media v2016.11.28)
Session: 2080392945
Range: npt=0.000-
RTSP/1.0 200 OK
RTP-Info: url=rtsp://wowzaec2demo.streamlock.net:554/vod/BigBuckBunny_115k.mp4/trackID=1;seq=1;rtptime=0,url=rtsp://wowzaec2demo.streamlock.net:554/vod/BigBuckBunny_115k.mp4/trackID=2;seq=1;rtptime=0
CSeq: 6
Server: Wowza Streaming Engine 4.8.18+1 build20220318091926
Cache-Control: no-cache
Range: npt=0.0-634.625
Session: 2080392945;timeout=60
PAUSE rtsp://wowzaec2demo.streamlock.net:554/vod/BigBuckBunny_115k.mp4/ RTSP/1.0
CSeq: 7
User-Agent: LibVLC/3.0.17.4 (LIVE555 Streaming Media v2016.11.28)
Session: 2080392945
RTSP/1.0 200 OK
CSeq: 7
Server: Wowza Streaming Engine 4.8.18+1 build20220318091926
Cache-Control: no-cache
Session: 2080392945;timeout=60
PLAY rtsp://wowzaec2demo.streamlock.net:554/vod/BigBuckBunny_115k.mp4/ RTSP/1.0
CSeq: 8
User-Agent: LibVLC/3.0.17.4 (LIVE555 Streaming Media v2016.11.28)
Session: 2080392945
RTSP/1.0 200 OK
RTP-Info: url=rtsp://wowzaec2demo.streamlock.net:554/vod/BigBuckBunny_115k.mp4/trackID=1;seq=182;rtptime=218496,url=rtsp://wowzaec2demo.streamlock.net:554/vod/BigBuckBunny_115k.mp4/trackID=2;seq=584;rtptime=1638720
11 cliente pkts,10 servidor pkts,10 cambios.
Conversación completa (5722 bytes)
Mostrar datos como ASCII
Buscar:


Wireshark · Seguir flujo TCP (tcp.stream eq 9) · b3e4-8.pcapng
User-Agent: LibVLC/3.0.17.4 (LIVE555 Streaming Media v2016.11.28)
Session: 2080392945
RTSP/1.0 200 OK
RTP-Info: url=rtsp://wowzaec2demo.streamlock.net:554/vod/BigBuckBunny_115k.mp4/trackID=1;seq=182;rtptime=218496,url=rtsp://wowzaec2demo.streamlock.net:554/vod/BigBuckBunny_115k.mp4/trackID=2;seq=584;rtptime=1638720
CSeq: 8
Server: Wowza Streaming Engine 4.8.18+1 build20220318091926
Cache-Control: no-cache
Range: npt=18.208-634.625
Session: 2080392945;timeout=60
PAUSE rtsp://wowzaec2demo.streamlock.net:554/vod/BigBuckBunny_115k.mp4/ RTSP/1.0
CSeq: 9
User-Agent: LibVLC/3.0.17.4 (LIVE555 Streaming Media v2016.11.28)
Session: 2080392945
RTSP/1.0 200 OK
CSeq: 9
Server: Wowza Streaming Engine 4.8.18+1 build20220318091926
Cache-Control: no-cache
Session: 2080392945;timeout=60
PLAY rtsp://wowzaec2demo.streamlock.net:554/vod/BigBuckBunny_115k.mp4/ RTSP/1.0
CSeq: 10
User-Agent: LibVLC/3.0.17.4 (LIVE555 Streaming Media v2016.11.28)
Session: 2080392945
Range: npt=289.960-
RTSP/1.0 200 OK
RTP-Info: url=rtsp://wowzaec2demo.streamlock.net:554/vod/BigBuckBunny_115k.mp4/trackID=1;seq=293;rtptime=3480576,url=rtsp://wowzaec2demo.streamlock.net:554/vod/BigBuckBunny_115k.mp4/trackID=2;seq=932;rtptime=26104320
CSeq: 10
Server: Wowza Streaming Engine 4.8.18+1 build20220318091926
Cache-Control: no-cache
Range: npt=289.96-634.625
Session: 2080392945;timeout=60
GET_PARAMETER rtsp://wowzaec2demo.streamlock.net:554/vod/BigBuckBunny_115k.mp4/ RTSP/1.0
CSeq: 11
User-Agent: LibVLC/3.0.17.4 (LIVE555 Streaming Media v2016.11.28)
Session: 2080392945
RTSP/1.0 200 OK
CSeq: 11
Server: Wowza Streaming Engine 4.8.18+1 build20220318091926
Cache-Control: no-cache
Public: DESCRIBE, SETUP, TEARDOWN, PLAY, PAUSE, OPTIONS, ANNOUNCE, RECORD, GET_PARAMETER
Supported: play.basic, con.persistent
Session: 2080392945;timeout=60
TEARDOWN rtsp://wowzaec2demo.streamlock.net:554/vod/BigBuckBunny_115k.mp4/ RTSP/1.0
CSeq: 12
User-Agent: LibVLC/3.0.17.4 (LIVE555 Streaming Media v2016.11.28)
Session: 2080392945
11 cliente pkts,10 servidor pkts,20 cambios.
Conversación completa (5722 bytes)
Mostrar datos como ASCII
Buscar:


a=fmtp:96 profile-level-id=1;mode=AAC-hbr;sizelength=13;indexlength=3;indexdeltalength=3;config=149056e500
a=control:trackID=1
m=video 0 RTP/AVP 97
a=rtpmap:97 H264/90000
a=fmtp:97 packetization-mode=1;profile-level-id=64000C;sprop-parameter-sets=Z2QADKzZQ8Vv/ACAAGxAAAADAEAAAAwDxQplgA==,aOvssiw=
a=cliprect:0,0,160,240
a=framesize:97 240-160
a=framerate:24.0
a=control:trackID=2
SETUP rtsp://wowzaec2demo.streamlock.net:554/vod/BigBuckBunny_115k.mp4/trackID=1 RTSP/1.0
CSeq: 4
User-Agent: LibVLC/3.0.17.4 (LIVE555 Streaming Media v2016.11.28)
Transport: RTP/AVP;unicast;client_port=63886-63887
RTSP/1.0 200 OK
CSeq: 4
Server: Wowza Streaming Engine 4.8.18+1 build20220318091926
Cache-Control: no-cache
Paquete 899.11 cliente pkts,10 servidor pkts,20 cambios.Clic para seleccionar.
Conversación completa (5722 bytes)
Mostrar datos como ASCII
Buscar:

EJERCICIO 5

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

Necesitaremos uno para la transmisión de audio y otro para la transmisión de vídeo. En la respuesta del servidor al comando DESCRIBE podemos observar que hay dos medios uno de tipo audio y otro de tipo video. Además como podemos ver en la respuesta al comando DESCRIBE los identifica como trackID1 y trackID2, y estos valores luego los utiliza en cada uno de los comandos SETUP.

```
c=IN IP4 34.227.104.115
t=0 0
a=sdplang:en
a=range:npt=0- 634.625
a=control:*
m=audio 0 RTP/AVP 96
a=rtpmap:96 mpeg4-generic/12000/2
a=fmtp:96 profile-level-id=1;mode=AAC-hbr;sizelength=13;indexlength=3;indexdeltalength=3;config=149056e500
a=control:trackID=1
m=video 0 RTP/AVP 97
a=rtpmap:97 H264/90000
a=fmtp:97 packetization-mode=1;profile-level-id=64000C;sprop-parameter-sets=Z2QADKzZQ8Vv/ACAAGxAAAADAEAAAAwDxQplgA==,aOvssiw=
a=cliprect:0,0,160,240
a=framesize:97 240-160
a=framerate:24.0
a=control:trackID=2
SETUP rtsp://wowzaec2demo.streamlock.net:554/vod/BigBuckBunny_115k.mp4/trackID=1 RTSP/1.0
CSeq: 4
User-Agent: LibVLC/3.0.17.4 (LIVE555 Streaming Media v2016.11.28)
Transport: RTP/AVP;unicast;client_port=63886-63887

RTSP/1.0 200 OK
CSeq: 4
Server: Wowza Streaming Engine 4.8.18+1 build20220318091926
Cache-Control: no-cache
Expires: Wed, 15 Jun 2022 10:04:31 UTC
Transport: RTP/AVP;unicast;client_port=63886-63887;source=34.227.104.115;server_port=7172-7173;ssrc=4696CE34
Date: Wed, 15 Jun 2022 10:04:31 UTC
Session: 2080392945;timeout=60

SETUP rtsp://wowzaec2demo.streamlock.net:554/vod/BigBuckBunny_115k.mp4/trackID=2 RTSP/1.0
CSeq: 5
User-Agent: LibVLC/3.0.17.4 (LIVE555 Streaming Media v2016.11.28)
Transport: RTP/AVP;unicast;client_port=63888-63889
Session: 2080392945
```

EJERCICIO 6

¿Qué comandos ha provocado adelantar la reproducción del vídeo? ¿Cómo indica por donde debe seguir la reproducción tras el cambio?
Primero provoca un pause y luego un play. Pero si nos fijamos en el play, vemos que está vez tiene un rango, que es el que nos indicará por donde debe seguir la reproducción.

```
PAUSE rtsp://wowzaec2demo.streamlock.net:554/vod/BigBuckBunny_115k.mp4/ RTSP/1.0
CSeq: 9
User-Agent: LibVLC/3.0.17.4 (LIVE555 Streaming Media v2016.11.28)
Session: 2080392945

RTSP/1.0 200 OK
CSeq: 9
Server: Wowza Streaming Engine 4.8.18+1 build20220318091926
Cache-Control: no-cache
Session: 2080392945;timeout=60

PLAY rtsp://wowzaec2demo.streamlock.net:554/vod/BigBuckBunny_115k.mp4/ RTSP/1.0
CSeq: 10
User-Agent: LibVLC/3.0.17.4 (LIVE555 Streaming Media v2016.11.28)
Session: 2080392945
Range: npt=289.960-
```

EJERCICIO 7

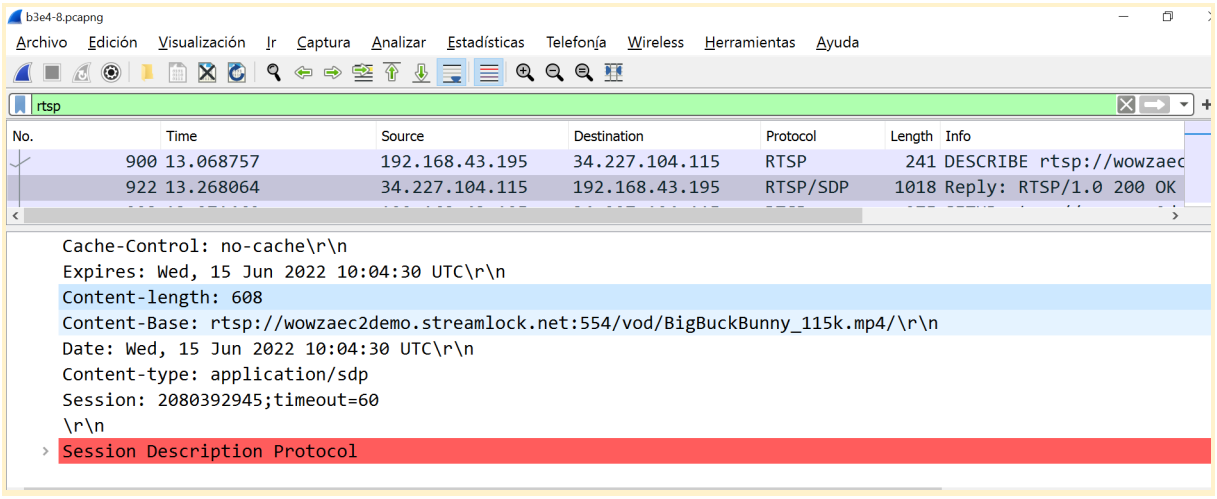
Si observa los comandos y las respuestas son muy similares a las que usa HTTP. Indique dos cabeceras que use RTSP que también se usen en HTTP y explique dos cabeceras de RTSP que no se usen en HTTP.

Cabeceras que usan ambos:

- Content-length
- Date

Cabeceras de RSTP que no se usan en HTTP:

- Expires: indica la fecha de expiración.
- Content-base:contiene la url al vídeo



EJERCICIO 8

Ahora añada al filtro que aparezcan también los paquetes del 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?

Se establecieron en el SETUP. En RTP los envíos no se confirman, ya que están basados en UDP.