

[Página Principal](#) ► [RSNG](#) ► [Prácticas](#) ► [Práctica entregable de IPv6](#)

## Navegación

[Página Principal](#)[■ Área personal](#)[Páginas del sitio](#)[Mi perfil](#)[Curso actual](#)[RSNG](#)[Participantes](#)[General](#)[Prácticas](#)[📄 Cuentas y grupos de prácticas](#)[🌐 Máquinas virtuales](#)[📄 Practica 1 - Configurando IPv6](#)[📄 \*\*Práctica entregable de IPv6\*\*](#)[Mis cursos](#)[Cursos](#)

## Ajustes

[Administración del curso](#)[Ajustes de mi perfil](#)

## Práctica entregable de IPv6

### Practica - Software IPv6 ready

#### Objetivo

El objetivo de esta práctica es implementar una aplicación distribuida que utilice IPv6 y multicast

#### Problema

Se pretende construir un sistema de anuncio de videos bajo demanda o canales de streaming para una red IPv6. Los elementos de este servicio serán dos

- Un programa servidor que dispone de una lista de programas (videos o canales de streaming) que es capaz de enviar a los clientes que lo pidan. En una red puede haber varias instancias de servidores cada uno con su lista de programas. Estos servidores se anunciarán en un grupo multicast y escucharán peticiones de los clientes para enviarles alguno de sus programas.
- Un programa cliente que es capaz de escuchar los anuncios multicast de todos los servidores y mostrarlos al usuario para que elija un programa de algún servidor. El cliente permitirá al usuario realizar una petición de uno de los programas ofrecidos y reproducirla.

El envío y la recepción y reproducción de los programa queda fuera de este trabajo. No se pide realizar el streaming ni reproducirlo. El servidor tendrá disponibles scripts que envíen un programa a una dirección y puerto y el cliente tendrá disponibles scripts que reciban en un puerto y reproduzcan el contenido multimedia.

Cada grupo tendrá que programar el programa cliente o el servidor.

#### Protocolo

## Anuncio de programas

Los servidores enviarán la lista de los programas que pueden servir en paquetes UDP al grupo multicast de dirección MDIR y puerto MPORT. La lista completa de programas puede ocupar varios paquetes UDP. En cada paquete la información que se enviara será texto con varias líneas con el siguiente formato

```
1: SSER puertoservidor [ipservidor]
2: PRG id titulo
3: PRG id titulo
...
N: [MORE|END]
```

Es decir la primera línea incluirá la palabra clave SSER separado por un espacio el puerto en el que escucha el servidor y opcionalmente la dirección IP en la que escucha al servidor. A continuación una serie de líneas con la palabra clave PRG separada por espacio de un identificador numérico y a continuación el resto de la línea será el título del programa. Esta lista terminará por una línea con la palabra MORE o END indicando si posteriormente habrá más paquetes con más líneas o este es el último bloque de la lista. En caso de que una lista ocupe varios bloques se debe repetir la línea de SSER.

Los números de línea son solo orientativos no se envían. Estos serían ejemplos de anuncios válidos.

### Ejemplo 1

```
En el paquete 1:
SSER 12345
PRG 1 Clase del martes
PRG 2 Camara de la sala 1
PRG 6 Camara de la sala de juntas
END
```

### Ejemplo 2

```
En el paquete 1:
SSER 5000 2001:720:1d10:fff7::123:aa1
PRG 1 Canal de video 1
PRG 2 Directo1
PRG 3 Directo sala 2
MORE
```

```
En el paquete 2
SSER 5000 2001:720:1d10:fff7::123:aa1
PRG 4 Camara de atras
PRG 6 Camara de la sala 8
END
```

## Petición de un programa

Un cliente que haya escuchado un anuncio de un servidor puede realizar una petición para recibir un programa de los anunciados. El protocolo para recibir un streaming es.

- El cliente realiza una conexión TCP a la dirección del servidor que hizo el anuncio y al puerto indicado. Si un servidor no ha anunciado una dirección IP, se entiende que acepta conexiones en la dirección IP desde la que envía el anuncio.
- El servidor es responsable por tanto de mantener un servidor TCP escuchando en el puerto indicado antes de empezar a hacer anuncios.
- Una vez abierta la conexión el servidor se queda a la espera y responderá a los envíos del cliente. Este dialogo sera siempre por lineas de texto. El envio de una linea de texto pasa el turno al otro lado

El cliente puede realizar las siguientes operaciones

- Para pedir un envío de un video el cliente en primer lugar preparara el script de recepción en un puerto indicado y a continuación pedirá el canal al servidor enviando la linea de texto

REQ id puertocliente [direccioncliente]

donde id es el identificador del canal visto en los anuncios y puertocliente el puerto en el que esta escuchando. Si especifica la direccion IP el servidor utilizara esa dirección o en caso contrario utilizara el origen de esta conexión. El servidor contestara a esta petición con una de estas lineas

REQ FAIL mensaje

Si no va a servir la peticion indicando el mens que quiere mostrar (Tipo: Canal inexistente, No el streamer o lo que sea)

REQ OK

Si ha iniciado correctamente el envio del strea

Una vez iniciado un stream el cliente y servidor mantendrán la conexión TCP abierta. El cierre de la conexión por el cliente se considera que implica la petición al servidor de que pare el stream asociado

## Direcciones y puertos

---

Parametro	Valor definicion
MDIR	ff05::1234:5678
MPORT	12345

El anterior protocolo se utilizará para la comunicación entre programas de diferentes grupos por lo que no puede cambiarse unilateralmente por ningún grupo

## Especificación del servidor

El programa servidor debe cumplir las siguiente especificaciones

Uso: `s_server -f conffile -p puerto -m mdir -o`

Lanza un servidor que realiza anuncios multicas indicados en el fichero de configuracion el servidor escucha en el puerto indicado en -p el servidor envia los anuncios a la direccion e y al puerto especificado en -o

El fichero de configuración tendra una linea por cada programa a servir y anunciar. Se sugiere el siguiente formato

```
1: id scriptenvio titulo
2: id scriptenvio titulo
...
N: id scriptenvio titulo
```

Por ejemplo este sería un ejemplo válido

```
1 /home/mikel/sserver/envia_canal1.bash Video d
5 /home/mikel/sserver/envia_canal2.bash Camara
6 /home/mikel/sserver/envia_escritorio.bash Str
```

## Especificación del cliente

El programa cliente debe cumplir las siguientes especificaciones

Uso:

`s_client -s scriptreproducir -m mdir -o mpu`

Lanza un cliente que escucha anuncios de progra en el grupo y puerto multicast indicados. Periodicamente muestra todos los canales que co un programa de entre todos los vistos en todos su reproduccion

## Scripts para streaming

Se puede considerar que se tendrán scripts para enviar y reproducir streaming. Los scripts de envío permitirán el uso con la siguiente especificación

Uso: `nombrescript direcciondestino puertodestin`

El script iniciará un envío en streaming hacia y el puerto indicados

El proceso que lance el script es responsable de el streaming y de detectar su finalización

Los scripts de recepción permitirán el uso con el siguiente formato

Uso: `nombrescript puerto`

El script escuchará un canal de streaming al puerto una ventana para mostrar el contenido. El proceso puede detenerlo si quiere interrumpir la reproducción finalización

Última modificación: martes, 23 de febrero de 2016, 19:05