Aplicaciones Multicast IPv6

Versión 2.0, abril 2018

**Alumno (apellidos, nombre (DNI) :\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Alumno (apellidos, nombre (DNI) :\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Fecha:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Duración estimada de la práctica: 2 sesiones de 2h.**

* 1. Entorno de trabajo
* Software de emulación de redes: GNS3 (Analizador de red: wireshark)
* Cisco IOS
* Lunix virtualizado (DebianAlumno)
* Framework de desarrollo en C para Linux que el alumno desee
  1. Objetivos
* Utilizando el modelo cliente-servidor realizar una aplicación multicast basada en sockets IPv6
* Analizar el tráfico que la misma genera
  1. Escenario de trabajo

En la fase de desarrollo se puede trabajar en el framework de desarrollo en C en Linux que el alumno desee. Para la realización de las pruebas de funcionamiento y análisis de tráfico se trabajará sobre un escenario prediseñado en GNS3. Descomprimir el fichero en el directorio GNS3/projects de la unidad Z. Se generará una carpeta llamada *06MulticastOSPFPIMIPv6* con los archivos del escenario. Abrirlo con GNS3 y se mostrará lo siquiente.

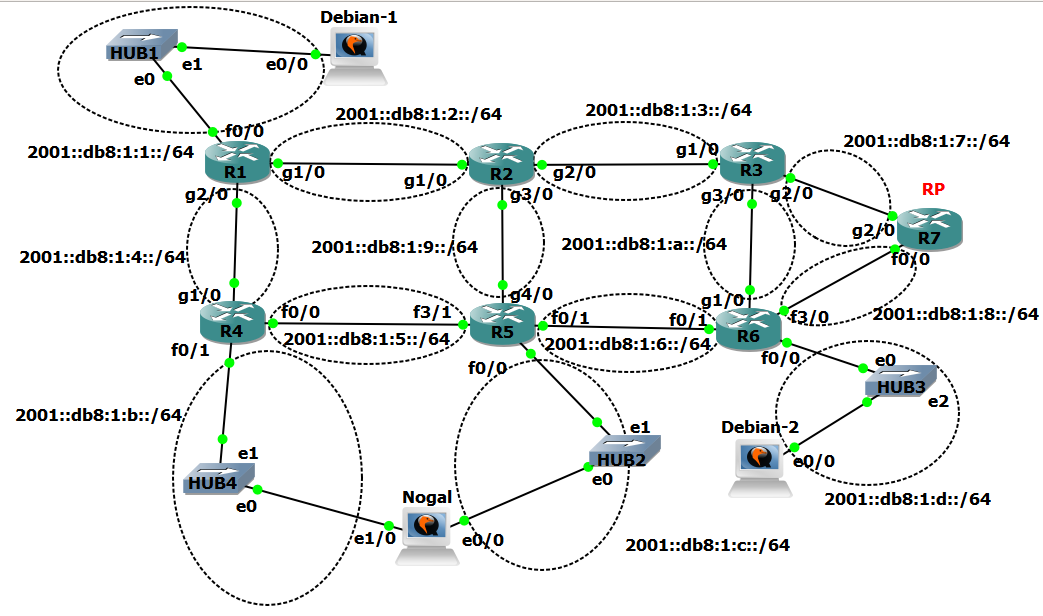


Figura 1: Escenario *multicast* IPv6.

* 1. Desarrollo de la aplicación

Utilizando el modelo cliente-servidor realizar una aplicación en red basada en sockets multicast para IPv6 con las siguientes especificaciones:

**Programa Difusor o Fuente**

Difundirá un mensaje que leerá por parámetros hasta que finalice ordenadamente capturando CTRL+C.

Se ajustará a la siguiente sintáxis:

difusor mensaje\_a\_difundir IPv6multicast interfaz puerto saltos intervalo

Si no se especifica ninguno de estos parámetros se difundirá un mensaje por defecto a una IPv6 multicast por defecto, por un interfaz por defecto, un puerto por defecto, un número de saltos y el intervalo de emisión de los mensajes.

Ejemplo:

difusor "Hola que tal" ff15::33 eth0 4343 10 15

**Programa Suscriptor o Receptor**

Se unirá al grupo multicast y leerá el mensaje que difunda el cliente mostrándo por pantalla junto con la IPv6 de la fuente. El cliente finalizará ordenadamente cuando se pulse Ctrl+C.

Se ajustará a la siguiente sintáxis:

suscriptor IPv6multicast interfaz puerto

Si no se especifica ninguno de estos parámetros la suscripción se realizará a una IPv6 multicast por defecto, por un interfaz por defecto y un puerto por defecto.

Ejemplo:

suscriptor ff15::33 eth0 4343

Las primeras líneas de todos los ficheros fuente contendrán unas líneas de comentarios donde se especifique el nombre del fichero fuente y el nombre completo, DNI y usuario con el que se ha subido la práctica. Ejemplo:

/\*

\*\* Fichero: multicast.c

\*\* Autores:

\*\* Antonio Pérez Puerto DNI 70876543M

\*\* Araceli Sánchez Álvarez DNI 06876701A

\*\* Usuario: i6876701

\*/

Responde a las siguientes preguntas:

1. ¿A qué dirección (IP+puerto) hace bind el receptor o suscriptor? ¿Por qué?

RESPUESTA: A la dirección (in6addr\_any, puerto especificado).

1. ¿A qué dirección (IP+puerto) hace bind la fuente o emisor? ¿Por qué?

RESPUESTA: A la dirección (in6\_addr\_any, puerto efímero).

1. ¿A qué dirección (IP+puerto) envía el emisor? ¿En que función del API de sockets se especifica? ¿Dónde se especifica el número de saltos? ¿Para qué sirve?

RESPUESTA: Envía la información a la dirección determinada por la dirección IP del grupo multicast y el puerto seleccionado por línea de órdenes.

Esta información se especifica con la función inet\_pton(), que convierte una representación textual de una dirección de red en una estructura de dirección de red de tipo struct in6\_addr.

El número de saltos se establece como cualquier otro parámetro del socket mediante la función setsockopt(), utilizando la opción IPV6\_MULTICAST\_HOPS. Esta función permite manipular las opciones asociadas a un socket en diferentes niveles de la pila de protocolos.

1. ¿Quién debe especificar el deseo de recibir datagramas de una determinada difusión multicast, la fuente o el receptor? ¿En qué función del API de sockets se especifica?

RESPUESTA: el receptor. Mediante la función setsockopt(), se puede especificar a nivel de red el deseo de unirse a un grupo multicast mediante la opción IPV6\_ADD\_MEMBERSHIP.

1. ¿Es cierto o falso que la fuente ha de suscribirse al grupo multicast?

RESPUESTA: falso. La fuente no tiene que suscribirse porque es ella la que emite los mensajes, y por tanto no es necesario llevar a cabo la configuración (árbol de expasión, etc.) para hacer llegar los mensajes que hay que establecer para los suscriptores.

1. Cuando un equipo tiene más de una interfaz de red si desea ser emisor de un grupo multicast, ¿en qué función del API de sockets debe especificar la interfaz por la que desea enviar? ¿Y si desea ser el receptor?

RESPUESTA: en setsockopt() con la opción IPV6\_MULTICAST\_IF. También en setsockopt(), como se señaló en la pregunta d).

1. ¿Qué sucede internamente en un equipo que se suscribe a una dirección multicast? ¿Y cuando lo abandona? Especifica la IP multicast que has utilizado en tu práctica y la MAC correspondiente.

RESPUESTA: nada. La información sobre los grupos multicast es gestionada por la red.

La dirección multicast es ff15::33. La dirección MAC se deriva de la dirección IPv6 configurada para la interfaz del equipo en que se ejecuta. Si la dirección fuera 2001:543:5::3456, la dirección IP multicast desde la que recibiría paquetes sería la ff15::3456, y la MAC correspondiente ?

1. ¿Se manda algún mensaje por la red cuando un equipo se suscribe a un grupo multicast? ¿Y cuando lo abandona? ¿Cúal es su propósito?

RESPUESTA: para unirse se mandan mensajes MLD. Para abandonarlo no se manda ningún mensaje.

1. ¿Cómo sabe el receptor desde donde le llega la difusión? ¿Con qué función del API de sockets la puede obtener?

RESPUESTA: mediante recvfrom() el receptor obtiene la dirección IP del emisor.

* 1. Probando la aplicación y analizando el tráfico multicast generado

En este apartado se detallan las pruebas de funcionamiento y el análisis del tráfico generado en la difusión que han de realizarse. En el análisis del tráfico hay que prestar atención a los mensajes ICMPv6 que se generan en la suscripción, así como los mensajes de la aplicación viajando desde la fuente a los suscriptores. Para ello realiza los siguientes pasos:

* 1. Una fuente en Nogal difundiendo por su interfaz eth1 y un suscriptor en Debian-1
  2. Una fuente en Nogal difundiendo por su interfaz eth1 y dos suscriptores uno en Debian-1 y otro en Debian-2

En todos los casos mostrar con capturas de pantalla que la aplicación funciona correctamente de forma que se vea como se han invocado la fuente y los suscriptores y la salida generada por ellos.

De igual forma, y para todos los casos, observar con las capturas de tráfico los mensajes de suscripción y abandono del grupo multicast y por qué ramas viajan los datagramas de nuestra aplicación. Incluye en el informe la explicación de estos mensajes y adjunta los ficheros de las capturas de tráfico realizadas (solo con las tramas que nos interesan filtrando adecuadamente desde wireshark). Consulta el estado de las tablas de rutas multicast (*show ipv6 mroute*) y la lista de suscriptores (*show ipv6 mld groups*) para apoyar tu explicación.

RESPUESTA

**Utilería:**

Fichero Makefile para compilar en los Debian:

CC = gcc

CFLAGS =

LIBS =

PROGS = emisor suscriptor

all: ${PROGS}

emisor: emisor.o

${CC} ${CFLAGS} -o $@ emisor.o ${LIBS}

suscriptor: suscriptor.o

${CC} ${CFLAGS} -o $@ suscriptor.o ${LIBS}

clean:

rm \*.o ${PROGS}

Para copiar ficheros entre los equipos del escenario de GNS se puede utilizar scp de esta forma:

scp fichero root@nombre\_equipo\_destino:/directorio\_destino

Ejemplo para copiar el fichero pp a nogal por su interfaz eth1 con R4 (IPv6 = 2001:db8:1:b:29a:61ff:fe58:4201)

scp pp root@nogalR4:/root

Para referinos por nombre a los equipos Debian del escenario se han dado de alta los nombres y sus IPsV6 en el fichero /etc/hosts de estos equipos. Su contenido es:

2001:db8:1:c:29a:61ff:fe58:4200 NogalR5

2001:db8:1:b:29a:61ff:fe58:4201 NogalR4

2001:db8:1:1:29a:61ff:fe02:6f00 Debian-1

2001:db8:1:d:29a:61ff:fe74:ae00 Debian-2