

Tarea: aplicaciones multidifusión

Redes multimedia

Alberto Mateos Checa

I. Telecomunicación

1. Introducción.-

Vamos a tratar de analizar el tráfico multicast para comprender cómo funciona éste. Para ello vamos a hacer uso del software Wireshark y de dos aplicaciones que permitirán el tráfico multicast: VLC y un programa en java realizado por mi.

De esta forma, vamos a dividir la tarea en dos partes. En la primera trataremos de abrir un flujo de tráfico multicast correspondiente a la radio BBC radio 1(Reino Unido). En la segunda se analizará lo que ocurre con el uso de los multicastsockets de java.

2. Uso de aplicación en multidifusión.-

Como ya se ha comentado en la introducción, en este caso vamos a abrir el flujo multicast mediante el programa VLC. Para ello disponemos de un archivo **.sdp**, que es un formato estándar IETF para describir flujos de audio y/o vídeo RTP. Dicho archivo ha sido descargado de la web oficial de BBC (<http://support.bbc.co.uk/multicast/streams.html>).

Tras abrir el archivo con VLC, en mi caso, he podido comprobar que no se recibe ni audio ni video alguno. Esto puede ser debido a que las emisiones multicast de BBC son de prueba. No obstante, he probado otras emisiones multicast, como las que ofrecen la web www.multicast.org.uk para test (<http://www.multicast.org.uk/teststreams/>) y las de Absolut Radio UK (<http://www.multicast.org.uk/absoluteradio/>), cuyo servicio es proveído por la universidad de Southampton, y siempre se han obtenido los mismos resultados. Además, estas pruebas han sido realizadas mediante conexión a partir de la red cviugr-2, de una ADSL de Orange y de una conexión HSPA de Vodafone.

A continuación se muestra una captura de pantalla de wireshark con el tráfico capturado durante las pruebas realizadas (multicast de BBC):

No..	Time	Source	Destination	Protocol	Info
1	0.00000	192.168.0.192	224.0.0.22	IGMP	V3 Membership Report / Join group 233.122.227.171 for any sources
2	9.64014	192.168.0.192	224.0.0.22	IGMP	V3 Membership Report / Join group 233.122.227.171 for any sources
3	21.0720	192.168.0.192	192.168.0.255	BROWSER	Local Master Announcement EMPUBUNTU, Workstation, Server, Print Que
4	21.0721	192.168.0.192	192.168.0.255	BROWSER	Domain/Workgroup Announcement WORKGROUP, NT Workstation, Domain Enu
5	23.7201	192.168.0.192	224.0.0.22	IGMP	V3 Membership Report / Leave group 233.122.227.171
6	25.0041	192.168.0.192	224.0.0.22	IGMP	V3 Membership Report / Leave group 233.122.227.171

Como podemos comprobar, no hay tráfico alguno de tipo UDP por lo que efectivamente es imposible que podamos percibir sonido ni vídeo en VLC. Además, también vemos que se envían dos mensajes IGMP para unirse al grupo multicast (233.122.227.171) y otros dos similares para dejar el grupo. En todas las pruebas realizadas se ha detectado que se envían dos mensajes de unión de grupo, incluso si se realizan con otros programas que no sean VLC. Lo más probable es que sea debido a que, al no obtenerse respuesta alguna al primer mensaje, se haga un reenvío de éste posteriormente, ya que hay una diferencia de casi 10 segundos en el envío de cada uno.

Por otro lado, vemos que todos los mensajes IGMP se envían desde la dirección IP 192.168.0.192 (dirección del equipo sobre el que se realizan las

pruebas) hacia la dirección 224.0.0.22. Esta dirección de tipo multicast es una dirección reservada para IGMP v3.

Si hacemos doble click sobre uno de los paquetes **join group**, podemos acceder a su contenido:

```

▼ Internet Protocol, Src: 192.168.0.192 (192.168.0.192), Dst: 224.0.0.22 (224.0.0.22)
  Version: 4
  Header length: 24 bytes
  ▸ Differentiated Services Field: 0xc0 (DSCP 0x30: Class Selector 6; ECN: 0x00)
  Total Length: 40
  Identification: 0x0000 (0)
  ▸ Flags: 0x02 (Don't Fragment)
  Fragment offset: 0
  Time to live: 1
  Protocol: IGMP (0x02)
  ▸ Header checksum: 0x4291 [correct]
  Source: 192.168.0.192 (192.168.0.192)
  Destination: 224.0.0.22 (224.0.0.22)
  ▸ Options: (4 bytes)
▼ Internet Group Management Protocol
  IGMP Version: 3
  Type: Membership Report (0x22)
  Header checksum: 0x0cd8 [correct]
  Num Group Records: 1
  ▼ Group Record : 233.122.227.171 Change To Exclude Mode
    Record Type: Change To Exclude Mode (4)
    Aux Data Len: 0
    Num Src: 0
    Multicast Address: 233.122.227.171 (233.122.227.171)

```

Como podemos ver, el mensaje que se envía es de la forma IGMP versión 3 y de tipo **Membership Report**. La versión 3 de IGMP permite que las aplicaciones envíen notificaciones explícitas a las fuentes de las que quieren recibir tráfico. En este caso, el mensaje pide un cambio a **modo de exclusión** (Exclude Mode). En este modo, la recepción de paquetes enviados a la dirección multicast es pedida desde todas las direcciones fuente IP excepto por aquellas que se encuentren en la lista de fuentes.

Además, vemos que el TTL está fijado a 1. Esta es la situación habitual para paquetes de este tipo, por lo que, en mi caso, como máximo la petición llegará hasta el router frontera de la red del ISP.

4 | Tarea: aplicaciones multidifusión

De la misma forma, podemos abrir uno de los paquetes **leave group** para ver su contenido:

```
· Internet Protocol, Src: 192.168.0.192 (192.168.0.192), Dst: 224.0.0.22 (224.0.0.22)
  Version: 4
  Header length: 24 bytes
  ▸ Differentiated Services Field: 0xc0 (DSCP 0x30: Class Selector 6; ECN: 0x00)
  Total Length: 40
  Identification: 0x0000 (0)
  ▸ Flags: 0x02 (Don't Fragment)
  Fragment offset: 0
  Time to live: 1
  Protocol: IGMP (0x02)
  ▸ Header checksum: 0x4291 [correct]
  Source: 192.168.0.192 (192.168.0.192)
  Destination: 224.0.0.22 (224.0.0.22)
  ▸ Options: (4 bytes)
· Internet Group Management Protocol
  IGMP Version: 3
  Type: Membership Report (0x22)
  Header checksum: 0x0dd8 [correct]
  Num Group Records: 1
  ▾ Group Record : 233.122.227.171 Change To Include Mode
    Record Type: Change To Include Mode (3)
    Aux Data Len: 0
    Num Src: 0
    Multicast Address: 233.122.227.171 (233.122.227.171)
```

Si nos fijamos, el contenido de dicho paquete es exactamente el mismo que el del paquete que se usa para unirse al grupo multicast con la única excepción de que la petición que se realiza es la de **cambiar al modo de inclusión**. En este modo, la recepción de paquetes enviados a una dirección multicast específica pedida sólo por aquellas direcciones fuente IP que estén incluidas dentro de la lista de fuentes.

3. Programación de sockets multidifusión.-

En este apartado nos vamos a centrar en la realización de un programa que haga uso de la multidifusión mediante sockets. Para ello vamos a utilizar el lenguaje de programación Java debido a que, a parte de que es el que mejor conozco, dicha tarea es bastante simple.

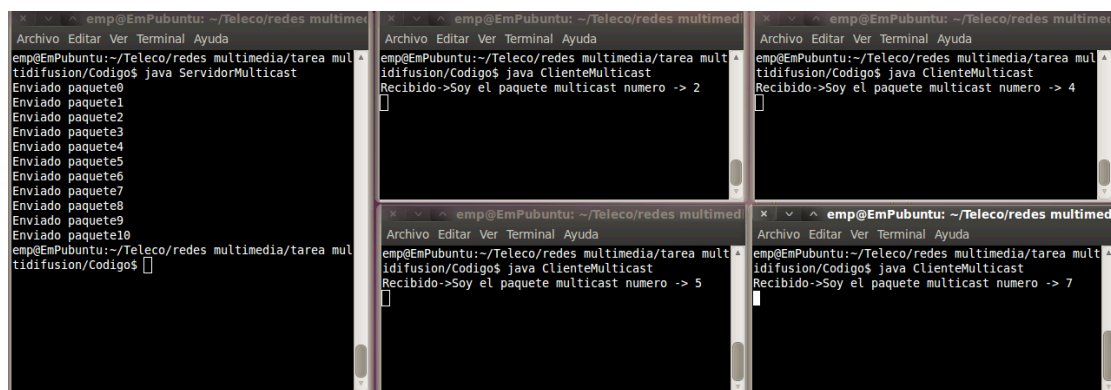
La programación de sockets multidifusión en Java es idéntica a la programación de un socket “normal”, con la salvedad de que hay que crear un socket de tipo **MulticastSocket** y la dirección IP a utilizar debe ser una dirección multicast (las comprendidas entre 224.0.0.0 y 239.255.255.255).

El código de los programas realizados no se muestra aquí ya que dichos programas serán adjuntados a este documento.

3.1. Funcionamiento del programa realizado.-

Para esta tarea se han realizado dos programas: **ServidorMulticast** y **ClienteMulticast**. El programa **ServidorMulticast** envía un mensaje de texto cada segundo, hasta un total de diez, asignando un número a cada uno de ellos para poder identificarlos posteriormente, a la dirección multicast **230.0.0.1**. El segundo de los programas, el **ClienteMulticast** permite que el ordenador donde se ejecuta se una al grupo multicast. Una vez que esté unido al grupo, tras recibir un mensaje multicast, deja el grupo y se queda en espera infinita, es decir, ejecutándose pero sin realizar acción alguna.

De esta forma, lo que he hecho para ver su correcto funcionamiento es lanzar el servidor y, posteriormente, varios clientes en diferentes instantes de tiempo. La siguiente captura de pantalla muestra este proceso:



Como vemos en la imagen anterior, los programas cliente (ventanas de consola pequeñas) sólo han recibido un paquete de los enviados por el servidor. Estos paquetes son los que tienen los números 2, 4, 5 y 7. Además, vemos que dichos programas siguen ejecutándose. Por tanto, se comprueba que los programas cliente únicamente reciben los mensajes dirigidos al grupo multicast cuando están unidos al grupo.

Además, en versiones previas del programa cliente se comprobó que si no se salía del grupo y se ejecutaban varios clientes a la vez, se recepcionaba la misma información en cada uno de ellos. No obstante, puesto que ese comportamiento parece obvio, se prefirió dejar el código como actualmente se

encuentra ya que creo que es más útil el ver el comportamiento de la unión y desunión a grupos multicast.

3.2. Análisis del tráfico con Wireshark.-

A continuación se muestra una captura del tráfico generado por los programas ejecutados en la prueba del funcionamiento de los sockets multicast:

No..	Time	Source	Destination	Protocol	Info
1	0.00000	192.168.0.192	230.0.0.1	UDP	Source port: 45030 Destination port: upnotifyp
2	1.00104	192.168.0.192	230.0.0.1	UDP	Source port: 52277 Destination port: upnotifyp
3	1.50243	192.168.0.192	224.0.0.22	IGMP	V3 Membership Report / Join group 230.0.0.1 for any sources
4	2.00207	192.168.0.192	230.0.0.1	UDP	Source port: 55166 Destination port: upnotifyp
5	2.00327	192.168.0.192	230.0.0.1	UDP	Source port: 55166 Destination port: upnotifyp
6	2.01032	192.168.0.192	224.0.0.22	IGMP	V3 Membership Report / Leave group 230.0.0.1
7	3.00266	192.168.0.192	230.0.0.1	UDP	Source port: 35060 Destination port: upnotifyp
8	3.27332	192.168.0.192	224.0.0.22	IGMP	V3 Membership Report / Join group 230.0.0.1 for any sources
9	4.00316	192.168.0.192	230.0.0.1	UDP	Source port: 44069 Destination port: upnotifyp
10	4.00417	192.168.0.192	230.0.0.1	UDP	Source port: 44069 Destination port: upnotifyp
11	4.01330	192.168.0.192	224.0.0.22	IGMP	V3 Membership Report / Leave group 230.0.0.1
12	4.94131	192.168.0.192	224.0.0.22	IGMP	V3 Membership Report / Join group 230.0.0.1 for any sources
13	5.00371	192.168.0.192	230.0.0.1	UDP	Source port: 54464 Destination port: upnotifyp
14	5.00470	192.168.0.192	230.0.0.1	UDP	Source port: 54464 Destination port: upnotifyp
15	5.01330	192.168.0.192	224.0.0.22	IGMP	V3 Membership Report / Leave group 230.0.0.1
16	6.00429	192.168.0.192	230.0.0.1	UDP	Source port: 40160 Destination port: upnotifyp
17	6.63031	192.168.0.192	224.0.0.22	IGMP	V3 Membership Report / Join group 230.0.0.1 for any sources
18	7.00480	192.168.0.192	230.0.0.1	UDP	Source port: 55400 Destination port: upnotifyp
19	7.00600	192.168.0.192	230.0.0.1	UDP	Source port: 55400 Destination port: upnotifyp
20	7.01030	192.168.0.192	224.0.0.22	IGMP	V3 Membership Report / Leave group 230.0.0.1
21	8.00525	192.168.0.192	230.0.0.1	UDP	Source port: 48878 Destination port: upnotifyp
22	9.00571	192.168.0.192	230.0.0.1	UDP	Source port: 35305 Destination port: upnotifyp
23	10.0061	192.168.0.192	230.0.0.1	UDP	Source port: 35913 Destination port: upnotifyp
24	12.2983	192.168.0.192	224.0.0.22	IGMP	V3 Membership Report / Leave group 230.0.0.1

Tal y como se aprecia, el comportamiento es el esperado tras ver la captura anterior en la que se mostraba el envío y recepción de mensajes en los clientes. Hay 4 mensajes de unión a grupo y otros tantos para dejar el grupo ya que se ejecutaron 4 clientes en la prueba realizada. Además, vemos que, a diferencia de lo que ocurría en el apartado 1 (uso de aplicación multicast), ahora si que se envían mensajes UDP desde la dirección del servidor a la dirección multicast. En este caso el funcionamiento si que es correcto ya que la ejecución del servidor y de los clientes se realiza en la misma máquina.

Además, también cabe destacar que de nuevo los mensajes IGMP se envían a la dirección 224.0.0.22 que como anteriormente se dijo es la dirección multicast reservada para tal fin.

A continuación vamos a abrir un **paquete UDP** para ver su contenido:

▶ Frame 1 (80 bytes on wire, 80 bytes captured)	
▶ Linux cooked capture	
▶ Internet Protocol, Src: 192.168.0.192 (192.168.0.192), Dst: 230.0.0.1 (230.0.0.1)	
▼ User Datagram Protocol, Src Port: 45030 (45030), Dst Port: upnotifyp (4445)	
Source port: 45030 (45030)	
Destination port: upnotifyp (4445)	
Length: 44	
▶ Checksum: 0x604a [validation disabled]	
▶ Data (36 bytes)	
0000	00 04 00 00 01 00 06 00 1b 63 02 43 cf 00 00 08 00 C.C....
0010	45 00 00 00 40 00 00 40 00 01 11 d2 43 c0 a8 00 c0 E..@..@...C....
0020	e6 00 00 01 af e6 11 5d 00 2c 60 4a 53 6f 79 20] ..JSoy
0030	65 6c 20 70 61 71 75 65 74 65 20 6d 75 6c 74 69 el paque te multi
0040	63 61 73 74 20 6e 75 6d 65 72 6f 20 2d 3e 20 30 cast num ero -> 0

Como podemos comprobar en la captura anterior, el programa utiliza tanto la dirección multicast de envío como el puerto que se especifican. Además, vemos que el contenido del paquete es el correcto.

Por último vamos a abrir un **paquete IGMP join group** para ver su contenido:

```

> Frame 8 (56 bytes on wire, 56 bytes captured)
> Linux cooked capture
  ▾ Internet Protocol, Src: 192.168.0.192 (192.168.0.192), Dst: 224.0.0.22 (224.0.0.22)
    Version: 4
    Header length: 24 bytes
    ▾ Differentiated Services Field: 0xc0 (DSCP 0x30: Class Selector 6; ECN: 0x00)
      1100 00.. = Differentiated Services Codepoint: Class Selector 6 (0x30)
      .... ..0. = ECN-Capable Transport (ECT): 0
      .... ...0 = ECN-CE: 0
    Total Length: 40
    Identification: 0x0000 (0)
    > Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 1
    Protocol: IGMP (0x02)
    > Header checksum: 0x4291 [correct]
    Source: 192.168.0.192 (192.168.0.192)
    Destination: 224.0.0.22 (224.0.0.22)
    > Options: (4 bytes)
  ▾ Internet Group Management Protocol
    IGMP Version: 3
    Type: Membership Report (0x22)
    Header checksum: 0xf3fc [correct]
    Num Group Records: 1
    ▾ Group Record : 230.0.0.1 Change To Exclude Mode
      Record Type: Change To Exclude Mode (4)
      Aux Data Len: 0
      Num Src: 0
      Multicast Address: 230.0.0.1 (230.0.0.1)

```

Como se ve en la captura anterior, el paquete es similar al que se obtuvo en el primer ejemplo (estudio de multicast con VLC) con la salvedad de que la dirección multicast es diferente.

Para el caso del mensaje IGMP para dejar el grupo ocurre lo mismo, por lo que no se muestra en este informe.

4. Conclusiones.-

Tras el estudio realizado hemos comprobado que la multidifusión funciona correctamente y que es realmente útil. No obstante, hemos visto también que por lo general no suele funcionar ya que los routers de los ISP u otros que estén entre el/los emisor/es y el/los receptor/es no suelen tener habilitadas sus capacidades multicast.

Además, hemos comprobado que, efectivamente, para poder recibir tráfico multicast hay que estar suscrito a un grupo.

5. Referencias.-

[1] RFC 3376 (<http://www.rfc-editor.org/rfc/rfc3376.txt>)

[2]
<http://download.oracle.com/javase/tutorial/networking/datagrams/broadcasting.html>

[3] <http://www.multicast.org.uk/>