



# Estudio de los mensajes de control del protocolo multicast PIM SM

Memòria del projecte  
d'Enginyeria Tècnica en  
Informàtica de Gestió  
realitzat per  
Marcos Piedra Osuna  
i dirigit per  
Daniel Franco

**Escola d'Enginyeria**  
Sabadell, juny de 2010

El sotasignat, Daniel Franco,  
Professor de l'Escola d'Enginyeria de la UAB,

**CERTIFICA:**

Que el treball al que correspon la present memòria  
ha estat realitzat sota la seva direcció  
per en Marcos Piedra Osuna

I per a que consti firma la present.  
Sabadell, juny de 2010

---

Signat: Daniel Franco

## **Resumen.**

OPNET es una herramienta de simulación de redes muy potente, donde se puede configurar cualquier tipo de red existente, hardware o software relacionado. Y por supuesto, la configuración de todas las clases de protocolos que existen, tanto a nivel experimental como ya existentes. Esta herramienta nos ayuda a “probar” sin tener que montar ninguna red física, una gran ventaja sobre todo para protocolos experimentales o para el diseño de los mismos. En este proyecto se utiliza el OPNET para explotar el protocolo multicast PIM-SM. Se establecerá una modelo de red, donde estarán configurados los nodos de forma estratégica para intentar extraer el máximo de información del comportamiento en esta situación del protocolo PIM-SM.

Para realizar la simulación en OPNET, es necesario de un modelo de red. Este modelo se generará de forma aleatoria mediante un pequeño software realizado especialmente para esta tarea. Este software, mediante archivos comunes XML, se importarán desde OPNET, donde se lanzará la simulación mediante un tiempo determinado, y se recogerán los datos que nos interesan para realizar el proyecto.

Además OPNET, se puede modificar el código fuente. En este caso, se realizar unas modificaciones en las estadísticas de OPNET para conocer los datos que nos interesen. Nos interesa conocer determinadas reacciones del protocolo y el comportamiento antes diferentes situaciones planteadas.

<b>1. Introducción</b>	
1.1. Descripción del proyecto .....	6
1.2. Motivación .....	7
1.3. Objetivos .....	7
1.4. Especificaciones .....	7
1.5. Estado del arte .....	8
<b>2. Planificación y desarrollo del proyecto.....</b>	<b>9</b>
2.1. Objeto .....	9
2.1.1. Descripción de la situación .....	9
2.1.2. Objetivos.....	10
2.2. Sistema a realizar.....	11
2.2.1. Descripción .....	11
2.2.2. Recursos.....	12
2.2.3. Evaluación de riesgos .....	12
2.2.4. Análisis Costos — Beneficios .....	13
2.2.5. Organización y costos del proyecto.....	15
<b>3. Base teórica y protocolos a estudiar .....</b>	<b>16</b>
3.1. Multicast, un primer enfoque.....	16
3.2. Tipos de direccionamiento. ....	17
3.3. Conceptos básicos de protocolos multicast. ....	18
3.3.1. Clasificación de protocolos multicast. ....	20
3.3.2. Introducción a PIM.....	23
3.3.3. Direccionamiento IP Multicast. ....	25
3.4. Arboles de distribución.....	26
3.5. IGMP.....	30
3.6. PIM-SM.....	36
3.7. Funcionamiento IGMP y PIM-SM.....	47
<b>4. Herramienta de simulación y experimentación .....</b>	<b>52</b>
4.1. Modelos y simulación .....	52
4.2. Tipos de simulación.....	53

4.3. OPNET .....	54
4.3.1. Modelar en OPNET.....	56
4.3.2. Salidas del simulador y análisis de resultados.....	58
4.4. Configuración del protocolo PIM-SM. ....	60
4.5. Estadísticas personalizadas para la experimentación. ....	63
4.6. Metas de la experimentación .....	68
4.7. Modelo base .....	70
5. Desarrollo de la aplicación GenNode .....	72
5.1. Lectura de los archivos XML .....	72
5.2. Estructura del proyecto en Visual Studio .....	74
5.3. Clases .....	74
5.4. Interface gráfica de usuario .....	77
5.5. Importación / Exportación XML.....	80
5.6. Modelo base .....	80
5.7. Funcionalidades extra.....	80
6. Análisis de resultados de simulación.....	82
6.1. Muestra de datos.....	82
6.2. Resultados.....	85
6.3. Comparativa de resultados.....	103
6.4. Conclusión.....	105
7. Bibliografía .....	106

## **1. Introducción**

### **1.1 Descripción del proyecto.**

La transmisión de datos desde un origen a un destino, en una red IP, se puede realizar vía unicast, broadcast y multicast. El envío multicast requiere de un origen a múltiples destinos, en ese trayecto es donde interviene las diferentes estrategias de los protocolos multicast. Durante una transmisión multicast es necesario de datos de control que dirijan cual es el camino óptimo por donde deben de ir los datos. Este proyecto realiza un análisis comparativo del flujo de datos de control con el protocolo PIM-SM y de un modelo analítico a describir. En definitiva lo que se pretende estudiar es el “ruido” que realiza los datos de control en el flujo de datos, cuando estos protocolos gestionan el flujo de la red multicast.

A partir de un escenario, realizado en un simulador de redes y que están configurados dichos protocolos, se lanza durante un tiempo determinado datos que viajan desde un origen a varios destinos. El simulador se encarga de recoger y analizar los diferentes tipos de datos de control, así se realiza las estadísticas que más convenga para el estudio. Básicamente el estudio se realizará con los mensajes de control que intervengan en la simulación, por lo tanto las estadísticas que se utilizan estarán orientadas a los datos de control.

El protocolo multicast que se utiliza en la simulación es PIM-SM y un modelo analítico con el que compararemos las estadísticas y se llega a conclusiones. Este proyecto también analiza superficialmente los diferentes protocolos multicast que existen actualmente e introduce una base teórica de su funcionamiento. El simulador de redes utilizado es el OPNET, donde se creará un modelo de red y se configurarán dichos protocolos para el estudio. Una vez lanzado el simulador y estudiado las estadística que interesan, se realizará un comparación y se buscará el protocolo multicast mas óptimo y será el que menos utilice los datos de control durante la transferencia de datos del origen al destino.

Para realizar el modelo es necesario generarlo, ya sea de manera manual, es decir mediante el editor de OPNET o de forma dinámica, por ejemplo generando archivos compatibles con OPNET y a continuación abrirlas. Este proyecto, además de realizar un análisis de los protocolo multicast PIM-SM, tratará de generar modelos de redes y se exportaran mediante XML a OPNET. Todo esto mediante un pequeño software de manejo de XML llamado GenNode.

## *1.2 Motivación*

PIM-SM es un protocolo multicast muy eficaz para los receptores que estén de forma esparcida en la red, pero ¿Cómo reacciona el protocolo a las caídas de sus clientes?, y los mensajes de control ¿Colapsan la red?, ¿qué sucede si existen muchas caídas de los nodos centrales? Estas preguntas se pueden responder mediante un sistema de simulación como es el OPNET y un modelo al que atacar. De estas dos bases se nutrirá el proyecto. Por otro lado, aunque OPNET facilita la tarea de no tener que montar una red física, siempre existirá la tarea de configuración del protocolo y de los nodos de los cuales está configurado. Para facilitar la generación de nodos existe software de generación de nodos de forma aleatoria. Este es un software muy específico según el simulador o sin el simulador, es decir, te organiza los nodos de forma aleatoria.

El XML es un lenguaje muy utilizado en la actualidad para la comunicación entre diferentes software, pero con unos criterios comunes. OPNET facilita la incorporación de redes mediante XML, y nos aprovecharemos para realizar un pequeño software que sepa exportar al OPNET mediante el XML.

## *1.3 Objetivos.*

Los objetivos que se persiguen básicamente son dos. Realizar una comunicación con el OPNET mediante el XML, para generar redes de forma aleatoria y a continuación realizar un estudio de un protocolo multicast que es el ya denominado PIM-SM. Para ello se introducirá en un marco teórico de las redes multicast existente y se profundizará con el protocolo PIM-SM, se estudiará superficialmente el OPNET y la parte que nos interesa para el proyecto que son las estadísticas y su modificación. A continuación se describe como se realiza una configuración del protocolo PIM-SM en un modelo de red realizado en OPNET. Y por último se lanzarán una serie de experimentos para conocer el protocolo PIM-SM, y recoger los resultados para analizarlos y sacar conclusiones finales.

## *1.4 Especificaciones*

La realización del modelo se hace mediante el OPNET, este es esencial para el desarrollo del experimento, se trabajarán con elementos muy estándares (como son los routers y las estaciones de trabajo). Además será necesario añadir algunas estadísticas en OPNET para el estudio del

experimento a realizar. Por otra parte el desarrollo del software generador de nodos se realiza mediante VB.NET. La facilidad de uso, ayudará a realizar una software en un tiempo corto, y además es uso del XML es considerablemente fácil en este lenguaje. Esto conlleva a tener el Microsoft Framework, aunque la versión no es importante del todo ya que se utilizan espacios de nombres muy comunes. Y como en cualquier desarrollo de software, será necesario algún editor multilenguaje, como puede ser el muy útil y gratuito notepad++.

### *1.5 Estado del arte.*

El software realizado para generar la experimentación se podría categorizar como generador de nodos. En la actualizar es un software muy específico, por lo tanto encontrar tanto a nivel comercial como a nivel profesional puede resultar complicado, y en algunos casos dichos software da sensación de ser muy “caseros”. A continuación se describirán algunos de los generadores que podemos encontrar.

GT-ITM: Sus siglas pertenecen a Georgia Tech Internetwork Topology Models. El funcionamiento principal se basa en diferentes scripts de los que indican cuando realiza la generación las características que tiene el modelo a crear. Generará un grafo siguiendo parámetros probabilísticos e indicándole el número de nodos de generar. Podemos indicar al generador la distribución aleatoria de cada nivel jerárquico y sus parámetros. La salida es un fichero de texto, que contiene la información sobre la posición de los nodos y las aristas que forman el grafo, es decir el modelo de red a generar. Dicha herramienta tiene la posibilidad de exportar a un formato compatible con el simulador de redes NS-2.

WaxMax: Está preparado para utilizarlo en MatLab. Este generador de topologías es un modelo geográfico donde existe una expansión de una red. En este modelo los nodos de la red se distribuyen uniformemente en el plano y las aristas son añadidas según las probabilidades de las distancias entre

nodos. La probabilidad de distancia entre nodos viene dada por  $P(u, v) = ae^{-d/(bL)}$ . Donde,  $0 < a < b = 1$ ,  $d$  es la distancia entre  $u$  y  $v$  (los nodos) y  $L$  es la distancia máxima entre nodos. Un incremento en el parámetro  $a$  aumenta la probabilidad de que las aristas estén más alejadas, mientras que un aumento en  $b$  puede repercutir en las diferencias entre numero de aristas largas y numero de aristas cortas. El uso de WaxMax es mediante MatLab por lo tanto como se trate el resultado deberá de decidirlo el usuario.

## **2. Planificación y desarrollo del proyecto**

Para realizar la simulación y así poder sustraer datos, es necesario poder generar una red con las características según estableciendo unos parámetros de entrada. GenNode es un generador de nodos cliente, y lo realiza a partir de un modelo base. Este modelo base tiene una serie de características las cuales se tienen que cumplir para que el programa pueda generar los nodos en la red base. Si tuviéramos que realizar la generación de nodos de forma manual, es decir desde el diseñador de OPNET, podría resultar muy larga la edición del modelo y posicionar los nodos según una función de distribución resultaría bastante difícil (pero no imposible.). Y una vez realizado los modelos a partir de este modelo base se continuará con la simulación en OPNET y la obtención de los resultados para la experimentación.

El XML será lenguaje de comunicación con OPNET y GenNode por tanto la importación como la exportación de los modelos será en XML.

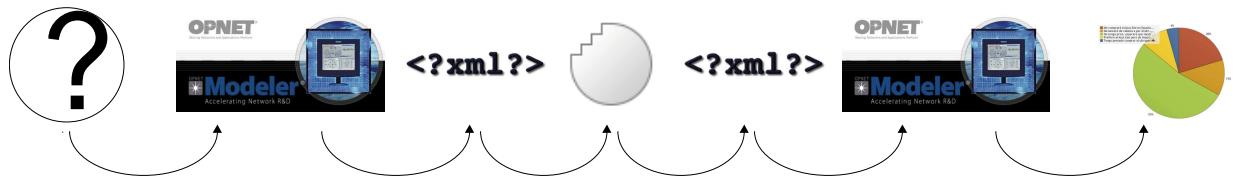
Una vez que el generador de redes sea funcional, será el momento de realizar la experimentación con OPNET. Todos los archivo que se han ido generando mediante GenNode se importarán a OPNET y se lanzarán las simulaciones correspondiente. Antes de establecer la simulación hay que preparar las estadísticas para que se recojan los datos que queremos para realizar el estudio. Una vez finalizada la simulación se continúa con la documentación.

### **2.1 Objeto**

#### *2.1.1 Descripción de la situación*

El requerimiento de GenNode vino a partir del planteamiento de los experimentos y las simulaciones a realizar. La realización de redes mediante OPNET es fácil y muy visual, pero si queremos realizar redundantemente tareas estableciendo una funciones de probabilidad, puede resultar un poco más complicado. Para eso GenNode puede facilitar un poco más la tarea de realización de los estos modelos. GenNode básicamente construye los modelos de red a partir de unos parámetros de entrada, que es tipo función de probabilidad de generación, para que a continuación se lance la simulación sobre estos y se puedan recoger los datos deseados. GenNode solo es un intermediario

entre la experimentación que se desee (orientado hacia la experimentación del este proyecto) y OPNET, como se puede observar en la siguiente figura.



Diferentes fases de forma simbólica

Se puede observar las diferentes fases que se realizarán en el experimento, y en el centro es donde interviene GenNode. En algunas fases intermedias, se simboliza con el XML que es el lenguaje con el cual se comunican las fases.

- La primera fase es plantear el problema. En nuestro caso es la experimentación a realizar.
- La segunda fase, es realizar el diseño base del modelo. Este diseño está configurado con el protocolo PIM-SM, routers, etc.
- La tercera fase es la importación a GenNode del modelo base. Con este modelo, definido de una determinada forma, se utilizará para generar más modelos según los parámetros establecidos.
- La cuarta fase es la importación del XML generado por GenNode desde OPNET. Y es donde se realiza la simulación de las estadísticas deseadas.
- La quinta fase, es la obtención de resultados y conclusiones finales.

### 2.1.2 Objetivos

El Objetivo final es conocer el comportamiento de los mensajes de control, del protocolo multicast PIM SM. Para realizar esta tarea se ha de generar varios escenarios los cuales se generarán mediante GenNode. Para generar los escenarios GenNode necesita unos parámetros de entrada, los cuales estarán relacionados con el tipo de simulación que se desea realizar. Estos parámetros son básicamente para generar los nodos clientes según una función de distribución elegida, donde mediante un interface de usuario se mostrarán para su posterior modificación.

También es necesario establecer un modelo base, es decir, cuales son las subredes, los nodos que intervienen, atributos, etc en el modelo que se quiere estudiar. Todo lo que se exporta desde OPNET, se podrá tratar desde GenNode para la exportación de nuevo a OPNET.

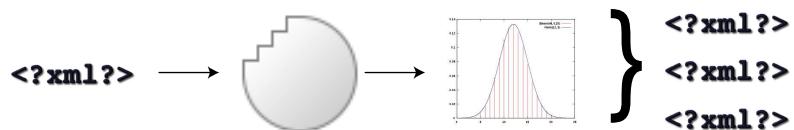
Además a medida que se ha ido experimentado mediante los modelos y sacando resultados, ha sido necesario de utilidades, donde están incluidas en GenNode y que se explicaran en el capítulo del desarrollo del proyecto.

## 2.2 Sistema a realizar

### 2.2.1 Descripción

La generación de nodos clientes es la tarea principal de GenNode, el cómo y el donde, se tendrán que especificar mediante los parámetros de entrada. Estos parámetros están adaptados a la experimentación multicast del protocolo PIM-SM, pero no es necesario realizar mucha modificación en el código fuente si se desea realizar otro tipo de experimentación con otro protocolo.

En la siguiente figura se puede observar un esquema del uso GenNode para la experimentación.



Uso de GenNode

El primer XML, representa el modelo base realizado en OPNET el cual se realiza una la lectura en GenNode. Este lo interpreta en forma de árbol, donde se puede navegar por todos los valores y atributos que contenga. Cuando se realiza de importación tiene que leer según los metatags que utilice OPNET para generar los archivos XML. Esta estructura se explicara con más detenimiento en el apartado de desarrollo de GenNode. En el momento que ya tiene la estructura del modelo base cargada, vendrá en momento de introducir los parámetros de entrada que deseemos. Estos parámetros están situados en un dialogo de opciones, donde se podrán modificar de forma visual. Los parámetros son básicamente funciones de probabilidad, con las cuales se van a generar los nodos mediante el modelo base. Una vez escogidos los parámetros de entrada, es la hora de realizar los modelos. En la figura se representan varios XML, ya que en nuestra experimentación será necesario generar varios modelos con diferentes parámetros. Estos modelos generados aun se tienen que importar en el OPNET y procesar su simulación.

### **2.2.2 Recursos**

El S.O. es recomendable un Windows XP SP2. El Framework 2.5 ya es suficiente para el uso de GenNode. El entorno de programación es Visual Studio 2005, y para la visualización y edición del XML con el notepad++ será suficiente. Además nos es necesario OPNET como software para la generación del modelo de red base, exportación XML, importación XML, simulación de los modelos y exportación de resultados.

Los datos se recogen mediante Excel, la documentación se ha generado mediante Word y la planificación del proyecto está realizada en Project 2007, así que el paquete Office 2007 es el mejor candidato para su uso en el proyecto.

Por la parte de Hardware, la maquina puede ser un Pentium IV 2800 MHz para el desarrollo y a partir de 512 Mb de RAM. GenNode no necesita mucho procesador, así que la maquina donde corra puede ser inferior a la del desarrollo (en cualquiera que funcione correctamente el Framework 2.5)

### **2.2.3 Evaluación de riesgos**

OPNET es un programa muy complejo y tiene muchas funcionalidades. Pero tiene la contra de que no es intuitivo aunque tenga un aspecto visual muy familiar. Antes de realizar cualquier tarea, hay que establecer una base de este software importante para poder determinar en qué grado se puede utilizar en el proyecto.

Otro problema que encontramos son metatags utilizado en OPNET para la importación/exportación en GenNode. No se ha encontrado ninguna documentación, así que se realizará de forma visual, es decir abriendo el archivo y examinando cada metatag. Esta forma puede resultar rápida, pero puede crear algunas dudas de los metatags. Este problema se encuentra tanto la lectura como la escritura, se tendrá que realizar una serie de pruebas antes de poder empezar a escribir código. Es decir, realizando una “simulación” de generación de nodos mediante un editor XML.

También existe el problema del tamaño del documento XML, ¿qué sucede si un documento es demasiado grande para la lectura? Hay muchas situaciones parecidas a estas de las cuales será necesario realizar ensayos para saber cómo funciona OPNET con el XML.

Si la exportación/importación en XML no resulta del todo factible, se puede estudiar otros tipos de generadores de redes. Pero por norma general, los generadores no exportan a OPNET, por lo cual se tendría que realizar mediante el diseño de OPNET los modelos a realizar.

#### 2.2.4 Análisis Costos — Beneficios

*Costos materiales:*

<b>Recurso</b>	<b>Costo</b>
<b>Servidor</b>	1700 €
<b>Licencia Visual Studio 2005</b>	2000 €
<b>Licencia OPNET</b>	2000 €
<b>Amortización PC programador</b>	276,25 €
<b>Amortización MS Project 2007</b>	20,83 €
<b>Amortización MS Office 2007</b>	29,16 €
<b>TOTAL</b>	<b>6026,40 €</b>

$$(2.325€+990,00/24)*2 = 276,25 \text{ €}$$

$$(250 \text{ €}/24)*2 = 20,83 \text{ €}$$

$$(350 \text{ €}/24)*2 = 29,16 \text{ €}$$

<b>Recurso</b>	
<b>Analista</b>	45 € /h
<b>Técnico programador</b>	30 € /h

*Costos de personal:*

<b>Tarea - GENNODE</b>	<b>Horas</b>	<b>Costo</b>	<b>Recurso</b>
Documentación de los metatags que utiliza OPNET	3 h.	135,00 €	Analista
Programación de clases en VB.NET	16 h.	480,00 €	Programador
Programación del Interface en VB.NET	8 h.	240,00 €	Programador
Programación de la importación /exportación a XML	8 h.	240,00 €	Programador
Primeras pruebas y retoques en el código	5 h.	150,00 €	Programador
Generación del modelo base mediante OPNET	2 h.	90,00 €	Analista
Comprobación del funcionamiento correcto del modelo base	2 h.	90,00 €	Analista

Banco de pruebas de la correcta importación/exportación XML	5 h.	150,00 €	Programador
Ultimo retoques en el código	10 h.	300,00 €	Programador
Genera documentación	10 h.	450,00 €	Analista
<b>TOTAL</b>	<b>69 h</b>	<b>2.325 €</b>	

<i>Tarea - OPNET</i>	<i>Horas</i>	<i>Costo</i>	<i>Recurso</i>
Implementación en OPNET de la simulación 1	1h	30,00 €	Analista
Simulación	1h	45,00 €	Analista
Tratamiento y recolecta de datos de la simulación 1	1h	45,00 €	Analista
Implementación en OPNET de la simulación 2	1h	45,00 €	Analista
Simulación	1h	45,00 €	Analista
Tratamiento y recolecta de datos de la simulación 2	1h	45,00 €	Analista
Implementación en OPNET de la simulación 3	1h	45,00 €	Analista
Simulación	1h	45,00 €	Analista
Tratamiento y recolecta de datos de la simulación 3	1h	45,00 €	Analista
Implementación en OPNET de la simulación 4	1h	45,00 €	Analista
Simulación	1h	45,00 €	Analista
Tratamiento y recolecta de datos de la simulación 4	1h	45,00 €	Analista
Implementación en OPNET de la simulación 6	1h	45,00 €	Analista
Simulación	1h	45,00 €	Analista
Tratamiento y recolecta de datos de la simulación 5	1h	45,00 €	Analista
Documentación de los datos	4h	180,00 €	Analista
<b>TOTAL</b>	<b>22 h</b>	<b>990,00 €</b>	

## 2.2.5 Organización y costos del proyecto

	Nombre de tarea	Duración	Comienzo	Fin	Tareas críticas	Predecesor	Costo	Nombres de los recursos
1	□ GenNode	9,13 días?	jue 01/04/10	mié 14/04/10		Sí	2.325,00 €	
2	Documentación de los metatags que utiliza OPNET	3 horas	jue 01/04/10	jue 01/04/10		No	135,00 €	Analista
3	Programación de clases en VB.NET	16 horas	jue 01/04/10	lun 05/04/10	No 2	480,00 €	Programador	
4	Programación del Interface en VB.NET	8 horas	lun 05/04/10	mar 06/04/10	No 3	240,00 €	Programador	
5	Programación de la importación /exportación a XML	8 horas	mar 06/04/10	mié 07/04/10	No 4	240,00 €	Programador	
6	Primeras pruebas y retoques en el código	5 horas	mié 07/04/10	mié 07/04/10	No 5	150,00 €	Programador	
7	Generación del modelo base mediante OPNET	2 horas?	jue 08/04/10	jue 08/04/10	No	90,00 €	Analista	
8	Comprobación del funcionamiento correcto del modelo bas	2 horas?	vie 09/04/10	vie 09/04/10	No 7	90,00 €	Analista	
9	Banco de pruebas de la correcta importación/exportación :	5 horas	vie 09/04/10	vie 09/04/10	Sí	150,00 €	Programador	
10	Ultimo retoques en el código	10 horas	vie 09/04/10	lun 12/04/10	Sí 9	300,00 €	Programador	
11	Generación de la documentación	10 horas	lun 12/04/10	mié 14/04/10	Sí 10	450,00 €	Analista	
12	□ OPNET	2,75 días	mié 14/04/10	vie 16/04/10	Sí 1	975,00 €		
13	Implementación en OPNET de la simulación 1	1 hora	mié 14/04/10	mié 14/04/10	Sí	30,00 €	Programador	
14	Simulación	1 hora	mié 14/04/10	mié 14/04/10	Sí 13	45,00 €	Analista	
15	Tratamiento y recolecta de datos de la simulacion 1	1 hora	mié 14/04/10	mié 14/04/10	Sí 14	45,00 €	Analista	
16	Implementación en OPNET de la simulación 2	1 hora	mié 14/04/10	mié 14/04/10	Sí 15	45,00 €	Analista	
17	Simulación	1 hora	mié 14/04/10	mié 14/04/10	Sí 16	45,00 €	Analista	
18	Tratamiento y recolecta de datos de la simulacion 2	1 hora	mié 14/04/10	mié 14/04/10	Sí 17	45,00 €	Analista	
19	Implementación en OPNET de la simulación 3	1 hora	mié 14/04/10	mié 14/04/10	Sí 18	45,00 €	Analista	
20	Simulación	1 hora	jue 15/04/10	jue 15/04/10	Sí 19	45,00 €	Analista	
21	Tratamiento y recolecta de datos de la simulacion 3	1 hora	jue 15/04/10	jue 15/04/10	Sí 20	45,00 €	Analista	
22	Implementación en OPNET de la simulación 4	1 hora	jue 15/04/10	jue 15/04/10	Sí 21	45,00 €	Analista	
23	Simulación	1 hora	jue 15/04/10	jue 15/04/10	Sí 22	45,00 €	Analista	
24	Tratamiento y recolecta de datos de la simulacion 3	1 hora	jue 15/04/10	jue 15/04/10	Sí 23	45,00 €	Analista	
25	Implementación en OPNET de la simulación 5	1 hora	jue 15/04/10	jue 15/04/10	Sí 24	45,00 €	Analista	
26	Simulación	1 hora	jue 15/04/10	jue 15/04/10	Sí 25	45,00 €	Analista	
27	Tratamiento y recolecta de datos de la simulacion 4	1 hora	jue 15/04/10	jue 15/04/10	Sí 26	45,00 €	Analista	
28	Implementación en OPNET de la simulación 6	1 hora	vie 16/04/10	vie 16/04/10	Sí 27	45,00 €	Analista	
29	Simulación	1 hora	vie 16/04/10	vie 16/04/10	Sí 28	45,00 €	Analista	
30	Tratamiento y recolecta de datos de la simulacion 5	1 hora	vie 16/04/10	vie 16/04/10	Sí 29	45,00 €	Analista	
31	Documentacion de los datos	4 horas	vie 16/04/10	vie 16/04/10	Sí 30	180,00 €	Analista	



### **3. Base teórica y protocolos a estudiar**

A continuación se estudiarán los tipos de direccionamiento que existen y se definirán conceptos referentes a los protocolos multicast existentes. Se profundizará más en los protocolos multicast que interviene utilizados en la simulación.

#### *3.1 Multicast, un primer enfoque*

El envío multicast (o multidifusión) permite enviar información a múltiples destinos, usando la estrategia más eficiente para el envío de mensaje sobre cada enlace de la red, creando copias de los paquetes de datos y distribuyéndolas según los destinos que estén interesado en recibirlo.

Las aplicaciones de multidifusión, incluyen, entre otros, la difusión de datos, distribución en tiempo real, datos bursátiles, televisión digital a través de una red IP (IPTV), radio de Internet, videoconferencias multipunto, aprendizaje a distancia, comunicaciones corporativas, etc. Otras aplicaciones incluyen simulación interactiva, grid computing (procesos que se distribuyen entre varios ordenadores), videojuegos (donde el emisor puede ser receptor). Multicast, nos permite un mejor uso del ancho de banda, reduce el tráfico entregando un único flujo de información a una gran mayoría de destinos, una menor carga en el procesamiento para el router y el host.

Entre las ventajas encontramos una mejor eficiencia en la utilización de recursos, controla el tráfico en la red y reduce la carga de la CPU, optimiza la calidad del servicio eliminando el tráfico redundante y hace posibles las aplicaciones multipunto. Por el contrario, las desventajas que conlleva el uso de multicast pueden ser la perdida de paquetes (las aplicaciones multicast no deben esperar una transmisión confiable), no existe el control de congestión (la falta de una ventana TCP puede resultar una red congestionada), algunos protocolos de multicast pueden generar paquetes duplicados y los cambios en la topología de la red puede alterar el orden de la transmisión de paquetes enviados.

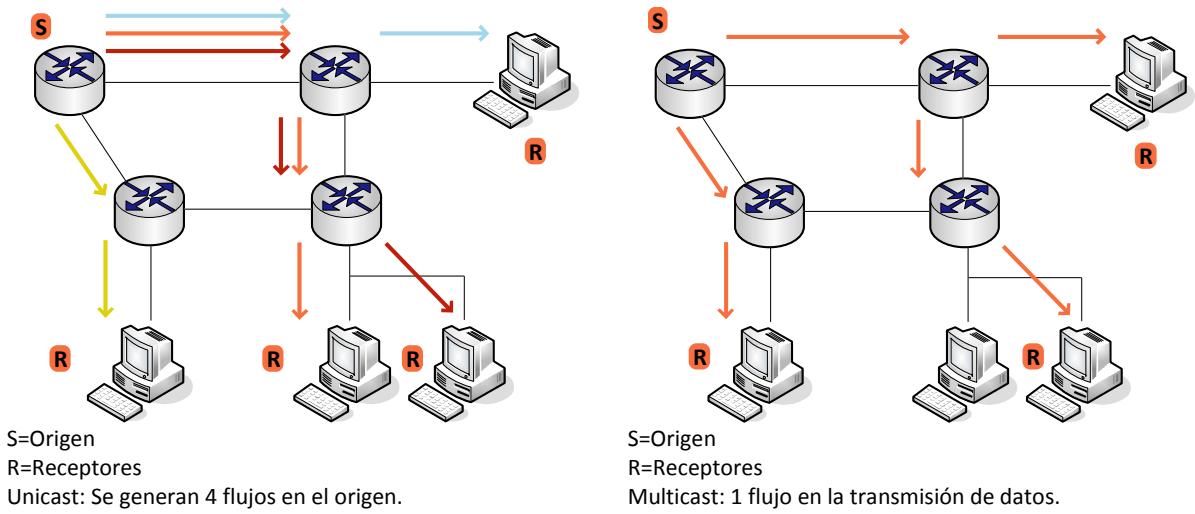
Podemos encontrar diferentes protocolos multicast, entre ellos está PIM (Protocol Independent Multicast) y demás protocolos de apoyo a la multidifusión, que más adelante describiremos, como es el IGMP para la gestión de grupos de difusión.

### *3.2 Tipos de direccionamiento.*

Existen cuatro formas de direccionamiento IP, es decir, diferentes estrategias para el envío de la información, desde un origen a uno varios destinos, según el tipo de direccionamiento. Cada una de ellas con sus propiedades que les corresponden.

- Unicast (unidifusión): Se trata de envío de información desde un único emisor a una máquina destino o receptor final. En el momento que se deseé enviar la información a varios receptores se establecerán las misma comunicaciones de envío como máquinas destino hayan. Se pueden encontrar que algunos PC tienen varias direcciones unicast, cada una para un propósito diferente. Para enviar la misma información a diferentes direcciones unicast el emisor debe enviar dichos datos una vez por cada receptor. Actualmente se la forma predominante de transmisión en Internet. Ejemplos básicos de aplicaciones unicast son los protocolos http, smtp, ftp o telnet.
- Broadcast (difusión): Consiste al envío de datos a todos los posibles destinos de una misma red, donde el emisor envía los datos una única vez a todos los receptores. Con diferencia al unicast, solo se transmite una vez los datos desde el origen, y lo reciben todos los receptores existentes en la red. En el protocolo IP, la dirección 255.255.255.255 representa un broadcast limitado de forma local. Para hacer un broadcast directo (y limitado) combinando el prefijo de red con el sufijo de hosts compuesto únicamente de 1 binarios. Por ejemplo para una red con el prefijo 192.0.2 la dirección IP a usar será la 192.0.2.255 (entendiendo que la máscara de red es la 255.255.255.0).
- Multicast (multidifusión): Permite envío de información a un grupo interesado de recibir dicha información. Igual que sucede en Broadcast, el envío es único e independientemente del número de máquinas destino. El receptor escuchará un grupo multicast y estarán escuchando los receptores que estén interesados en recibir esta información. De acuerdo al RFC 3171 las direcciones desde la 224.0.0.0 a la 239.255.255.255 están destinadas para ser direcciones de multicast. Este rango se llama formalmente "Clase D". El emisor envía un único datagrama (desde la dirección unicast del emisor) a la dirección multicast y el router se encargará de hacer copias y enviarlas a todos los receptores que hayan informado de su interés por los datos de ese emisor.
- Anycast: Igual que el broadcast y el multicast, con anycast se representa una topología de uno a muchos pero teniendo en cuenta que el receptor es único. Es decir, en el flujo de datos se seleccionará a uno como receptor de la información. El receptor seleccionado es el que considere

que esté más cerca en la red. Anycast se suele usar para dar alta disponibilidad y balanceo de carga en protocolos sin gestión del estado, como por ejemplo, en el acceso a información replicada.



En un escenario donde la emisión de datos es de forma masiva a determinados receptores, el multicast comparándolo con unicast es mucho más eficiente porque solo realiza una conexión para realizar el envío de datos, además hace un uso menor de los recursos existentes en la red, como puede ser el ancho de banda. Con broadcast enviará los datos a todos los destinatarios de la red y en el escenario particular no interesa este tipo de direccionamiento. Por otra parte una desventaja del multicast (y broadcast), es que se utilizan transmisiones de tipo UDP, por lo tanto perdemos la confiabilidad que nos puede otorgar una transmisión de tipo TCP.

### 3.3 Conceptos básicos de protocolos multicast.

Los protocolos multicast resuelven el problema de cómo hacer llegar el tráfico multicast desde los emisores allí donde se encuentren a los receptores que están interesados de recibir la información. Los encargados de realizar dicha tarea son los routers y los protocolos multicast. Todos los protocolos de ruteo multicast parte de la premisa que existe una ruta viable (unicast) que permite hacer llegar los datagramas de los emisores a los receptores, y el protocolo multicast se encarga de optimizarla tanto para todos orígenes de datos y como para todos los receptores interesados.

En una comunicación multicast se emiten todo tipo de datos, ya sea voz, video o datos de interés. Esta emisión no tiene límite geográfico o físico y los destinatarios se pueden encontrar en cualquier lugar de Internet o cualquier red interna privada. En cuestiones de ancho de banda la

conectividad puede ser de naturaleza heterogénea (por ejemplo los receptores conectados a través de internet), o homogénea (por ejemplo IPTV).

Cuando enviamos información vía multicast llega a los receptores prácticamente al mismo tiempo, por lo tanto la espera del receptor será más corta. Además, los emisores de datos no necesitan saber la dirección por cada cliente interesado porque usan una única dirección multicast. Es decir, los receptores interesados escucharán una dirección multicast y los emisores “enviarán” datos hacia una única dirección multicast. De esto hace que el servicio sea anónimo tanto para el emisor como para el receptor.

Los receptores, pendientes la recepción de datos, están unidos a una dirección multicast gestionado por el protocolo IGMP (Internet Group Management Protocol). Según la jerarquía TCP/IP el protocolo IGMP está situado en la capa de comunicaciones por encima del IP en el nivel de red o internet. Además el IGMP esta tan íntimamente ligado al IP que se puede ver como una parte integrada al IP, es decir, un módulo más dentro del propio módulo o proceso IP.

Los mecanismos de transmisión multicast están disponibles la capa de red (capa 3). En los últimos tiempos, la atención se ha centrado en la capa 3, la IP. A finales de 1980 se definieron los primeros servicios de multicast IP y si crearon algoritmos que permitían a los ordenadores unirse (join) o separarse (leave) del grupo de multidifusión [RFC1054, RFC1112, RFC2201].

Durante la transmisión multicast es necesario un árbol de distribución. Un árbol de distribución es el enrutamiento que utilizará el protocolo multicast desde el origen de datos, al receptor de estos datos mediante una dirección multicast. La creación del árbol de distribución debe de garantizar que las copias de cada paquete se envíen a cada rama del árbol y asegurarnos que se enrutan los paquetes de forma de que cada router reciba el tráfico multicast requerido. Cada protocolo multicast tiene su estrategia para la construcción del árbol de distribución, que no siempre es la misma y se utiliza para un uso determinado.

En resumen, en un flujo multicast existirán tres de condiciones principales:

- Una dirección multicast, donde están implicados un origen de datos y cero o varios receptores desconocidos.

- Un receptor podrá solicitar la unión en cualquier momento participar en un grupo multicast ofrecido por una fuerte origen. Esto lo hará mediante mensajes de IGMP que envía a su router local.
- La red tiene que ser capaz de responder ante esta petición haciendo llegar al cliente estas tramas del grupo de multicast que ha solicitado. Esto se conseguirá mediante que se comentaran a continuación.

### 3.3.1 Clasificación de protocolos multicast

Para los diferentes aspectos de la multidifusión IP existen una gran variedad de protocolos. La siguiente tabla podemos ver los más conocidos y cuál es la capa OSI que le corresponde.

Protocolo	Descripción	RFC	OSI
<b>IGMP</b>	Internet Group Management Protocol. Permite gestionar los grupos y sus miembros en un flujo multicast. Los router utilizan este protocolo para realizar consultas de los miembros en una red.	1112	Red
<b>DVMRP</b>	Distance Vector Multicast Routing Protocol. Es un protocolo de enrutamiento.	1075	Red
<b>MOSPF</b>	Multicast Extensions to Open Shortest Path First. Es un protocolo de enrutamiento.	1584	Red
<b>PIM-SM</b>	Protocolo Independent multicast – Sparse Mode. Protocolo de enrutamiento.	2362	Red
<b>PIM-DM</b>	Protocolo Independent multicast – Dense Mode. Protocolo de enrutamiento.	3973	Red
<b>PGM</b>	Pragmatic General Multicast. Garantiza que el receptor perteneciente a un grupo puede recibir todos los paquetes o detectar paquetes perdidos irrecuperables.	3208	Red
<b>BGMP</b>	Border Gateway Multicast Protocol. Permite realizar enrutamientos entre dominios.	3913	Aplicación
<b>CBT</b>	Core Base Tree. Protocolo de enrutamiento.	2189	Red

Aunque estén clasificados como protocolos multicast, no todos realizan la misma función. En la tabla que se muestra a continuación, se clasifican los protocolos con su función.

Protocolo	Función
<b>IGMP</b>	Gestión de receptores y emisores.
<b>PIM-SM,PIM-DM,DVMRP ,MOSPF</b>	Topología de router a router.
<b>BGMP</b>	Comunicación de router a router entre dominios.
<b>MADCAP (Multicast Address Dynamic Client Allocation Protocol)</b>	Asignación de direcciones multicast.
<b>MASC (Multicast Address Set Claim Protocol)</b>	
<b>CGMP (Cisco Group Management Protocol)</b>	Comunicación del router al switch. (Cisco specific)
<b>GMRP (GARP Multicast Registration Protocol)</b>	
<b>IGMP snooping</b>	
<b>RGMP (Router-Port Group Management Protocol)</b>	

Los protocolos multicas que hemos mencionado en las figuras de arriba, los podemos clasificar según el modo de operación en dos grupos: protocolos DENSE MODE y protocolos PARSE MODE. Los protocolos DENSE MODE están diseñados para trabajar sobre redes preferiblemente con un ancho de banda amplio y los miembros del grupo están densamente distribuidos a través de la red; un extremo diferente es considerar que los miembros del grupo están ampliamente dispersos a través de la red, lo cual se refiere a un protocolo de enrutamiento SPARSE-MODE.

A continuación describiremos los dos grupos de protocolos.

- **DENSE MODE:** Están diseñados para entornos en los que existe una buena representación del grupo en la red, y se cuenta con ancho de banda suficiente. Es el más antiguo y el más sencillo y se caracteriza por utilizar inundaciones periódicas y podas para la construcción de los árboles de distribución multicast; a estos árboles se les conoce con el nombre de árboles basados en el origen, y existe uno de ellos por cada origen mediante el algoritmo de árbol de expansión mínima (Spanning Tree).

Estos protocolos muestran un menor retardo porque existe un árbol por cada origen de datos, pero tienen el inconveniente de consumen una mayor memoria en los routers multicast porque mantiene todos los arboles de distribución en las tablas del mismo. Además no es escalable y no es eficiente cuando el número de receptores es minoritario ni cuando están distribuidos de forma dispersa.

Los protocolos de DENSE MODE son aconsejables utilizarlos cuando hay un gran ancho de banda o cuando una mayoría de los routers quieren recibir el grupo de multicast.

Protocolos que utilizan el modo denso:

- ➔ DVMRP (Distance Vector Multicast Routing Protocol).
- ➔ PIM-DM (Protocol Independent Multicast – Dense Mode).
- ➔ MOSPF (Multicast OSPF).

Inconvenientes del modo denso:

- ➔ Cada router de la red ha de mantener:
  - Por cada emisor hay un árbol de expansión mínima, y este ha de estar registrado en cada router intermedio.
  - La relación de las ramas que han sido podadas para cada emisor y cada grupo (cada par (S, G), Source, Group).
- ➔ La gran cantidad de información de estado hace difícil establecer un servicio multicast en una red grande para un número elevado de emisores y grupos
- ➔ Para la construcción del árbol de distribución se procede por inundación (flooding) a todos los routers multicast y a continuación empezará la poda (prune). Para estar adaptado a los cambios orígenes, destinos o routers intermedios en la red este proceso se repite cada 2-3 minutos, por lo tanto generará mucho tráfico.

La evolución de DENSE MODE a SPARSE MODE es por evitar que todos los routers de Internet tengan que mantener información sobre grupos multicast en los que no están interesados.

- SPARSE MODE: Están orientados a redes que estén sometidos a tener una gran presencia de miembros del grupo de forma esparsa. Se basa explícitamente en árboles unidireccionales compartidos, centralizándolo en un punto de encuentro por grupo y distribuyéndolo al receptor, y además, según el caso puede comutarse y crear árboles del camino más corto hacia la fuente. Aunque la característica principal, como hemos comentado, es el uso de árboles compartidos (llamados puntos de reunión o Rendezvous point, RPs), donde los receptores escuchan al router origen y mantiene el estado del árbol multicast. Por cada grupo multicast existe un árbol que se comunica con el RP y sus receptores.

En términos generales el RP está situado en centro de la red multicast, y por lo tanto, el tráfico multicast siempre pasará por la situación del RP en la red. Esto sucede puede dar lugar a un mayor retardo porque nunca garantiza que la ruta hacia los destinos sea la más óptima. A medida que transcurre el tiempo, se irá generando el árbol de distribución a medida que aparecen más receptores. Los receptores, estando de forma esparcida, se comunican con el RP para unirse al grupo y recibir tráfico multicast. Este mantenimiento del árbol de distribución supondrá un menor uso de las sobrecargas en la red (tablas de ruta en los routers). A este modo de operación se le denomina "dirigido a miembros".

El SPARSE MODE es preferible al DENSE MODE cuando el número de receptores es minoritario y de forma esparcida aunque actualmente es el más utilizado en Internet, pues es escalable.

Protocolos que utilizan el modo disperso:

- ➔ PIM-SM v2 (Protocol Independent Multicast – Sparse Mode)
- ➔ CBT v2 (Core Based Trees)
- ➔ BGMP (Border Gateway Multicast Protocol)

Inconvenientes del modo denso:

- ➔ No siempre existirá el camino más óptimo hacia el destino.
- ➔ Si existen muchos orígenes de datos y miembros, el establecimiento de datos directo a la fuente mediante árboles dedicados puede acabar saturando las tablas de los routers que mantiene el estado del encaminamiento.

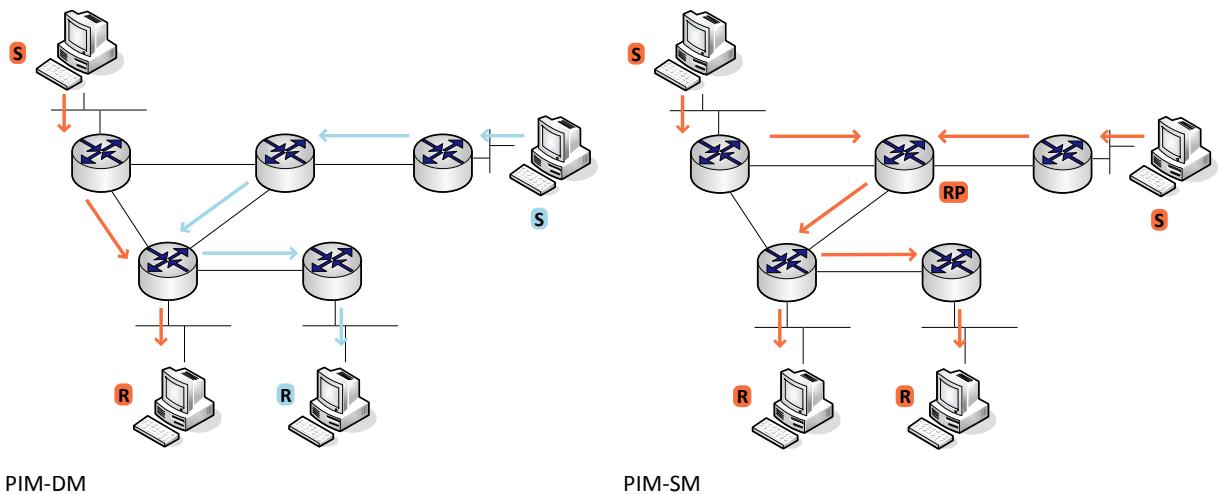
### 3.3.2 Introducción a PIM

El protocolo multicast independiente (PIM) es una familia de protocolos de enrutamiento multidifusión de Internet (IP). Se denomina protocolo independiente, porque no incluye su propio mecanismo de detección de la topología de red, sino que utiliza la información de enrutamiento suministrados por otros protocolos de enrutamiento tradicionales, como el Border Getaway (BGP), Enhanced Interior Gateway Routing Protocol (EIGRP) y Open Shortest Path First (OSPF). PIM utiliza esta información de enrutamiento para realizar el envío de la multidifusión.

Los protocolos PIM utilizan el modo denso (PIM-DM, *Dense mode*) y esparcido (PIM-SM, *Sparse mode*), que como se ha definido anteriormente, según el modo se adapta a la densidad de equipos de un grupo multidifusión presentes en un sistema autónomo. El modo denso presupone que todos los routers están interesados en recibir tráfico multicast e inicialmente inunda toda la red con datagramas de control siguiendo un árbol óptimo sin bucles, y en el momento que un router deje de estar interesado enviará un mensaje de “prune” o podado de la rama donde está dicho router. El modo disperso, los routers son los que demandan la unión al grupo multicast, por lo tanto no hay ningún árbol de distribución creado. En el momento que algún router quiera formar parte del grupo, se solicitará un mensaje “join” o de unión al grupo. Los dos protocolos pueden convivir en un sistema autónomo (por ejemplo, para distintos grupos multidifusión), siendo incluso posible comutar entre ambos modos, lo que permitiría usar ambas soluciones dentro de un mismo grupo.

Aunque PIM-DM es relativamente simple y optimiza al máximo el árbol de distribución pero presenta problemas de escalabilidad, ya que cada router de la red debe mantener la relación con todas ‘ramas’ que cuelgan de él, así como la relación de las ramas que han sido podadas por cada emisor y grupo multicast en el árbol de distribución. Si por cada router existe esta información y la red contiene muchos routers intermedios, el servicio multicast puede ser deficiente. El modo denso es el más sencillo de implementar; fue el primero que se utilizó en Internet y el primero que se estandarizó.

En el caso de PIM-SM, no existe la inundación para establecer un árbol inicial, sino que se establece un punto de encuentro (Rendezvous point, RPs) donde los emisores son los que se tienen que registrar (Join) para poder emitir datos y los receptores se tendrán que registrar para recibir el flujo multicast del grupo. Los árboles que se crean, serán compartidos partiendo desde el RP, por lo tanto la cantidad de información será mínima manteniendo solo la información de  $(*, G)$ , donde \* es la IP de la fuente de información y G que es la IP del grupo multicast. Además PIM-SM puede establecer cuál es el umbral de tráfico multicast a partir del cual los routers puedan comutar de árbol compartido a árbol de distribución basado en el origen o SPT.



### 3.3.3 Direcciónamiento IP Multicast

Las direcciones IP que están clasificadas como grupos multicast y están identificadas como clase D (rango 224.0.0.0 a 239.255.255). Si pensamos en una analogía podemos hablar de los canales de TV. Cuando un usuario sintoniza el canal uno se unirá al conjunto de espectadores (receptores) del canal uno. El momento que cambie al canal dos, el usuario se unirá al conjunto de espectadores del canal dos. En la analogía cada canal puede ser una dirección multicast, es decir por ejemplo el canal uno tiene la IP 239.10.10.1 y el canal dos la IP 239.10.10.2. Por lo tanto cada proveedor de contenido, tendrá una dirección u otra asignada.

Se puede encontrar cero o mas destinatarios activos por cada dirección multicast, y estos serán los que reciban la copia del datagrama que se envía a dicha dirección. El grupo multicast está formado por los miembros activos, y se mantiene de forma dinámica mediante conexiones y desconexiones de los mismos (orientado al receptor). Independientemente de su naturaleza dinámica, los grupos multicast pueden ser permanentes o transitorios.

- **Grupo permanente:** En este caso la dirección IP multicast es fija por IANA e independiente del número de receptores que tenga el grupo. Se asocian a aplicaciones normalizadas. Estas direcciones estarán en el rango 224.0.0.0 – 224.0.0.255 y son para uso específico de protocolos.

Algunas direcciones conocidas (well known):

- 224.0.0.1 = Todos los sistemas de la subred
- 224.0.0.2 = Todos los routers de la subred

- 224.0.0.4 = Todos los routers DVMRP de la subred
- 224.0.0.5 = Todos los routers OSPF del dominio
- 224.0.0.13 = Todos los routers PIM
- 224.0.0.22 = Todos los routers IGMPv3-capable

Se puede hacer uso del servicio DNS para localizar la dirección asociada a un grupo multicast permanente (dominio mcast.net) y lo mismo para sus resoluciones inversas (224.in-addr.arpa.)

- Grupo transitorio: Son creadas dinámicamente (en el momento que se lanza una aplicación multidifusión), y dejará de existir cuando no tenga miembros activos. El grupo multicast se considera siempre de receptores, no de emisores. Existen estas direcciones desde 224.0.1.0 – 238.255.255.255 y son conocidas como direcciones de ámbito global.

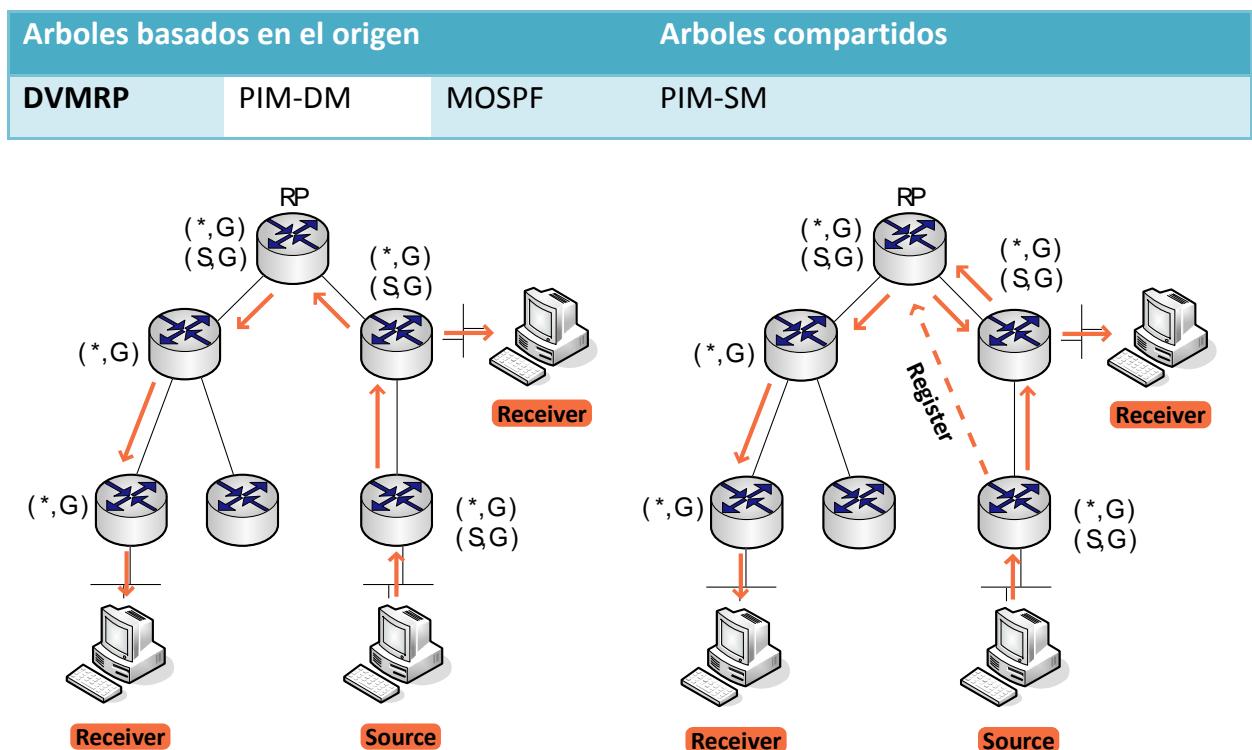
### *3.4 Árboles de distribución*

Una red multicast requiere un mecanismo para la creación de árboles de distribución que definan una única ruta de transmisión entre la subred de la fuente de contenido y cada red que contiene los miembros del grupo multicast, es decir, los receptores. Existen diferentes mecanismos para la creación y mantenimiento (por ejemplo la poda) de los árboles de distribución. La creación del árbol de distribución debe garantizar que las copias de cada paquete se envíen a cada rama del árbol y asegurar que se enrutan los paquetes de forma de que cada router reciba el tráfico multicast que necesite. Por lo tanto de enrutamiento multicast debe obtener un árbol más óptimo que conecte todos los routers del grupo multicast. Cada protocolo utiliza un algoritmo de enrutamiento y determina cuál es el árbol de distribución más correcto para transmitir la información a los nodos destinos del grupo de multidifusión.

Un miembro de un grupo multicast, podrá unirse (join) o dejar (leave o dejjoin) el flujo multicast durante el transcurso del tiempo, así pues, el árbol de distribución se cambiará dinámicamente. Cuando todos los receptores, que están en una subred, dejen el grupo multicast, esta rama (la interficie) hacia la subred se podará (prune) por lo tanto se actualizará de nuevo el árbol de distribución y pararán de recibir tráfico multicast los receptores. En el momento que un receptor se quiera unir al flujo multicast, se modificará el árbol de distribución agregando una nueva rama al árbol (la interficie) y empezará el reenvío de paquetes al receptor.

Se diferencian dos grupos de árboles de distribución, los árboles basados en origen (SPT, Short Path Tree) o los árboles de punto de reunión o árboles compartidos (RPT, Rendezvous Point Tree). Los SPT, son árboles que para cada origen se conectan con los receptores por el camino más corto posible. Los protocolos DVMRP, PIM-DM y MOSPF usan el SPT. El RPT, en cambio, usa un punto central único compartido para todos los grupos multicast. Por el punto de reunión pasa todo el tráfico y desde allí se distribuye a los receptores, y los emisores de datos envían al punto de reunión los datos para que se distribuyan.

En la siguiente figura muestra la clasificación de los diferentes protocolos según su grupo de árbol de distribución:



Arboles basados en el origen. Busca el camino más corto hacia el destino.

Arboles compartidos. En un punto de reunión (RP) se centraliza todos los datos para después distribuirlos

A continuación se describen las diferentes estrategias que utilizan los protocolos multicast para la generación de árboles de distribución.

### Flooding

Llamado también algoritmo por inundación, consiste en enviar cada paquete a todas los routers vecinos, exceptuando el que se ha recibido. Cuando un enrutador recibe un paquete multicast, primero revisara si el paquete a que ha llegado es la primera vez que llega o si ha llegado

anteriormente. Si es la primera vez entonces procederá a reenviarlo por hacia todos los routers vecinos excepto por el que llegó, si ya lo había recibido anteriormente el paquete será descartado. De esta forma se asegura que un enrutador reciba una sola copia del paquete multicast. La inundación puede multiplicar el tráfico si existen bucles en la topología, ya que en ese caso se envían paquetes duplicados.

### **Spanning Trees**

Un árbol de expansión es una subred de la red donde están incluidos todos los routers de la red en un único camino, sin contener ciclos. Por cada router del árbol de expansión, puede enviar una copia de paquete de entrada, exceptuando por el interface de entrada, sin temer que ese paquete se duplique en la llegada a otro router del árbol de expansión. Es decir, asegura que se generen paquetes duplicados. Conociendo el árbol de expansión, podemos llegar a todos los routers garantizando que se utilizan la cantidad mínima de paquetes necesarios. Este método puede ser deficiente a la hora de tener múltiples orígenes de información.

### **Reverse Path Forwarding (RPF)**

El algoritmo de difusión por el camino inverso, es una modificación mejorada del Spanning Tree. Construye un árbol desde el origen hasta cada miembro del grupo, es decir creará el camino más corto desde el origen a cada uno de los destinos. En el momento que un paquete llega al router, hace la comprobación para el reenvío el paquete, si no lo considera correcto el paquete lo eliminará. Cuando recibe el paquete, el algoritmo procede de la siguiente manera:

1. El router comprueba la dirección origen de la tabla unicast para determinar si la interface por donde ha llegado el paquete pertenece al camino más corto hacia el origen.
2. Si pertenece al camino más corto entonces reenviará el paquete por todos los interfaces menos por el cual ha llegado.
3. En caso contrario el paquete se eliminará.

Podemos encontrar dos extensiones del RPF, que son el algoritmo TRPB (Truncated Reverse Path Broadcasting) y RPM (Reverse Multicast Path).

### **Core-Base Tree (CBT)**

El enrutamiento se hará a partir de un nodo situado en el centro de la red multicast, donde se crea un árbol que es único para el grupo, independientemente de los orígenes de datos. El algoritmo realiza los siguientes pasos:

1. Se elige un router (nodo) situado en el centro del la red multicast, donde se gestionarán todas las consultas.
2. Por cada router receptor, donde están situados los host destinos, enviarán un mensaje de "join" al router central a la red, de pertenencia a un grupo multicast. El "join" o unión se envía mediante mensajes IGMP.
3. La solicitud "join" es procesada por todos los routers intermedios que están entre el router receptor y el router central, y a medida que "asciende" la petición hacia el router central, se crea el árbol.
  - 3.1. En el caso que los routers intermedios que reciben la solicitud ya forma parte del árbol de entrega de ese grupo multicast, comprobará si el interface por el que ha recibido la petición está asociado al grupo, sino es así lo asociara (es decir añadirá una nueva rama al árbol).
  - 3.2. En el caso que no formen parte del árbol, el router enviará la petición a otro router que este en el camino hacia router central.
4. A partir de ahora los paquetes de multidifusión se enviaran al router elegido como central. Una vez allí se distribuirán hacia los destinos, mediante la ruta más óptima.

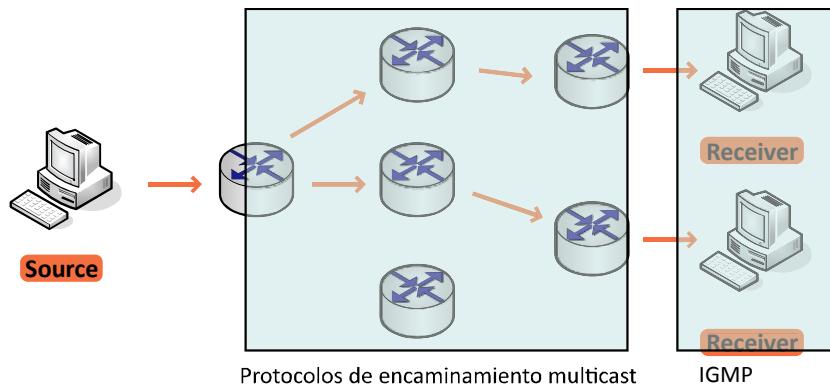
A pesar de que existen múltiples destinos y/o múltiples orígenes existirá solo un árbol de distribución, donde todos los orígenes puedan emitir hacia todos los receptores pertenecientes al grupo multicast. El camino será el más óptimo desde el router central a los receptores, pero el origen tendrá que emitir los paquetes al router central, por lo tanto no siempre será el camino más óptimo del origen al destino, será sub-óptimo.

En la siguiente figura, muestra la relación de los protocolos mencionados anteriormente y las estrategias para la creación del árbol de distribución.

	DENOS				ESPARCIDOS	
Protocolos	DVMRP	MOSPF	PIM-DM	BGMP	PIM-SM	CBT
Algoritmos de creación del árbol	TRPB(RPF)	Dijkstra	SPT	RP (CBT)	RP (CBT)	CBT

### 3.5 IGMP

En la siguiente figura observamos donde está situado el protocolo para la gestión de grupo de internet (Internet Group Management Protocol).



Está recogido en las RFC 1112, 2236, 3376 (las versiones 1, 2 y 3). Es un mecanismo para que el receptor informe de la unión (join) o desunión (leave) a un grupo multicast de forma dinámica. Esta información la mantienen los routers en una lista de los grupos multicast que están interesados los hosts conectados a sus interfaces. En cualquier momento un host puede estar interesado en pertenecer a un grupo multicast o dejar el grupo multicast. Esta información que recibe el router y se está actualizando de forma periódica. Los mensajes IGMP van encapsulados dentro de datagramas IP, con número de protocolo IP = 2, TTL = 1 y con la opción *IP Router Alert* en la cabecera IP. Por lo tanto estos mensajes solo pueden llegar al primer router directamente conectado entre el host y el router de una misma LAN.

En definitiva las funciones de IGMP realizan dos acciones básicas.

- Cuando un receptor se une a un nuevo grupo multicast, envía un mensaje IGMP de inscripción de grupo (Membership Report) con la dirección IP multicast del grupo deseado. Este mensaje IGMP también lo escucha el router de multicast local, que está conectado a la misma red, para realizar el correspondiente registro de inscripción en una lista de grupos que contiene el router. Asimismo, el datagrama IP que encapsula dicha inscripción lleva un TTL = 1 para indicar expresamente que este tipo de mensaje IGMP está restringido al primer salto, es decir no puede salir de la misma LAN y no debe ser reenviado por ningún otro router multicast. A su vez, el router de multidifusión local se pone en contacto con otros routers de multidifusión vecinos por Internet, pasando la correspondiente información y registrando, en sus tablas, las oportunas rutas en función de dichas pertenencias para la posterior fase de transferencia de datos multicast. Un receptor sólo tiene que emitir un informe IGMP de

pertenencia por cada grupo al que pertenezca. Cuando una máquina final o destinataria se une a un determinado grupo multicast, envía inmediatamente un informe de pertenencia sin esperar al siguiente sondeo.

- Teniendo en cuenta que la pertenencia es dinámica, el router de multidifusión local sondea de forma periódica, mediante un mensaje IGMP de solicitud de pertenencia a grupos (Host Membership Query message) a las máquinas vecinas de su red de área local para determinar aquéllas que se mantienen como miembros activos de grupos. Este mensaje IGMP de solicitud o sondeo se envía en un datagrama IP con la dirección de destino conteniendo la dirección reservada de multidifusión, 224.0.0.1, que semánticamente quiere decir “a todas las máquinas en esta red de área local”. Asimismo, dicho datagrama IP lleva un TTL = 1 para indicar que este tipo de mensaje IGMP no debe ser reenviado a ninguna otra red por ningún otro router de multidifusión local que pudiera haber. En este escenario, para que un router de multidifusión local pueda difundir alguna información de pertenencia a otros routers de multidifusión intermedios por Internet, debe determinar si una o más máquinas, en su red de área local, han decidido unirse a un grupo de multidifusión. Si para un determinado grupo, no se reciben informes de miembros después de varios sondeos, el router de multidifusión asume que no hay destinatarios activos en su red y deja de anunciar miembros del grupo a otros routers de multidifusión vecinos en Internet. Cuando una máquina destinataria recibe una solicitud, debe responder con un informe por cada grupo al que pertenece. Asimismo, una máquina destinataria no emite ningún informe si quiere abandonar el grupo. Se resalta, además, que un router de multidifusión local no tiene porqué conocer cuántas máquinas pertenecen a un grupo particular; sólo le debe interesar saber que al menos una máquina pertenece a un determinado grupo.

Dentro de la arquitectura OSI está en la misma línea que el protocolo ICMP un mismo subnivel de comunicaciones por encima de IP en el nivel de red o Internet. IGMP es un modulo dentro del IP, es decir el protocolo IGMP encapsula un mensaje IGMP en un datagrama IP. De ahí que IGMP ocupe un subnivel superior al ocupado por el protocolo IP en el mismo nivel de Internet o red de la arquitectura TCP/IP.

De las tres que existen actualmente las más extendidas es la versión 2 y 3, ya versión 1 fue la más básica y la base de las siguientes. Cada versión amplia a la anterior, así que la primera versión la

pueden utilizar sistemas operativos antiguos, como por ejemplo Windows 95, y la versión 3 se puede encontrar en Windows XP o 2003 server.

### **IGMP Versión 1**

Está recogido en el RFC 1112, y es un simple protocolo consta de dos mensajes de control; informe de pertenencia al grupo (MR), emitidos por los hosts, y consulta de pertenencia al grupo (MQ), que son emitidos por los routers.

Los receptores interesados en formar parte del grupo envían el informe indicando que está interesado, que contiene referencias a la dirección multicast. El router lo registra y enruta los paquetes multicast al interface que soporta esa subred donde está el receptor registrado. Por otra parte el router periódicamente envía un MQ para comprobar que al menos un receptor está interesado en recibir tráfico a ese grupo. Este mensaje se envía a todos los host de la red multicast, es decir la dirección 224.0.0.1. En esta versión no existe ningún mecanismo para dejar el grupo multicast, en el momento que no hay respuesta a los tres consecutivos MQ, el router deja de enviar tráfico a dicho grupo. Por último hay que decir que ni el MQ ni el MR salen del ámbito de la LAN pues los mensajes IGMP llevan siempre TTL=1.

El formato de mensajes de la versión 1 de IGMP es el siguiente:

0	4	7	15 31
Versión	Type	Unused	Checksum
<b>Group address</b>			

Tipos de mensaje:

- Membership Query (Router)
- Membership Report (Host)

### **IGMP Versión 2**

La versión 2 tiene la misma funcionalidad básica de la versión 1. La principal diferencia es la introducción de un nuevo tipo de mensaje para dejar el grupo o Leave Group (LG). Con este mensaje, el router conoce activamente los receptores que tengan la intención de abandonar el grupo.

El host que quiera abandonar un grupo manda un mensaje de tipo leave a la dirección de todos los routers multicast (224.0.0.2). A continuación, el router envía un mensaje MQ al grupo que quiere abandonar el host. Si algún host contesta con un Report, entonces el router mantiene el grupo. Si

ningún host contesta en un tiempo dado, se considera que el que ha abandonado era el último del grupo y el router lo elimina.

Otra característica de esta versión es el tipo de consulta que podemos realizar. Puede ser de dos modalidades: General Query (equivalente al Membership Query de IGMP v1) y el Group-Specific Query, por el cual permitirá a los routers lanzar una consulta solamente exclusiva a los miembros de un grupo multicast determinado.

Además se puede modificar los timeouts para la recepción de mensajes MR, así se adaptara a la situación y necesidad de la LAN.

Esta versión de IGMP reduce significativamente la latencia en comparación con la versión 1. El formato del mensaje tiene una pequeña variación respecto a la versión 1:

0	7	15 31
Type	Max. Resp. time	Checksum
<b>Group address</b>		

Tipos de mensaje:

- General Query (antiguo Membership query, Router)
- Group-Specific-Query (Router)
- Member Report (Host)
- Leave Group (Host)

### IGMP Versión 3

En esta versión permite a los receptores pertenecer o dejar un conjunto de fuentes dentro de un grupo multicast, en lugar de una fuente individual (source-specific multicast). La elección de flujos multicast no solo se limita a direcciones destino además se puede especificar la dirección origen. Con esta característica, IGMPv 3 agrega soportes para “filtrado de fuentes”, es decir la capacidad de recibir paquetes de una particular dirección multicast, direcciones específicas o de todas las direcciones multicast. Esta información se puede utilizar en protocolos de enrutamiento multicast para evitar el envío de paquetes multicast desde ciertas direcciones de origen a redes donde no hay receptores interesados.

La selección de algunos emisores con el comando Member Report de IGMP v3, además de indicar el grupo multicast interesado, se especifica un filtro que puede tener dos modalidades:

- INCLUDE: En este modo el receptor anuncia que desea recibir tráfico de las fuentes indicadas mediante una lista.
- EXCLUDE: En este modo el receptor indica la o las direcciones que no desea recibir tráfico establecidas en una lista. Por lo tanto, recibirá información de las fuentes que no estén en dicha lista de direcciones multicast. Para recibir el tráfico de todas las fuentes, igual al comportamiento de IGMPv2, el receptor enviará una lista de exclusión vacía.

Además existe otro tipo de mensaje que es el Group-and-Source-Specific Query por el cual permite preguntar a los routers que receptor sigue estando interesando en recibir tráfico multicast de un grupo determinado originado por una fuente concreta. Estos mensajes se enviarán a la dirección multicast para que lo reciban todos los host asignados a dicho grupo en la LAN.

En la siguiente figura, se puede observar el formato de mensaje de 'query'.

0		7		15 31	
Type				Max. Resp. time	Checksum
<b>Group address</b>					
Resv	S	QRV	QQIC		Source of number
<b>Source address [1]</b>					
....					
<b>Source address [N]</b>					

Tipos de mensajes:

- Membership Query. (Router)
- General Query. (Router)
- Group-Specific Query. (Router)
- Group-and-Source-Specific Query (Router)
- Membership Report (Hosts)
- Membership Leave Group (Host)

A continuación se muestra el formato de ‘report’:

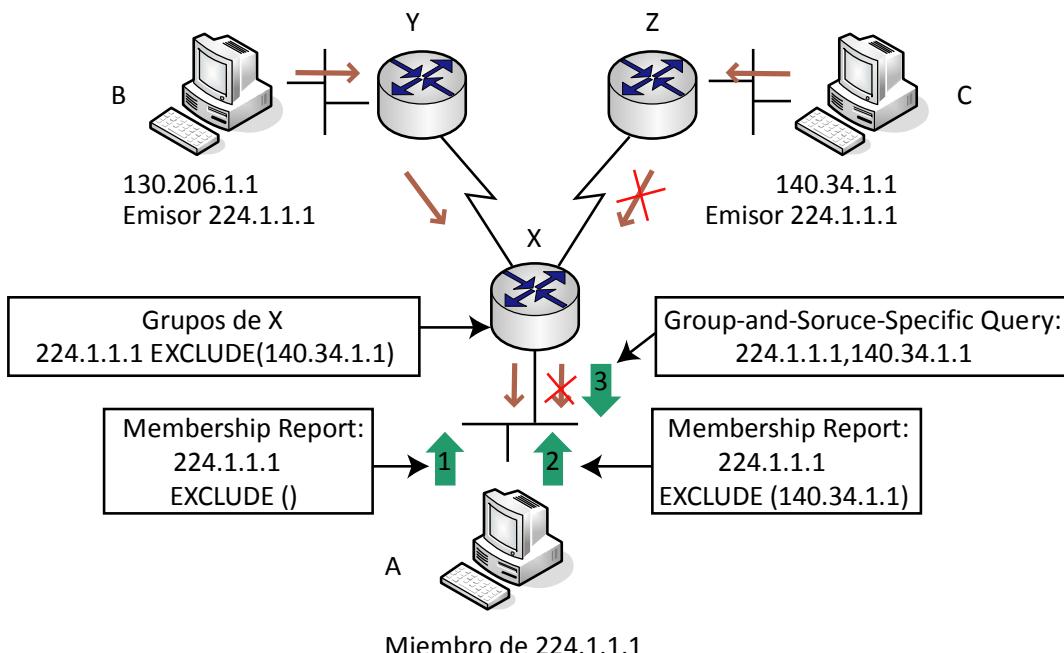
0	7	15	31
Type	Reserved	Checksum	
Reserved	Source of number (N)		
Group Record [1]			
....			
Group Record [N]			

0	7	15	31
Record type	Data length	# of sources (N)	
Group address			
Source address [1]			
Source address [N]			
Auxiliary data			

Record type:

- Estado actual
- Cambio de modo filtro (EXCLUDE)
- Cambio de lista de fuentes (INCLUDE)



En esta figura podemos ver el funcionamiento de la versión 3 de IGMP. Se pueden observar dos fuentes de datos multicast y un receptor, con tres routers intermedios. El escenario representado es que el receptor no quiere recibir datos de una determinada fuente. Y para ello el protocolo realiza el siguiente procedimiento:

- Inicialmente el host A está interesado en recibir datos del grupo multicast 224.1.1.1, entonces le envía un report a su router mediante el ‘Membership Report’ correspondiente. Como aun no se conoce los emisores de ese grupo, en el reporte no se excluyen ninguno, de esta forma se recibe de todas las fuentes emisoras.
- Una vez que el router X recibe el ‘Membership Report’ anota en su tabla el nuevo grupo. Una vez que ya está inscrito recibirá datos multicast tanto de B como de C.

- A continuación el receptor A no le interesa recibir tráfico que genera el emisor C, entonces debe de comunicárselo al router X para que tome las medidas oportunas y eventualmente suprima el tráfico del emisor. Para realizar esta tarea el host A envía un ‘Membership Report’ indicando el filtro EXCLUDE (140.34.1.1).
- Cuando recibe el router X el mensaje, envía un ‘Group-and-Source-Specific Query’ para confirmar que ningún receptor está interesado en recibir tráfico del 140.34.1.1.
- Como no se recibe ninguna respuesta y se supera el timeout oportuno de X, decide de aplicar el EXCLUDE (140.34.1.1) en su lista. Ya no se recibirá más tráfico de la fuente C.

### *3.6 PIM-SM*

Mediante el protocolo IGMP se conoce cuales son las fuentes (grupos multicast) y quiénes son los receptores interesados. Una vez resuelto esto lo siguiente es conocer la ruta que debe seguir el tráfico multicast desde el origen al destino. Esta labor es la que se va a ocupar PIM-SM.

El protocolo independiente multicast en modo disperso, se describe en el RFC 2362, y funciona de manera diferente a PIM-DM (modo denso). Es un protocolo más útil para grupos de receptores dispersos y alejados. Utiliza una variante del algoritmo CBT. En comparación con el protocolo en modo denso, no hay inundación seguida de una poda, por lo tanto no almacena información de control que es innecesaria (por lo tanto un overhead disminuye).

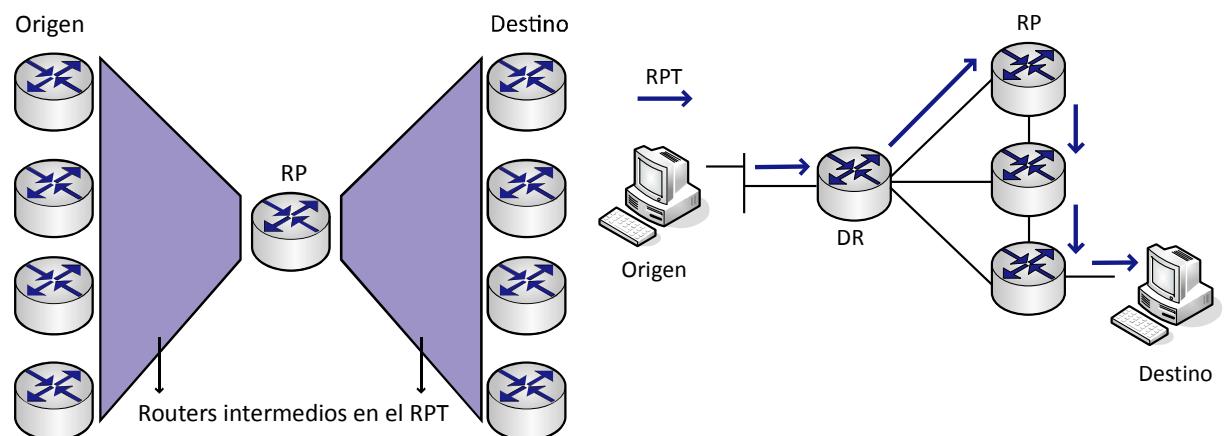
Actualmente PIM-SM es el protocolo de ruteo más utilizado en internet, dada su escalabilidad a una red de cualquier tamaño y además de poder establecer un número de fuentes de origen de datos para el mismo grupo. Al ser modo esparcido, tiene sentido tener un pequeño conjunto de receptores potenciales que estén distribuidos por todo el mundo.

PIM-SM hace uso de los árboles de los arboles compartidos y está basado en un RP, el llamado punto de reunión (Rendezvous Point), que tiene una función equivalente al Center Point del CBT. El RP es un punto en la red donde se conectan tanto los emisores como los receptores interesados. Opcionalmente puede construir un árbol de camino más corto por cada fuente, ya que algunos casos puede interesar usar esta estrategia en vez del RP. Las tablas de ruta que mantiene el protocolo son necesarias para conocer el próximo paso en la ruta a lo largo de un posible camino hasta la subred de destino. Esta información de las tablas además se utiliza para conocer cuál es el próximo paso que se va a seguir para la unión o la poda del router vecino.

El RP es un punto en común que tienen todos los nodos de la red. En el momento que un emisor desee enviar datos, el primer paso es comunicárselo a RP, y en el caso de la recepción de datos, el receptor debe de estar registrado en el RP. Todo el tráfico que fluye se envía desde el origen al RP, y a través del árbol de distribución (o llamado RP Tree), se reenvía a los destinatarios, optimizando la ruta que va a seguir el flujo multicast. El beneficio del modo esparcido es que está orientado a emisores, quiere decir que no existe flujo multicast hasta que al menos un receptor este interesado en recibir la información. Por lo tanto se reducirá bastante el tráfico trasversal de la red si existen poco receptores. Además el ‘recorte’ del árbol de distribución se va manteniendo a medida que van apareciendo receptores o van abandonando el grupo multicast.

### Estrategias de distribución de los orígenes de datos.

Cuando el router DR recibe tráfico para un particular grupo multicast desde un origen de datos, el DR encapsula el tráfico y lo reenvía hacia el RP que soporta el grupo multicast. El RP desencapsula el tráfico y lo reenvía hacia los receptores. De esta forma se crea el RPT que se extiende desde el router DR, a través de los routers intermedios y dirigiéndolo a los receptores que están interesados en recibir el tráfico.

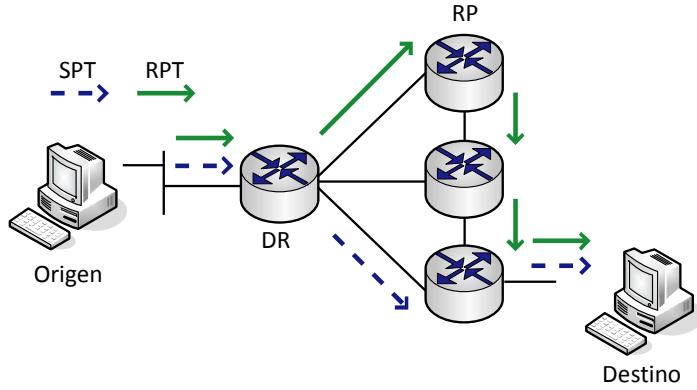


### Shortest-Path Tree (SPT)

SPT es especialmente útil cuando el tráfico entre el origen y el receptor es alto y existe una ruta alternativa que es más eficiente que la del RPT. En el protocolo se define un umbral de tráfico multicast, si se supera este umbral se comutará del RPT a SPT.

**Cambio a SPT:** Después de recibir el primer paquete del tráfico multicast pedido por un receptor, se utiliza el Reverse Path Forwarding (RPF) que conocer la ruta más corta hacia la fuente del grupo. A

continuación el router termina de utilizar el RPT para empezar a usar el SPT contactando el emisor al receptor. En ese momento, cuando el router empieza a recibir tráfico del origen a través del SPT, envía un mensaje de poda al RPT para terminar de enviar el tráfico del grupo multicast de esa ruta. Una vez completado este proceso se reducirá la concentración de tráfico innecesario y los retrasos de recepción de datos en los receptores.



### **Tipología de routers en PIM-SM.**

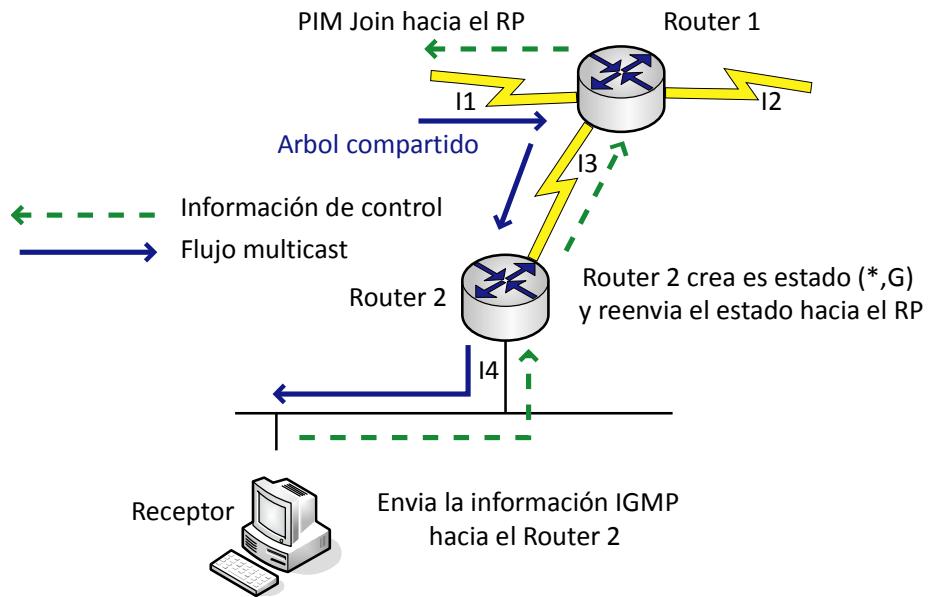
En un dominio PIM-SM, los routers se pueden identificar como Designated Router (DR, router designado), Rendezvous Router (RP o punto de encuentro) y Bootstrap Router (BSR). Estos participan durante la creación y mantenimiento del árbol RTP y el envío del flujo multicast.

- Designated Router (DR): Realiza la comunicación con el RP, donde le indica los estados de las fuentes de datos y los receptores.
- Bootstrap Router (BSR): El router seleccionado para esta función mantiene todos los routers del dominio PIM-SM informados del actual RP para cada grupo multicast.
- Rendezvous Point (RP): Cuando una fuente de datos desea enviar datos, primero se envía al RP, y cuando un receptor desea recibir datos, se registra en el RP. Cuando empieza la secuencia de datos, el emisor enviará al RP y después al receptor.
  - Static Rendezvous Point (Static-RP): Trabaja de igual forma que el RP, pero esta es una opción para la configuración manual del RP, donde todos los routers del dominio conocerán la ubicación de antemano.

#### *Designated Router (DR)*

En una LAN donde uno o mas routers están trabajan con PIM-SM, un router es elegido como DR en esta LAN. Cuando un receptor desea unirse al grupo multicast, se lo debe de notificar al DR. Se encarga de enviar dichas peticiones, tanto la unión como el abandono del grupo, al RP. Generalmente el DR es el router con la dirección más alta dentro de la sub-red. Mediante mensajes IGMP se comunica el deseo de la unión al grupo multicast al DR. Para conocer la ubicación del RP, el

DR lo calcula mediante funciones Hash, y a continuación envía un mensaje PIM unicast y crea una tabla de reenvío para el grupo multicast en el caso que sea necesario. En el caso que un receptor desea abandonar el grupo, envía mediante IGMP el comando correspondiente al DR y a continuación se realiza la poda del RPT.



En esta figura se puede observar el funcionamiento del DR. El receptor envía el mensaje IGMP Membership Report al router 2 e intenta añadirse al grupo multicast G. El primer router que recibe esta información IGMP es el router 2 y el registro de estado  $(*, G)$ . Se añade a una lista con el interface I4 en la lista de salidas de interface, y envía esta información hacia los router vecinos superiores (upstream). Este proceso se irá repitiendo hasta llegar al RP, donde también se añadirá esta información. El recorrido del receptor al RP se añadirá al RPT, por lo tanto ya existirá una ruta para enviar flujos multicast.

En el momento que el RP recibe la información  $(*, G)$ , y no tiene dicho estado, se cargará la entrada  $(*, G)$  en la tabla de rutas multicast, y se añadirá el interface del router 2 a la lista de salida de interfaces. Por lo tanto el árbol compartido del grupo multicast G se crea entre el RP, el router 2 y el receptor como se muestra en la figura de arriba. A partir de ahora cualquier mensaje desde el origen al RP se reenviará por todo el árbol compartido hasta el receptor.

En una LAN los criterios para la selección del DR son los siguientes:

- El router selecciona con la alta prioridad de selección.
- En el caso que exista varios con alta prioridad, se seleccionará el router con la IP más alta.

### *Bootstrap Router (BSR)*

Antes de que el DR reenvíe paquetes encapsulados a un grupo de multidifusión específica a un RP, debe conocer RP que esta seleccionado en el dominio para ese grupo. Por lo tanto debes existir un mecanismo para conocer el RP actual. El router bootstrap permite descubrir el RP mediante el siguiente procedimiento.

1. Conoce todos los C-RP (Candidate Rendezvous Points) por medio de los propios C-RP. Los C-RP se hacen conocer mediante un mensaje “C-RP-Advs” (C-RP Advertisement) que lo realiza periódicamente el propio C-RP.
2. Distribuye la información de todos los C-RP a todos los routers PIM-SM del dominio. Esto es seguido por una asignación de un grupo multicast específico o un rango de grupos del C-RP en el dominio.

El BSR envía periódicamente mensajes (Bootstrap messages) a todos los routers PIM-SM del dominio para mantener y actualizar el dato RPs en todo el dominio, además de mantener el estatus del BSR actual. Cuando el router recibe el BSM lo almacena, y en el momento que el DR se comunica con el RP (mediante un mensaje IGMP), el DR utiliza una función Hash para conocer la dirección del RP. A partir de ahora el DR ya puede enviar el mensaje de Join o Prune hacia el RP.

Dentro del dominio existen routers configurados como candidatos a BSR (C-BSR), y a través de un mecanismo de selección simple se elige un BSR. La elección del BSR será el que tenga la más alta prioridad configurada, si existen múltiples candidatos que tienen configurada la selección alta, será el que tenga la dirección IP más alta.

### *Rendezvous Point (RP)*

PIM-SM utiliza el RP para que controle el tráfico que se envía por cada grupo multicast. El RP realiza las siguientes tareas:

- Recibe el tráfico desde un origen mediante el DR.
- Recibe el Join desde los routers que desean recibir el tráfico multicast.
- Reenvía el tráfico recibido a los routers que lo peticionan.

Los routers que se interesan en recibir tráfico usan el IGMP para realizar la petición del tráfico multicast del grupo, envía dicha petición al RP por los routers intermedios, y si está conectado

directamente el DR al RP el interface de salida será el mismo. Esta operación forma el RP Tree (RPT) en el que solo la dirección multicast del destino aparece en la tabla de ruta del RP. Cada entrada en el router se representa de la siguiente forma:

- ( \*, G ), donde:

\* = Representa la dirección ip del origen multicast. El asterisco representa todos los orígenes.

G = La dirección IP del grupo del cual desea recibir tráfico.

El RP se elige de un dominio de forma estática o dinámica, para cada grupo en particular. Cuando es de forma dinámica, existen unos routers configurados como C-RP (que suelen coincidir con el C-BSR). Cada uno de los candidatos envían su petición al BSR con su IP y los grupos donde puede prestar servicio. En el momento que un DR necesita informar a un RP, procesará una función hash para decidir cuál es el RP para dicho grupo y así poder continuar con el protocolo.

La ubicación del RP no es crítica en cuestiones de rendimiento, porque generalmente los arboles SPT desde la fuente se establecen con el primer paquete enviado. Sin embargo si el RP falla el multicast en la red deja de funcionar.

#### Rango de direcciones de grupos Multicast:

Se ha comentado en el punto 1.3.3 los rangos IP que existen para los grupos multicast. En este caso el RP soporta dichos rangos IP para los grupos para la difusión multimedia. Estos rangos se expresan mediante una dirección IP y una máscara, donde la máscara define si la dirección es de una sola dirección o un rango de direcciones contiguas:

Multicast Address	Mask	Address Range
224.0.0.0	224.0.0.0	224.0.0.0 - 239.255.255.255

Una forma alternativa para expresar la dirección y máscara es 224.0.0.0/4

Un RP permite hasta cuatro rangos contiguos de grupos de multidifusión y / o grupos individuales de multidifusión.

RP Candidate Configuration	Supported Range of Multicast Groups
235.0.240.0/12	235.0.240.1 - 235.0.255.255

<b>235.0.0.1/28</b>	235.0.0.1 - 235.0.0.15
<b>235.0.0.128/32</b>	235.0.0.128 only
<b>235.0.0.77/32</b>	235.0.0.77 only

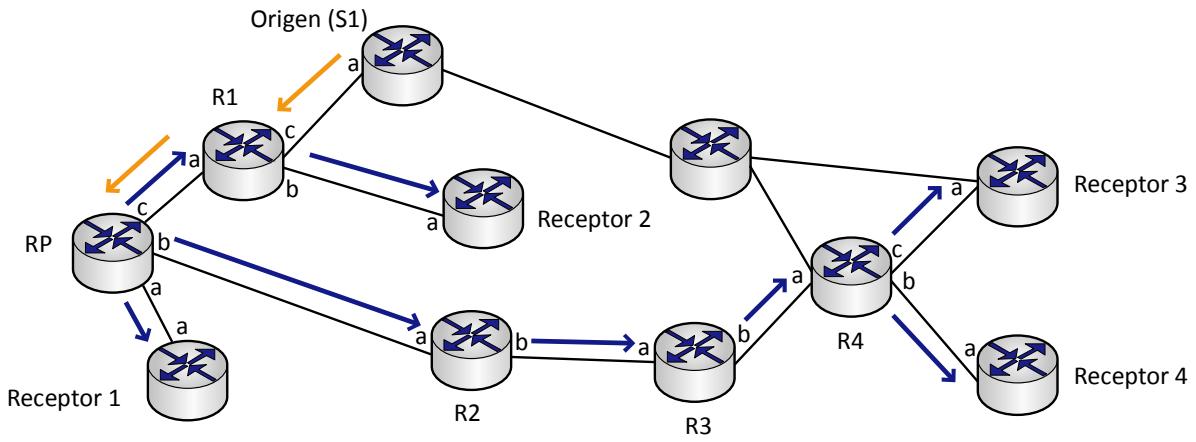
Si un determinado grupo multicast está excluido de todo RPs de un dominio, entonces ese grupo no estará disponible para los receptores de multidifusión conectados al dominio.

#### **Creación del RPT.**

Cuando varios receptores han pedido unirse a un grupo multicast y todos los mensajes han convergido al RP, estos formarán el árbol de distribución para ese grupo multicast, con origen G. Este árbol de distribución es el denominado RPTree (RPT), por el cual se comparte por todas las fuentes que envían datos de ese mismo grupo. Si todos los receptores conectados a un router no desean recibir tráfico, entonces se podará (Prune) esa rama del RPT.

Cuando el DR envía un mensaje hacia el RP de Join/Prune, utiliza se hace uso de dos flags por cada router intermedio en el recorrido; el bit WC (Wildcard o asterisco) y el bit RP-Tree. El WC-bit indica que se puede reenviar datos de cualquier fuente y que los receptores esperan recibir paquetes de todos los orígenes. El RPT-bit indica que esa entrada está asociada con un árbol compartido RPT y por lo tanto el mensaje de Join/Prune se propaga por todo el del RPTree. Por cada router de subida (o upstream) se crea o actualiza la entrada de router (\*,G) cuando recibe el mensaje de Join/Prune, con el RPT-bit y el WC-bit cargado. Y el interface en el que el mensaje Join/Prune llega se añade a la lista de interface de salida (oifs) con (\*,G).

En el momento que una fuente multicast empieza a enviar paquetes a un grupo, el router local a la fuente (DR) los recepciona, los encapsula y los envía como unicast al RP. El RP recibe dichos paquetes, los desencapsula y los distribuye sobre el RPT que esté formado y que correspondan a su grupo, es decir, que tenga el estado (\*,G). Por cada paquete que recibe el router se por las ramas del árbol hasta llegar a su destino.



RTP → STP →

ROUTER	Entrada	RPT	SPT	lif (input)	Oifs (output)
RP	(*,G)	1	-	Enc.	a, b, c
RP	(S1,G)	0	1	c	a, b
R1	(*,G)	1	-	a	b
R1	(S1,G)	0	1	c	a, b
R2	(*,G)	1	-	a	b
R3	(*,G)	1	-	a	b, c
R4	(*,G)	1	-	a	b, c
Receptor 1	(*,G)	1	-	a	L
Receptor 2	(*,G)	1	-	a	L
Receptor 3	(*,G)	1	-	a	L
Receptor 4	(*,G)	1	-	a	L
Origen (S1)	(S1,G)	-	1	L	a

### Unión a un Grupo.

En el momento que un router desea recibir tráfico para un grupo determinado, y con la dirección RP ya resuelta, envía un mensaje de unión al nodo vecino (si un paquete va del receptor al RP es de subida) que se encuentra en dirección hacia el RP (ya comentado anteriormente). Este mensaje de unión lleva la entrada  $(*,G)$  y la dirección del RP. En el momento que el nodo vecino recibe el mensaje  $(*, G)$  para el grupo  $G$  comprueba si existe el mismo estado en su tabla de enrutamiento. En el caso que exista entonces se registra en la interfaz por donde llegó el mensaje a la interfaz de salida el estado  $(*,G)$ . Si no existe se creará una nueva entrada con el estado  $(*,G)$  y se registrará en la interfaz de salida por la interfaz donde recibió el mensaje IGMP. Por cada mensaje de unión existe también información de la dirección IP del router PIM superior.

### **Poda de un Grupo.**

En el momento que los receptores que no desean recibir tráfico, envían un mensaje IGMP Leave o abandono del grupo G que está en el árbol compartido. Entonces se eliminará las entradas (\*, G) y (S, G) de la OIL (lista de interfaces de salida) de la interface por donde fue recibido el mensaje leave. Si realizando el procedimiento anterior la OIL queda en estado nulo, entonces el enrutador enviará un mensaje de poda (\*, G) Prune UP en dirección al RP y por cada router se repetirá el procedimiento hasta llegar al RP. Si por algún motivo el mensaje de poda no es enviado, el estado del árbol tiene un tiempo de expiración y en ese momento serán borrados de la tabla.

### **Emisión a un Grupo.**

El router empieza a recibir tráfico de la fuente de emisión S para un determinado grupo, este router no necesariamente tiene que ser receptor del grupo G, es decir que no es necesario que este unido al grupo. En ese momento el router sabe donde está situado el RP, y el enrutador encapsula el paquete multicast original en un mensaje “PIM Register message” enviado en transmisión unicast con dirección al RP. Desde este momento, cualquier paquete multicast que el nodo emisor envíe al enrutador será encapsulado hasta que el enrutador reciba el mensaje Register Stop del RP.

Cuando el RP recibe el primer paquete de datos del registro encapsulado desde una fuente S y un grupo G, lo desencapsula e iniciará un mensaje (S, G) Source-Specific Join hacia S, en el caso que no esté registrado. El mensaje de registro irá salto a salto hacia el origen S, y creará el árbol multicast (S, G) en los routers a lo largo del recorrido. Esta ramificación del árbol de distribución, servirá solo para la emisión al grupo G que vengan desde la fuente S. Los mensajes de unión que alcancen la subred de S o algún router que ya tiene la entrada (S,G), entonces los paquetes de S empezarán a fluir siguiendo rutas por el (S, G) hacia el RP. Estos paquetes puede alcanzar routers con el estado (\*, G) a lo largo del camino al RP, en ese caso pueden tomar un atajo sobre el árbol a este punto.

El router emisor continúa enviando paquetes encapsulados al RP, mientras se está en proceso de unión al árbol de fuente específica. Cuando los paquetes de S también llegan de forma “nativa”, es decir sin encapsular y por la ramificación del árbol multicast, el RP estará recibiendo paquetes de la fuente S duplicados, teniendo que descartar los paquetes encapsulados. En ese momento el RP enviará un mensaje de Register-Stop hacia al router origen de datos, así no se enviarán paquetes por duplicado desde la fuente. Por lo tanto la fuente específica de datos dejará de encapsular tráfico y lo enviará a través de la rama SPT creada hacia el RP, y los paquetes que reciban serán nativos.

Hay que tener en cuenta que una fuente específica puede empezar a enviar datos antes o después de que un destino se une a un grupo, por esto mismo puede suceder que el árbol compartido aun no esté construido pero recibiendo datos de un grupo.

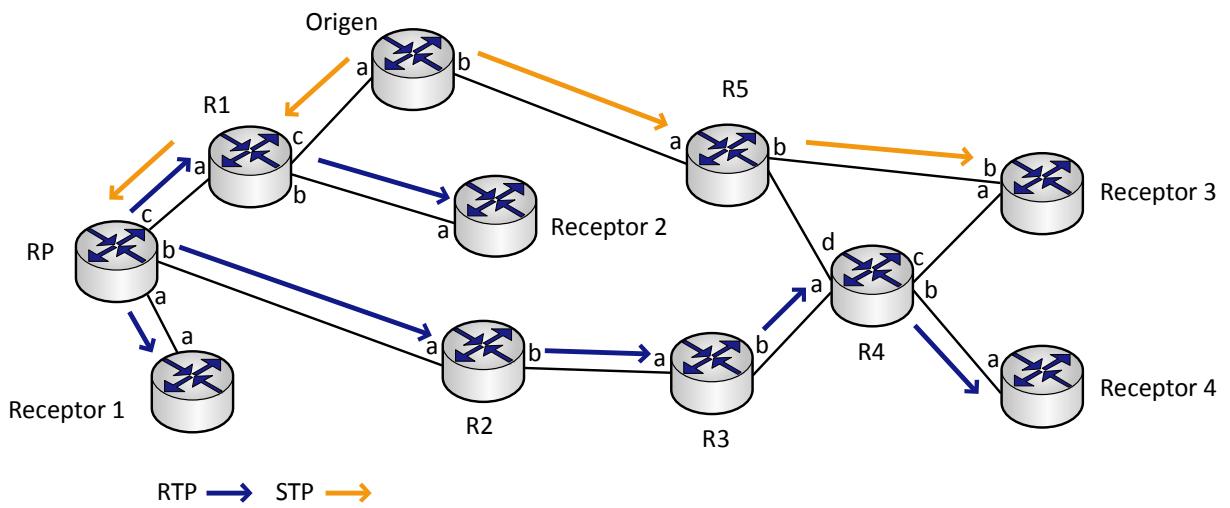
### **Cambio del SPT.**

Hay que tener en cuenta que no siempre existirá la ruta óptima del origen al destino. Para muchos destinos, la ruta a través del RP podría ser un desvío significativo si lo comparamos con la ruta más rápida hacia la fuente. Por eso existe la posibilidad de definir un umbral para realizar el cambio de un árbol compartido a un árbol mediate el SPT, es decir, buscar una ruta más optima cuando se dé este caso. Los valores se especifican en kilobits por segundo y pueden ser usados con una lista de acceso para saber cuáles son los valores de umbrales para cada grupo. Por defecto el valor umbral, o llamado SPT-Threshold, es 0 kbps, y quiere decir que cualquiera de los routers está conectado al SPT. Si existe con umbral infinito significa que no se unirán al SPT y que quedarán en el árbol compartido.

Si la cantidad de tráfico excede el SPT-Threshold de un determinado grupo, el enrutador que está conectado al receptor procederá a enviar un mensaje Join (S,G) hacia el origen del paquete, y se construirá un SPT desde el router emisor hasta el router que está conectado al receptor para unirlo al SPT. Se construirá un SPT que va desde el router emisor hasta el enrutador conectado directamente al receptor. En el momento que el mensaje Join alcanza el router emisor o algún router que tiene el estado (S, G), los paquetes de datos empiezan a fluir siguiendo la ruta del árbol de (S, G) hasta llegar al destino. Por lo tanto los destinos (o un router camino a los destinos) estarán recibiendo información duplicada, una desde el SPT y otra desde el RPT.

En el momento que el primer tráfico empieza a llegar desde el SPT, el router receptor empieza a descartar los paquetes para el grupo G que llegan desde el árbol RP. Además enviará un mensaje de Prune (S, G) (un mensaje de poda de la fuente S en el grupo G) hacia el RP. El mensaje de poda, va salto a salto hacia el RP informando por el camino dicha poda hasta alcanzar el RP o un router que requiera de S para otros destinos.

Este método tiene la ventaja de que se utiliza la ruta más corta para enviar los paquetes, y esto ayudará a reducir el retardo de la red sustancialmente. La reducción dependerá de la ubicación del router emisor con relación al RP. Pero el establecer otro camino hacia un destino requiere de memoria de los router, por lo tanto es una desventaja.



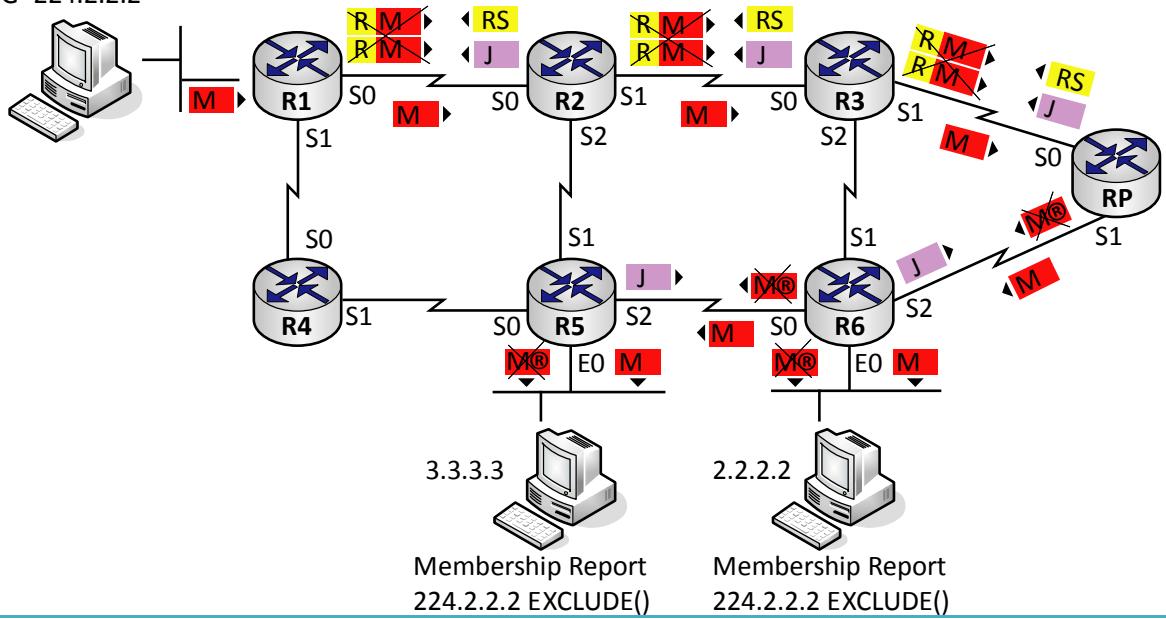
ROUTER	Entrada	RPT	SPT	iif	oifs
RP	(*,G)	1	-	Enc.	a, b, c
RP	(S1,G)	0	1	c	a, b
R1	(*,G)	1	-	a	b
R1	(S1,G)	0	1	c	a, b
R2	(*,G)	1	-	a	b
R3	(*,G)	1	-	a	b, c
R4	(*,G)	1	-	a	b, c
R5	(S1,G)	0	1	a	b
Receptor 1	(*,G)	1	-	a	L
Receptor 2	(*,G)	1	-	a	L
Receptor 3	(S1,G)	-	1	b	L
Receptor 4	(*,G)	1	-	a	L
Origen (S1)	(S1,G)	-	1	L	a, b

### 3.7 Funcionamiento IGMP y PIM-SM

Escenario: primero los receptores y después el emisor.

Origen S1 - 1.1.1.1

G=224.2.2.2



J	M	M®	R M	RS	▶	✗
Join msg	Datagrama multicast	Datagrama multicast desencapsulado	Register msg encapsulado	Register stop msg	Dirección del mensaje	Mensaje descartado

Tabla de ruta por router:

R1	Entrada	Salida	R2	Entrada	Salida
(S1,G)	E0	S0	(S1,G)	S0	S1
R3	Entrada	Salida			
(S1,G)	S0	S1	RP	Entrada	Salida
RP	Entrada	Salida	(*,G)	S0	S1
R6	Entrada	Salida	Registros de emisor en el RP	Entrada	Salida
(*,G)	S2	E0, S0	(S1,G)	S0	
	R5	Entrada	Registros de emisor en el RP	Entrada	Salida
	(*,G)	S2	(S1,G)	S0	E0

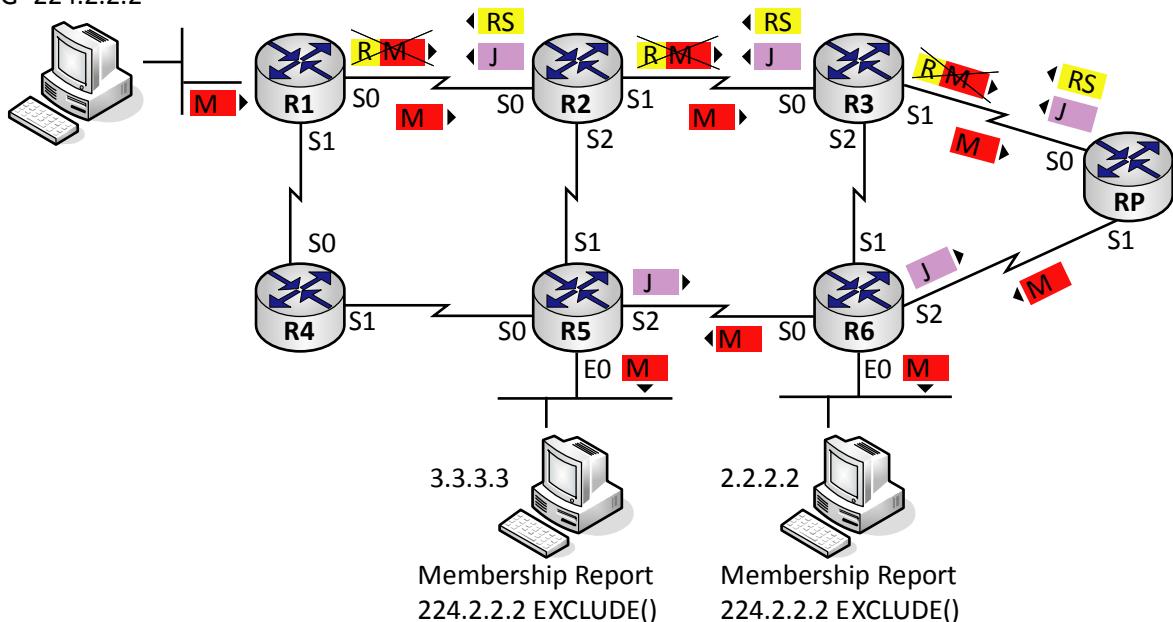
- Un cliente del router R6 empieza uniéndose al grupo multicast G, lo realiza mediante el mensaje IGMP ‘Membership Report 224.2.2.2 EXCLUDE ()’. El router envía una petición con un mensaje ‘join’ al RP y el RP recibe dicha suscripción.

- Un cliente del router R5 quiere suscribirse al grupo multicast G, enviará un mensaje IGMP MR al R5. Entonces el router R5 enviara el mensaje de 'join' dirección al RP. Pero no llega al RP porque cuando llega al router R6 ya reconoce que dicho router está en el mismo grupo por el cual quiere recibir el tráfico.
- Aparece una fuente de datos para el grupo multicast G. Para poder enviar datos al grupo G, envía un mensaje de 'register' (unicast) hacia el RP. El mensaje encapsulado contiene el datagrama multicast de 1.1.1.1. El RP al recibir un envío que va dirigido al grupo 224.2.2.2 y ya existen receptores para dicha emisión enviará un 'join' a R1. Los routers intermedios que reciben el 'join', actualizan su estado para que establezca una ruta hacia el RP. Mientras tanto el emisor continúa enviando tráfico encapsulado en mensajes 'register'.
- En el momento que el RP empieza a recibir tráfico sin encapsulación, es decir nativos propios de la fuente, el RP enviará la confirmación del registro con un mensaje 'register stop'. Esto le indicará que ya puede dejar de realizar envíos multicast.
- Desde el momento que el RP recibe cualquier dato de la fuente, ya sea encapsulado o nativo, lo reenviará a sus emisores registrados.

#### Escenario: Primero emisor después receptores.

Origen S1 - 1.1.1.1

G=224.2.2.2



Las tablas de rutas es la misma que el escenario anterior, excepto en el RP.

RP	Entrada	Salida	Registros de emisor en el RP
(S1,G)	S0	S1	(S1,G) S0

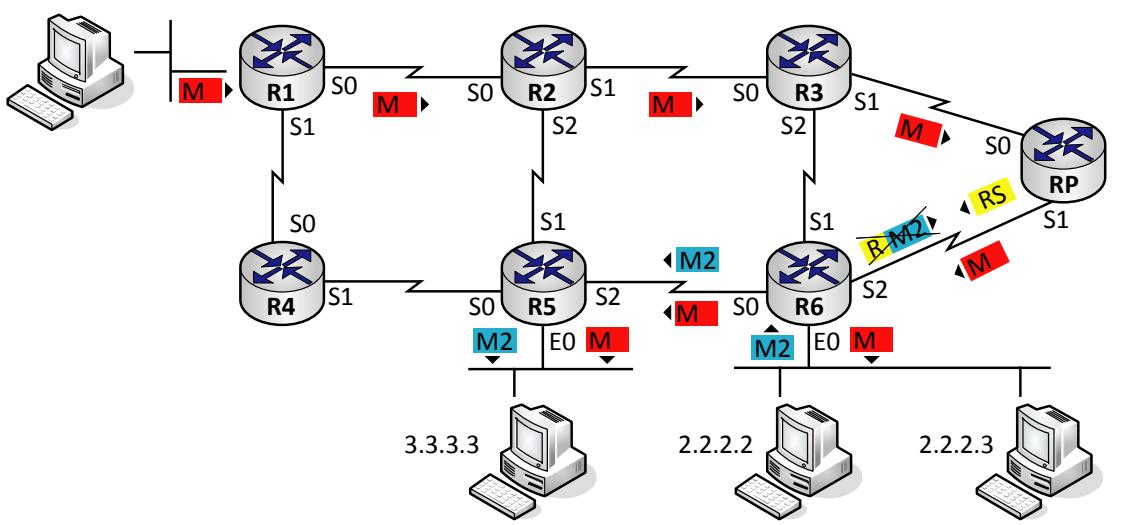
- La fuente de datos 1.1.1.1 empieza a emitir antes de que aparezca cualquier receptor al grupo G (224.2.2.2). El router R1, que conoce la dirección del RP, envía el tráfico multicast con el mensaje encapsulado, el mensaje de registro.
- En el momento que el primer ‘register’ llega al RP, registra la fuente de datos y analiza si tiene receptores para ese grupo multicast. Como no los tiene envía un mensaje de ‘register stop’ (RS) al router R1. Cuando R1 le llega el primer mensaje RS, deja de enviar mensaje encapsulado.
- Un host de R6 le interesa recibir tráfico del grupo G. R6 le envía un el correspondiente ‘join’ al RP. Y el RP le envía hacia R1 el ‘join’ y cuando lo recibe empieza a enviarle tráfico de forma nativa al RP, el cual lo distribuye al receptor.
- A continuación aparece un receptor del R5, esto provoca que R5 envíe un ‘join’ hacia el RP. Cuando en R6 recibe el ‘join’ de R5, R6 se encarga de no enviar el ‘join’ hacia al RP porque ya está recibiendo tráfico de la misma fuente.

El mecanismo de los mensajes ‘register’ y ‘register stop’ permite optimizar el tráfico, si no hay receptores el tráfico no viajará inútilmente desde la fuente hasta el RP. Otra ventaja que se observa es que el RP no tiene un conocimiento detallado de los receptores, en este ejemplo el router sabe que hay receptores en S1, pero no sabe ni cuantos ni quiénes son. Esto es importante de cara a la escalabilidad.

#### Escenario: Emisión de dos fuentes.

Origen S1 - 1.1.1.1

G=224.2.2.2



M2

Datagrama multicast (de S2)

R M2

Register msg encapsulado (de S2)

Las tablas de rutas es la misma que el escenario anterior, excepto en el RP y el R6.

RP	Entrada	Salida
(*,G)	S0	S1

Registros de emisor en el RP	
(S1,G)	S0
(S2,G)	S1

R6	Entrada	Salida
(*,G)	S2	E0, S0
(F2,G)	E0	S0

- Partimos de que ya existe la fuente de datos emitiendo tráfico y los dos receptores del grupo G recibiendo dicho tráfico. Un segundo emisor aparece en la LAN donde se encuentra uno de los receptores. El router R6 ya forma parte del árbol de distribución compartido.
  - Como el router R6 ya pertenece al grupo donde la fuente S2 empieza a emitir, enviará por la interface S0.
  - Además R6 enviará un mensaje de Register al RP, sin embargo el RP devolverá un Register Stop porque observa que no tiene receptores registrados para ese grupo por ninguna interface salvo S1, que es por la que ha llegado el propio Register.

## Escenario: Árbol SPT.

Origen S1 - 1.1.1.1

G=224.2.2.2

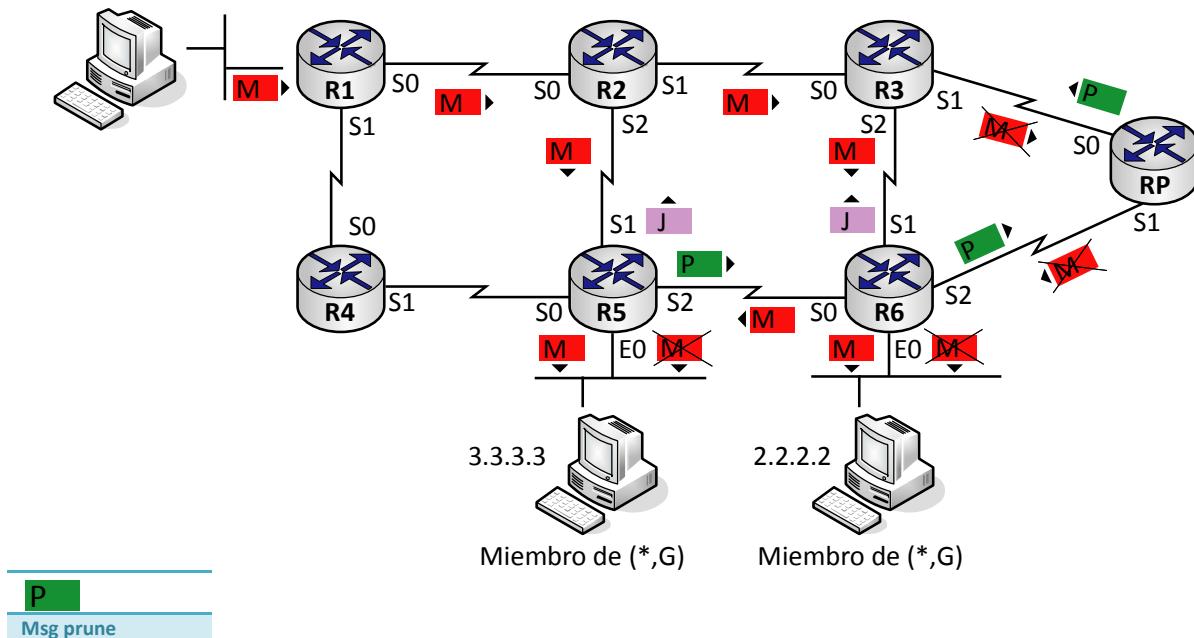


Tabla de ruta por router:

R1	Entrada	Salida	R2	Entrada	Salida
(S1,G)	E0	S0	(S1,G)	S0	S1,S2
Registros de emisor en el RP					
R3	Entrada	Salida			
(S1,G)	S0	S1, S2			
RP	Entrada	Salida	R5	Entrada	Salida
(F1,G)	S0	S1	(*,G)	S2, S1	E0
R6	Entrada	Salida			
(*,G)	S2, S1	E0, S0			

- En escenario parte de que existe un emisor y dos receptores para el grupo G.
- Existe un umbral configurado para el cambio al SPT, y en este escenario es superado por los routers R5 y R6. Los dos routers procederán al cambio.
- Cada receptor establece el SPT emitiendo un ‘join’ hacia la fuente.
- Cuando el receptor detecta que el tráfico está llegando por el camino SPT, emite un ‘prune’ hacia el RP para que deje de enviarle tráfico. En este caso puede recibir tráfico duplicado.
- En el momento que el ultimo receptor emite el ‘prune’ hacia el RP, este emite a su vez emite un ‘prune’ hacia la fuente S1. De esta forma existía el tráfico que va desde S1 al host de R5 y otro que va desde S1 al host de R6.

## **4. Herramienta de simulación y experimentación**

Las herramientas de simulación permiten tener simular una configuración topológica sin tener que realizar una configuración física que resulta más costosa. Por lo tanto se puede analizar y observar características en una configuración sin necesidad de acudir al sistema real.

Durante los últimos años se ha podido observar la aparición de servicios los cuales su consumo de recursos es muy alto, tanto a nivel de red como de las propias maquinas además del tiempo que conlleva en ponerlo en marcha. Esto puede provocar funcionamientos defectuosos debido a la baja capacidad de las redes y los propios equipos. Además la implementación de protocolos experimentales sería muy costosa y en algunos además puede resultar en vano.

Todo esto no lleva a la aparición de simuladores de red, los cuales nos ayudan a obtener resultados previos y así poder estipular una conclusión respecto a la configuración topológica.

### **4.1 Modelos y simulación**

La simulación es una técnica que permite imitar el comportamiento del sistema analizado conforme va evolucionando en el tiempo. Mediante una simulación se puede realizar observaciones de las características del sistema que se interesan analizar. Sobre este concepto surgen dos términos.

- **Modelo de simulación:** Es una representación simplificada de la realidad de la cual existen un conjunto de hipótesis acerca el funcionamiento del sistema y expresada como relaciones matemáticas y/o lógicas entre los elementos del sistema. El modelo hace referencia a la representación del sistema real estableciendo las condiciones de la simulación y las variables que se emplean. La finalizada es estudiar el sistema con más facilidad y deducir propiedades difíciles de estudiar en la realidad.
- **Proceso de simulación:** Es la ejecución del modelo de simulación, el cual va transcurriendo a través del tiempo y se van capturando todos los comportamientos y así recopilar los datos provenientes de la simulación. Mediante unos parámetros iniciales de simulación se esperan tener unos resultados y poder especificar el comportamiento del sistema.

Una simulación no se considera una técnica de optimización, es decir, no proporciona una maximización del rendimiento o prestaciones del mismo, sino que únicamente se limita a informar cual es el comportamiento del sistema a analizar en las condiciones iniciales hasta la finalización de la simulación.

Los resultados obtenidos en una simulación son datos establecidos por el simulador. El proceso se basa en el muestreo aleatorio, esto es, los resultados que se extraigan de ella, están sujetos a variaciones aleatorias y por ese motivo los resultados que se obtienen han de ser analizados. Por lo tanto estos resultados tienen que analizarse y comprobar que son fiables o no atienden las finalidades que prevean que tendría que realizar dicho sistema o si n definitiva no tienen validez alguna.

Cuando hablamos del término “sistema” se entiende como cualquier conjunto de elementos que actúan e interactúan de forma que logren un fin lógico. De esta forma si nos referimos al estado del sistema es el conjunto de variables que son necesarias para la descripción del estado de un sistema en un determinado instante del tiempo. Entre estas variables podemos distinguir las entradas y las salidas. Al referirse a las salidas son la finalidad de nuestro estudio, es decir los objetivos, expresados mediante valores numéricos. Por otra parte las entradas son los valores numéricos iniciales por los cuales parte la simulación. En las entradas se incluyen:

- Condiciones iniciales: Valores que expresan el estado del sistema al inicio de una simulación
- Datos determinísticos: valores conocidos necesarios para realizar los cálculos que producen las salidas.
- Datos probabilísticos: Cantidades cuyos valores son inciertos pero necesarios para obtener las salidas del sistema. Los valores específicos de estos datos deben conocerse a través de una distribución de probabilidad.

#### *4.2 Tipos de simulaciones.*

Según la característica se puede clasificar las diferentes simulaciones. En este apartado se mostraran algunas de estas posibles clasificaciones:

- Simulación estática: Es representación de un sistema en un instante de tiempo determinado o bien representa un sistema en el que el tiempo no es importante.
- Simulación dinámica: Es representación de un sistema cuando evoluciona con el tiempo y en el que el tiempo es una variable de interés.
- Simulación determinista: Es la representación de un sistema que no contiene absolutamente ninguna variable aleatoria y comportándose de acuerdo con una ley probabilística.
- Simulación aleatoria: Es la representación de un sistema que contendrá algún componente aleatorio de entrada.
- Simulación continua: Es la representación de un sistema donde el cambio de valores se basa directamente en los cambios de tiempo.
- Simulación Discreta: Es la representación de un sistema donde su comportamiento cambia únicamente en instantes de tiempo concretos, eventos. El estado de los cambios en los modelos sólo se dan cuando dichos eventos ocurren.

#### **4.3 OPNET**

Es un programa muy utilizado en la industria para el modelaje y simulación de sistemas de telecomunicaciones, es un lenguaje orientado a las comunicaciones. Permite realizar un diseño y estudiar redes, dispositivos, protocolos y aplicaciones vinculada a su gran escalabilidad y flexibilidad, además del acceso al código fuente. OPNET es una potente herramienta con la que analizar, por medio de simulaciones de eventos discretos, el comportamiento de redes de comunicaciones.

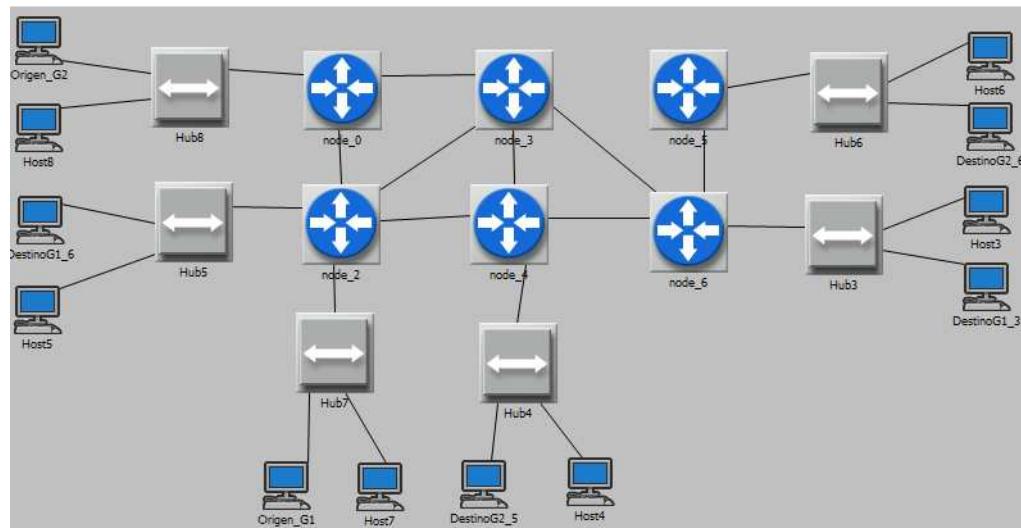
Es un simulador orientado a objetos por lo tanto interactuar al usuario sin problemas y ofrece una gran facilidad de interpretación y creación de escenarios además de tener en cada objeto una serie de atributos para la configuración del modelo a simular.

Además permite entre otras cosas realizar una animación en 2d de los nodos que existan en la red, editar y compilar el código fuente de los protocolos existentes, contenidos en diferentes librerías. Se puede definir el tipo de tráfico o comportamiento que debe de tener, cargar la red debido a los

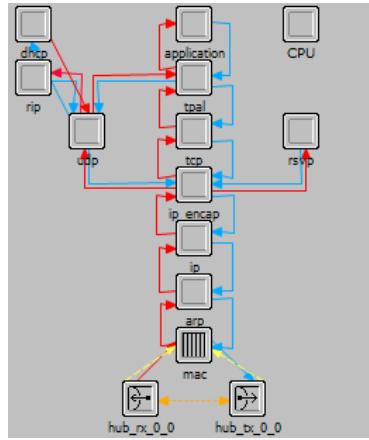
servicios. Y como no configurar servicios como HTTP, correo, VoIP, Video Streaming, Bases de datos, etc.

Está basada en la teoría de redes de colas e incorpora las librerías para facilitar el modelado de las topologías de red. El desarrollo de los modelos se realiza mediante la conexión de diferentes tipos de nodos, utilizando diferentes tipos de enlaces. En un escenario de simulación se deben especificar tres tipos de modelos jerarquizados, definidos a continuación:

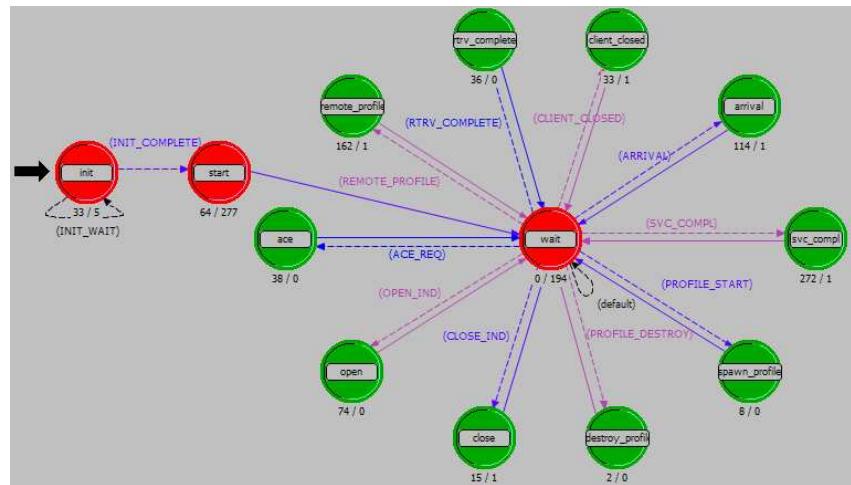
- **Modelo de red:** Aquí irán definidas las redes y subredes que hayan en la simulación. Es decir aquí se representan la topología de la red de comunicaciones y con sus enlaces correspondientes. En esta parte interviene la parte física de la modelación, donde existirán gran cantidad de condiciones a proporcionar. Como por ejemplo las distancias geográficas, posición de satélites, obstáculos que pueden alterar el funcionamiento de los nodos inalámbricos, Etc. Tiene el siguiente aspecto.



- **Modelo de nodo:** Se define la estructura interna de los modelos de red. Para ver los nodos de un modelo de red, se hace realizando doble-clic sobre cualquier nodo. Se mostrará una estructura de elementos funcionales de los que está compuesto. Cada uno es el denominado modulo. Como se puede observar en la siguiente figura, representa los módulos dentro de un host:



- **Modelo de procesos:** Son los estados que pueden existir dentro cada nodo. Se definen protocolos, algoritmos, aplicaciones, etc. representándolo mediante máquinas de estados finitos (FSM Finite State Machine). La FSM está compuesta normalmente por estados forzados y no forzados. Los estados forzados son estados en los que el proceso está bloqueado esperando a que un nuevo evento se genere para cambiar a un nuevo estado. Las transiciones entre estados pueden ser condicionales o incondicionales. A cada estado se le puede asignar una serie de código que se ejecutará al entrar o al salir del estado. El funcionamiento interno tanto de estados como de transiciones implica la programación en lenguaje C/C++. En caso de utilizar modelos de nodos estándar no será necesaria la programación de procesos, ya estarán definidos en las librerías.

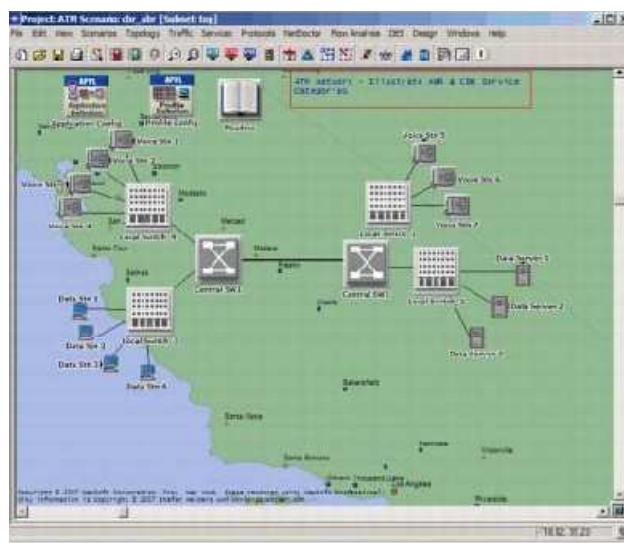


#### 4.3.1 Modelar en OPNET.

Una vez que existe un diseño de la implementación topológica, se realiza el modelo mediante editores que proporciona OPNET. Cada editor está destinado para una tarea específica:

## Project Editor

Es la parte principal del simulador, donde se crea la simulación de red. Se utilizan librerías que contienen objetos estándar para la colocación en el proyecto. Estos objetos pueden ser routers, switches, conexiones, definición de paquetes de datos, etc. Y en cada objeto tiene sus propiedades las cuales se modifica según el modelo que se quiere representar. Al definir un nuevo proyecto, le indicamos el tipo de escenario con el que queremos trabajar (la topología base), hasta el fondo como por ejemplo un mapamundi. Dentro del mismo proyecto se podrán tener varios escenarios con distintas configuraciones, por ejemplo para comparar los resultados obtenidos en cada uno de ellos. El modelo de proyecto será por tanto el nivel donde se especificará el sistema que será simulado y es donde se podrá obtener los resultados de la tipología que se está estudiando.

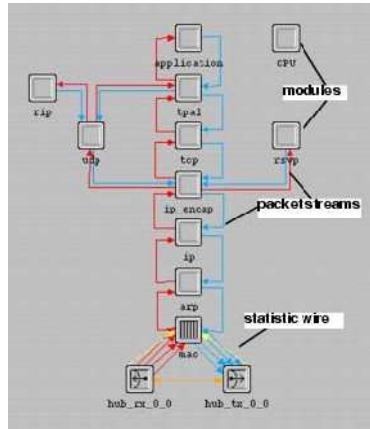


## Node editor

Permite editar el modelo a nivel de nodo por cada elemento de la red. Por cada nodo (o cada objeto) mediante el node editor se ven el comportamiento de ese nodo, representándolo con unas cajas conectadas entre ellas e interactúan entre sí realizando una función en el nodo. Las cajas que se pueden encontrar son:

- Procesadores: El comportamiento viene especificado por el editor de procesos. Existen modelos ya configurados como la fuente de datos.
  - Colas: Con distintos atributos para definir su comportamiento.
  - Transmisores y receptores: La entrada y salida de paquetes al nodo.
  - Stream de paquetes: Transporta el flujo de paquetes entre cajas.
  - Statistics Wire: Flujo de las interrupciones.

- Cable de asociación lógica transmisor-receptor: Se utiliza para crear un vínculo entre transmisores y receptores de un mismo elemento. Un ejemplo de utilidad de esta asociación es la necesidad de identificar en un hub el transmisor y receptor de un enlace, dado que en un hub se tienen varios dispositivos conectados.



Mediante este editor se puede representar aplicaciones, protocolos, algoritmos o recursos como: buffers, puertos y buses, entre otros; es decir que todo nodo es asignado a un proceso o evento dentro de la simulación, y esto se logra, mediante el editor de procesos.

### **Process editor**

Edita máquinas de estados finitos para modelar el comportamiento que se desee. Esta herramienta soporta las especificaciones, detalles, protocolos, recursos y aplicaciones que se desean configurar en la red a modelar gráficamente con los editores de proyecto y de nodos respectivamente. Los objetos más importantes presentes en este editor:

- Estados: cada uno de ellos representa un estado del proceso. Se definen en él las funciones a realizar durante su ejecución.
- Transiciones: marcan la condición que se necesita para pasar de un estado a otro.
- Bloques: sirven para la programación, esto es, declaración de variables.

#### *4.3.2 Salidas del simulador y análisis de resultados.*

El entorno de simulación OPNET utiliza la tecnología DES (Discrete event simulation). Cuando se realiza una simulación DES se utilizan eventos para describir sucesos o acciones mientras dura la simulación. Por cada uno de estos eventos tiene un instante de incidencia puntual en la escala del tiempo, esta escala, al igual que el resto de magnitudes, es discreta.

Cuando se realiza la simulación, existen dos tipos de contadores que distinguen el tiempo real transcurrido y el tiempo que lleva la simulación. Además de utilizar varias variables de estado que

representan la fase del sistema simulado. Si hablamos de sistema, va evolucionando en la memoria del ordenador, produciéndose diferentes eventos que se construyen en variables de estado, y estas a su vez determinan los futuros eventos.

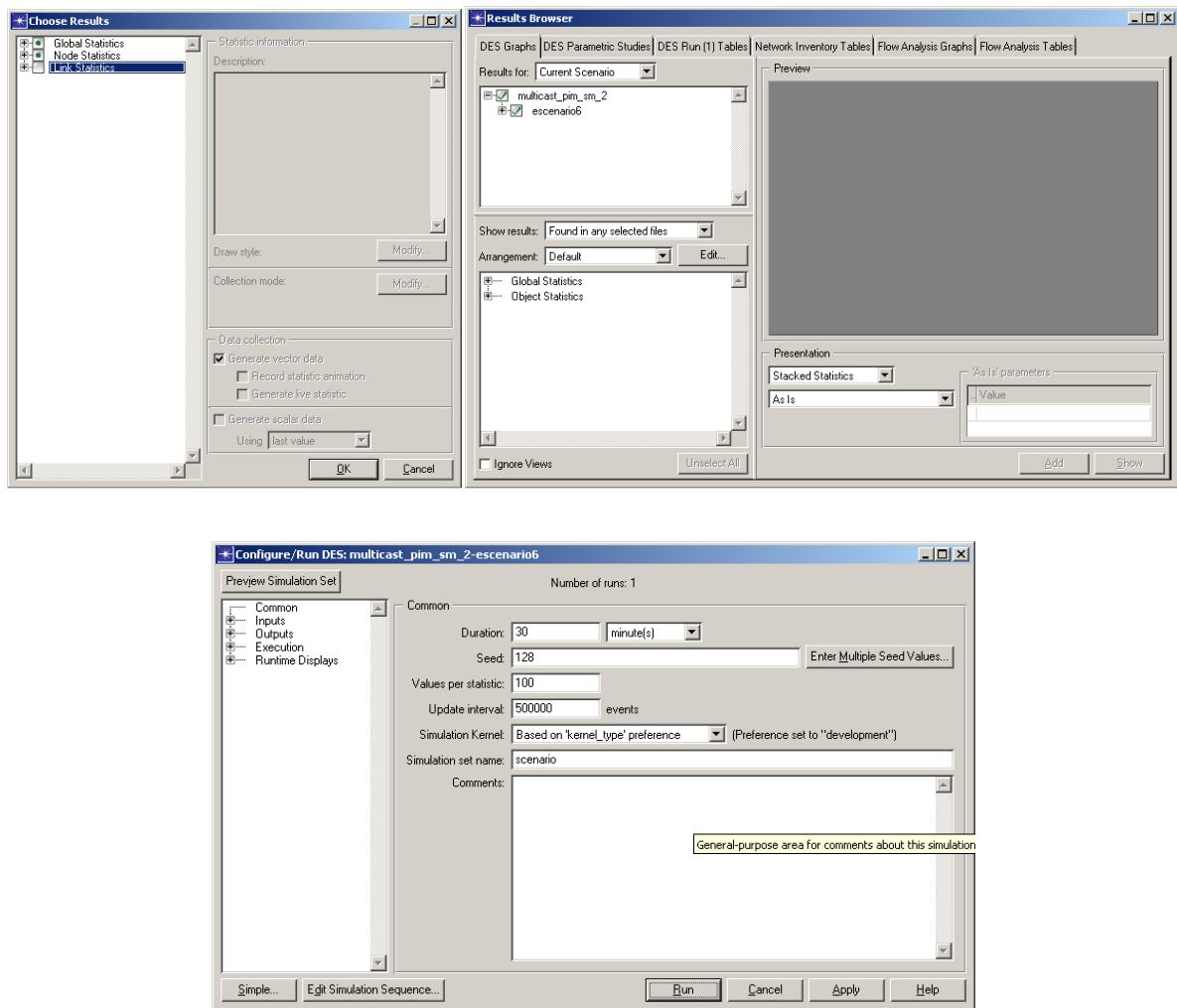
Por cada evento que “salta” tiene un instante de incidencia puntual, es decir representa sobre la escala temporal discreta de la simulación una única posición de tiempo. Por lo tanto los eventos se logran ser ordenados cronológicamente, según su instante de incidencia, para ser procesados. Por cada evento se describe una acción, y el resultado de esta modifica las variables de estado. Esta característica esta soportada por los lenguajes de OOP (programación orientada a objetos).

En el momento que se realiza la simulación, OPNET tiene de forma predeterminada estadísticas que se puede realizar en el momento de la simulación. En la propia escena si se presiona botón derecho del ratón y se selecciona el menú contextual “Choose Individual Statistics”, mostrará todas las estadísticas que tiene por defecto. En este punto se seleccionan las estadísticas. Están clasificadas de tipo nodo, globales o de interconexión. Según el estudio que se quiere realizar se seleccionaran unas u otras. Hay que tener cuidado a la hora de seleccionar la estadísticas, por ejemplo, si se selecciona la media del retardo global no será el mismo resultado que si seleccionar la media del retardo por nodo.

En definitiva, los pasos fundamentales que siguen a la hora de realizar la simulación de eventos discretos con OPNET son:

- 1- Se crea un proyecto nuevo con la configuración a realizar. Si el modelo de red a estudiar está compuesto de elementos existentes (routers, hosts, etc.) se escogerán del paleta de objetos, por el contrario se utilizarán los editores que correspondan en la definición estos elementos (nodos).
- 2- Si el estudio lo requiere, será necesario crear una estadística en el momento que se realice la simulación. Como hemos comentado existen configuraciones por defecto, pero no siempre es del todo útil si se tiene que profundizar en algún aspecto del modelo. En ese caso se puede modificar el código para crear estadísticas nuevas e incluirlas en la simulación.
- 3- En el momento que ya esté creada, con los nodos, enlaces y sus configuraciones se escogerán las estadísticas desde el menú “Choose Individual Statistics”.

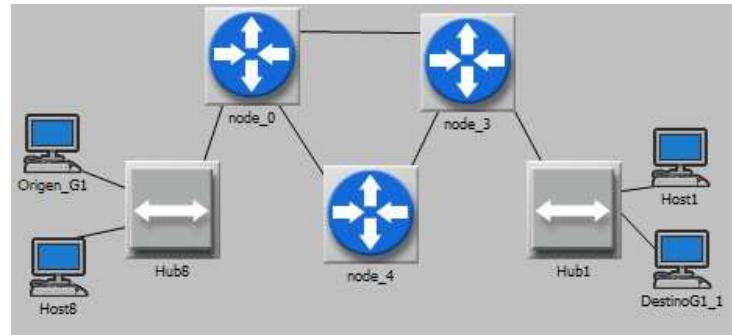
- 4- A partir de ahora se editan los parámetros de la simulación. Desde el menú “DES” se selecciona “Configure Discrete Simulation”. Mostrara un dialogo de configuración antes de correr la simulación. Y botón para iniciar la simulación es “Run”.
- 5- Una vez finalizada la simulación, se observan los resultados. Para ello hay que ir a través del menú desplegable del escenario y clicando en “View Results”. Se mostrara una ventana donde se mostrarán los resultados de las estadísticas previamente seleccionadas a la simulación.



#### 4.4 Configuración del protocolo PIM-SM.

A continuación describiremos como se manipula OPNET para configurar una red con el protocolo PIM-SM. Los pasos son simples y automatizados, pero hay que tener mucho cuidado, porque una equivocación puede repercutir en los resultados finales de la simulación. Para que resulte más fácil la explicación seguiremos el ejemplo de la implementación de una red pequeña.

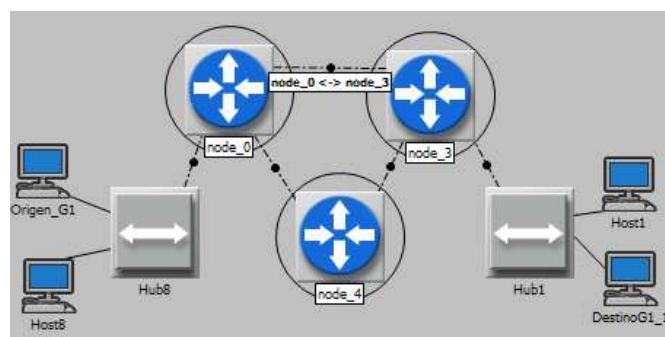
En el primer paso se debe insertar los routers y las estaciones de trabajo tanto los receptores y los emisores. Además se debe de insertar los links entre estos. Para el ejemplo, se ha preparado dos hubs, tres routers y cuatro estaciones de trabajo, y la conexión entre ellas es como se muestra en la siguiente figura.



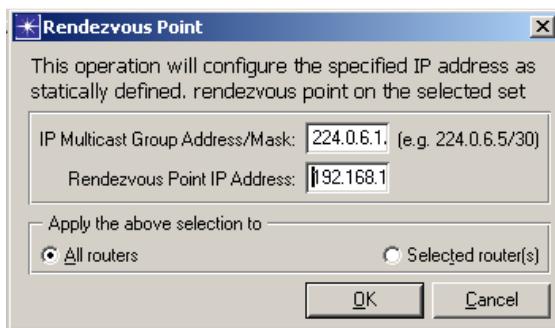
Los routers son del modelo “bn\_bcn\_13s\_a2\_e8\_f3\_fe4\_sl13\_tr6”, se puede realizar una búsqueda por la paleta de objetos. El hub son del modelo “ethernet4\_hub\_adv”, y las estaciones “Ethernet\_wkstn\_adv”, si se editan los atributos todos los modelos tienen algún apartado específico para el multicast. Los links son del modelo “100BaseT\_int”. Como se puede observar se va a configurar un origen de datos y un único receptor.

Ahora se tiene que establecer todos los nodos destinos como multicast, para ello se debe de seleccionar el nodo destino llamado “DestinoG1\_1” e ir al menú “Protocols->IP Multicast” y seleccionar la opción “Enable Multicasting on Selected End Nodes”.

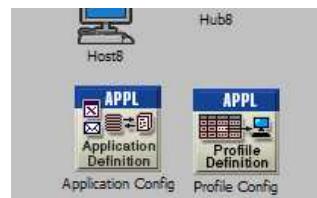
A continuación se debe de configurar todos los routers para establecer todos los links (interfaces) que intervienen en la red multicast. Para ello se tiene que seleccionar tanto los links como los routers e ir al menú “Protocols->IP Multicast->Enabled multicasting on interfaces across selected link”.



El siguiente paso es la configuración de un grupo multicast. Como se ha comentado en apartados anteriores, utilizamos una IP de clase D, como por ejemplo 224.0.6.1, además hay conocer el RP, así que le asignaremos el valor 192.168.1.1. Para dicha configuración se tiene que ir al menú “Protocols -> IP -> Multicast -> Configure Rendezvous Point Using Static RP Configuration”



A partir de ahora la red que hemos configurado ya puede emitir hacia un grupo multicast, y recibir dicha emisión. Pero es necesario saber cuál va a ser la aplicación que vamos a simular, y que tipos de datos van a intervenir en la simulación. Para ello se va a introducir dos nodos de configuración en la red. Estos nodos no tienen ninguna relación física en la red, solo realizan la tarea de configuración de cuál va a ser la aplicación y como serán los datos tanto de la emisión como la recepción. En la siguiente imagen se puede ver los nodos introducidos.



- Application Config: En este nodo se configuran las aplicaciones que intervienen en la simulación. Se definirá una aplicación entrando en los atributos del nodo. Se añadirá una aplicación en la propiedad “Aplication Definition” con el nombre de “Video”. Editaremos de nuevo esta nueva propiedad y buscaremos la propiedad “Video Conferencing”. A continuación se establecerá la opción “High resolution video”, que es una configuración por defecto de alta resolución de datos de video, es decir los datos a transferir serán de un gran peso.
- Profile Config: Establece como se va a comportar las aplicaciones configuradas. Para ello se tiene que entrar en las propiedades del nodo y añadir una fila en las propiedades “Profile configuration”. En la columna “Profile name”, se le indicara un nombre alias,

como puede ser “Video profile”. La otra columna que hay que tener en cuenta es “Applications”, en esta se indicará el nombre de la aplicación que se ha configurado anteriormente, es decir “Video”.

Una vez configurado las aplicaciones y perfiles, se aplicarán en los nodos destinos y los orígenes. Primero se configurarán los nodos origen. Se tiene que editar los atributos del nodo “Origen\_G1” y en la rama de “Applications” existe una opción llamada “Application: Supported Profiles”, donde se editará y se establecerá como profile la configurada anteriormente llamada “Video profile”. Existe otro atributo llamado “Application: Destination Preferences”, se editará y se configurarán el tipo de aplicación destino, es decir “Video” y en la columna “Actual Name” se introducirá la IP “224.0.6.1”. En este momento ya está el emisor preparado. Ahora se configurará el receptor, es decir, en qué momento el receptor se unirá a la red. Para ello se tiene que editar el nodo “Destino\_G1” y editar el atributo “Applications” → “Application: Multicasting Specification”, se inserta una fila en la tabla y se modificará la columna “Application Name” con el valor “Video” y la columna “Membership Addresses” se establecerá el valor 224.0.6.1. Por último se tiene que configurar la aplicación, en la rama de los atributos “Applications” → “Applications: Supported Services” se establecerá el nombre de la aplicación que hemos creado, llamada “Video”.

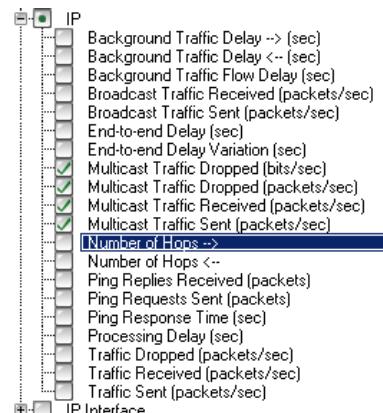
La red ya está configurada, y se han realizado todos los pasos necesarios para que permita la comunicación Multicast entre los nodos. Ahora es el momento de decidir cuáles son las estadísticas que se van a mostrar como resultados de la simulación. En la experimentación del proyecto se centra en los mensajes de control. Pero esto se describe en el siguiente apartado.

#### *4.5 Estadísticas personalizadas para la experimentación.*

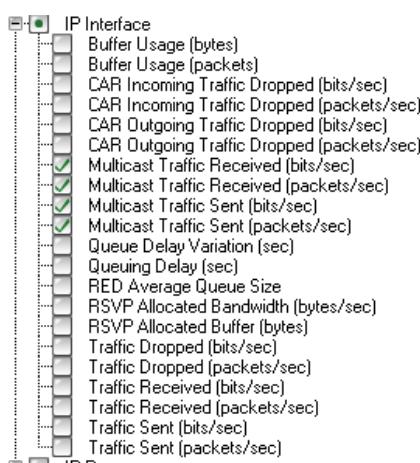
A continuación mostraremos la forma de modificar algunas estadísticas para el estudio de nuestra experimentación. Las estadísticas ya preestablecidas en OPNET, se muestran mediante el menú “DES→Choose Individual Statistics”. Se pueden encontrar clasificadas en función de en qué grado afecta a la red, es decir a nivel global, a nivel de nodo o a nivel de link, y según el tipo de protocolo/aplicación que se quiera estudiar. Las relacionadas con PIM-SM, que se pueden encontrar son las siguientes imágenes:



En la opción de “Global Statistics”



En la rama de “Node Statistics”



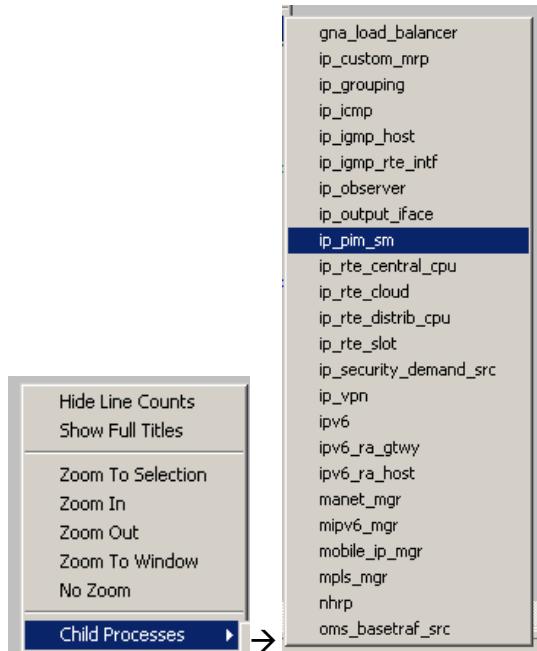
En la rama de “Node Statistics”



En la rama de “Node Statistics”

En la opción de PIM-SM, se muestran estadísticas sobre los mensajes de control y el registro de mensajes, tanto de recepción como de emisión. En todos los casos muestra bits por segundo o paquetes por segundo. Pero, nos hace falta una modificación de las estadísticas para conocer de los datos de control de forma desglosada según su tipo. Estos pueden ser el “Hello”, “Join/Prunte”, “Register” y “Register-Stop”. Cada uno se ha descrito en apartados anteriores.

La modificación será sobre las “Global Statistics”. Se seleccionara cualquier nodo router, y se clicará dos veces en el icono del router. Ahora aparecerán todos los procesos que intervienen dentro del router, y se puede apreciar en algunos casos el nombre de sus protocolos. Todos pasan por el proceso IP, este se tiene que editar con doble clic. Dentro de este se verá un diagrama de estados el cual es el definido en el protocolo IP. Ahora se tiene que buscar el diagrama de estado del protocolo pim\_sm, para ello se presionará botón derecho y se seleccionará la siguiente opción:



El diagrama de estados que se muestra es el referente al PIM-SM. Una vez aquí se tiene que añadir las estadísticas globales, para ello se tiene que ir al menú “Interfaces” → “Global Statistics”. En la tabla que se muestra a continuación, se añadirán las estadísticas que deseemos dar alta, en este caso son:

Stat Name	Mode	Count	Description	Group	Capture M...	Draw Style	Low Bound	High Bound
Network Convergence Activity	Dimensioned	20	Records a square wave alternating between...	PIM-SM	normal	square-wave	0.0	1.0
Network Convergence Duration (sec)	Dimensioned	20	Records the duration of convergence cycle...	PIM-SM	normal	discrete	0.0	disabled
PIM-SM Control Traffic Received (bits/sec)	Single	N/A	Number of PIM-SM control messages in	PIM-SM	bucket/de...	linear	0.0	disabled
PIM-SM Control Traffic Received (packets/sec)	Single	N/A	Number of PIM-SM control messages in	PIM-SM	bucket/de...	linear	0.0	disabled
PIM-SM Control Traffic Sent (bits/sec)	Single	N/A	Number of PIM-SM control messages in	PIM-SM	bucket/de...	linear	0.0	disabled
PIM-SM Control Traffic Sent (packets/sec)	Single	N/A	Number of PIM-SM control messages in	PIM-SM	bucket/de...	linear	0.0	disabled
PIM-SM Register Messages Received (bits/sec)	Single	N/A	Number of PIM-SM Register messages in	PIM-SM	bucket/de...	linear	0.0	disabled
PIM-SM Register Messages Received (packets/sec)	Single	N/A	Number of PIM-SM Register messages in	PIM-SM	bucket/de...	linear	0.0	disabled
PIM-SM Register Messages Sent (bits/sec)	Single	N/A	Number of PIM-SM Register messages in	PIM-SM	bucket/de...	linear	0.0	disabled
PIM-SM Register Messages Sent (packets/sec)	Single	N/A	Number of PIM-SM Register messages in	PIM-SM	bucket/de...	linear	0.0	disabled
PIM-SM MSJ HELLO	Single	N/A	Mensajes de control Hello	PIM-SM	normal	linear	0.0	disabled
PIM-SM MSJ JOIN-PRUNE	Single	N/A	Mensajes de control Join/Prune	PIM-SM	normal	linear	0.0	disabled
PIM-SM MSJ REG	Single	N/A	Mensajes de control Register	PIM-SM	normal	linear	0.0	disabled
PIM-SM MSJ REG-STOP	Single	N/A	Mensajes de control Register-Stop	PIM-SM	normal	linear	0.0	disabled

Como se puede observar, las que se han dado de alta son las que están señaladas en rojo. Ahora llega el momento de modificar el código. Para ello se tiene que ir al menú “Code Blocks” → “Header Blocks”, esta contiene las definiciones en la cabecera. Se busca el apartado de estadísticas globales y se añade:

```
/* Global PIM Statistics. */
Stathandle Gbl_Num_Pim_Sm_Sent_Bps_Hndl;
Stathandle Gbl_Num_Pim_Sm_Sent_Pps_Hndl;
Stathandle Gbl_Num_Pim_Sm_Recv_Bps_Hndl;
Stathandle Gbl_Num_Pim_Sm_Recv_Pps_Hndl;
Stathandle Gbl_Num_Register_Sent_Bps_Hndl;
Stathandle Gbl_Num_Register_Sent_Pps_Hndl;
Stathandle Gbl_Num_Register_Recv_Bps_Hndl;
```

```

Stathandle Gbl_Num_Register_Recv_Pps_Hndl;
Stathandle Gbl_Num_Hello; /*contador de mensajes hello*/
Stathandle Gbl_Num_Join_Prune; /* contador de mensajes join prune*/
Stathandle Gbl_Num_Register; /* contador de mensajes register */
Stathandle Gbl_Num_Register_Stop; /* contador de mensajes register-stop */

```

A continuación se tiene que inicializar las variables, para ello guardamos los cambios hechos en la cabecera y retornamos al menú “Code Blocks” → “Function Blocks”. Se mostrarán todas las definiciones de función que se llevan a cabo en el protocolo, se tiene que buscar especialmente una llamada “ip\_pim\_sm\_sv\_init”. Es una parte del código de la función se registran las estadísticas globales, en la cual se introducirá el siguiente código:

```

Gbl_Num_Hello = op_stat_reg ("PIM-SM.PIM-SM MSJ HELLO",OPC_STAT_INDEX_NONE,
OPC_STAT_GLOBAL);

Gbl_Num_Join_Prune = op_stat_reg ("PIM-SM.PIM-SM MSJ JOIN-
PRUNE",OPC_STAT_INDEX_NONE, OPC_STAT_GLOBAL);

Gbl_Num_Register = op_stat_reg ("PIM-SM.PIM-SM MSJ REG",OPC_STAT_INDEX_NONE,
OPC_STAT_GLOBAL);

Gbl_Num_Register_Stop = op_stat_reg ("PIM-SM.PIM-SM MSJ REG-
STOP",OPC_STAT_INDEX_NONE, OPC_STAT_GLOBAL);

```

Una vez ya registrada las variables, solo añadir datos que se reflejarán en las graficas. En este caso registraremos un 1 por cada mensaje de control. Para cada tipo de mensajes hay que buscar cual es la función donde realiza en envío del mensaje de control. En el caso del “Hello” se busca una función llamada “ip\_pim\_sm\_send\_hello\_msg”. Al final de la función realiza el envío mediante otra función llamada “ip\_pim\_sm\_send\_pkt”. Inmediatamente antes estableceremos con un 1 en el contador de la estadística “Gbl\_Num\_Hello” mediante la siguiente función:

```
op_stat_write (Gbl_Num_Hello, 1.0);
```

Esta sentencia realiza una llamada una función nativa de OPNET, en este caso escribe en la estadística un 1 y cada vez que pase por aquí escribirá un uno en la estadística, por lo tanto se podrán contabilizar los mensajes “hello” que se envían.

Para cada tipo de mensaje se deberá realizar la tarea descrita, pero a su global correspondiente. En el caso de join/prune, se tiene que buscar la función ip\_pim\_sm\_create\_and\_send\_join\_prune\_msg, y se deberá poner antes de la llamada ip\_pim\_sm\_send\_pkt. Para el tipo register su busca la función ip\_pim\_sm\_send\_register\_msg y se añade antes de llamar a ip\_pim\_sm\_send\_pkt, y para register-

stop se tiene que buscar ip\_pim\_sm\_send\_register\_stop\_msg y se añade antes de llamar a ip\_pim\_sm\_send\_pkt.

Las estadísticas creadas hasta ahora son referentes al envío de mensajes de control. También son necesarias las referentes a la recepción de mensajes de control, que para ello el proceso el mismo que el explicado pero hay que conocer el lugar donde recibe los paquetes protocolo. La función donde se tienen que escribir las estadísticas es ip\_pim\_sm\_transition\_code\_get, y la referencia es el switch case para determinar el siguiente estado de transición:

```

switch (pim_sm_pkt_type)
{
    case IpC_Pim_Sm_Hello_Msg:
        {
            /* A Hello message has been received */
            trans_code = IpC_Pim_Sm_Hello_Msg_Recv;
            op_stat_write (Gbl_Num_Hello_rcv, 1.0);
            break;
        }
    case IpC_Pim_Sm_Join_Prune_Msg:
        {
            /* A Join/Prune message has been received */
            _code = IpC_Pim_Sm_Join_Prune_Msg_Recv;
            op_stat_write (Gbl_Num_Join_Prune_rcv, 1.0);
            break;
        }

    case IpC_Pim_Sm_Register_Msg:
        {
            /* A Register message has been received */
            trans_code = IpC_Pim_Sm_Register_Msg_Recv;
            op_stat_write (Gbl_Num_Register_rcv, 1.0);
            break;
        }

    case IpC_Pim_Sm_Register_Stop_Msg:
        {
            /* A register-Stop message has been received */
            trans_code = IpC_Pim_Sm_Register_Stop_Msg_Recv;
            op_stat_write (Gbl_Num_Register_Stop_rcv, 1.0);
            break;
        }

    default:

```

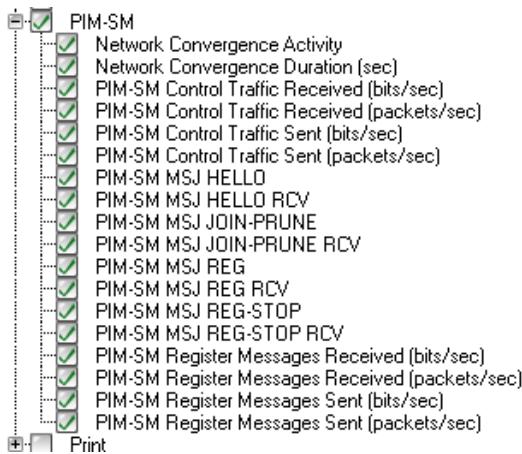
```

    {
        /* An invalid control message has been received */
        trans_code = IpC_Pim_Sm_Invalid;

        break;
    }
}

```

El resultado final tiene el siguiente aspecto:



#### 4.6 Metas de la experimentación

El estudio que se va a realizar es respecto a los mensajes de control que intervienen en el flujo multicast. Como hemos comentando en el marco teórico existen varios mensajes de control que utiliza el protocolo. Los mensajes de control que se quieren diferenciar son “Hello”, “Join/Leave”, “Register” y “Register-Stop”. Estos utilizan una parte del flujo total de datos que se envían o que se recibe. Se trata de saber cuál la congestión de estos mensajes respecto al tráfico total. Para ello se ha planteado varios escenarios, y en los cuales se va incrementando el número de nodos clientes y el número de grupos multicast, y por lo tanto orígenes de datos.

Además este estudio realiza una ejecución de fallo y recuperaciones de nodos. Los nodos que fallan y se recuperan pueden ser de tipo hub o de tipo router. El fallo y la recuperación de los nodos establecen una similitud a la realidad ya que se rigen en una función de distribución y determina el tiempo del fallo y la recuperación del nodo.

Otra característica de la red a simular son los nodos receptores de datos. Por cada hub que sea de tipo cliente, se generarán destinos de datos según una función de distribución. Esto ayuda a aproximarse a la realidad, ya que las estaciones aparecen de forma aleatorias. Por cada estación de

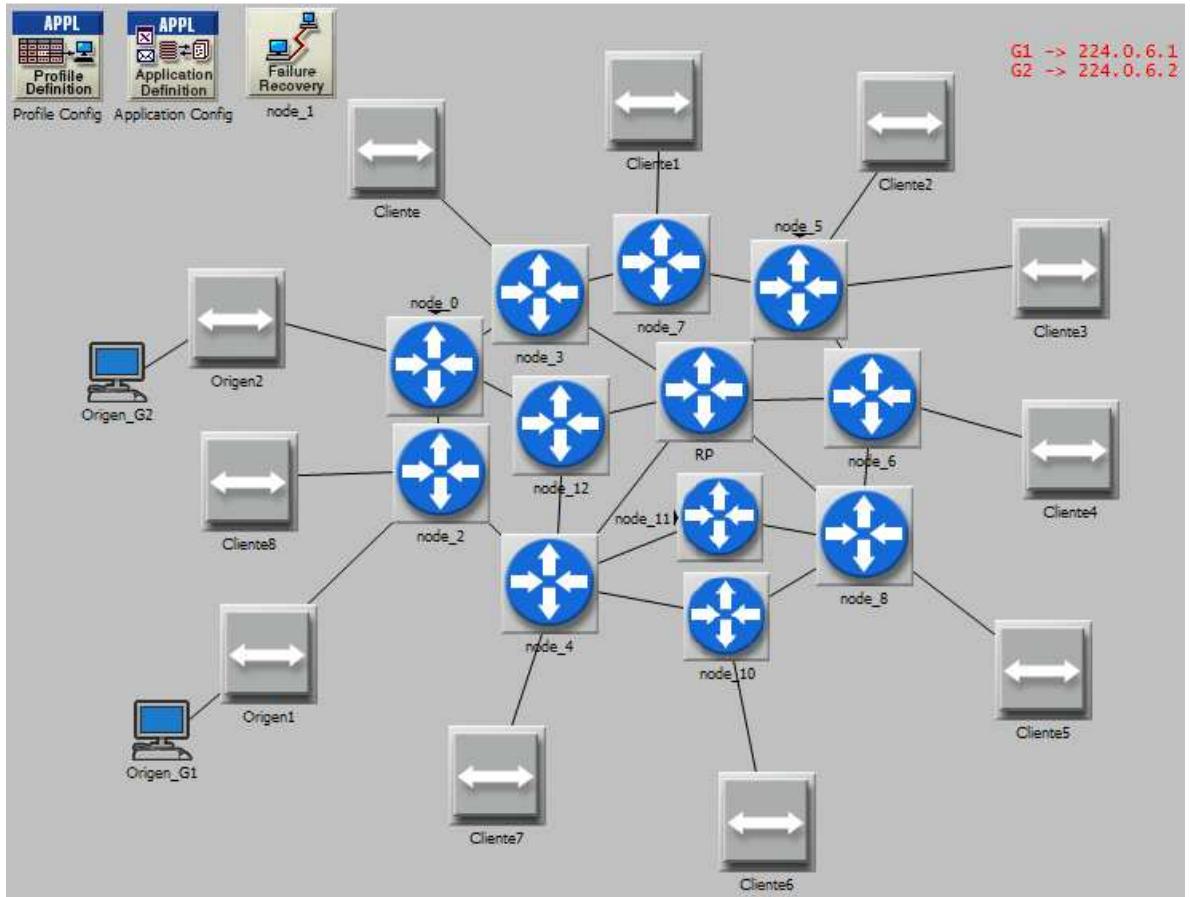
trabajo, realiza continuamente una unión y desunión del grupo multicast en según a una función de distribución.

Aprovecharemos la infraestructura para conocer un dato más que puede resultar útil, que es el retardo medio de los nodos clientes a nivel de aplicación. Esta dato en definitiva lo que nos indica cual es retraso que lleva esa carga de datos en toda la red, de un extremo que es el cliente al otro extremo que es el emisor.

En resumen los escenarios que se plantean están diseñados principalmente para la captura de los datos de control del protocolo PIM-SM, las cuales intentan semejarse a la realidad.

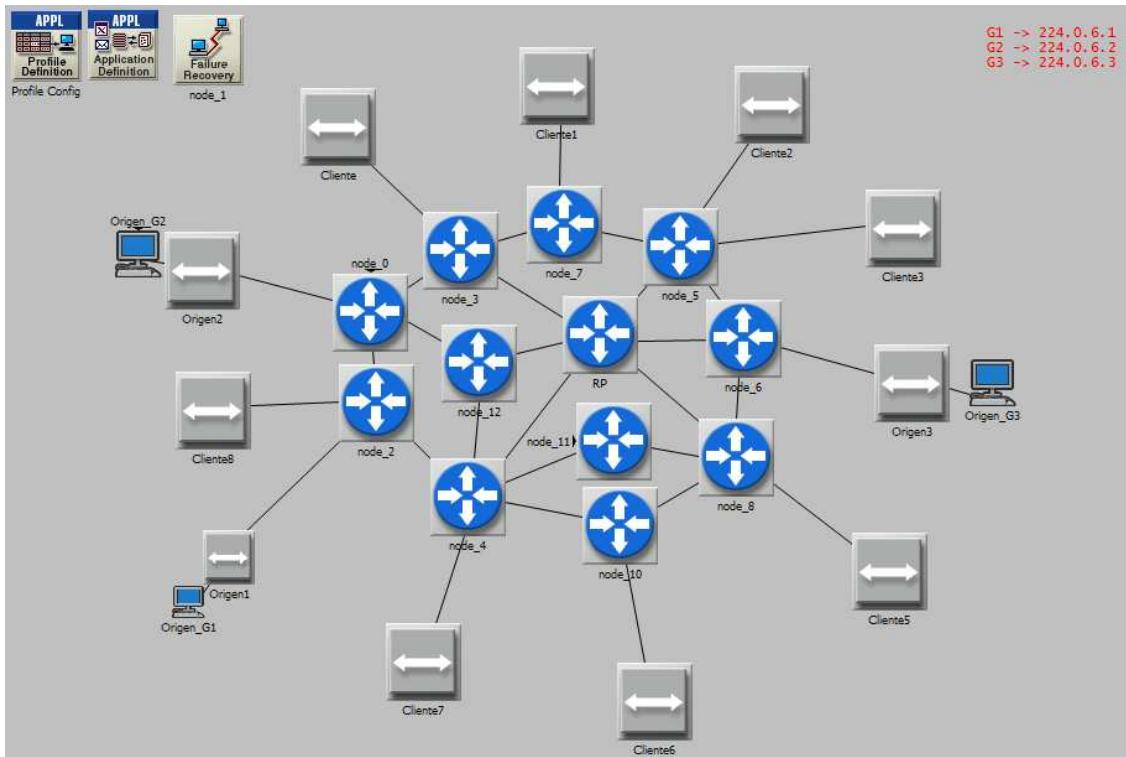
#### 4.7 Modelo base

El modelo base que proponemos se puede ver en la siguiente figura.



Este primer modelo tiene 9 hubs que son de clientes, donde tendrán las estaciones de trabajo receptoras. Los orígenes están situados en la parte de la izquierda, el Origen\_G1 y Origen\_G2, y el RP es estático y está situado en el centro. Los routers y los links que están alrededor del RP están configurados con el RP estático y con permiso para los dos grupos multicast que se han definido, estos son 224.0.6.1 y 224.0.6.2.

El otro modelo que proponemos es el mismo que el anterior pero con grupo multicast más.



El nuevo origen de datos está situado a la derecha de la red. En el momento que aprovechamos un hub para situar un origen de datos, se reduce el número de hub por lo tanto ahora hay 8 hubs con estaciones de trabajo receptoras. El grupo multicast que se agrega es 224.0.6.3.

## 5. Desarrollo de la aplicación GenNode

### 5.1 Lectura de los archivos XML

La lectura de los metatags del XML tiene que ser compatible con el uso de OPNET. Cuando exportar un modelo de red a XML tiene la siguiente estructura.

```
<network>
    <subnet>
        <node>
            </attr>
            ... lista de atributos del nodo
        </node>
        ... lista de nodos
        <link>
            </attr>
            ... lista de atributos del link
        </link>
        ... lista de links
        <attr>
            ... lista de atributos de subnet
    </subnet>
    ... lista de subnets
    <attr>
        ... lista de atributos de network
</network>
```

Descripción de cada metatag:

- Network: Representa la red completa. Contiene las subredes que intervienen, atributos, características y propiedades de la vista (como se visualiza en OPNET). La red de trabajo puede tener varias subnet y varios atributos.
- Subnet: Es la representación de la sub-red. Contiene los nodos y las conexiones (links) entre los nodos. Además tiene una lista de atributos de la sub-red.
- Node: El nodo representa una entidad dentro de la subred. Esta entidad puede ser un router, un hub, una estación de trabajo, por ejemplo. Pero además no es necesario que

sea un elemento físico, puede ser un elemento de describa determinado comportamiento en la simulación. Un ejemplo puede ser el nodo “Fallo/Recuperación”, en el cual se define que nodos o links van a fallar o recuperar durante la simulación. Los nodos están compuestos por varios atributos.

- Links: Son los links tienen un origen y un destino, por ejemplo de un router a un hub. Y estos tienen que estar definidos en el modelo. Los links están compuestos por varios atributos.
- Atributos: Definen la característica para ese elemento. Hay que tener en cuenta que los atributos se definen de una forma determinada para la lectura del atributo en OPNET, por ejemplo:

```
<attr name="x position" value="872"/>
```

Este atributo define la posición con el valor indicado. Hay otros atributos que definen que un array como por ejemplo:

```
<attr name="Application Definitions.count" value="17"/>
<attr name="Application Definitions [0].Name" value="Database Access (Heavy)"/>
```

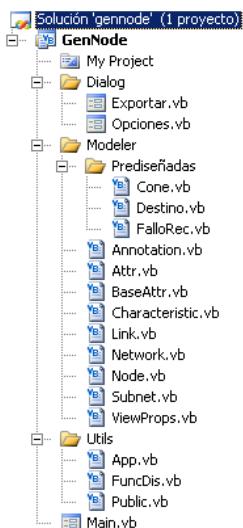
Cuando lea el OPNET el nombre del atributo entiende que es un atributo vector y tiene 17 registros los cuales cada registro tiene una “columna”, en el caso del ejemplo la columna es “name” y el valor es “Database Access (Heavy)”

Los metatags que se ha comentado son los más importantes pero existen otros que indican cómo se tiene que mostrar la red, definir unidades de medida, imagen de fondo, etc.

## 5.2 Estructura del proyecto en Visual Studio

La solución en visual estudio está estructurada de la siguiente forma:

- Carpeta Dialog: Contiene los formularios para la configuración de GenNode. Esta tiene dos formularios que son Opciones y Exportar.
- Carpeta Modeler: Contiene todas las clases que se utilizar para interpretar los datos XML de OPNET. Cada clase se puede importar y exportar a XML. Además contiene otra carpeta:
  - Prediseñadas: Son modelos que ya están prediseñados, es decir es un código “hard-code” de los nodos que interesen.
- Carpeta Utils: Contiene clases que se utilizan en varios partes del código y que su uso es genérico.
- Formulario Main: Es el formulario principal.



## 5.3 Clases

A continuación se explicará que es cada clase y para que se utilicen, está organizado según las carpetas de la solución en Visual Studio:

### Carpeta Utils:

- App: Es una clase compartida (es decir global). Esta clase es útil para funciones genéricas de uso en la aplicación. En esta clase hay funciones para guardar o recuperar la última configuración con la cual se utilizó GenNode. Los datos de los formularios se guardan una vez se cierra la aplicación, y cuando se inicia la aplicación recoge los valores guardados en el XML de configuración, llamado config.xml. La clase App contiene:
  - Datos y estructuras para el uso de forma global.
  - Funciones públicas globales para guardar y recuperar datos de configuración (guardados en el archivo XML).
  - Funciones públicas para importar/exportar valores de OPNET.
- FuncDis: Contiene métodos para generar según la función de distribución. En el proyecto existen dos tipos, poisson y binomial. Tiene una función public compartida la cual se llama para conocer el siguiente número aleatorio (según la función de distribución de la configuración).
- Public: Es un modulo. Contiene variables constantes las cuales principalmente son los literales de metatags que se utilizan para importar desde OPNET.

### Carpeta Modeler:

- Network: Clase donde contiene todos las subredes que se leen desde el fichero xml. Es la clase principal, ya que al importar se llama al método public LoadXML que es donde empieza la carga y la pondera a las subredes. También tiene el método público ToXML, el cual se pasa el path donde se desea guardar el XML. Para realizar el experimento, hay un método llamado experimento1, el cual se le pasa un path donde se quiere guardar el resultado del experimento.
- Subnet: Clase que contiene una lista de nodos y links (entre las más importantes) y otros atributos recogidos en el xml, contiene métodos para la importación y exportación del XML y métodos para realizar el experimento.

- Node: Clase que contiene el listado de todos los atributos, además de algunas propiedades, como el nombre el modelo, etc. Contiene métodos para la importación / exportación del XML.
- Link: Clase que contiene las propiedades de los links. También tiene otras propiedades como el origen y el destino del link, y estos serán dos nodos definidos en la subred.
- Attr: Clase que contiene un atributo específico. El atributo puede ser array o registro. Cuando es array quiere decir que contiene una lista de más atributos con sus propiedades. Si no es array contiene el nombre y el valor.
- BaseAttr: Clase base de Network, Subnet, Node, Link, Attr. Esta clase base la utilizan aquellos nodos que tengan varios atributos para definir. También trata los elementos que son de tipo array, pero que no son nodos XML. Hay que tratar los valores de los atributos para definir la array y todos los registros con sus correspondientes columnas.

## Carpeta Prediseñadas

- Cone: Clase que contiene propiedades para realizar una conexión (link) entre dos nodos. Es necesario para crear el XML que se genera para la conexión.
- Destino: Clase que contiene las estaciones de destino. Genera un XML con de tipo nodo y de clase estación de trabajo. Además tiene métodos para añadirle los grupos multicast los cuales se quiera conectar. Otra de las funciones que hace es generar los Join/Leave de forma aleatoria. Esta clase hace falta para la generación de destinos.
- FalloRec: Clase que genera los fallos y recuperaciones de los nodos que existen en la subred. Lo genera según una función de distribución configurada. Tiene métodos para la exportación a XML.

## 5.4 Interface grafica de usuario

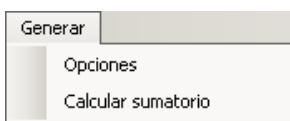
Formulario principal:



Este es el formulario inicial. Como se observa hay un control de tipo árbol de visualización en la parte izquierda, y un grid en la parte derecha. Arriba esta los menús de la aplicación, y se pueden observar dos:



- Abrir XML: Emplea un dialogo para abrir el archivo XML.
- Generar XML: Llama al dialogo Exportar.vb
- Salir: Salir de la aplicación



- Opciones: Llama al dialogo Opciones.vb
- Calcular sumatorio: Función extra, llama a un diálogo para buscar un archivo de texto para el cual hacer el sumatorio.

**Árbol de visualización:** En la pantalla principal, cuando se importa un modelo, muestra un árbol con todo el contenido del modelo. Cada nodo está representado por un ícono diferente. A continuación se mostrarán los íconos posibles que existen:

Icono	Nombre	Icono	Nombre	Icono	Nombre	Icono	Nombre	Icono	Nombre
	Lista de links		Link		Red de trabajo		Atributo o lista de atributos		Lista de subredes



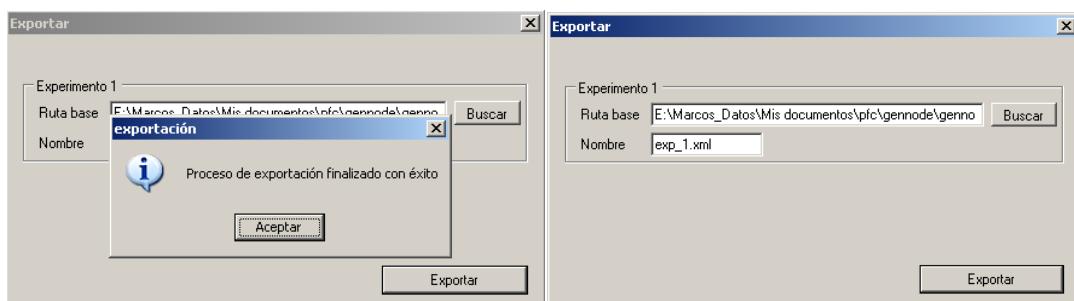
**Grid:** En el momento que se presiona un nodo del árbol, muestra todo su contenido en el grid. Si el elemento en el grid tiene referencia a uno o más elementos, se puede clicar y continúa la navegación al siguiente nodo. En el siguiente ejemplo se muestra los datos de los atributos.

	Campo	Valor
Locale	C	
Version	1.6	
Attribute Processing	explicit	
Subnet	1	
Atributos	7	
Características	units	
ViewProps		

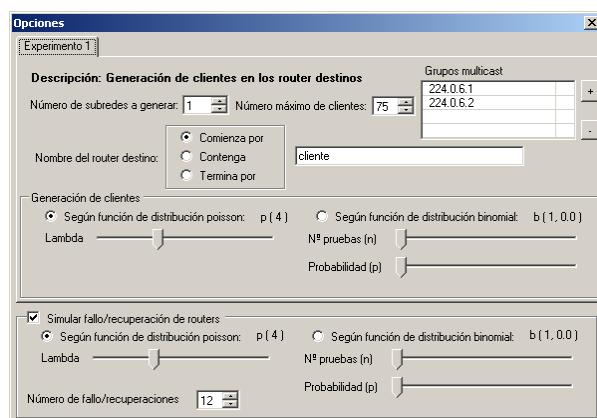
→

	Campo	Valor
x position	0.0	
y position	0.0	
x span	360	
y span	180	
creation source	Object Palette	
creation timestamp	Unknown	
creation data		

**Generar XML:** Es el formulario llamado Exportar XML. Este formulario, contiene dos campos, donde se introduce la ruta a exportar el XML y la dirección de la carpeta donde se quiere guardar el XML. El botón exportar realiza la exportación al archivo indicado. Una vez exportado da un mensaje de finalización.



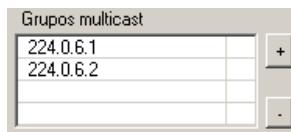
**Opciones:** Es el formulario llamado Opciones.vb. Contiene los parámetros de entrada para la generación del archivo xml. Todos los controles se guardan de forma global, y posteriormente las globales se guardan en un xml. Así estos valores se conservan aunque se cierre la aplicación.



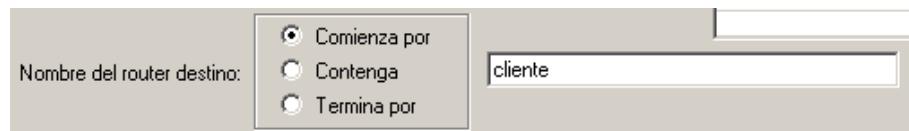
Definición de los controles:

Número de subredes a generar:  Número máximo de clientes:

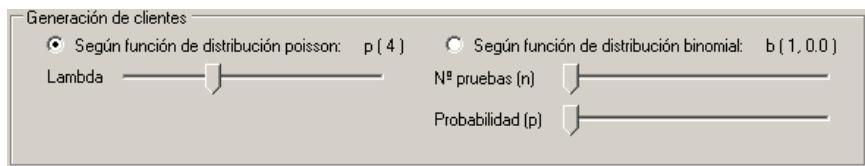
Estos dos controles son para indicar el número de subredes a generar y el número máximo de clientes que se generarán.



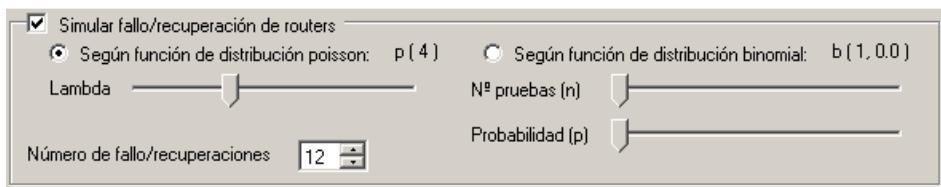
En este control agregan o se borra los grupos multicast que intervienen en la simulación. Si no hay ningún grupo, por defecto escoge 224.0.6.1



En este control se tiene que indicar el nombre del nodo hub que contiene los clientes. Cuando genera el xml, tiene como referencia este nombre en donde generará las estaciones de trabajo que son receptoras. Se puede escoger si empieza, contiene o termina por el nombre indicado.



En este control se define la función de distribución para la generación de nodos clientes. Se puede escoger entre poisson y binomial, y por cada uno se puede definir los parámetros según un control de desliz.



Este control es igual que el control anterior, y es donde se definen las funciones de distribución pero son para los fallo/recuperaciones que hay durante la simulación. Tiene un check para activar esta función, y un campo para indicar el número de fallo/recuperación que se van a generar.

### *5.5 Importación / Exportación XML*

La importación del modelo base tiene cumplir una serie de características. Una vez importado la tarea siguiente será introducir los parámetros de entrada que se deseen para la generación de los nodos. A continuación se generará el archivo, y para ello se muestra un dialogo donde se escoge en que carpeta se quiere guardar y con qué nombre, y entonces se genera.

En la exportación o generación del XML la tarea que realiza es buscar todos los nodos de tipo hub que existen en la red y que tenga un criterio de nombre el cual se ha indicado en los parámetros de entrada. En el momento que encuentra ese hub realiza la generación de nodos clientes. Para ello se basa en el tipo de función de distribución que se ha seleccionado en los parámetros de entrada y por cada nodo que genere, realiza una conexión (un link) y una estación de trabajo destino. Cuando se crea una estación de trabajo, esta contiene todos los join/leave que hay durante una duración de 3600 segundos. Y cuando se crea el link, va desde el hub hasta el cliente creado. Este proceso lo realiza hasta llegar al tope de número máximo de clientes que se ha seleccionado en los parámetros de entrada. Cuando termina el proceso de creación de clientes empieza con los fallo/recuperaciones. Para ello se basa en el número de fallo/recuperaciones que se quieran generar y la función de distribución que se aplica en el tiempo. Todo este proceso de generación, hasta llegar máximo número de subredes a generar.

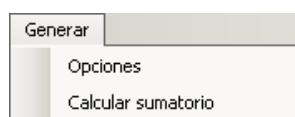
### **5.6 Modelo base**

Es importante que el modelo que está en XML tenga una determinada estructura ya que de esto depende la generación. El modelo tiene que estar compuesto de routers, hubs, y estaciones de trabajo. Las estaciones de trabajo tienen que ser orígenes o destino de un grupo multicast ya definido. Se tiene que definir las aplicaciones y los profiles en OPNET. Además todas las conexiones entre routers y su configuración correspondiente para permitir flujo multicast. Es importante que los nombres de los hub destinos y orígenes estén diferenciados, ya que la generación del XML emplea este nombre para generar los nodos destinos.

### *5.7 Funcionalidades extra*

A medida que se ha ido desarrollando el proyecto, se han hecho en falta algunas funcionalidades, las cuales al final se han implementado ya que ahorraban tiempo.

- Guardado/Recuperación de los parámetros de entrada: Al iniciar la aplicación, genera un archivo llamado config.xml. Este archivo está en la misma ruta donde está la aplicación. Los valores que contiene son los parámetros de entrada últimos que se han guardado, y además guarda la ruta donde se ha generado la ultima exportación.



- Calcular sumatorio: Esta opción está desarrollada para lo archivo de estadística que genera OPNET. El archivo que genera se rige en un patrón “Tiempo” “Valor”, y esta opción lo que realiza es recorrer el archivo y sumar todos los valores, esto nos resultara eficiente para realizar las estadísticas. Cuando se presiona en la opción, aparece un dialogo de búsqueda del archivo .txt donde está la estadística. Una vez seleccionado, aparece una dialogo con el resultado del sumatorio.

## **6. Análisis de resultados de simulación**

### **6.1 Muestra de datos.**

Según la necesidad experimentación variará la recolecta de datos. La principal meta es estudiar los mensajes de control en base a diferentes escenarios, y cuál es el uso de mensajes de control respecto al total de mensajes enviados o recibidos. En cada escenario variará un poco respecto al otro, así que se podrá realizar la comparación. La muestra de datos se basará en el siguiente modelo:

#### Parámetros de entrada:

	Cantidad	Descripción
<b>Routers</b>	N	
<b>Hubs</b>	N	
<b>Links</b>	N	
<b>Host</b>	N	
<b>Clientes</b>	N	Poisson(N)
<b>Orígenes</b>	N	
<b>Grupos Multicast</b>	N	N.N.N.N N.N.N.N
<b>Join/Leave por cliente</b>	N	Random
<b>Fallo/Recuperaciones</b>	N	Poisson(N)
<b>Tiempo Simulación (min.)</b>	N	

*Los parámetros de entrada son los que se utilizan tanto en GenNode como en OPNET. En la columna de cantidad indica la cantidad del elemento, y la descripción es una información añadida para ese elemento:*

*Routers: Cantidad de routers que intervienen en la simulación. Estos estarán basados del modelo base y serán fijos en todas las simulaciones.*

*Hubs: Cantidad de Hubs que intervienen en la simulación. Los hubs ya están predefinidos en el modelo base, así que será una cantidad fija en todas las simulaciones.*

*Links: Son las conexiones entre el router y el hub o entre los routers. El número vendrá dado por la cantidad de host en la simulación.*

*Host: Es el número total de host, puede ser de tipo cliente o de tipo origen.*

- *Clientes: Son los receptores de datos, e indica el número total receptores. La descripción la función de distribución utilizada y sus parámetros.*
- *Orígenes: El numero de orígenes de datos.*

*Grupos multicast: La cantidad de grupos multicast y la descripción es la IP del grupo multicast que intervienen.*

*Fallo/Recuperaciones: Son los fallo/recuperaciones que hay durante la simulación, viene dada por una función de distribución y su parámetro.*

*Tiempo simulación: Es el tiempo de simulación en minutos.*

## Resultados

Información de control:

<b>Msj. "Hello" enviados</b>	$N_1$	<b>Msj. "Hello" recibidos</b>	$N_1$
<b>Msj. "Join-Prune" enviados</b>	$N_1$	<b>Msj. "Join-Prune" recibidos</b>	$N_1$
<b>Msj. "Register" enviados</b>	$N_1$	<b>Msj. "Register" recibidos</b>	$N_1$
<b>Msj. "Register-Stop" enviados</b>	$N_1$	<b>Msj. "Register-Stop" recibidos</b>	$N_1$
<b>Total mensajes enviados</b>	$N_1$	<b>Total mensajes recibidos</b>	$N_1$

<b>Total control recibido (MB)</b>	$N_1$	<b>Total registro recibido (MB)</b>	$N_1$
<b>Total control recibido (paquet.)</b>	$N_1$	<b>Total registro recibido (paquet.)</b>	$N_1$
<b>Total control enviado (MB)</b>	$N_1$	<b>Total registro enviado (MB)</b>	$N_1$
<b>Total control enviado (paquet.)</b>	$N_1$	<b>Total registro enviado (paquet.)</b>	$N_1$

*La información de control que existe durante la simulación esta es recogida y guardada durante toda la simulación. Se puede observar que se diferencian los tipos de mensajes, que son Hello,Join/Prune,Register y Register-Stop. Tambien existe una tabla con las estadísticas propias de opnet, que son el envío/recibo de todos los mensajes de control y el envío/recibo de los registros a los grupos multicast.*

Información del tráfico del contenido:

<b>Total MB recibidos</b>	N
<b>Total Mensajes recibidos</b>	N
<b>Total MB enviados</b>	N
<b>Total Mensajes enviados</b>	N

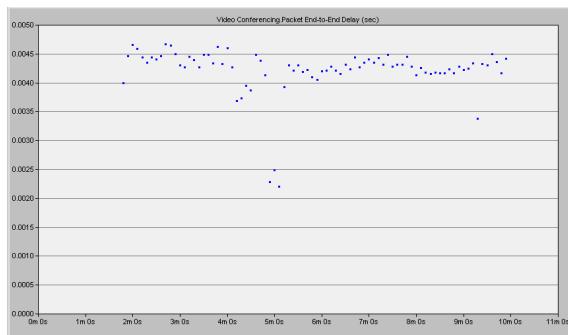
*Aquí puede observar el tráfico total de video recibido a las estaciones que son receptoras. Este valor es útil para conocer la diferencia entre el total recibido y los mensajes de control.*

Mensajes de control respecto al total recibido y enviado (en porcentaje)

<b>Msj. "Hello" enviados</b>	N <sub>1</sub>	<b>Msj. "Hello" recibidos</b>	N <sub>1</sub>
<b>Msj. "Join-Prune" enviados</b>	N <sub>1</sub>	<b>Msj. "Join-Prune" recibidos</b>	N <sub>1</sub>
<b>Msj. "Register" enviados</b>	N <sub>1</sub>	<b>Msj. "Register" recibidos</b>	N <sub>1</sub>
<b>Msj. "Register-Stop" enviados</b>	N <sub>1</sub>	<b>Msj. "Register-Stop" recibidos</b>	N <sub>1</sub>
<b>Total mensajes enviados</b>	N <sub>1</sub>	<b>Total mensajes recibidos</b>	N <sub>1</sub>
<b>Total control recibido (bytes)</b>	N <sub>1</sub>	<b>Total registro recibido (bytes)</b>	N <sub>1</sub>
<b>Total control recibido (paquet.)</b>	N <sub>1</sub>	<b>Total registro recibido (paquet.)</b>	N <sub>1</sub>
<b>Total control enviado (bytes)</b>	N <sub>1</sub>	<b>Total registro enviado (bytes)</b>	N <sub>1</sub>
<b>Total control enviado (paquet.)</b>	N <sub>1</sub>	<b>Total registro enviado (paquet.)</b>	N <sub>1</sub>

*Esta información refleja en tanto porciento la relación entre el total enviado o recibido y los mensajes de control, tanto en bits como en número de mensajes.*

Retraso medio punto a punto:



*Esta gráfica viene establecida por defecto como estadísticas en OPNET. Es un dato útil y muy práctico, ya que nos dice el retraso que hay de un punto a otro de forma global.*

## 6.2. Resultados

### SIMULACIÓN 1

#### Parámetros de entrada

	Cantidad	Descripción
Routers	12	
Hubs	11	
Links	82	
Host	52	
Clientes	50	Poisson(4)
Orígenes	2	
Grupos Multicast	2	224.0.6.1    224.0.6.2
Join/Leave por cliente	8	Random (librería de framework)
Fallo/Recuperaciones	10	Poisson(3)
Tiempo Simulación (min.)	30	

#### Resultados

Información de control:

Msj. "Hello" enviados	94	Msj. "Hello" recibidos	62
Msj. "Join-Prune" enviados	177	Msj. "Join-Prune" recibidos	175
Msj. "Register" enviados	10376	Msj. "Register" recibidos	9640
Msj. "Register-Stop" enviados	1226	Msj. "Register-Stop" recibidos	1226
<b>Total mensajes enviados</b>	<b>11873</b>	<b>Total mensajes recibidos</b>	<b>11103</b>

<b>Total control recibido (bytes)</b>	21056	<b>Total registro recibido (bytes)</b>	1400240
<b>Total control recibido (paquet.)</b>	1463	<b>Total registro recibido (paquet.)</b>	9640
<b>Total control enviado (bytes)</b>	21	<b>Total registro enviado (bytes)</b>	13378672

<b>Total control enviado (paquet.)</b>	1497	<b>Total registro enviado (paquet.)</b>	10376
--	------	---	-------

Información del contenido:

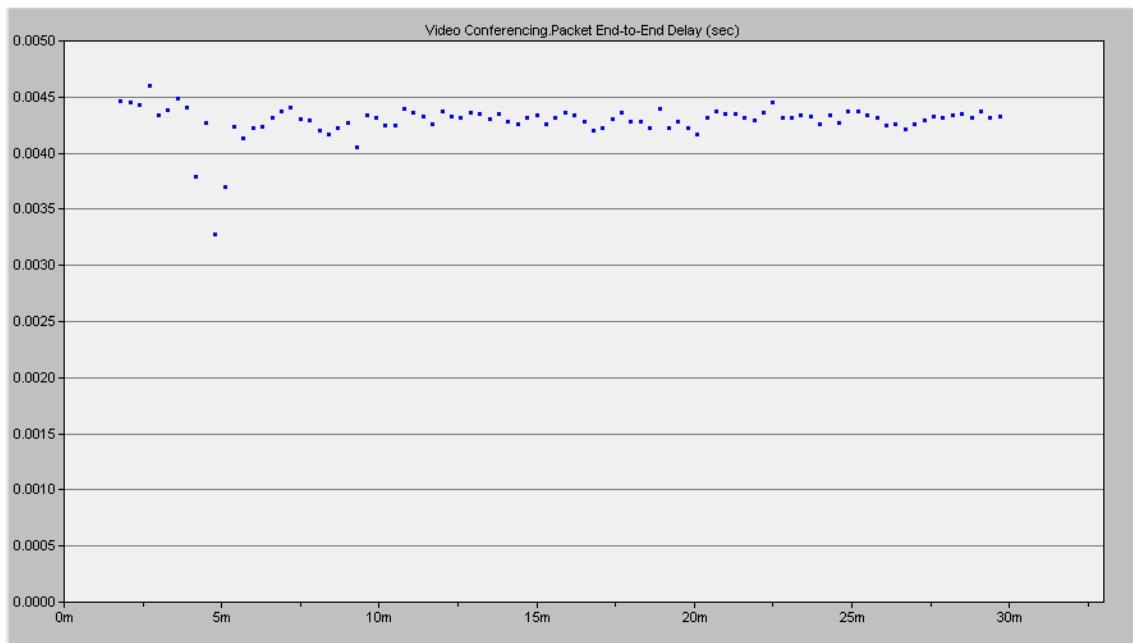
<b>Total bytes recibidos</b>	3167156728
<b>Total Mensajes recibidos</b>	633458
<b>Total bytes enviados</b>	2617126856
<b>Total Mensajes enviados</b>	523454

Mensajes de control respecto al total recibido y enviado (en porcentaje)

<b>Msj. "Hello" enviados</b>	0,01795764	<b>Msj. "Hello" recibidos</b>	0,00978755
<b>Msj. "Join-Prune" enviados</b>	0,03381386	<b>Msj. "Join-Prune" recibidos</b>	0,02762614
<b>Msj. "Register" enviados</b>	1,98221811	<b>Msj. "Register" recibidos</b>	1,52180571
<b>Msj. "Register-Stop" enviados</b>	0,23421351	<b>Msj. "Register-Stop" recibidos</b>	0,19354085
<b>Total mensajes enviados</b>	2,268203128	<b>Total mensajes recibidos</b>	1,752760246

<b>Total control recibido (bytes)</b>	0,00066482	<b>Total registro recibido (bytes)</b>	0,04421126
<b>Total control recibido (paquet.)</b>	0,23095454	<b>Total registro recibido (paquet.)</b>	1,52180571
<b>Total control enviado (bytes)</b>	0,00	<b>Total registro enviado (bytes)</b>	0,51119692
<b>Total control enviado (paquet.)</b>	0,28598501	<b>Total registro enviado (paquet.)</b>	1,98221811

Retraso medio punto a punto:



## SIMULACIÓN 2

Parámetros de entrada:

	Cantidad	Descripción
<b>Routers</b>	12	
<b>Hubs</b>	11	
<b>Links</b>	107	
<b>Host</b>	77	
<b>Clientes</b>	75	Poisson(4)
<b>Orígenes</b>	2	
<b>Grupos Multicast</b>	2	224.0.6.1    224.0.6.2
<b>Join/Leave por cliente</b>	8	Random (librería de framework)
<b>Fallo/Recuperaciones</b>	12	Poisson(4)
<b>Tiempo Simulación (min.)</b>	30	

Resultados

Información de control:

<b>Msj. "Hello" enviados</b>	129	<b>Msj. "Hello" recibidos</b>	64
<b>Msj. "Join-Prune" enviados</b>	71	<b>Msj. "Join-Prune" recibidos</b>	56
<b>Msj. "Register" enviados</b>	67664	<b>Msj. "Register" recibidos</b>	67664
<b>Msj. "Register-Stop" enviados</b>	29	<b>Msj. "Register-Stop" recibidos</b>	29
<b>Total mensajes enviados</b>	67893	<b>Total mensajes recibidos</b>	67813

<b>Total control recibido (bytes)</b>	2908	<b>Total registro recibido (bytes)</b>	10283824
<b>Total control recibido (paquet.)</b>	149	<b>Total registro recibido (paquet.)</b>	67664
<b>Total control enviado (bytes)</b>	4168	<b>Total registro enviado (bytes)</b>	86608816
<b>Total control enviado (paquet.)</b>	229	<b>Total registro enviado (paquet.)</b>	67664

Información del tráfico del contenido:

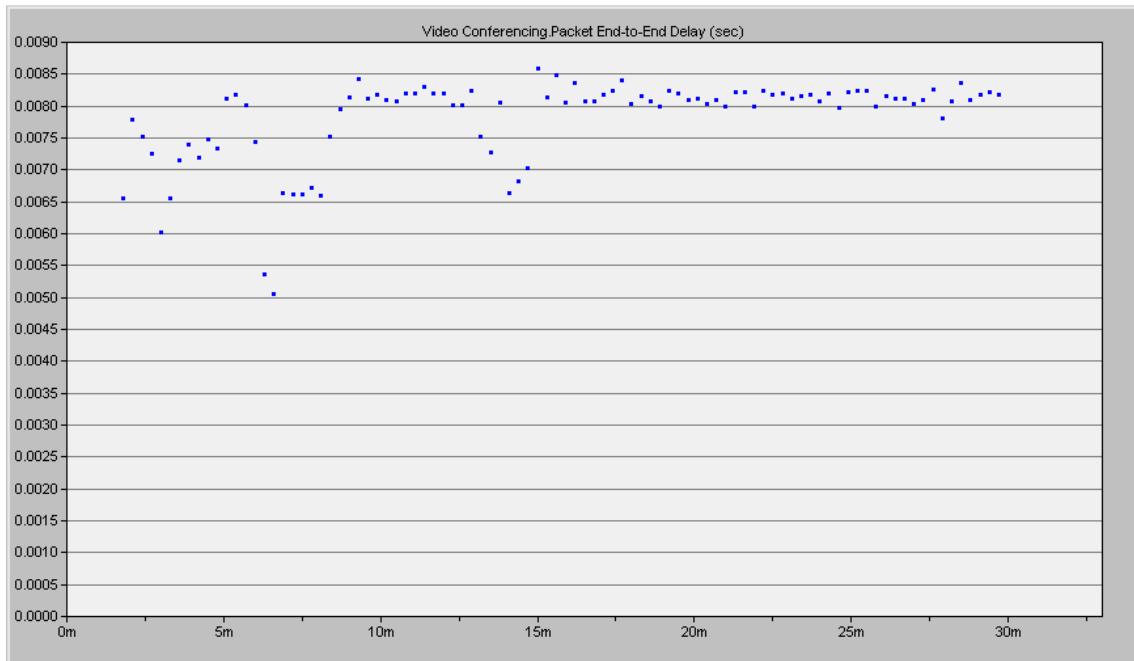
<b>Total bytes recibidos</b>	6036333520
<b>Total Mensajes recibidos</b>	1207321
<b>Total bytes enviados</b>	4972613648
<b>Total Mensajes enviados</b>	994579

Mensajes de control respecto al total recibido y envidado (en porcentaje)

<b>Msj. "Hello" enviados</b>	0,0129703	<b>Msj. "Hello" recibidos</b>	0,0053010
<b>Msj. "Join-Prune" enviados</b>	0,0071387	<b>Msj. "Join-Prune" recibidos</b>	0,0046384
<b>Msj. "Register" enviados</b>	6,8032806	<b>Msj. "Register" recibidos</b>	5,6044747
<b>Msj. "Register-Stop" enviados</b>	0,0029158	<b>Msj. "Register-Stop" recibidos</b>	0,0024020
<b>Total mensajes enviados</b>	6,8263054	<b>Total mensajes recibidos</b>	5,6168161

<b>Total control recibido (bytes)</b>	0,0000482	<b>Total registro recibido (bytes)</b>	0,1703654
<b>Total control recibido (paquet.)</b>	0,0123414	<b>Total registro recibido (paquet.)</b>	5,6044747
<b>Total control enviado (bytes)</b>	0,0000838	<b>Total registro enviado (bytes)</b>	1,7417162
<b>Total control enviado (paquet.)</b>	0,0230248	<b>Total registro enviado (paquet.)</b>	6,8032806

Retraso medio punto a punto:



### **SIMULACIÓN 3**

Parámetros de entrada:

	Cantidad	Descripción
<b>Routers</b>	12	
<b>Hubs</b>	11	
<b>Links</b>	132	
<b>Host</b>	102	
<b>Clientes</b>	100	Poisson(5)
<b>Orígenes</b>	2	
<b>Grupos Multicast</b>	2	224.0.6.1    224.0.6.2
<b>Join/Leave por cliente</b>	8	Random (librería de framework)
<b>Fallo/Recuperaciones</b>	15	Poisson(4)
<b>Tiempo Simulación (min.)</b>	30	

Resultados

Información de control:

<b>Msj. "Hello" enviados</b>	193	<b>Msj. "Hello" recibidos</b>	38
<b>Msj. "Join-Prune" enviados</b>	76	<b>Msj. "Join-Prune" recibidos</b>	76
<b>Msj. "Register" enviados</b>	63703	<b>Msj. "Register" recibidos</b>	254
<b>Msj. "Register-Stop" enviados</b>	69	<b>Msj. "Register-Stop" recibidos</b>	69
<b>Total mensajes enviados</b>	64041	<b>Total mensajes recibidos</b>	437

<b>Total control recibido (bytes)</b>	7432	<b>Total registro recibido (bytes)</b>	8652
<b>Total control recibido (paquet.)</b>	366	<b>Total registro recibido (paquet.)</b>	71
<b>Total control enviado (bytes)</b>	11152	<b>Total registro enviado (bytes)</b>	81108148
<b>Total control enviado (paquet.)</b>	676	<b>Total registro enviado (paquet.)</b>	63365

Información del contenido:

<b>Total bytes recibidos</b>	3907453520
<b>Total Mensajes recibidos</b>	781545
<b>Total bytes enviados</b>	4985078648
<b>Total Mensajes enviados</b>	997072

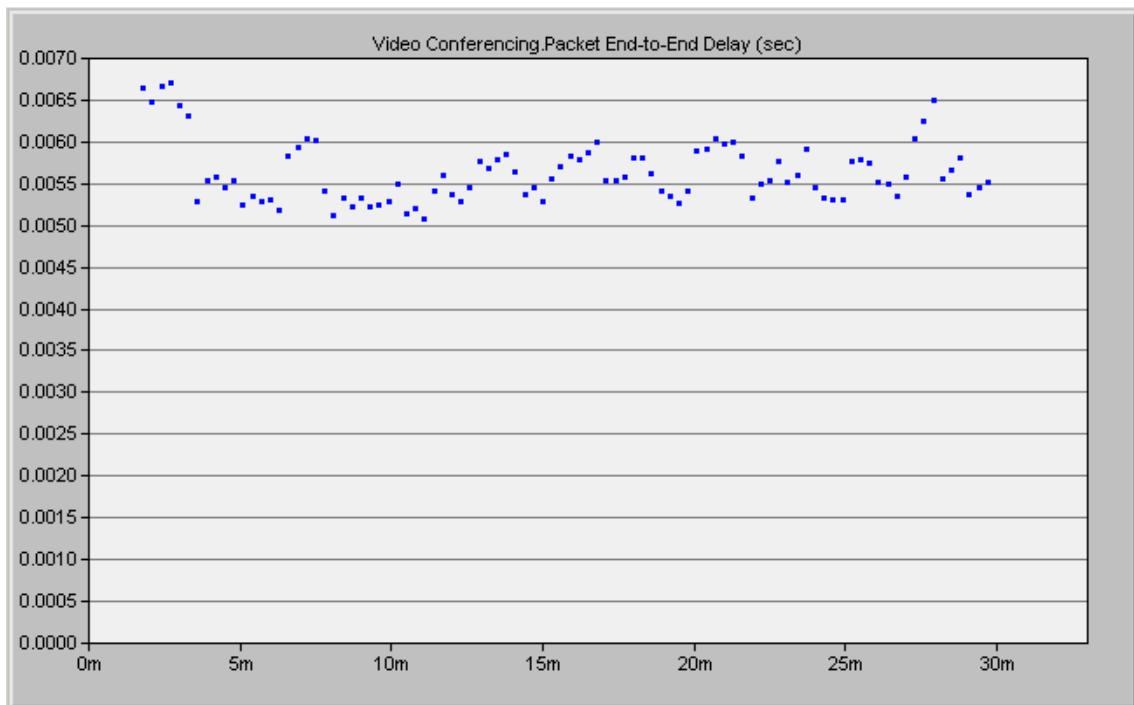
Mensajes de control respecto al total recibido y enviado (en porcentaje):

<b>Msj. "Hello" enviados</b>	0,0193567	<b>Msj. "Hello" recibidos</b>	0,0048622
<b>Msj. "Join-Prune" enviados</b>	0,0076223	<b>Msj. "Join-Prune" recibidos</b>	0,0097243
<b>Msj. "Register" enviados</b>	6,3890070	<b>Msj. "Register" recibidos</b>	0,0324997
<b>Msj. "Register-Stop" enviados</b>	0,0069203	<b>Msj. "Register-Stop" recibidos</b>	0,0088287
<b>Total mensajes enviados</b>	6,4229063	<b>Total mensajes recibidos</b>	0,0559149

<b>Total control recibido (bytes)</b>	0,0001902	<b>Total registro recibido (bytes)</b>	0,0002214
<b>Total control recibido (paquet.)</b>	0,0468303	<b>Total registro recibido (paquet.)</b>	0,0090846
<b>Total control enviado (bytes)</b>	0,0002237	<b>Total registro enviado (bytes)</b>	1,6270184
<b>Total control enviado (paquet.)</b>	0,0677985	<b>Total registro enviado (paquet.)</b>	6,3551078

Retraso medio punto a punto:



## SIMULACIÓN 4

Parámetros de entrada:

	Cantidad	Descripción
<b>Routers</b>	12	
<b>Hubs</b>	11	
<b>Links</b>	83	
<b>Host</b>	53	
<b>Clientes</b>	50	Poisson(4)
<b>Orígenes</b>	3	
<b>Grupos Multicast</b>	3	224.0.6.1    224.0.6.2    224.0.6.3
<b>Join/Leave por cliente</b>	8	Random (librería de framework)
<b>Fallo/Recuperaciones</b>	10	Poisson(3)
<b>Tiempo Simulación (min.)</b>	30	

Resultados

Información de control:

<b>Msj. "Hello" enviados</b>	126	<b>Msj. "Hello" recibidos</b>	80
<b>Msj. "Join-Prune" enviados</b>	166	<b>Msj. "Join-Prune" recibidos</b>	156
<b>Msj. "Register" enviados</b>	68602	<b>Msj. "Register" recibidos</b>	68402
<b>Msj. "Register-Stop" enviados</b>	12638	<b>Msj. "Register-Stop" recibidos</b>	12638
<b>Total mensajes enviados</b>	81532	<b>Total mensajes recibidos</b>	81276

<b>Total control recibido (bytes)</b>	157608	<b>Total registro recibido (bytes)</b>	10250704
<b>Total control recibido (paquet.)</b>	12874	<b>Total registro recibido (paquet.)</b>	68402
<b>Total control enviado (bytes)</b>	158480	<b>Total registro enviado (bytes)</b>	88031888
<b>Total control enviado (paquet.)</b>	12930	<b>Total registro enviado (paquet.)</b>	68602

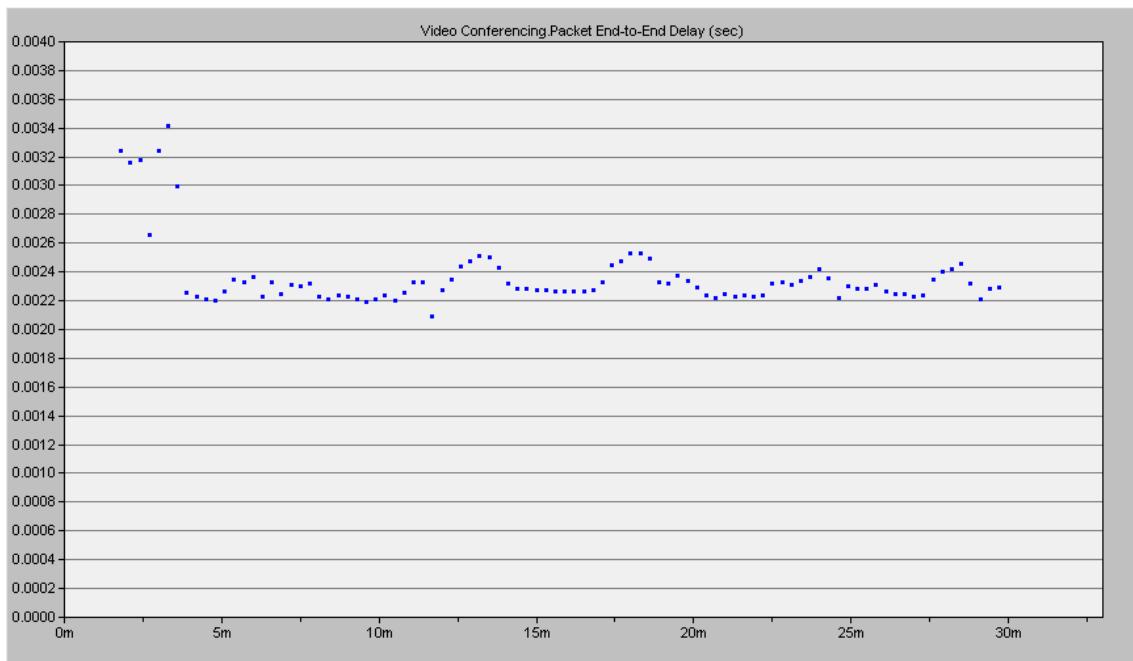
Información del contenido:

<b>Total bytes recibidos</b>	1633911536
<b>Total Mensajes recibidos</b>	326806
<b>Total bytes enviados</b>	2358921664
<b>Total Mensajes enviados</b>	471810

Mensajes de control respecto al total recibido y enviado (en porcentaje):

<b>Msj. "Hello" enviados</b>	0,0267057	<b>Msj. "Hello" recibidos</b>	0,0244794
<b>Msj. "Join-Prune" enviados</b>	0,0351837	<b>Msj. "Join-Prune" recibidos</b>	0,0477347
<b>Msj. "Register" enviados</b>	14,5401751	<b>Msj. "Register" recibidos</b>	20,9304603
<b>Msj. "Register-Stop" enviados</b>	2,6786206	<b>Msj. "Register-Stop" recibidos</b>	3,8671261
<b>Total mensajes enviados</b>	17,2806850	<b>Total mensajes recibidos</b>	24,8698004
<b>Total control recibido (bytes)</b>	0,0096461	<b>Total registro recibido (bytes)</b>	0,6273720
<b>Total control recibido (paquet.)</b>	3,9393402	<b>Total registro recibido (paquet.)</b>	20,9304603
<b>Total control enviado (bytes)</b>	0,0067183	<b>Total registro enviado (bytes)</b>	3,7318699
<b>Total control enviado (paquet.)</b>	2,7405100	<b>Total registro enviado (paquet.)</b>	14,5401751

Retraso medio punto a punto:



## SIMULACIÓN 5

Parámetros de entrada:

	Cantidad	Descripción
<b>Routers</b>	12	
<b>Hubs</b>	11	
<b>Links</b>	108	
<b>Host</b>	78	
<b>Clientes</b>	75	Poisson(4)
<b>Orígenes</b>	2	
<b>Grupos Multicast</b>	3	224.0.6.1    224.0.6.2    224.0.6.3
<b>Join/Leave por cliente</b>	8	Random (librería de framework)
<b>Fallo/Recuperaciones</b>	12	Poisson(4)
<b>Tiempo Simulación (min.)</b>	30	

Resultados

Información de control:

<b>Msj. "Hello" enviados</b>	157	<b>Msj. "Hello" recibidos</b>	98
<b>Msj. "Join-Prune" enviados</b>	157	<b>Msj. "Join-Prune" recibidos</b>	156
<b>Msj. "Register" enviados</b>	30297	<b>Msj. "Register" recibidos</b>	30297
<b>Msj. "Register-Stop" enviados</b>	30244	<b>Msj. "Register-Stop" recibidos</b>	30244
<b>Total mensajes enviados</b>	60855	<b>Total mensajes recibidos</b>	60795

<b>Total control recibido (bytes)</b>	369096	<b>Total registro recibido (bytes)</b>	394220
<b>Total control recibido (paquet.)</b>	30498	<b>Total registro recibido (paquet.)</b>	30297
<b>Total control enviado (bytes)</b>	369836	<b>Total registro enviado (bytes)</b>	45263052
<b>Total control enviado (paquet.)</b>	30558	<b>Total registro enviado (paquet.)</b>	30297

Información del contenido:

<b>Total bytes recibidos</b>	1335545576
<b>Total Mensajes recibidos</b>	267118
<b>Total bytes enviados</b>	1096135704
<b>Total Mensajes enviados</b>	219238

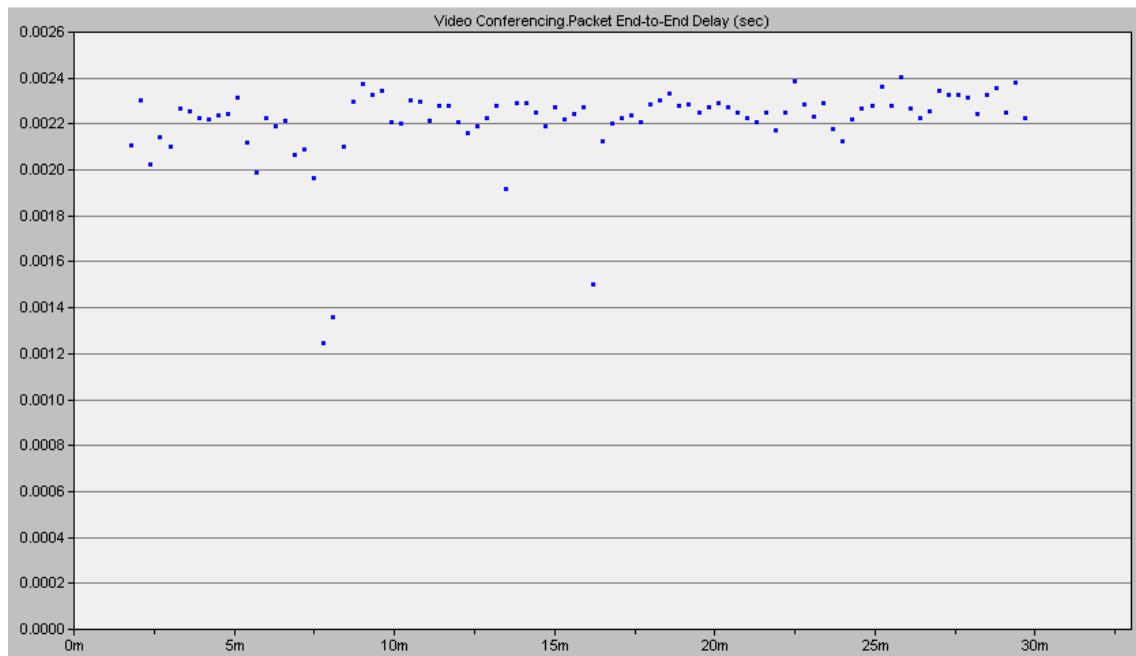
Mensajes de control respecto al total recibido y enviado (en porcentaje):

<b>Msj. "Hello" enviados</b>	0,0716117	<b>Msj. "Hello" recibidos</b>	0,0366879
<b>Msj. "Join-Prune" enviados</b>	0,0716117	<b>Msj. "Join-Prune" recibidos</b>	0,0584012
<b>Msj. "Register" enviados</b>	13,8192284	<b>Msj. "Register" recibidos</b>	11,3421784
<b>Msj. "Register-Stop" enviados</b>	13,7950538	<b>Msj. "Register-Stop" recibidos</b>	11,3223369
<b>Total mensajes enviados</b>	27,7575055	<b>Total mensajes recibidos</b>	22,7596044

<b>Total control recibido (bytes)</b>	0,0276363	<b>Total registro recibido (bytes)</b>	0,0295175
<b>Total control recibido (paquet.)</b>	11,4174260	<b>Total registro recibido (paquet.)</b>	11,3421784
<b>Total control enviado (bytes)</b>	0,0337400	<b>Total registro enviado (bytes)</b>	4,1293292
<b>Total control enviado (paquet.)</b>	13,9382771	<b>Total registro enviado (paquet.)</b>	13,8192284

Retraso medio punto a punto:



## SIMULACIÓN 6

Parámetros de entrada:

	Cantidad	Descripción
<b>Routers</b>	12	
<b>Hubs</b>	11	
<b>Links</b>	133	
<b>Host</b>	103	
<b>Clientes</b>	100	Poisson(N)
<b>Orígenes</b>	3	
<b>Grupos Multicast</b>	3	224.0.6.1    224.0.6.2    224.0.6.3
<b>Join/Leave por cliente</b>	8	Random (librería de framework)
<b>Fallo/Recuperaciones</b>	15	Poisson(4)
<b>Tiempo Simulación (min.)</b>	30	

Resultados

Información de control:

<b>Msj. "Hello" enviados</b>	155	<b>Msj. "Hello" recibidos</b>	98
<b>Msj. "Join-Prune" enviados</b>	289	<b>Msj. "Join-Prune" recibidos</b>	287
<b>Msj. "Register" enviados</b>	50533	<b>Msj. "Register" recibidos</b>	43334
<b>Msj. "Register-Stop" enviados</b>	30622	<b>Msj. "Register-Stop" recibidos</b>	30042
<b>Total mensajes enviados</b>	81599	<b>Total mensajes recibidos</b>	73761

<b>Total control recibido (bytes)</b>	2966912	<b>Total registro recibido (bytes)</b>	37840640
<b>Total control recibido (paquet.)</b>	30427	<b>Total registro recibido (paquet.)</b>	43334
<b>Total control enviado (bytes)</b>	3028576	<b>Total registro enviado (bytes)</b>	540304864
<b>Total control enviado (paquet.)</b>	31066	<b>Total registro enviado (paquet.)</b>	50533

Información del contenido:

<b>Total MB recibidos</b>	8013369928
<b>Total Mensajes recibidos</b>	1602750
<b>Total MB enviados</b>	7004820120
<b>Total Mensajes enviados</b>	1401043

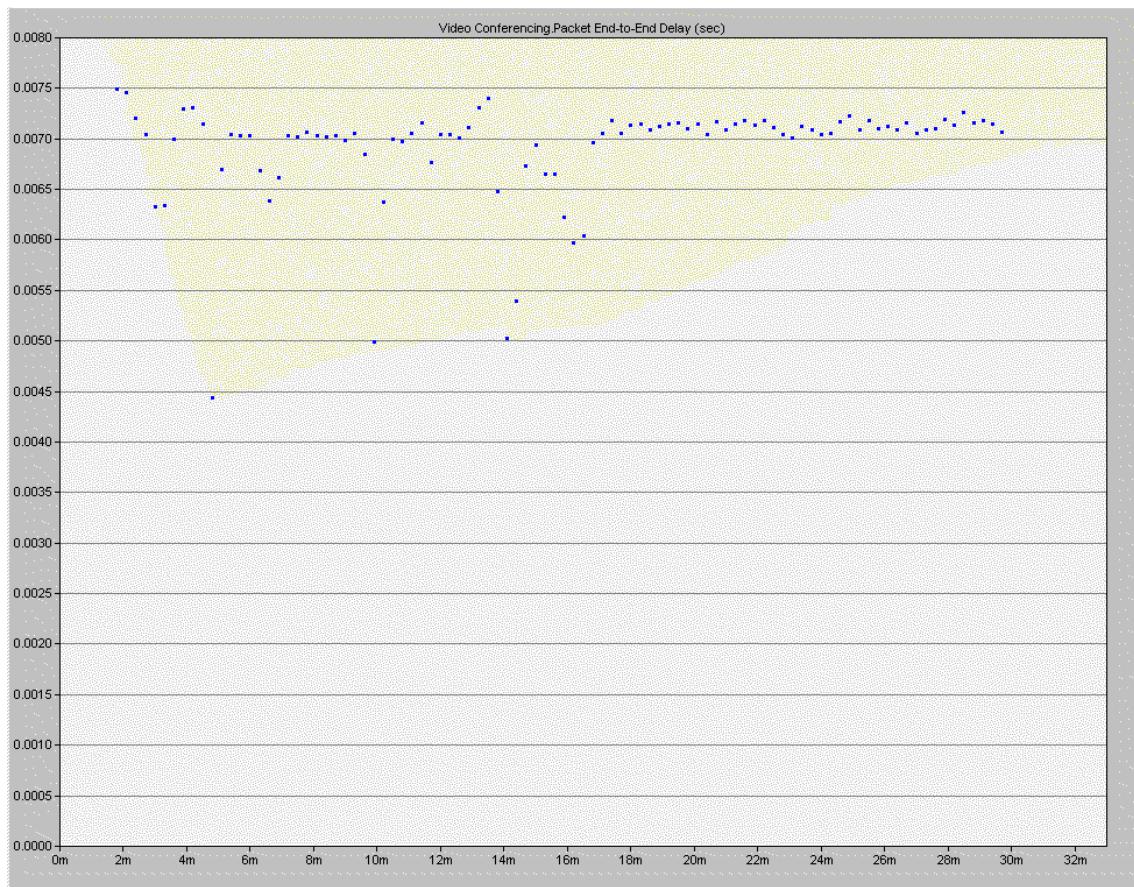
Mensajes de control respecto al total recibido y enviado (en porcentaje):

<b>Msj. "Hello" enviados</b>	0,0110632	<b>Msj. "Hello" recibidos</b>	0,0061145
<b>Msj. "Join-Prune" enviados</b>	0,0206275	<b>Msj. "Join-Prune" recibidos</b>	0,0179067
<b>Msj. "Register" enviados</b>	3,6068129	<b>Msj. "Register" recibidos</b>	2,7037280
<b>Msj. "Register-Stop" enviados</b>	2,1856574	<b>Msj. "Register-Stop" recibidos</b>	1,8744034
<b>Total mensajes enviados</b>	5,8241610	<b>Total mensajes recibidos</b>	4,6021526

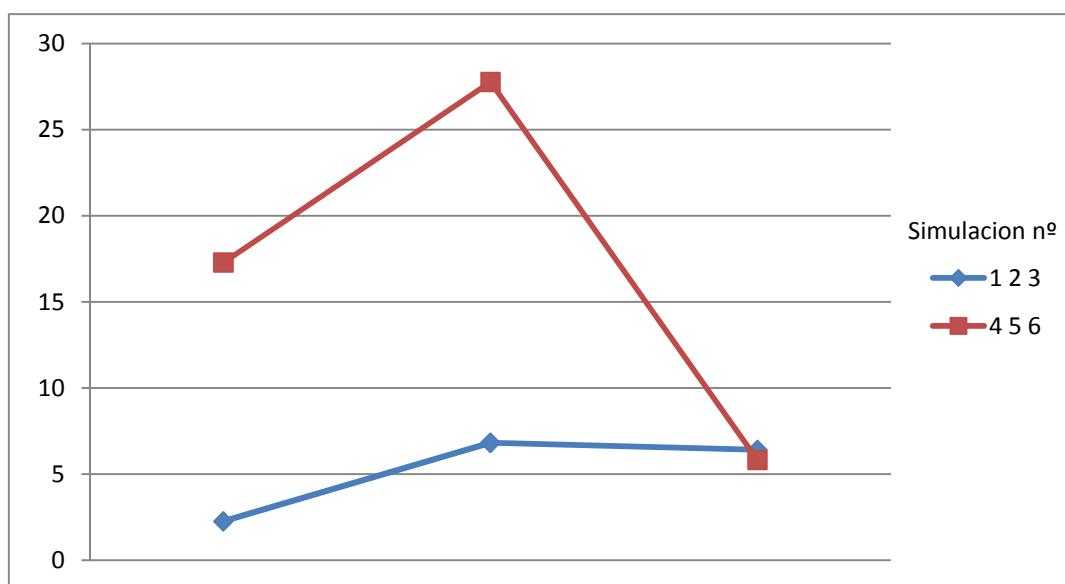
<b>Total control recibido (bytes)</b>	0,0370245	<b>Total registro recibido (bytes)</b>	0,4722188
<b>Total control recibido (paquet.)</b>	1,8984246	<b>Total registro recibido (paquet.)</b>	2,7037280
<b>Total control enviado (bytes)</b>	0,0432356	<b>Total registro enviado (bytes)</b>	7,7133296
<b>Total control enviado (paquet.)</b>	2,2173481	<b>Total registro enviado (paquet.)</b>	3,6068129

Retraso medio punto a punto:

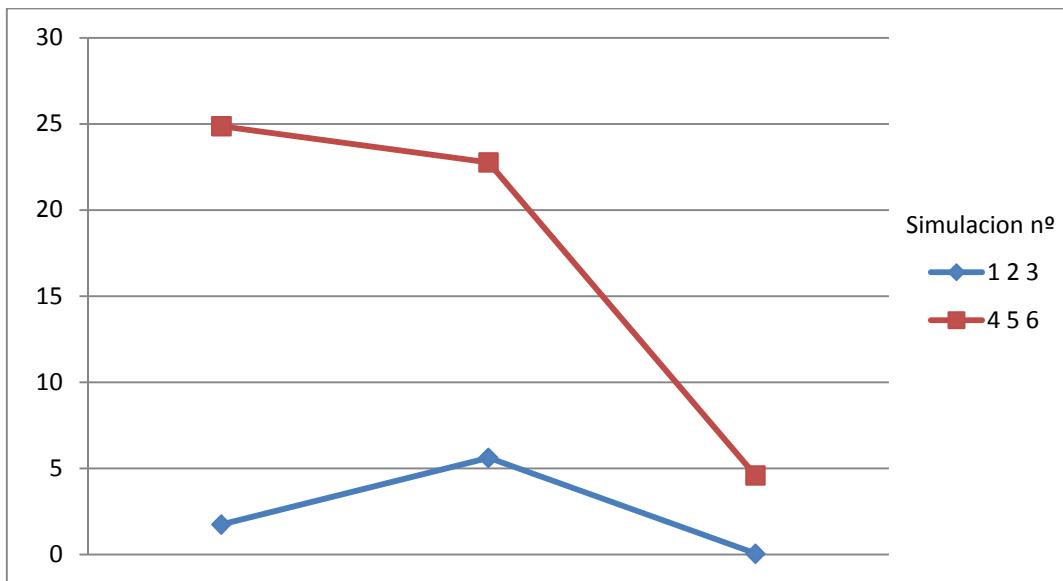


### 6.3 Comparativa de resultados

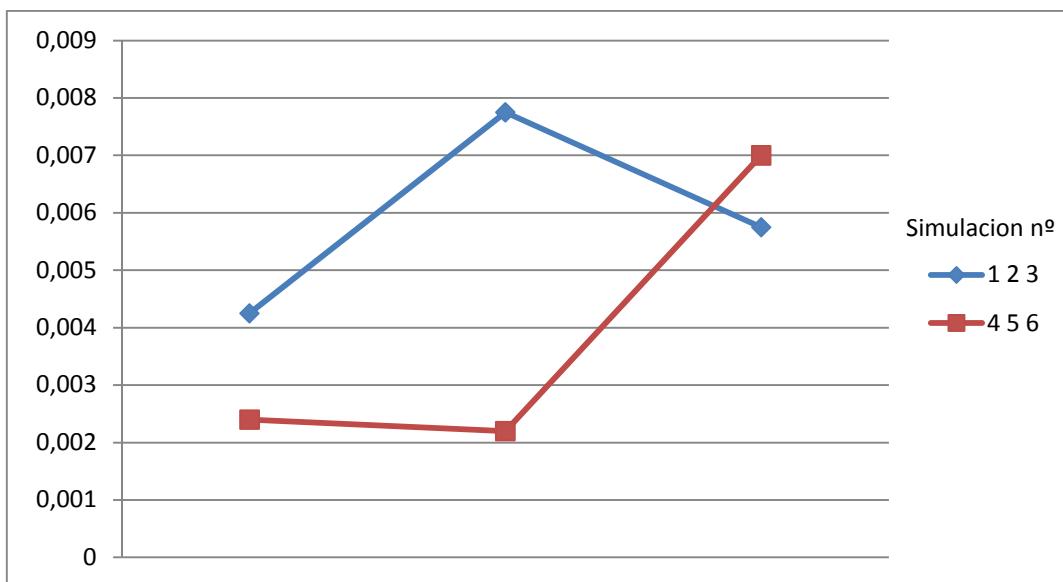
Total mensajes enviados en porcentaje



*Total mensajes recibidos en porcentaje*



*Retraso medio punto a punto*



#### *6.4 Conclusión*

Podemos observar que la simulación 4,5 y 6 respecto a flujos de control tienen una mayor congestión que las simulaciones 1,2 y 3, esto quiere decir que al añadir un grupo mas multicast, ayuda a una mayor congestión. Las simulaciones 1,2 y 3 la media de congestión están en torno al 5%, y las de la simulaciones 4,5 y 6 están en torno al 15% de congestión. A medida que aumentamos los clientes, en los dos grupos de simulaciones, los datos de control aumentan, por lo tanto este hecho repercute en el ancho de banda tanto la recepción como en el envío. El ancho de banda experimenta un descenso tanto en el envío de datos como la recepción y esto es debido al aumento del flujo de control.

Otro dato a destacar, aunque exista mayor congestión de flujos de control el retraso medio de extremo a extremo es mayor con dos orígenes de datos. Es posible que sea porque la red está más seccionada, es decir que hay más grupos multicast, y por lo tanto es más probable que un cliente esté más cerca del origen y pueda realizar un árbol de distribución más óptimo.

## **7. Bibliografía.**

- ❖ <http://www.ietf.org/rfc/rfc2362.txt> — *Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification*
- ❖ <http://www.rfc-editor.org/rfc/rfc3973.txt> — *Protocol Independent Multicast — Dense Mode (PIM-DM): Protocol Specification (Revised)*
- ❖ *Libing Wu, Shengchao Ding, Yanxiang He, Simeng Wang, Qiang Cao1 — The Research and Emulation on PIM-SM Protocol*
- ❖ *Yu Zhenwei, Zhang Yi, Wang Jilu — Research on Simulation of multicast Protocol*
- ❖ <http://cdn.procurve.com/training/Manuals/3500-5400-6200-8200-MRG-Jan08-4-PIM-SM.pdf> — *PIM-SM (Sparse Mode)*
- ❖ *Yezid Donoso Meisel, Adriana Ortiz, Alvaro Herrera — Análisis de rendimiento y pruebas de configuración de transmisiones de datos multipunto bajo árboles Dense-Mode y Sparse-Mode*
- ❖ *Daniel Minoli — IP MULTICAST WITH APPLICATIONS TO IPTV AND MOBILE DVB-H*
- ❖ [https://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6552/prod\\_white\\_paper0900aecd804d5fe6.pdf](https://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6552/prod_white_paper0900aecd804d5fe6.pdf) — *IP Multicast Technology Overview*
- ❖ [http://www.opnet.com/university\\_program/teaching\\_with\\_opnet/textbooks\\_and\\_materials/materials/OPNET\\_Modeler\\_Manual.pdf](http://www.opnet.com/university_program/teaching_with_opnet/textbooks_and_materials/materials/OPNET_Modeler_Manual.pdf) — *OPNET Manual de usuario.*
- ❖ *Tomasz Bartczak, Piotr Zwierzykowski — Simulation study of PIM DM and PIM SM protocols.*
- ❖ *Dan Li1, Jianping Wu, Ke Xu, Yong Cui, Ying Liu, Xiaoping Zhang — Performance Analysis of Multicast Routing Protocol PIM-SM*