# APLICACIONES MULTICAS

Alejandro Hernández de la Iglesia 70900084-P

Índice

Cliente 3

Servidor 4

Tráfico generado 6

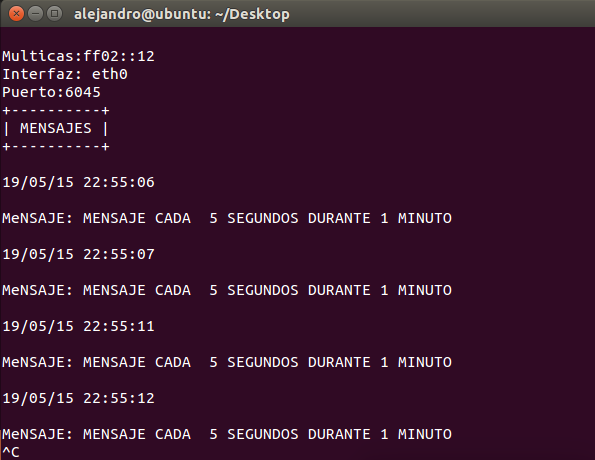
Nogal 7

## Cliente

En el programa Cliente, se comienza creando el Socket en UDP, a continuación se configura para que se una al grupo multicas en IPv6. Después hay que hacer el Bind para asignar al socket creado la dirección, el puerto y la interfaz a la que conectarse y nos unimos al grupo. Una vez hecho eso, se debe mantener conectado leyendo lo que el socket mande, esto se consigue con un bucle infinito. Para podernos salir de forma adecuada del cliente, es necesario interceptar la señal SIGINT que se manda al hacer Ctrl+C, cuando se pulsa esa unión de teclas, se manda a la manejadora “leave” que finaliza todo de forma correcta.



Configuramos el cliente con otros parámetros para comprobar que también funciona correctamente.



## Servidor

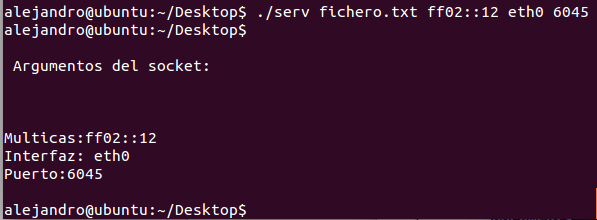
Este servidor, está hecho de forma Daemond para liberar la terminal. Al liberar la terminal, no tenemos conciencia de si ocurren errores o lo que está sucediendo, por lo que hacemos que escriba todo lo que pueda pasar, en Syslog.

El servidor funciona con argumentos preestablecidos o se pueden seleccionar introduciéndolos al arrancarlo, fichero, Direccion multicast, interfaz y puerto.

Una vez pasado este tramo, se crea el socket en UDP con los parámetros establecidos, se comienza leyendo los mensajes desde el fichero introducido, creando, por cada línea del fichero, un hilo (con un máximo de 20 líneas, máximo de hilos), se separa el tiempo de repetición, la duración y el mensaje en cuestión, que se introducen en una estructura que se pasa a la manejadora que crea los hilos llamada “createThread”, en ella, se manda el mensaje el tiempo y las repeticiones establecidas en los parámetros (ojo, se ha introducido semáforos para que vaya de forma ordenada y cada hilo espera a que termine el anterior para mandar su mensaje). Por cada hilo creado, debemos esperar a que acabe su ejecución, una vez acaben todos su ejecución, el servidor acabará de forma adecuada, si por el contrario, se desea acabar antes, al estar en Daemond, tendría que matarse el proceso, si se elimina la parte de Daemond, con hacer Ctrl+C, bastaría pues, se intercepta la señal SIGINT para que finalice de forma correcta.

## 

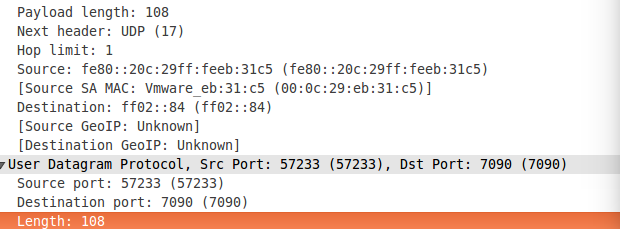
Configuramos el servidor con los mismos parámetros con los que configuramos el cliente anteriormente para que pueda mandar los mensajes al grupo multicast y puerto especificados.



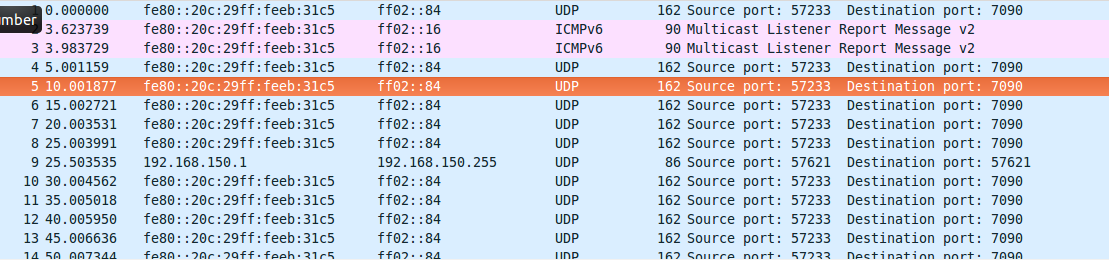
## Tráfico generado



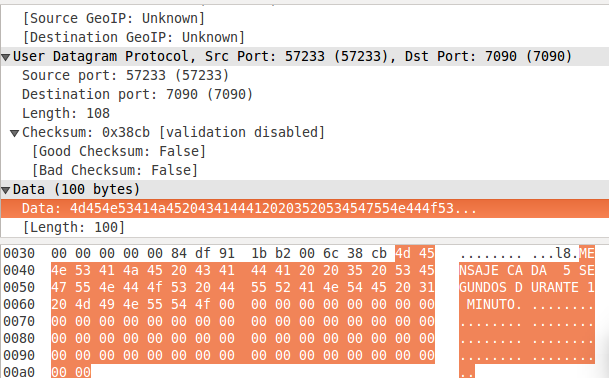
Podemos observar en la primera imagen, como se crean unos primeros mensajes para hacer un join al grupo multicas ff02::0084 por el puerto 7090. Este mensaje, es generado por el cliente cuando se arranca para poderse unir al grupo multicast deseado y que así le puedan llegar los mensajes que el servidor está tranamitiendo.



Podemos ver otro tipo de mensajes:



Son mensajes UDP mandados por el servidor a la dirección multicas ff02::84 y al puerto 7090, podemos ver también el contenido de los mensajes.

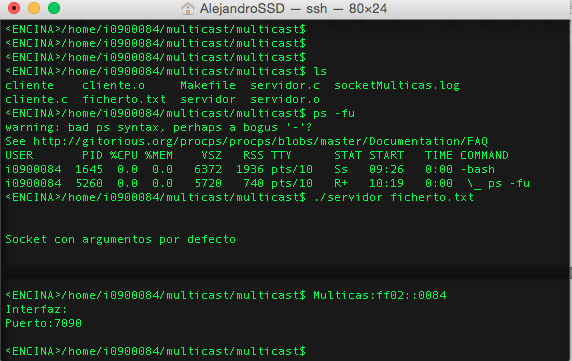


Observamos también dos mensajes mandados por el cliente en el momento en el que se cierra el programa, estos son mandados para salirse del grupo multicast.

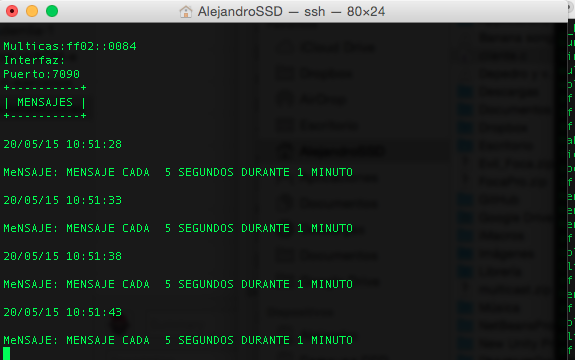


## Nogal

Debemos comprobar que también funciona en nogal, para ello, debemos subir nuestro servidor y cliente y ejecutarlo en el servidor. Como hemos hecho el servidor Daemon, solo tendremos que abrir una terminal para poder comprobar su funcionamiento.



Ejecutamos nuestro servidor en nogal y a continuación lanzamos el cliente.



Comprobamos como efectivamente, el funcionamiento es adecuado.