

PRÁCTICA 2: INTRODUCCIÓN A LA PROGRAMACIÓN MULTIMEDIA.

Héctor Menéndez

AIDA Research Group
Computer Science Department
Universidad Autónoma de Madrid

20 de febrero de 2013

Index

- 1 Objetivos
- 2 Ejercicio Obligatorio: Incorporar Audio al Chat
- 3 RTSP
- 4 RTP
- 5 Módulos a implementar
- 6 Manual de Ejecución

Objetivos

Los objetivos de la presente práctica son:

- Familiarizarse con los protocolos multimedia.
- Crear un chat con comunicación de voz por streaming.

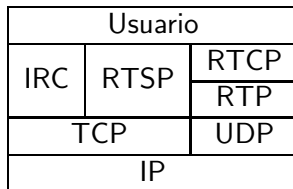
Ejercicio Obligatorio: Incorporar Audio al Chat

- Se debe implementar un servidor y un cliente de streaming de vídeo y audio.
- Para ello se utilizarán los siguientes protocolos:
 - RTSP (*Real-Time Streaming Protocol*): es un protocolo de comunicación ideado para el intercambio de material multimedia en tiempo real (ver detalles en RFC 2326¹).
 - RTP (*Real-Time Transfer Protocol*): es un protocolo de transporte que se ejecuta sobre UDP y que se utiliza para la transmisión de los datos multimedia (ver detalles en RFC 3550²).

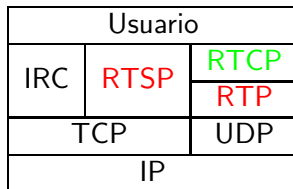
¹RFC 2326: <http://tools.ietf.org/html/rfc2326>

²RFC 3550: <http://tools.ietf.org/html/rfc3550>

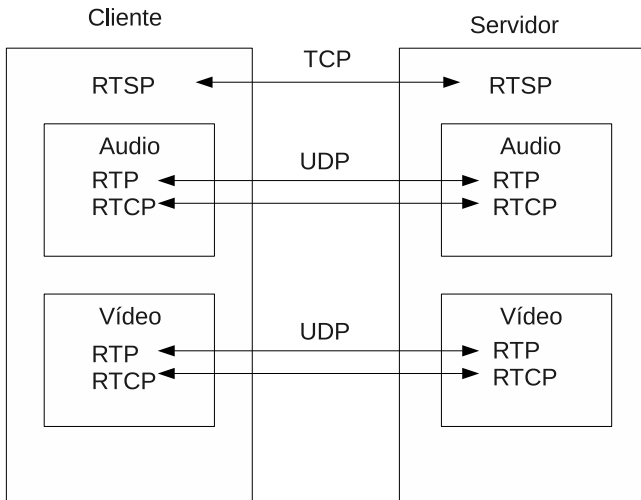
Esquema General: Capas



Esquema General: Capas



Esquema General: Comunicación



RTSP

- RTSP normalmente se ejecuta sobre TCP y sirve para controlar la transmisión de streams de vídeo y audio mediante el envío de “comandos” desde el cliente que solicitan la ejecución de una determinada acción al servidor.
- Es necesario implementar obligatoriamente los siguientes comandos:

Comandos RTSP

- **DESCRIBE** (RFC 2326 10.2). Comando que sirve para recoger la información pertinente de los distintos canales multimedia. En nuestro caso audio. El alumno debe tener en cuenta el protocolo SDP (*Session Description Protocol*, ver detalles en RFC 2327³) a la hora de responder a este comando.
- **SETUP** (RFC 2326 10.4). Comando que se utiliza para comenzar la sesión e inicializar los parámetros del protocolo.
- **PLAY** (RFC 2326 10.5). Este comando es el que inicia la reproducción.
- **PAUSE** (RFC 2326 10.6). Este comando es el encargado de parar la reproducción.
- **TEARDOWN** (RFC 2326 10.7). Este comando termina la sesión y cierra la conexión.

³RFC 2327: <http://tools.ietf.org/html/rfc2327>

Ejemplo RTSP

```
C->S: DESCRIBE rtsp://foo/twister RTSP/1.0
      CSeq: 1
S->C: RTSP/1.0 200 OK
      CSeq: 1
      Content-Type: application/sdp
      Content-Length: 164
      v=0
      o=- 2890844256 2890842807 IN IP4 172.16.2.93
      s=RTSP Session
      i=An Example of RTSP Session Usage
      a=control:rtsp://foo/twister
      t=0 0
      m=audio 0 RTP/AVP 0
      a=control:rtsp://foo/twister/audio
      m=video 0 RTP/AVP 26
      a=control:rtsp://foo/twister/video
```

Ejemplo RTSP

```
C->S: SETUP rtsp://foo/twister/audio RTSP/1.0
      CSeq: 2
      Transport: RTP/AVP;unicast;client_port=8000-8001
S->C: RTSP/1.0 200 OK
      CSeq: 2
      Transport: RTP/AVP;unicast;client_port=8000-8001;
                server_port=9000-9001
      Session: 12345678
C->S: SETUP rtsp://foo/twister/video RTSP/1.0
      CSeq: 3
      Transport: RTP/AVP;unicast;client_port=8002-8003
      Session: 12345678
S->C: RTSP/1.0 200 OK
      CSeq: 3
      Transport: RTP/AVP;unicast;client_port=8002-8003;
                server_port=9004-9005
      Session: 12345678
```

Ejemplo RTSP

```
C->S: PLAY rtsp://foo/twister RTSP/1.0
      CSeq: 4
      Range: npt=0-
      Session: 12345678
S->C: RTSP/1.0 200 OK
      CSeq: 4
      Session: 12345678
      RTP-Info: url=rtsp://foo/twister/video;
                seq=9810092;rtptime=3450012
```

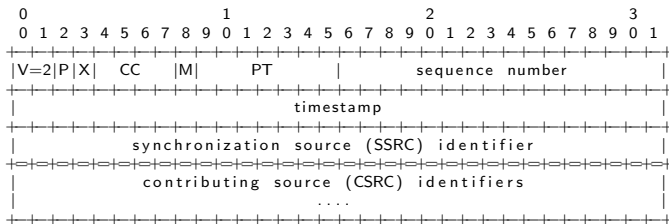
Ejemplo RTSP

```
C->S: PAUSE rtsp://foo/twister/video RTSP/1.0
      CSeq: 5
      Session: 12345678
S->C: RTSP/1.0 460 Only aggregate operation allowed
      CSeq: 5
C->S: PAUSE rtsp://foo/twister RTSP/1.0
      CSeq: 6
      Session: 12345678
S->C: RTSP/1.0 200 OK
      CSeq: 6
      Session: 12345678
```

RTP

- RTP es el protocolo que utilizará el servidor para encapsular los datos multimedia solicitados por un cliente con el protocolo RTSP.
- Los paquetes RTP se encapsularán dentro de una trama UDP y constarán de dos partes: una cabecera y un campo, que se suele denominar de “carga útil”, en el que se almacenarán los distintos bloques del elemento multimedia que se están transmitiendo.

Cabecera RTP



Cabecera RTP

- Versión del protocolo RTP (V). La versión actual es la 2, por lo que este campo siempre tendrá este valor.
- Padding (P). Indica si el paquete contiene bytes de relleno. En nuestro caso esto no va a ocurrir, por lo que siempre valdrá 0.
- Extensión (X). Indica si se utiliza o no una cabecera ampliada. En nuestro caso vamos a utilizar cabeceras simples, por lo que siempre valdrá 0.
- Número de orígenes de datos (CC). Siempre consideraremos fuentes de datos únicas, por lo que por convenio propio el valor de este campo siempre será 0.

Cabecera RTP

- Marcador de eventos significativos (M). En nuestra implementación siempre valdrá 0.
- Tipo de carga (PT). Indica el tipo de codificación para que las aplicaciones lo interpreten de forma correcta. Los posibles valores a utilizar en este campo serán decisión del alumno.
- Número de secuencia. Identificador único del mensaje.
- Timestamp. Indica el instante en el que se generó el primer bit de datos del paquete RTP.
- Identificador de fuentes de sincronización (SSRC).

En esta práctica el alumno deberá implementar cinco módulos principales:

- **RTP:** Este módulo se utilizará para manipular los paquetes RTP.
- **RTSP:** Este módulo se utilizará para manipular los paquetes RTSP.
- **Voz:** Este módulo se encargará de gestionar todo lo referente a la emisión y recepción del audio.

Mensajes: Escenario



IRC Server

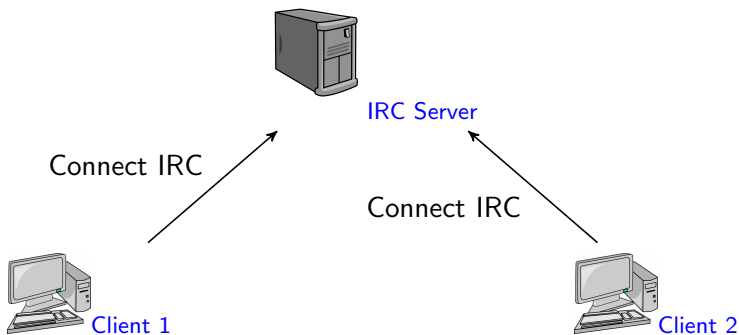


Client 1

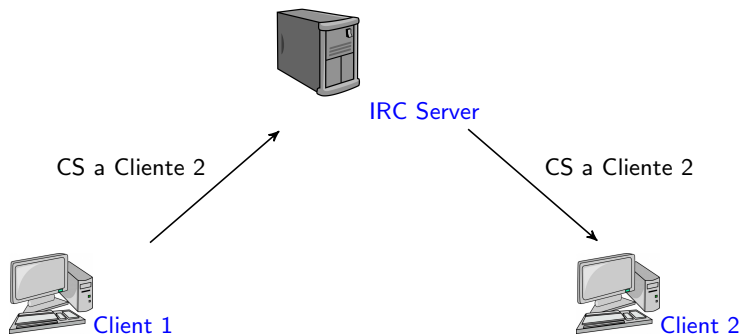


Client 2

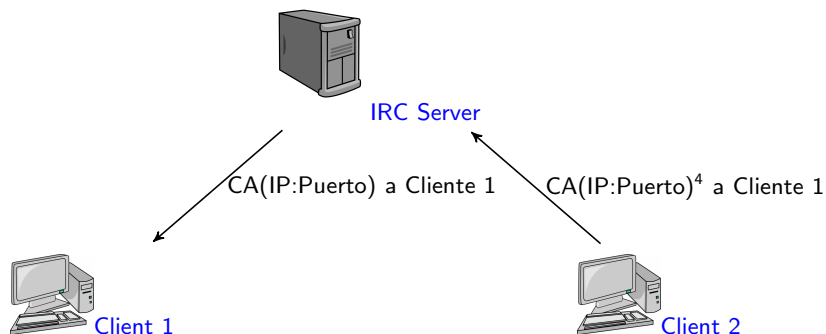
Mensajes: Inicio



Mensajes: Mensaje CS



Mensajes: Mensaje CA



⁴Usar la función `getsockname` para recoger la IP

Mensajes: Conexión RTSP



IRC Server



Client 1

Conecta RTSP



Client 2

Mensajes: DESCRIBE



IRC Server



Client 1

Envía DESCRIBE (con información RTP)



Client 2

Mensajes: Rep. DESCRIBE

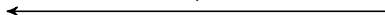


IRC Server



Client 1

Responde DESCRIBE (con información RTP)



Client 2

Mensajes: SETUP



IRC Server



Client 1

Envía SETUP



Client 2

Mensajes: Res. SETUP



IRC Server



Client 1



Responde SETUP



Client 2

Mensajes: PLAY



IRC Server



Client 1



Envía PLAY



Client 2

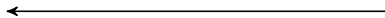
Mensajes: Res. PLAY



IRC Server



Client 1



Responde PLAY



Client 2

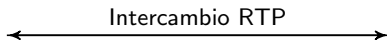
Mensajes: Intercambio RTP



IRC Server



Client 1



Client 2

Mensajes: PAUSE



IRC Server



Client 1

Envía PAUSE



Client 2

Mensajes: Res. PAUSE



IRC Server



Client 1

Responde PAUSE, detiene el intercambio



Client 2

Mensajes: TEARDOWN



IRC Server



Client 1



Client 2

Mensajes: Res. TEARDOWN



IRC Server



Client 1

Responde TEARDOWN, destruye socket



Client 2

Calificación

La puntuación de la práctica se repartirá en:

- 1** Implementación. Se podrá obtener un máximo de 10 puntos si se implementa de forma correcta y funcional todo lo que se solicita para esta práctica. Además, habrá que realizar un manual que explique el funcionamiento de la práctica.

Recomendaciones

Realizar una implementación de arriba a abajo:

- 1 Crear un cliente y un servidor que se envíen comandos RTSP.
- 2 Incorporar a la interfaz los mensajes para las funcionalidades de audio.
- 3 Crear un servicio por RTP para que el cliente reciba un fichero que le pida al servidor.
- 4 Incorporar PortAudio al RTP.
- 5 Recibir los datos (audio) y reproducirlos.