ANÁLISIS DE REQUERIMIENTOS DE QoS SOBRE REDES IP MULTICAST

por

David Alejandro Perez

Trabajo de Tesis

Dirigido a

Universidad Nacional de La Plata, Facultad de Informática

en total cumplimiento con los requerimientos

para el titulo de

MAGÍSTER EN REDES DE DATOS

Facultad de Informática

2005

Director: Ing. Luis Marrone

ÍNDICE DE TEMAS

ÍNDICE DE TEMAS 2

ÍNDICE DE FIGURAS 4

ÍNDICE DE TABLAS 6

INTRODUCCIÓN. 7

OBJETIVO. 9

1. IP MULTICAST. 10

1.1. DIRECCIONAMIENTO MULTICAST. 10
1.2. DIRECCIONES MULTICAST DE CAPA 2. 11
1.3. ÁRBOLES MULTICAST DE DISTRIBUCIÓN. 13
1.3.1. SOURCE TREE. 13
1.3.2. SHARED TREE. 14
1.4. MULTICAST FORWARDING. 16
1.5. IGMP (INTERNET GROUP MANAGEMENT PROTOCOL). 17
1.5.1. IGMP Versión 1. 18
1.5.1.1. REPORT SUPRESION MECHANISM. 19
1.5.1.2. IGMPv1 QUERIER. 19
1.5.1.3. IGMPv1 JOIN PROCESS. 19
1.5.2. IGMP Versión 2. 19
1.6. DISTANCE VECTOR MULTICAST ROUTING PROTOCOL, DVMRP. 21
1.6.1. DVMRP MULTICAST FORWARDING. 22
1.7. PROTOCOL INDEPENDENT MULTICAST, PIM, DENSE MODE. 23
1.7.1. PIM NEIGHBOR DISCOVERY. 23
1.7.2. PIM-DM SOURCE DISTRIBUTION TREES. 24
1.7.3. PIM-DM MULTICAST FORWARDING. 24
1.7.4. PIM-DM ASSERTS. 25
1.8. PROTOCOL INDEPENDENT MULTICAST SPARSE MODE, PIM-SM. 25
1.8.1. EXPLICIT JOIN MODEL. 25
1.8.2. PIM-SM SHARED TREES. 26
1.8.3. PIM-SM SHORTEST PATH TREE, SPT. 26
1.8.4. PIM-SM STATE-REFRESH. 27
1.8.5. SOURCE REGISTRATION. 27
1.8.6. PIM-SM DESIGNATED ROUTER, PIM-SM DR. 28
1.9. MULTICAST OPEN SHORTEST PATH FIRST, MOSPF. 28
1.9.1. MOSPF INTRA-AREA MULTICAST ROUTING. 28
1.9.1.1. GOUP MEMBERSHIP LINK-STATE ADVERTISEMENT. 28
1.9.1.2. INTRA-AREA SHORTEST PATH TREE, SPT. 29
1.9.2. MOSPF INTER-AREA MULTICAST FORWARDING. 30
1.9.2.1. MULTICAST AREA BORDER ROUTER. 30
1.9.3. MOSPF INTER-AS MULTICAST ROUTING. 31
1.10. INTER-DOMAIN MULTICAST ROUTING. 31
1.10.1. MULTIPROTOCOL BGP, MBGP. 32
1.10.2. MULTICAST SOURCE DISCOVERY PROTOCOL, MSDP. 32
1.10.3. BORDER GATEWAY MULTICAST PROTOCOL, BGMP. 34
1.11. ANYCAST RP. 34
1.12. CICLO DE VIDA DE UN GRUPO IP MULTICAST. 35

2. MULTIPROTOCOL LABEL SWITCHING, MPLS. 39

2.1. ARQUITECTURA. 39
2.2. COMPONENTES FUNCIONALES DE RUTEO EN LA CAPA DE RED. 40

2.2.1. COMPONENTE DE FORWARDING. 40
2.2.1.1. TRANSPORTE DE ETIQUETAS. 43
2.2.2. COMPONENTE DE CONTROL. 43
2.2.2.1. DISTRIBUCIÓN DE LA INFORMACIÓN DE LABEL BINDING. 45
2.2.2.2. MULTICAST FORWARDING. 46
2.3. LABEL SWITCHED PATH, LSP. 47
2.4. PREVENCIÓN Y DETECCIÓN DE LOOPS. 48
2.5. ENCAPSULACIÓN. 49
2.6. LABEL DISTRIBUTION PROTOCOL, LDP. 50
2.6.1. LSR NEIGHBOR DISCOVERY. 51
2.6.2. TRANSPORTE CONFIABLE. 52
2.6.3. MENSAJES LDP. 52
2.6.4. MODOS DE DISTRIBUCIÓN DE ETIQUETAS. 54
2.7. DISTRIBUCIÓN DE ETIQUETAS UTILIZANDO BGP. 55
2.8. MULTICAST. 55

3. TÉCNICAS DE COMPRESIÓN DE VÍDEO. 57

3.1. RECOMENDACIÓN ITU-T H.323. 57
3.2. NORMAS ISO MPEG-X. 58
3.3. PROTOCOLOS PARA LA TRANSMISIÓN REAL-TIME MULTIMEDIA. 60

4. IP MULTICAST SOBRE MPLS. 62

4.1. PRINCIPALES CARACTERÍSTICAS DE IP MULTICAST SOBRE REDES MPLS. 62
4.2. FORWARDING EN CAPA 2 Y CAPA 3 EN UN NODO. 66
4.3. CREACIÓN DE LSPs PARA TRÁFICO IP MULTICAST. 67
4.3.1. REQUEST DRIVEN. 67
4.3.1.1. MENSAJES DE CONTROL DE RUTEO MULTICAST. 68
4.3.1.2. MENSAJES DE CONTROL MULTICAST RESOURCE RESERVATION. 68
4.3.2. TOPOLOGY DRIVEN. 69
4.3.3. TRAFFIC DRIVEN. 69
4.4. PIGGY-BACKING. 69
4.5. EXPLICIT ROUTING. 71
4.6. DIFFSERV. 71
4.7. REDES MULTIACCESO. 71
4.8. MECANISMOS DE CONTROL PARA LA DISTRIBUCIÓN DE ETIQUETAS. 73
4.9. CONSIDERACIONES DE SEGURIDAD. 73
4.10. DISTRIBUCIÓN DE ETIQUETAS EN PIM-SM. 73
4.11. PROTOCOLOS IP MULTICAST DENSE-MODE EN IP MPLS. 76

5. AGGREGATED MULTICAST TREE. 79

5.1 ESQUEMA DE DISTRIBUCIÓN IP MULTICAST. 80
5.2 AGGREGATED IP MULTICAST. 82
5.2.1. GROUP-TREE MATCHING. 85
5.2.2 ARQUITECTURA ASSM, AGGREGATED SOURCE SPECIFIC MULTICAST. 86
5.2.3 PROTOCOLO ASSM, AGGREGATED SOURCE SPECIFIC MULTICAST. 87
5.2.3.1. SOURCE MA ROUTERS. 88
5.2.3.2. SOURCE ADVERTISEMENT. 88
5.2.3.3. PROCESO DE SUSCRIPSIÓN DE UN HOST LOCAL A UN CANAL MULTICAST. 88
5.2.3.4. DES-SUBSCRIPCIÓN DE HOST LOCALES. 89
5.2.3.5. CAMBIO DE AGGREGATED TREES. 90
5.2.3.6 PROCEDIMIENTO DE MATCHING GRUPO IP MULTICAST- ÁRBOL DE DISTRIBUCIÓN. 91
5.2.3.6.1 FUNCIÓN OVERHEAD. 92
5.2.3.6.2 DYNAMIC TREE SHARING WITH AGGREGATION OVERHEAD THRESHOLD CONTROL. 95
5.2.3.6.3 MÉTRICAS DE PERFORMANCE. 97

6. ANÁLISIS DE TÉCNICAS Y MECANISMOS DE QOS SOBRE REDES IP MULTICAST. 100

6.1. TÓPICOS QoS EN IP MULTICAST. 100

6.1.1. PROBLEMAS DE RUTEO MULTICAST. 104
6.1.2. RECUPERACIÓN ANTE FALLAS EN ESQUEMAS MULTICAST CON SOPORTE DE QoS. 109
6.2. PROTOCOLOS, MÉTODOS Y ESQUEMAS MULTICAST CON SOPORTE DE REQUERIMIENTOS DE QOS. 111
6.2.1. ANÁLISIS DEL MÉTODO AQoSM. 111
6.2.1.1. PERFORMANCE DE AQoSM Y COMPARACIÓN CON IP MULTICAST NATIVO. 117
6.2.1.1.1. DESCRIPCIÓN DE LAS COSIDERACIONES Y MÉTODOS A SER UTILIZADOS EN LA COMPARACIÓN DE AQoSM CON IP MULTICAST NATIVO. 117
6.2.1.1.2. COMPARACIÓN DE AQoSM E IP MULTICAST NATIVO. 121
6.2.1.2. CONCLUSIONES. 122
6.2.2. ANÁLISIS DEL PROTOCOLO QoSMIC. 124
6.2.2.1. DESCRIPCIÓN DEL PROTOCOLO. 125
6.2.2.2. DESCRIPCIÓN DE MESAJES Y MECANISMOS QoSMIC. 127
6.2.2.3. MECANISMOS DE BUSQUEDA DE CANDIDATOS, SEARCH PHASE. 128
6.2.2.4. ESTABLECIMIENTO DE LA CONEXIÓN. 129
6.2.2.5. DEJAR UN GRUPO. 130
6.2.2.6. SELECCIÓN DEL CANDIDATO EN MULTICAST TREE SEARCH. 130
6.2.2.7. CONCLUSIONES. 132
6.2.3. ANÁLISIS DEL PROTOCOLO QMRP. 132
6.2.3.1. MODELO DE RED. 133
6.2.3.2. DESCRIPCIÓN DEL PROTOCOLO. 133
6.2.3.2.1. ESTADOS DE QMRP. 135
6.2.3.3. QMRP-m. 140
6.2.3.4. PERFORMANCE DE QMRP Y COMPARACIÓN CON QoSMIC. 142
6.2.3.5. CONCLUSIONES. 147
6.2.4. ANÁLISIS DEL PROTOCOLO SoMR. 148
6.2.4.1. MODELO DE RED. 149
6.2.4.2. DESCRIPCIÓN DEL PROTOCOLO. 149
6.2.4.3. SoMR-m 154
6.2.4.4. PERFORMANCE DE SoMR Y COMPARACIÓN CON QoSMIC. 156
6.2.4.5. CONCLUSIONES. 159
6.2.5. ANÁLISIS DEL PROTOCOLO RIMQoS. 160
6.2.5.1. DESCRIPCIÓN DEL PROTOCOLO. 161
6.2.5.1.2. OPERACIÓN INTRA-DOMAIN. 162
6.2.5.2. RIMQoS GENERALIZADO. 165
6.2.5.3. EVALUACIÓN DE LA PERFORMANCE DE RIMQoS. 167
6.2.5.4. CONCLUSIONES. 168
6.2.6. CONCLUSIÓN. 168

7 GLOSARIO. 171

8. BIBLIOGRAFÍA. 173

ÍNDICE DE FIGURAS

Figura 1.2.1.- Formato Dirección LAN. 12
Figura 1.2.2.- Ejemplo del Mapeo de Direcciones de Capa 3 en Direcciones de Capa 2 (Ref.:) 12
Figura 1.2.3.- Mapeo de Direcciones. 13
Figura 1.3.1.1.- Ejemplo de Source Tree (Ref.:) 14
Figura 1.3.2.1.- Ejemplo Topología Shared Tree (Ref.:). 15
Figura 1.4.1.- Ejemplo Funcionamiento RFP. 17
Figura 1.5.1.- Formato Mensaje IGMPv1 18
Figura 1.5.2.1- Formato Mensaje IGMPv2 19
Figura 1.6.1.- Ejemplo Ruteo DVRMP. 22
Figura 1.7.2.1.- Ejemplo Árbol de Distribución PIM-DM. 24
Figura 1.9.2.1.1.- Ejemplo SPTs para las Redes N4 y N3 29
Figura 1.10.2.1.- Funcionamiento de MSDP entre RPs (Ref.:). 33
Figura 1.12.1.- Ciclo de Vida de una Sesión Multicast (Ref.:). 36

Figura 2.2.1.1.- Ejemplo de Tabla de Forwarding. 42
Figura 2.2.1.2.- Label.Switching 42
Figura 2.2.1.1.1.- Shim Label Header. 43
Figura 2.2.2.1.- Componente de Control. 44
Figura 2.2.2.2.- Bindings Upstream y Downstream. 44
Figura 2.5.1.- Formato de una Entrada en la Pila de Etiquetas. 49
Figura 2.7.1.- Codificación de Etiqueta en BGP-4 55
Figura 3.1.1.- Arquitectura Recomendación H.323 (Ref.:) 58
Figura 3.2.1.- Estructura Sistema MPEG-4 (Ref.:) 58
Figura 3.3.1.- Stack de Protocolos para Transmisión Multimedia sobre Internet (Ref.:) 60
Figura 4.2.1.- Forwarding Multicast en Capa 2 y Capa 3. 66
Figura 4.3.1.1.- Creación de LSPs Mensajes de Control Recibidos. 67
Figura 4.3.1.2.- Creación de LSPs Mensajes de Control Enviados. 68
Figura 4.10.1.- Ejemplo PIM-SM en MPLS. 74
Figura 4.10.2.- Ejemplo de Piggybacking para un LSR Rd que se une a un Árbol Source Tree (S,G). 75
Figura 4.10.3.- Ejemplo de Piggybacking para un LSR Rd que se une a un Árbol Shared Tree (*,G). 75
Figura 4.11.1.- Ejemplo de IP Unicast utilizando Método de Control Topology Driven. 76
Figura 5.1.1.- Esquema Source Specific Tree. 80
Figura 5.1.2.- Esquema Shared Tree Unidireccional. 80
Figura 5.1.3.- Esquema Shared Tree Bi-direccional. 81
Figura 5.2.1.- Ejemplo Aggregated Multicast. 82
Figura 5.2.2.1.- Arquitectura ASSM. 87
Figura 5.2.3.3.1.- Esquematización del Proceso de Subscripción del Receptor R al grupo (S,G). 88
Figura 5.2.3.4.1.- Esquematización del Proceso de Des-Subscripción del Receptor R al grupo (S,G). 89
Figura 5.2.3.5.1.- Ejemplo de Operación de Cambio o Intercambio de Arbol de Distribución. 90
Figura 6.1.1.- Técnicas de Red para la Provisión de QoS (Ref:). 101
Figura 6.1.1.1.- Componentes de Ruteo Multicast (Ref:). 106
Figura 6.1.2.1.- Aspectos Necesarios en un Mecanismo de Recuperación ante Fallas (Ref:). 110
Figura 6.2.1.1.- Pedido de Inclusión de un Nuevo Grupo Multicast. 113
Figura 6.2.1.2.- Envío de Paquetes sobre Árbol Multicast. 113
Figura 6.2.1.1.1.1.- Esquema de Red Propuesto. 118
Figura 6.2.1.1.2.1.- Gráfico Número de Árboles MPLS vs Número Promedio de Grupos Activos Concurrentes. 122
Figura 6.2.2.1.1.-Procedimiento QoSMIC de Identificación de Rama del Árbol Multicast (Ref.:). 125
Figura 6.2.2.1.2.- Procedimientos QoSMIC Local Search y Multicast Tree Search (Ref.:). 126
Figura 6.2.2.3.1.- Procedimiento de Take-Over (Ref.:). 129
Figura 6.2.3.2.1.- Ejemplo QMRP (Ref:). 135
Figura 6.2.3.2.1.1.- Transición de Estados en QMRP (Ref:). 135
Figura 6.2.3.2.1.2.a.- Selección de Camino en QMRP (Ref:) 138
Figura 6.2.3.2.1.2.b.- Selección de Camino en QMRP (Ref:). 139
Figura 6.2.3.2.1.2.c.- Selección de Camino en QMRP (Ref:). 139
Figura 6.2.3.2.1.2.d.- Selección de Camino en QMRP (Ref:). 140
Figura 6.2.3.3.1.- Search Tree en QMRP-2 (Ref.:). 141
Figura 6.2.3.4.1.a.- Tasas de Éxito considerando un Árbol de Distribución de 180 Nodos (Ref.:) 144
Figura 6.2.3.4.1.b.- Tasas de Éxito considerando un Árbol de Distribución de 45 Nodos (Ref.:) 145
Figura 6.2.3.4.1.c.- Tasas de Éxito considerando un Árbol de Distribución de 6 Nodos (Ref.:). 145
Figura 6.2.3.4.2.a.- Overhead de Mensajes considerando un Árbol de Distribución de 6 Nodos (Ref.:). 146
Figura 6.2.3.4.2.b.- Overhead de Mensajes considerando un Árbol de Distribución de 45 Nodos (Ref.:). 146
Figura 6.2.3.4.2.c.- Overhead de Mensajes considerando un Árbol de Distribución de 6 Nodos (Ref.:). 147
Figura 6.2.4.2.1.- Mecanismo de Detección de Loops (Ref:). 154
Figura 6.2.4.3.1.- Ejemplo de SoMR – 3 (Ref.:). 155

Figura 6.2.4.4.1.a.- Comparación de Performance entre SoMR-3 y QoSMIC (Ref.:) 158
Figura 6.2.4.4.1.b.- Comparación de Performance entre SoMR-3 y QoSMIC (Ref.:) 159
Figura 6.2.5.1.2.1.- Procedimiento de Join (Ref.:). 163
Figura 6.2.5.1.2.2.- Procedimiento de Join, donde el Camino Existente no Cumple con el Delay, caso a. (Ref.:). 164
Figura 6.2.5.1.2.3.- Procedimiento de Join, donde el Camino Existente no Cumple con el Delay, caso b. (Ref.:). 164
Figura 6.2.5.2.1.- Procedimiento de Cálculo de un Nuevo Camino con Requerimientos de Jitter y Delay (Ref.:). 166

ÍNDICE DE TABLAS

Tabla 4.11.1.- Tabla de Switching de Etiquetas en R3. 76
Tabla 6.1.1.- Comparación entre Protocolos SPR y MPR (Ref:). 103
Tabla 6.2.2.2.1.- Funciones de los Nodos bajo el Protocolo QoSMIC (Ref.:). 127
Tabla 6.2.2.2.2.- Descripción de los Mensajes en QoSMIC (Ref.:). 127
Tabla 6.2.3.2.1.1.- Pseudo Código de Estados del Protocolo QMRP (Ref:). 138

INTRODUCCIÓN.

El presente trabajo de Tesis propone presentar un análisis sobre los protocolos y mecanismos que permiten el manejo y administración de requerimientos de calidad de servicio, QoS, sobre flujos de datos IP Multicast. Dentro de estos podemos citar el caso de los requermientos aditivos como son el *delay* y el *jitter*, y requerimientos no aditivos como lo son el ancho de banda y la capacidad de almacenamiento en los *buffers* de los elementos de la red.

El trabajo comienza con una descripción de las tecnologías IP Multicast, IP MPLS, IP Multicast sobre redes IP MPLS, Aggregated Multicast Tree, también se presenta una breve descripción de las metodologías de compresión de vídeo sobre redes IP.

La Tesis culmina con un análisis de distintos protocolos y mecanismos que proporcionan el soporte de calidad de servicio sobre flujos de datos IP Multicast.

A continuación se detallan los principales temas a ser cubiertos por el presente trabajo de Tesis:

Multicast.

a. Protocolos de Ruteo.

b. Métodos de Ingresar a grupos Multicast.

c. Direccionamiento.

d. Topología.

e. Aggregated IP Multicast.

MPLS.

a. Protocolos.

b. Distribución de Etiquetas.

c. Topología.

Métodos de Codificación.

a. Descripción de los métodos de codificación.

b. Ventajas y desventajas de los métodos de codificación.

Multicast en MPLS.

a. Descripción.

b. Topología.

c. Arquitectura.

OBJETIVO.

Actualmente, las aplicaciones que no presentan requerimientos de Real Time, como Noticias, Distribución de *Software*, etc.; pueden ser soportadas utilizando técnicas de *Web Caching*, las cuales evitan las complicaciones que pueden presentar su transmisión utilizando IP Multicast. La misma técnica de distribución puede utilizarse para el caso de aplicaciones *Real Time* no interactivas, como el caso de *Video on Demand*. En cambio, para el caso de aplicaciones en *Real Time* interactivas, como vídeo conferencia, juegos en red, etc; no podrían ser eficientemente soportadas por técnicas Unicast como el *Web Caching*. Estas requieren, en general, NxN conexiones, cada una con sus requerimientos específicos de QoS. Por esto, que para este tipo de aplicaciones, es deseable el poder contar con una tecnología que soporte la provisión de parámetros de QoS sobre redes IP Multicast.

Basado en lo anterior, el objetivo del presente trabajo de Tesis es el de analizar distintos protocolos y mecanismos que permiten tener en cuenta los requerimientos de calidad de servicio impuestos por distintos flujos y aplicaciones basadas sobre IP Multicast. Para llegar a este análisis final, a través del desarrollo de la Tesis, se presentarán distintas técnicas, arquitecturas, protocolos y mecanimos que nos permitirán entender y desarrollar los métodos por los cuales se logra la capacidad de poder administrar los parámetros de calidad de servicio sobre el tráfico de redes IP Multicast.

1. IP MULTICAST.

Multicast se basa en el concepto de grupo. Un grupo arbitrario de receptores esperan recibir un particular flujo de datos. Los grupos no tienen ningún sentido físico de ubicación, pueden localizarse en cualquier parte de Internet. Los *Host*s que estén interesados en ser receptores de un flujo de datos, deben unirse al Grupo Multicast usando la dirección IGMP del grupo. Los *host*s deben ser miembros del grupo para recibir el flujo de datos. Las direcciones IP Multicast especifican un conjunto arbitrario de *host*s que pertenecen a un grupo Multicast, los cuales recibirán el tráfico enviado a ese grupo.

IP Multicast es una tecnología basada en la optimización del uso del ancho de banda, entregando simultáneamente un sólo flujo de información a los destinatarios del mismo. Entrega tráfico desde una fuente, *source*, a los múltiples receptores, *receivers*, sin agregar carga adicional en la fuente o los receptores, utilizando el menor ancho de banda de la red cualquiera sea la tecnología.

1.1. DIRECCIONAMIENTO MULTICAST.

En IPv4 las direcciones IP Multicast son direcciones clase D:

224.0.0.0 a 239.255.255.255

Este rango de direcciones sólo se utiliza para identificar la dirección de grupo o la dirección IP de destino del tráfico IP Multicast. La dirección de la fuente para el datagrama Multicast es su dirección IP Unicast.

El rango de direcciones:

224.0.0.0 a 224.0.0.255

se definió como rango para ser utilizado dentro de un segmento LAN, direccionamiento Multicast Privado. Los paquetes con estas direcciones no serán ruteados por *router*s en Internet. Se transmiten con un *TTL* de 1. Los protocolos de red usan estas direcciones para el descubrimiento automático de los *router*s y para transmitir información sobre la asignación de rutas IP.

El rango de direcciones:

224.0.1.0 a 238.255.255.255

es denominado *Globally Scoped Address*. Estas pueden utilizarse para direccionar datagramas IP Multicast entre *host*s a través de Internet. Algunas de estas direcciones han sido reservadas para el uso de aplicaciones IP Multicast específicas. Puede encontrarse información sobre este rango de direcciones IP en el siguiente sitio Web:

http://www.isi.edu/in-notes/iana/assignments/multicast-dirección

El rango de direcciones:

239.0.0.0 a 239.255.255.255

es denominado de Alcance Limitado o *Administratively Scoped Addresses*. Están definidas en la RFC 2365, para ser utilizadas en un grupo local u organización. Los *router*s son configurados con filtros para prevenir tráfico IP Mullticast, en este rango de direcciones, hacia fuera de un Sistema Autónomo o cualquier dominio de usuario definido. Dentro de un Sistema Autónomo, o dominio, el rango de direcciones de Alcance Limitado puede ser dividido en sub-redes de modo que puedan definirse límites Multicast locales. Esto también permitirá la reutilización de direcciones entre dominios más pequeños.

La RFC 2770 propone que el rango de direcciones:

233.0.0.0/8

Este rango se reserva para las direcciones definidas estáticamente por organizaciones que posean un Sistema Autónomo, *AS,* reservado. El número del *AS* se encuentra embebido dentro del segundo y tercer octeto del rango.

1.2. DIRECCIONES MULTICAST DE CAPA 2.

Las placas de Red, *NICs*, en un segmento de LAN sólo recibirán paquetes destinados a su dirección MAC, o a la dirección MAC de *broadcast*. Se implementaron diferentes medios para que los múltiples *host*s pudieran recibir el mismo paquete y puedan así diferenciar entre los grupos IP Multicast.

El standart IEEE 802.3 implementa lo anterior a través de la utilización del bit diferencia entre paquetes *Broadcast* y Multicast. La siguiente figura muestra esta situación:

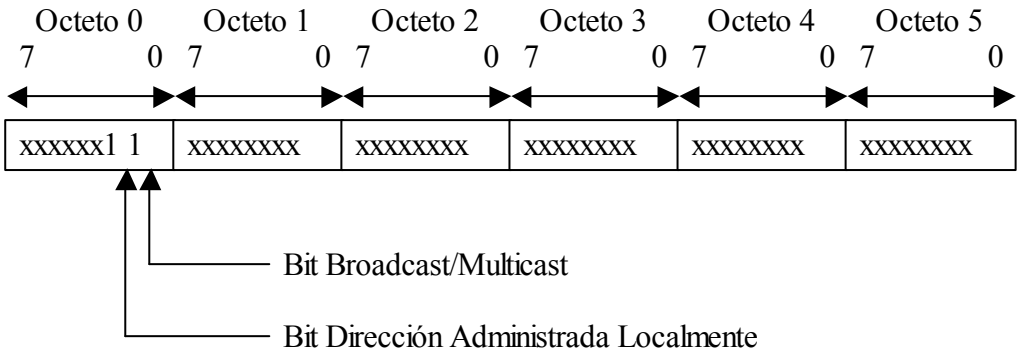


Figura 1.2.1.- Formato Dirección LAN.

IP Multicast hace uso de esta capacidad para transmitir paquetes de IP a un grupo de *host*s en un segmento de LAN.

El IANA posee un bloque de direcciones de MAC que empiezan con 01:00:5E en hexadecimal. La mitad de este bloque se asigna para las direcciones IP Multicast. Esto da un rango:

0100.5e00.0000 a 0100.5e7f.ffff

de direcciones MAC validas para tráfico IP Multicast.

Esta asignación permite tener 23 bits en la dirección MAC disponibles para direccionar un grupo IP Multicast dentro de un entorno LAN. Así se mapean los 23 bits más bajos de la dirección IP Multicast del grupo con los 23 bits disponibles en la dirección MAC. Debido a que los tres bits superiores de la dirección IP Multicast son desechados de este mapeo, la dirección resultante no es única. De hecho, 32 direcciones IP Multicast diferentes de grupo se mapean a la misma dirección MAC.

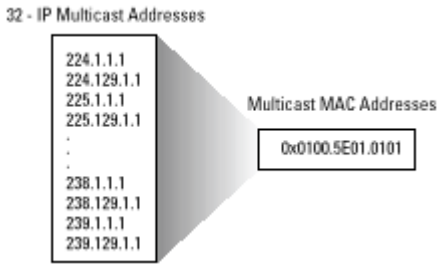


Figura 1.2.2.- Ejemplo del Mapeo de Direcciones de Capa 3 en Direcciones de Capa 2 (Ref.:1)

[1] Tomado Referencia Bibliográfica 3.

Debido a que no todos los 28 bits de la dirección IP Multicast de Capa 3 pueden ser mapeados en los 23 bts disponibles en la dirección de capa 2 (*MAC Address*), da como resultado que 32 direcciones de capa 3 no sean mapeadas en forma unívoca en direcciones de capa 2. Esto causará problemas cuando se tengan dos *host* en una misma LAN que mapean su dirección *broadcast* de capa 3 en la misma dirección de capa 2, lo cual puede impactar en la *performance* de los elementos de la red, al tener que analizar paquetes que luego tendrán que descartar.

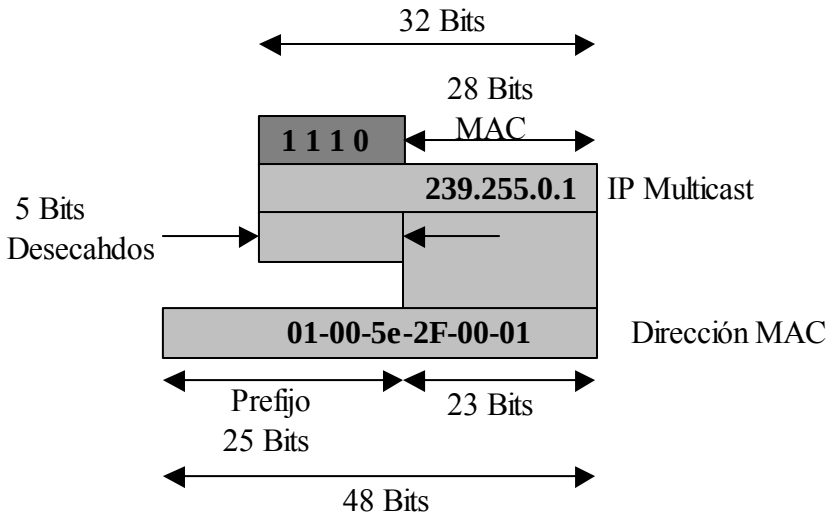


Figura 1.2.3.- Mapeo de Direcciones.

1.3. ÁRBOLES MULTICAST DE DISTRIBUCIÓN.

Los *router*s Multicast crean árboles de distribución que controlan el camino de los paquetes IP Multicast a través de la red para entregar el tráfico a todos los receptores. Los dos tipos básicos de árboles de distribución IP Multicast son:

- *Source Tree*.
- *Shared Tree*.

1.3.1. SOURCE TREE.

La forma más simple que toma un árbol de distribución en IP Multicast es *Source Tree*, formando una topología tipo *Spanning Tree* entre las fuentes y los receptores. Esta topología utiliza el concepto del camino más corto, se lo llama también *SPT* (*Shortest Path Tree*).

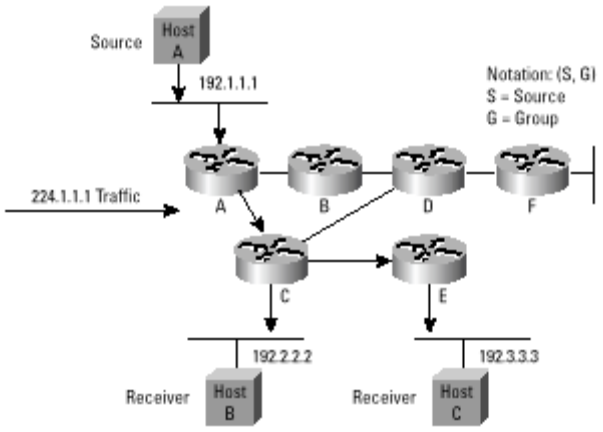


Figura 1.3.1.1.- Ejemplo de Source Tree (Ref.:2)

La notación utilizada para especificar un *host* dentro de esta topología es:

(S,G); donde: S: indica la dirección IP Unicast de la fuente.
G: indica la dirección IP Multicast del grupo.

De aquí surge que existe un *SPT*, denotado por el par (S,G) para cada flujo Multicast entre la fuente y el grupo.

1.3.2. SHARED TREE.

A diferencia de la topología *Source Tree*, la topología *Shared Tree* utiliza una única raíz, *root*, situada en algún punto de la red. Dependiendo del protocolo IP Multicast utilizado, esta es a menudo denominada *Rendezvous Point*, *RP*, o *Core*. Al utilizar esta topología de distribución, las fuentes envían el tráfico al *RP* definido por el protocolo IP Multicast, y este lo direcciona al grupo IP Multicast destino.

Debido a que todas las fuentes utilizan la misma raíz, *root*, la notación utilizada para este caso es la siguiente:

(*,G); donde: *, indica todas las fuentes.
G, indica la dirección IP Multicast del grupo.

[2] Tomado Referencia Bibliográfica 3.

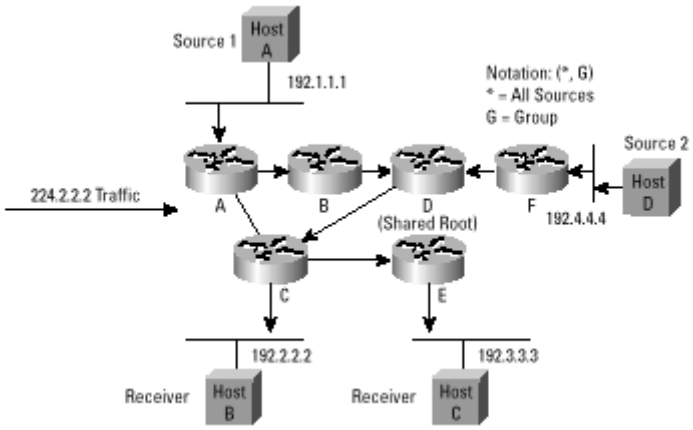


Figura 1.3.2.1.- Ejemplo Topología Shared Tree (Ref.:3).

Esta topología de distribución IP Multicast puede dividirse en dos tipos:

- Direccional.
- Bidireccional.

SHARED TREE BIDIRECCIONAL.

El tráfico IP Multicast puede ser transmitido en dirección *upstream*, hacia arriba, o *downstream*, hacia abajo, a través del árbol de distribución Multicast, para alcanzar los receptores del grupo IP Multicast al cual este dirigido.

SHARED TREE UNIDIRECCIONAL.

Esta topología de distribución sólo permite el flujo de tráfico IP Multicast en sentido *downstream*, desde la raíz hacia los receptores. Las fuentes de tráfico IP Multicast deben utilizar algún otro medio para primero enviar el tráfico IP Multicast al *RP*, para que luego este envíe el tráfico a los receptores dentro del grupo IP Multicast.

Un método para enviar el tráfico desde las fuentes al *RP* puede ser la utilización de *SPT*. Este método es utilizado por el protocolo *PIM*.

Otro método es utilizar ruteo IP Unicast entre la fuente de tráfico IP Multicast y el *RP*. El protocolo CBT utiliza esta metodología.

[3] Tomado Referencia Bibliográfica 3.

1.4. MULTICAST FORWARDING.

En el modelo IP Multicast, una fuente envía tráfico a un grupo de *host*s representados por una dirección IP Multicast. En el campo *Address* del paquete IP se inserta esta dirección IP Multicast.

Los *router*s no pueden utilizar esta dirección IP para realizar el *forwarding* de los paquetes IP. Virtualmente todos los protocolos de ruteo IP Multicast utilizan algún método *RPF*, *Reverse Path Forward*, o realizan el chequeo de la interfase de entrada del paquete como primer mecanismo para decidir si deben *"forwardear"* o desechar el paquete IP Multicast. Por lo cual, cuando un paquete IP Multicast arriba a un *router*, este realiza un chequeo *RFP*, si el chequeo resulta exitoso, el paquete es *"forwardeado"*, de lo contrario será desechado por el *router*.

Para tráfico fluyendo en sentido *dowstream* en un árbol IP Multicast, el funcionamiento del mecanismo *RFP* es el siguiente:

1. El *router* examina la dirección IP origen del paquete Multicast para determinar si el mismo arribó en una interfase que esta en el camino de reversa*, reverse path back*, hacia la fuente.

2. Si el chequeo anterior es correcto, el paquete es *"forwardeado"* por el *router*.

3. Si el chequeo del punto 1 no es correcto, el paquete es descartado por el *router*.

El mecanismo por el cual el *router* determina que interfase se encuentra en el camino de reversa hacia la fuente dependerá del protocolo de ruteo IP Multicast. En el caso del protocolo DVMRP, este utiliza una tabla de ruteo IP Multicast en forma separada de la tabla de ruteo IP Unicast, y esta es utilizada por el *router* para realizar la verificación *RFP*. En el caso de los protocolos PIM y CBT, estos utilizan la misma tabla de ruteo IP Unicast para realizar el *RFP*.

Tabla de Ruteo Multicat	
Red	Interface
151.10.0.0/16	S1
198.14.32.0/24	So
204.1.16/24	Eo

El paquete ingreso por una Interfase incorrecta, será descartado

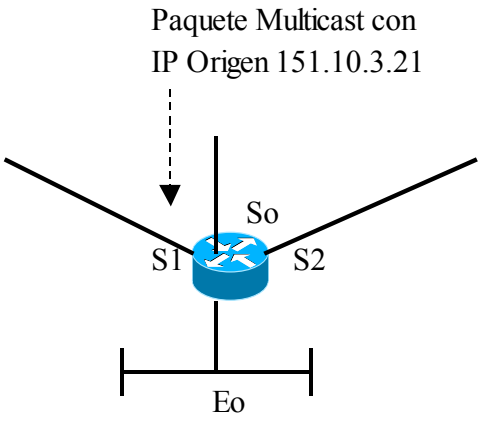


Tabla de Ruteo Multicat	
Red	Interface
151.10.0.0/16	S1
198.14.32.0/24	So
204.1.16/24	Eo

El paquete ingreso por una Interfase correcta, será forwardeado

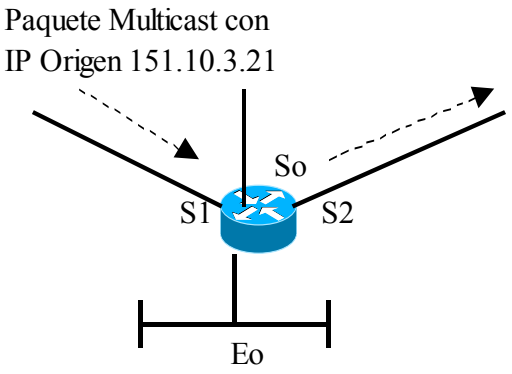


Figura 1.4.1.- Ejemplo Funcionamiento RFP.

En el caso de árboles *Shared Tree* bidireccionales, no se hace una distinción entre las interfases de entrada y salida, *Dowstream* y *Upstream*, ya que el tráfico puede fluir en ambas direcciones a través del árbol IP Multicast.

El chequeo de cada paquete IP Multicast a través del mecanismo *RFP* resulta en un aumento del procesamiento del *router*, lo cual atenta contra la *performance* del mismo. Por tal motivo, en algunos *router*s se utiliza una tabla de memoria *Cache* la cual se actualiza cuando se verifica algún cambio, esto disminuye la carga de procesamiento debida al chequeo *RFP*.

1.5. IGMP (INTERNET GROUP MANAGEMENT PROTOCOL).

IGMP es utilizado como protocolo para la registración de un *host* con su *router* local, especificando que se "unen" a un determinado grupo IP Multicast y así comenzar a recibir tráfico destinado a ese grupo. Los *hosts* informan su pertenencia al grupo enviando mensajes IGMP al *router* IP Multicast local. Bajo IGMP, los *router*s "escuchan" los mensajes y periódicamente envían mensajes preguntando si los miembros del grupo se encuentran activos o inactivos. Utilizando IGMPv2, los *hosts* pueden enviar mensajes a su *router* local indicando que ya no

pertenecen más a un determinado grupo IP Multicast, y de esta forma dejan de recibir el tráfico destinado a dicho grupo.

Utilizando la información obtenida a través de los mensajes IGMP, los *router*s mantienen una lista con los grupos IP Multicast y sus miembros junto con la información de la interfase por la cual se los alcanza. Un grupo se considera activo, sobre una determinada interfase, cuando existe al menos un *host* que pertenezca a dicho grupo y al cual se lo alcanza por esta interfase.

1.5.1. IGMP Versión 1.

Esta especificado en la RFC 1112.

Los mensajes IGMP son trasmitidos dentro de los datagramas IP y están denotados por el número 2 dentro del campo *Protocol Type* del paquete IP. Son transmitidos con el campo *TTL* seteado en 1 y no son *"forwardeados"* por los *router*s fuera del ambiente LAN.

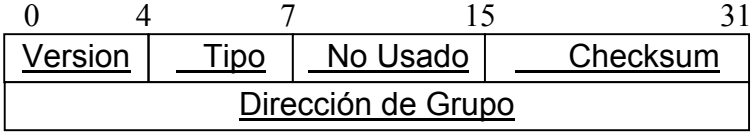


Figura 1.5.1.- Formato Mensaje IGMPv1.

Existen sólo dos tipos de mensajes:

- *Membership Querry*.
- *Membership Report*.

Los *host*s envían un mensaje *Membership Report* indicando que están interesados en "unirse" a un grupo en particular. El *router* envía periódicamente mensajes *Membership Querry* con la finalidad de averiguar si existe al menos un *host* del grupo IP Multicast interesado en recibir el tráfico destinado al grupo. Cuando no recibe respuesta a tres mensajes IGMP consecutivos, el *router* dejará de enviar tráfico IP Multicast a ese grupo en particular.

IGMPv1 utiliza un modelo de Pregunta/Respuesta, por el cual los *router*s determinan si un determinado grupo esta activo o no. Este modelo es implementado dentro de un ambiente LAN.

1.5.1.1. REPORT SUPRESION MECHANISM.

Este mecanismo ayuda a reducir la cantidad de tráfico IGMP en una determinada red LAN al mínimo necesario. A continuación se detalla el funcionamiento de este mecanismo:

1. Cuando un *host* recibe un mensaje IGMP *Membership Querry*, este inicializa un *Timer* por cada grupo IP Multicast al cual pertenece. Cada *Timer* es inicializado con un valor entre 0 y el máximo tiempo de respuesta. El valor por *default* es de 10 segundos.

2. Cuando el *Timer* llega a su fin, el *host* envía un mensaje IP Multicast IGMP *Membership Report* asociado al grupo IP Multicast activo con el *Timer* de referencia.

3. Si un *host* "escucha" a otro *host* enviando un mensaje IGMP *Membership Report*, cancela el *Timer* asociado a este grupo y deja de enviar mensajes IGMP *Membership Report* a dicho grupo.

1.5.1.2. IGMPv1 QUERIER.

La RFC 1112 no especifica como debe ser elegido el IGMP *Querier*, de esta manera un *router* dentro de un ambiente LAN sería el responsable por el envío de los mensajes IGMP *Queries*. IGMPv1 deja esta tarea para ser resuelta en la capa 3, capa IP, por el protocolo de ruteo IP Multicast utilizado.

1.5.1.3. IGMPv1 JOIN PROCESS.

Para reducir la latencia producida por la asociación a un grupo IP Multicast, *Join Process*, no es necesario esperar por el envío, por parte del *router*, del siguiente mensaje *Membership Query* antes que un *host* envíe el correspondiente mensaje *Membership Report* para unirse a un determinado grupo IP Multicast. Así se reduce la latencia asociada a este proceso.

1.5.2. IGMP Versión 2.

Esta especificado en la RFC 2236.

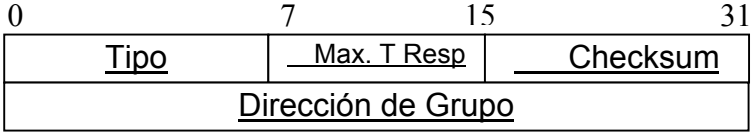


Figura 1.5.2.1- Formato Mensaje IGMPv2.

Existen cuatro tipos de mensajes:

- *Membership Query*.
- *Version1 Membership Report*.
- *Version2 Membership Report*.
- *Leave Group*.

Esta versión funciona básicamente en forma idéntica a la versión 1, la diferencia principal radica en el mensaje de *Leave Group*. Así los *host*s comunican activamente al *router* IP Multicast local que están dejando el grupo. Entonces el *router* IP Multicast local envía un mensaje *Membership Query*, dirigido al grupo IP Multicast, preguntando si existe algún *host* del grupo interesado en seguir recibiendo tráfico IP Multicast. Si no recibe respuesta, dejará de enviar tráfico IP Multicast hacia el grupo. Esta funcionalidad reduce el tráfico de control generado en la versión 1.

Los mensajes de *Query Membership* y *Report Membership* son idénticos a los mensajes de IGMPv1 con dos excepciones. Una diferencia es que en IGMPv2 los mensajes de *Query Membership* se dividen en dos categorías:

- *General Queries*: análogo al definido en IGMPv1.
- *Group-Specific Queries*: son dirigidos a un grupo en particular.

La segunda diferencia radica que el hecho de que los mensajes *Membership Report* en IGMPv2 tienen un tipo de código diferente al de IGMPv1.

IGMPv2 agrega básicamente las siguientes mejoras con respecto a IGMPv1:

Querier Election Process.
Provee la posibilidad que los *router*s elijan al *Querier Router* sin tener que depender del protocolo de ruteo IP Multicast para realizar este proceso.

Máximo Tiempo de Respuesta.
Permite al *Querier Router* especificar el máximo tiempo de respuesta, permitiendo controlar la latencia producida por el proceso de “dejar” un grupo y el *burtiness* producido por dichos mensajes, lo cual produce tráfico innecesario en la red.

Group-Specific Query Messages.

Permite al *Querier Router* efectuar el proceso de consulta sobre un grupo determinado en lugar de enviar mensajes a todos los grupos IP Multicast activos. Además estos mensajes permiten reducir aun más la latencia generada por el proceso de dejar un grupo ya que utilizan un menor valor de tiempo de respuesta.

Leave Group Messages.

Permiten a los *host*s informar al *router* IP Multicast local que dejan de recibir tráfico de un determinado grupo. En conjunto con la determinación del Máximo Tiempo de Respuesta, permiten reducir a sólo unos segundos la latencia generada por un *host* que deja de pertenecer a un dado grupo IP Multicast.

1.6. DISTANCE VECTOR MULTICAST ROUTING PROTOCOL, DVMRP.

Tiene las siguientes características principales:

- Es un protocolo de ruteo *Distance Vector* similar a RIP.
- Envía Uptades periódicos cada 60 segundos.
- Considera como métrica infinita a 32 *hops*.
- Es un protocolo *Classless*.
- Los *router*s necesitan mantener información sobre los *router*s adyacentes

La tabla de ruteo utilizada por este protocolo se almacena en los *router*s en forma separada a la tabla de ruteo IP Unicast. Estas tables son tenidas en cuenta para:

- Construir los árboles de distribución IP Multicast.
- Realizar el mecanismo de chequeo *RPF*.

La métrica se basa en la cuenta de *hops*, siendo el máximo de 32 *hops*. Al igual que RIP, los *router*s incrementan en un *hop* la métrica al publicar las rutas que aprendieron a partir de sus *router*s vecinos.

La utilización del mecanismo de *Poison Reverse* es diferente al que se tiene en el caso de RIP, en este se utiliza para informar al *router* adyacente, en sentido *upstream*, que el *router* es *downstream* para el sentido de transmisión del tráfico en el árbol de distribución IP Multicast.

DVMRP es un protocolo *Dense Mode*, utiliza árboles de distribución IP Multicast *SPT* para *"forwardear"* el tráfico IP Multicast. Los árboles de distribución son construidos por los *router*s utilizando el mecanismo *Truncated Broadcast Tree*, el cual se basa en las métricas de la tabla de ruteo de DVMRP. Para esto, los *router*s informan al *router* adyacente en sentido *upstream* que se encuentran en sentido *downstream* enviándole un mensaje *Poison Reverse*, con métrica 32. Así, se informa al *router* en sentido *upstream* que *"forwadee"* el tráfico IP Multicast, de la red fuente, por la interfase en la cual recibió el mensaje *Poison Reverse*, hacia el *router downstream*.

Debido a que es un protocolo *Distance Vector* y al limitado valor máximo de métrica que utiliza, 32 *hops*, este protocolo no es aconsejable para utilizar en grandes redes IP Multicast ni en Internet.

1.6.1. DVMRP MULTICAST FORWARDING.

Como el ruteo del tráfico IP Multicast cumple con el sentido de distribución *upside-down*, la información almacenada en la tabla de ruteo del protocolo DVMRP, es utilizada para determinar si el paquete IP Multicast entrante fue recibido en la interfase correcta. De no ser así, será descartado para prevenir Loops.

Tabla de Ruteo DVMRP		
Red	Interface	Metrica
151.10.0.0/16	E1	4
198.14.32.0/24	So	3
204.1.16/24	Eo	11

El paquete ingreso por una Interfase incorrecta, será descartado

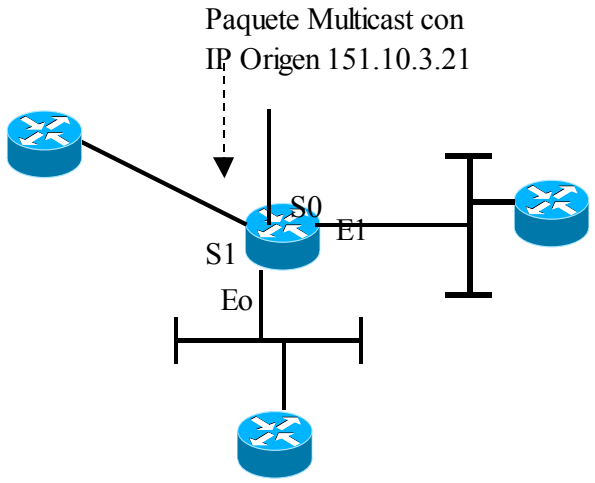


Figura 1.6.1.- Ejemplo Ruteo DVRMP.

En el ejemplo anterior se muestra el caso en que un paquete IP Multicast es recibido sobre una interfase equivocada, este caso no es normal en redes bajo condiciones de operación

estables, y puede ocurrir en casos inmediatamente después que cambia la topología de la red y el protocolo DVMRP aun no haya "convergido".

1.7. PROTOCOL INDEPENDENT MULTICAST, PIM, DENSE MODE.

Independientemente del protocolo de ruteo IP Unicast utilizado para generar la tabla de ruteo IP Multicast, el protocolo PIM utiliza esta información para *"forwardear"* el tráfico IP Multicast.

Utiliza la tabla de ruteo IP Unicast para ejecutar el mecanismo *RFP* y así *"forwardear"* los paquetes IP Multicast, con lo cual no genera una tabla de ruteo IP Multicast, por este motivo no envía ni recibe *Updates* Multicast, reduciendo de esta manera el *overhead*. Es un protocolo *Classless*.

En redes bien diseñadas y con un buen plan jerárquico de asignación de direcciones IP y agregación de rutas, PIM-DM escala mejor que DVMRP. La razón de esto es que PIM-DM utiliza la tabla de ruteo generada por protocolos IP Unicast para realizar el chequeo *RFP*, y así *"forwardear"* el tráfico IP Multicast. No obstante, utiliza el mismo mecanismo de *flood-and-prune* que DVMRP, lo cual puede generar transmisión de tráfico no deseado en la red.

PIM-DM es aconsejable para redes de alta velocidad con gran ancho de banda. No es aconsejable utilizar PIM-DM en redes con gran número de fuentes y grupos IP Multicast, o redes con aplicaciones multimedias basadas en *RTP*, debido a que puede traer problemas de saturación en los enlaces.

1.7.1. PIM NEIGHBOR DISCOVERY.

Al igual que DVMRP, PIM utiliza un mecanismo para establecer los *router*s adyacentes. Para esto envía paquetes de *Hello* cada 30 segundos, valor por *default*, utilizando paquetes IP Multicast con dirección destino 224.0.13.0

Los mensajes de *Hello* contienen un valor de *Holdtime*, el cual indica a los receptores del mismo cuando deben dejar de considerar al *router* adyacente como activo si no reciben otro mensaje de *Hello* una vez expirado dicho valor. Este valor es de 90 segundos, valor por *default*.

Los mensajes de *Hello* son también utilizados para designar al *Designated Router*, *DR*, para los casos de redes con características *Multi-Access*, por ejemplo redes Ethernet. De esta manera, se indica el *router* con la mayor IP, el cual se convertirá en el *DR* de la red.

1.7.2. PIM-DM SOURCE DISTRIBUTION TREES.

Utiliza árboles de distribución *SPTs*, para distribuir el tráfico IP Multicast. Estos árboles de distribución son construidos utilizando un mecanismo *flood-and-prune* en el instante que una fuente Multicast comienza a transmitir. Los *router*s de la red utilizan la información de sus *router*s adyacentes para construir el árbol de distribución. Inicialmente los *router*s adyacentes se consideran que se encuentran dentro del árbol *SPT*, donde la interfase de entrada se considera la interfase en dirección de la fuente, y todos los otros *router*s se consideran en sentido *downstream* hacia la fuente. A esto se lo conoce como *Broadcast Tree*, ya que un *router* envía un paquete IP Multicast a sus vecinos en modo *Broadcast*.

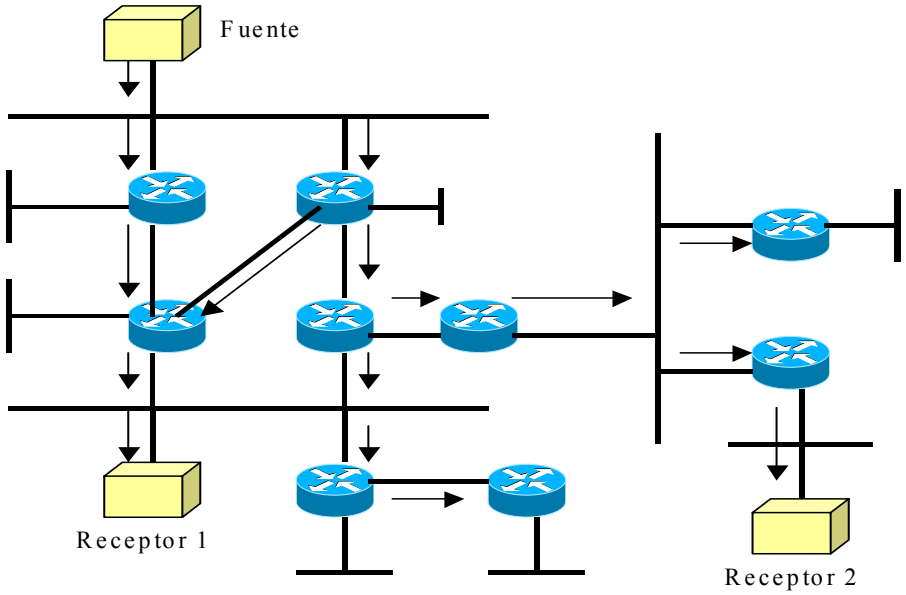


Figura 1.7.2.1.- Ejemplo Árbol de Distribución PIM-DM.

1.7.3. PIM-DM MULTICAST FORWARDING.

Cuando un *router* inicialmente recibe un paquete IP Multicast, el paquete es sometido al chequeo *RFP* para verificar que haya sido recibido por la interfase correcta en dirección a la fuente. Este chequeo es realizado tomando los datos de la tabla de ruteo IP Unicast. Si existe más de una entrada en la tabla de ruteo IP Unicast para una dada fuente, el *router* elegirá sólo una interfase para *"forwardear"* el tráfico IP Multicast y realizar el chequeo *RFP*. La entrada que elegirá será aquella que presente el valor más alto de *next-hop*.

1.7.4. PIM-DM ASSERTS.

PIM utiliza un mecanismo denominado *Assert* para elegir el *router forwarder* para una fuente IP Multicast en particular. Este mecanismo es inicializado según la siguiente regla:

- Si un *router* recibe un paquete IP Multicast vía una interfase en la lista de interfases de salida asociada con la fuente IP Multicast, envía un mensaje PIM *Assert* para resolver cual será el *router* que continuará con el *"forwardeo"* del tráfico IP Multicast.

Cuando el mecanismo *Asser*t es inicializado en una dada interfase, el *router* envía un mensaje *Assert* indicando su métrica a la fuente. Todos los PIM *Router* en la red examinan la métrica para determinar quien tiene la mejor métrica a la fuente IP Multicast. El *router* con la mejor métrica continúa con el *forwarding* del tráfico IP Multicast. Si existe más de un *router* con métricas igualadas, se utiliza el direccionamiento IP de la interfase para determinar quien continúa con el *"forwardeo"* del tráfico, este será el *router* que posea la IP más alta.

1.8. PROTOCOL INDEPENDENT MULTICAST SPARSE MODE, PIM-SM.

Al igual que PIM-DM, utiliza la información almacenada en las tablas de ruteo generadas por los protocolos IP Unicast para realizar el chequeo *RFP*.

No sufre de las ineficiencias de *overhead* provocadas por el mecanismo *flood-and-prune* como en el caso de DVMRP y PIM-DM. Es adecuado para ser utilizado en redes IP Multicast con un gran número de miembros al final de un enlace WAN.

Permite reducir la latencia generada por la utilización de árboles de distribución *Shared Tree* a través de la utilización de *SPTs*.

Es la mejor elección para el caso de redes que utilizan protocolos IP Multicast Intra-Dominios.

1.8.1. EXPLICIT JOIN MODEL.

PIM-SM envía el tráfico IP Multicast sólo a los sitios de la red donde este es requerido. Esto es realizado mediante el mecanismo *Join Model*, por el cual los mensajes *Join* son enviados *hop* por *hop* hacia el nodo raíz del árbol de distribución, este nodo es denominado *RP* en el caso

de árboles de distribución *Shared Tree*, o es el primer *router* directamente conectado a la fuente IP Multicast en el caso de árboles de distribución *SPT*.

A medida que el mensaje *Join* viaja en sentido *Upstream* a través del árbol de distribución, los *router*s que se encuentran en el camino del mismo, establecerán un estado *Multicast Forwarding* tal que el tráfico IP Multicast pueda fluir en sentido *Downstream* a través de la topología del árbol de distribución.

Cuando no se requiere recibir más tráfico IP Multicast, un *router* envía un mensaje PIM *Prune*, en sentido *Upstream*, hacia el nodo raíz del árbol de modo que no continúe enviando tráfico IP Multicast hacia dicho destino. A medida que el mensaje *Prune* PIM viaja en sentido *Upstream*, *hop* por *hop*, a través del árbol de distribución, cada *router* actualiza su estado de *forwarding* Multicast. Esta actualización a menudo resulta en la eliminación del estado de *forwarding* asociado con un grupo o fuente IP Multicast.

1.8.2. PIM-SM SHARED TREES.

La operación de PIM-SM se centra sobre un simple y unidireccional árbol de distribución *Shared Tree*, cuyo nodo raíz es denominado *RP*, *Redezvous Point*.

Los *router*s directamente conectados a un grupo receptor de tráfico IP Multicast, que necesitan recibir tráfico IP Multicast de un determinado grupo, se unen al árbol de distribución *Shred Tree*. Cuando ya no requiere seguir recibiendo tráfico IP Multicast, el *router* realiza una acción de *Prune* del árbol al cual estaba conectado.

Debido a que PIM-SM utiliza un árbol de distribución *Shared Tree* unidireccional, donde el tráfico sólo fluye en sentido *Downstream*, las fuentes IP Multicast deben registrarse con el *RP* para enviar el tráfico a través del árbol de distribución. Esta registración genera el envío de un mensaje *SPT Join* del *RP* hacia la fuente cuando detecta receptores activos pertenecientes al grupo destino del tráfico.

Cuando un grupo deja de recibir tráfico IP Multicast, o una fuente deja de transmitir tráfico, PIM-SM utiliza el envío de mensajes *Prune* para deshabilitar el árbol de distribución.

1.8.3. PIM-SM SHORTEST PATH TREE, SPT.

Así como es posible utilizar el mecanismo *Explicit Join* para vincularse con un árbol de distribución *Shared Tree*, PIM-SM permite utilizar este mecanismo de registración para registrarse a un *SPT* cuyo nodo raíz es una fuente en particular. De esta manera el tráfico IP

Multicast es “ruteado” directamente hacia los receptores, sin la necesidad de tener que contar con un nodo *RP* para que realice el *forwarding* del mismo. De esta manera se puede reducir la latencia causada por el nodo *RP* y la posible congestión del mismo. La desventaja es que los *router*s deben crear y mantener estados (S,G) en sus tablas de ruteo IP Multicast, lo cual consume recursos en los mismos. Igualmente esta información es mucho menor que la que deben mantener para el caso de protocolos *Dense Mode*. Aun cuando parezca que la utilización de PIM-SM *SPT* es más ventajosa, es necesario utilizar árboles de distribución *Shared Tree* para entregar los primeros paquetes IP Multicast desde una fuente a los receptores, ya que de otra forma los *router*s no tienen manera de saber cuales son los receptores que se encuentran activos.

1.8.4. PIM-SM STATE-REFRESH.

Para prevenir que tráfico IP Multicast quede circulando por la red sin un destino, PIM-SM utiliza un *Timer*, *Lifetime*, de 3 minutos, valor por *default*, luego del cual el paquete es desechado. Este *Timer* es establecido por la asociación de cada estado (*,G) y (S,G) en la tabla de ruteo IP Multicast de los *router*s. Cuando este *Timer* expira, la entrada en la tabla de ruteo es eliminada. Como resultado, los *router*s en sentido *downstream* del tráfico, deben enviar periódicamente la información de estos estados de modo de evitar que el *Timer* expire en los casos que no deba expirar. Para esto, envían mensajes PIM *Join/Prune* a su *router* adyacente en sentido *upstream* del tráfico, una vez cada minuto. De esta manera el *router upstream*, actualiza su tabla de ruteo IP Multicast y re-inicializa el *Timer* en 3 minutos.

Los *router*s envían estos mensajes *Join/Prune* siempre que posean una lista no vacía en su interfase de salida asociada con las entradas (*,G) o (S,G), o en caso que posean un grupo G de receptores directamente conectados a su interfase de salida.

1.8.5. SOURCE REGISTRATION.

Como el nodo *RP* debe ser notificado de la existencia de las fuentes IP Multicast, PIM-SM utiliza el envío de mensajes *Register* y *Regsiter-Stop* para la registración de las fuentes IP Multicast. Estos mensajes son enviados a los *RP* por los *router*s directamente conectados a las fuentes IP Multicast, denominados *DR*.

Cuando una fuente IP Multicast comienza a transmitir, el *DR* al cual esta conectada recibe los paquetes IP Multicast enviados por la fuente y crea una entrada (S,G) en su tabla de ruteo IP Multicast. Encapsula cada paquete IP Multicast en un paquete PIM *Register* y lo envía en forma Unicast al *RP*. El *RP* desencapsula el paquete IP Multicast, si este paquete es para un grupo activo, el *RP “forwardeara”* el paquete en sentido *downstream* a través del árbol de

distribución *Shared Tree*. Luego registra la fuente en el SPT, de manera que esta pueda recibir el tráfico (S,G) en forma nativa sin tener que ser enviado al *DR*, y que este tenga luego que encapsularlo en un paquete PIM *Register*. En caso de no existir receptores activos para dicho grupo IP Multicast, el *RP* descarta el paquete recibido por el *DR* y no genera ninguna acción de registración de la fuente.

En el caso que los receptores de un grupo IP Multicast ya no se encuentren activos, el *RP* envía un mensaje Unicast al *DR* correspondiente a la fuente, de manera que detenga el envío de paquetes IP Multicast para dicho grupo.

1.8.6. PIM-SM DESIGNATED ROUTER, PIM-SM DR.

PIM-SM selecciona un *router DR* para cada red multiacceso utilizando el envío de mensajes PIM *Hello*.

1.9. MULTICAST OPEN SHORTEST PATH FIRST, MOSPF.

Se encuentra definido en la RFC 1584, es una extensión del protocolo Unicast OSPFv2. Estas extensiones al protocolo OSPFv2 le permiten a MOSPF *"forwardear"* tráfico IP Multicast dentro de un área OSPF utilizando *SPT*, ya sea para un conjunto parcial de los *router*s MOSPF o para el total de estos.

La mayor ventaja de MOSPF es que comparte la capacidad de OSPF para responder rápidamente a cambios en la topología de red, debido a que utiliza el concepto de *link-state routing* para la generación de árboles de distribución.

1.9.1. MOSPF INTRA-AREA MULTICAST ROUTING.

Se encuentra definido en la RFC 2238.

1.9.1.1. GOUP MEMBERSHIP LINK-STATE ADVERTISEMENT.

El mayor cambio en el formato del paquete de datos de OSPF para poder soportar el *forwarding* de tráfico IP Multicast es la definición de un nuevo *Link-State Adevertisement*, *LSA*, el cual es utilizado para *"forwardear"* información acerca de los miembros de un grupo a través de un área de OSPF. Cada mensaje *SLA* contiene la siguiente información:

- *Multicast Group Address, Link-State ID.*

- *Adevertisement Router ID*.

- Listado de las interfases del *router* que tienen miembros activos para un dado grupo.

Los paquetes *LSA Group Membership* son transmitidos a través del área OSPF y son almacenados en la base datos locales de los *router*s MOSPF. Son similares a los anuncios *LSA* de OSPF, sólo que en MOSPF el *DR* origina los *LSA Goup Membership* en el caso de redes multiacceso.

1.9.1.2. INTRA-AREA SHORTEST PATH TREE, SPT.

Luego que las bases de datos de todos los *router*s de un área se han sincronizado, la combinación de anuncios *LSA Group Membership*, junto con los anuncios *LSA Network* le proveen a cada *router* MOSPF la información necesaria para construir *Intra-Area SPTs* para cada par (fuente-red, grupo) dentro del área OSPF. Para la construcción de estos árboles de distribución, los *router*s MOSPF utilizan el algoritmo de *Dijkstra* que les permiten construir un único *SPT* Multicast, que es ruteado a la fuente. Este algoritmo debe ser implementado por cada par (fuente-red, grupo).

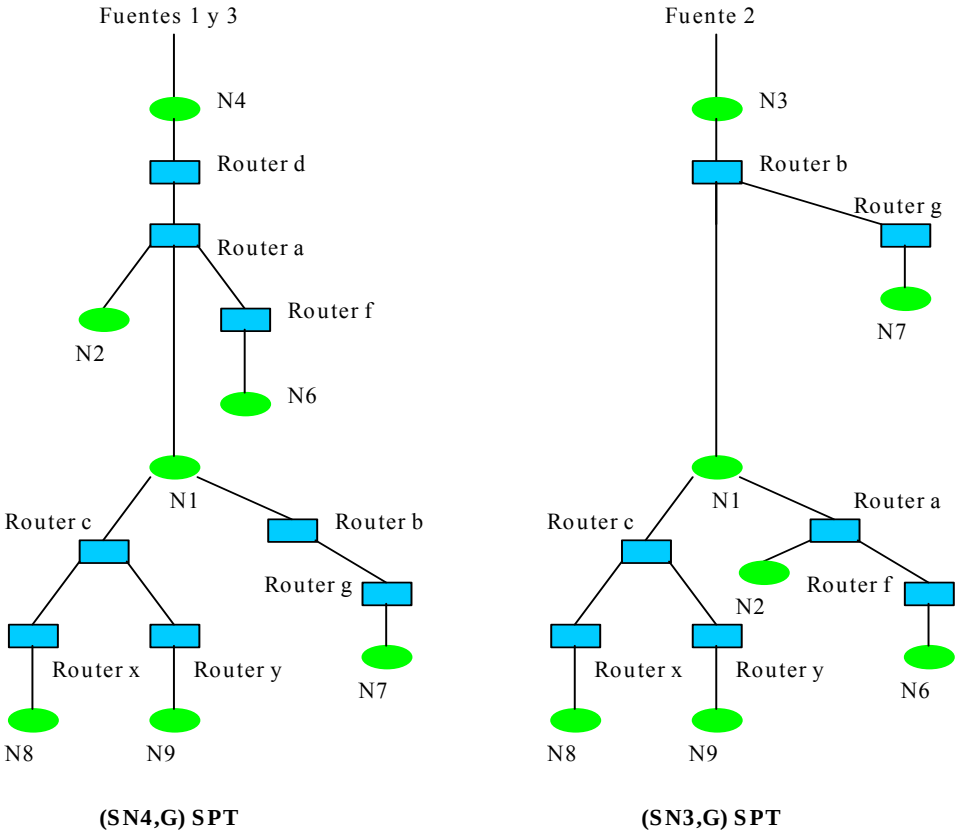


Figura 1.9.2.1.1.- Ejemplo SPTs para las Redes N4 y N3

Los árboles *SPTs* son utilizados como base para el *forwarding* del tráfico IP Multicast dentro de un área OSPF siempre que la información de *Group Membership* permanezca sin cambios. Si existe algún cambio en la información de Grupo, *Group Membership*, o si existe algún cambio en la topología de la red, los *SPTs* deben ser recalculados utilizando el algoritmo de *Dijkstra.*

1.9.2. MOSPF INTER-AREA MULTICAST FORWARDING.

Esto ocurre cuando la fuente de tráfico IP Multicast se encuentra en un área y los receptores en una o más áreas diferentes.

El mecanismo utilizado por MOSPF para el *forwarding* de paquetes IP Multicast entre áreas es muy similar al utilizado por OSPF para el caso de paquetes IP Unicast.

1.9.2.1. MULTICAST AREA BORDER ROUTER.

Para soportar el *forwarding* de paquetes entre áreas, la RFC 1584 define *Interareas Multicast Forwarders*, las cuales son un conjunto de *router*s *OSPF ABR*, *Area Border Router*. Estos realizan las siguientes funciones:

- Sumarización de anuncios *Group Membership* dentro del Área 0.
- *Forwarding* de paquetes IP Multicast entre áreas.

Para que el tráfico IP Multicast sea transmitido en sentido *downstream*, desde el área 0 hacia otras áreas OSPF, los *router*s del área 0 deben conocer los *MABR, Multicast Area Border Router*, que están conectados a áreas que tienen grupos IP Multicast con miembros activos. Para esto, los *MABR* sumarizan la información de *Goup Membership* de áreas OSPF de menor jerarquía, y la transmiten dentro del área 0, o *backbone*, a través de anuncios *LSA Membership Group*. A diferencia de OSPF, estos anuncios fluyen desde áreas de menor jerarquía hacia el *Backbone* en forma asimétrica.

Los *MABR* son los nodos raíz de los árboles *SPTs*, con lo cual se ven como fuente para todos los *router*s dentro del área.

En el caso que la fuente de tráfico IP Multicast no se encuentre dentro del área 0, y los receptores se encuentren fuera de su área, MOSPF define una *Wilcard Multicast* a través de un *flag* en los paquetes *LSA*, dentro del campo *rtype (W-bit)*. Con este bit se indica que el *router* desea recibir todo el tráfico IP Multicast, con lo cual cuando un *router* debe calcular un nuevo

SPT para un paquete IP Multicast de entrada, este busca a través de su base de datos local por anuncios *LSA* que posean seteado el *W-bit,* y tratara a estos *router*s como si fuesen miembros del grupo IP Multicast de destino. Todos lo *MABRs* setean el *W-bi*t en 1 en los anuncios *LSA* que ellos originan dentro de áreas de menor jerarquía. Esto tiene como efecto el enviar todo el tráfico IP Multicast originado por fuentes Multicast, dentro de áreas no *Backbone*, hacia los *MABRs*.

1.9.3. MOSPF INTER-AS MULTICAST ROUTING.

La RFC 1584 define *Inter-AS Multicast Forwarders*, los cuales son responsables del *forwarding* del tráfico IP Multicast dentro y fuera de los dominios MOSPF. Además estos *router*s ejecutan otros protocolos IP Multicast, los cuales les permiten que los árboles de distribución IP Multicast "crucen" el borde dentro y fuera de un dominio MOSPF, o *AS*.

Los *Inter-AS Multicast Forwarders* son configurados como *Wilcard Multicast Forwarders* y anuncian esto a los otros *router*s dentro del área mediante el envío de mensajes *LSA* con el *W-bit* seteado en 1. Esto causa que todo el tráfico IP Multicast dentro del área fluya a través de estos, de manera que lo puedan *"forwardear"* fuera del área hacia otros *ASs* en los casos que sean necesarios.

1.10. INTER-DOMAIN MULTICAST ROUTING.

Los protocolos más utilizados para *"forwardear"* tráfico IP Multicast entre distintos dominios son:

- *Multiprotocol BGP*, *MBGP*.
- *Multicast Source Discovery Protocol*, *MSDP*.

La IETF, *Internet Engineering Task Force*, esta trabajando en el desarrollo de protocolos con el objeto de brindar mayor escalabilidad que los anteriormente citados, ellos son:

- *Border Gateway Multicast Protocol*, *BGMP*.
- *Multicast Address Set-Claim*, *MASC*.

La necesidad de poder contar con un protocolo *Spare-Mode*, el cual evite el comportamiento *flood-and-prune* de los protocolos *Dense Mode*, llevo a la definición de protocolos *Inter-Domain* para ser utilizados sobre Internet para la transmisión de tráfico IP Multicast entre distintos *AS*.

A continuación se listan algunos de los requerimientos que deben cumplir los protocolos *Inter-Domain*:

- Proporcionar un protocolo *Explicit Join* para los anuncios dentro de un dado *AS*.
- Utilizar alguno de los protocolos Unicast definidos hoy para los *Peering* Multicast.
- No depender de los *RPs* de otros *ASs*.
- Flexibilidad en cuanto a la ubicación de los *RPs* dentro de la Red.

El primer ítem es satisfecho con la utilización del PIM-SM. Los restantes tres ítems son resueltos con la utilización de MBGP y MSDP.

1.10.1. MULTIPROTOCOL BGP, MBGP.

Se encuentra definido en la RFC 2283, *Multiprotocol Extensions to BGP-4*. La característica principal de MBGP es la definición de dos nuevos atributos:

- *MP_REACH_NLRI*.
- *MP_UNREACH_NLRI*.

Estos atributos se utilizan para mejorar la información acerca de la "alcanzabilidad" entre diferentes familias de direcciones IP, y son enviados dentro de los mensajes de *Update* de BGP. Dentro de estos atributos se encuentran los campos *Address Family Identifier*, *AFI*, *y Subsequent Address Family Identifier*, *Sub-AFI*, los cuales se utilizan para identificar el protocolo para el cual la información de "alcanzabilidad" es aplicable. De esta manera es posible que BGP publique diferentes *paths*, o caminos, para el tráfico IP Unicast e IP Multicast entre distintos *AS*.

1.10.2. MULTICAST SOURCE DISCOVERY PROTOCOL, MSDP.

En el modelo PIM- SM, las fuentes y receptores de los mensajes Multicast se deben registrar con su *RP* local. Generalmente, el *router* más cercano a una fuente o a un receptor se registra como el *RP*, pero lo importante de notar es que el *RP* conoce sobre todas las fuentes y receptores para cualquier grupo particular. Los *RPs* en otros dominios IP Multicast no tienen ninguna manera de saber sobre las fuentes de los otros dominios. MSDP provee una manera para resolver este problema.

MSDP es un mecanismo que le permite a los *RPs* compartir información sobre las fuentes activas. Los *RPs* conocen los receptores en su dominio local. Cuando *RPs* en dominios remotos "oyen" hablar de las fuentes activas, ellos pueden pasar esa información a sus receptores locales. Pueden remitirse datos de IP Multicast entre los dominios.

Una característica útil de MSDP es que cada dominio pueda mantener un *RP* independiente de otros dominios, pero le permite a los *RPs* que "trafiquen" entre dominios. PIM-SM es utilizado para reenviar tráfico entre dominios IP Multicast.

El *RP* en cada dominio establece una sesión MSDP *peer* que usa una conexión de TCP con *RPs* en otros dominios, o con un *router* de frontera que lleva a los otros dominios. Cuando el *RP* "aprende" sobre una nueva fuente Multicast dentro de su propio dominio, a través del mecanismo de registro normal de PIM, el *RP* encapsula el primer paquete de datos en un mensaje *SA* y les envía el *SA* a todos los pares de MSDP.

Cada par receptor, usa un *RPF* modificado para Transmitir el *SA*, hasta que el *SA* alcanza cada *Router* de MSDP en la red IP Multicast. Si el par receptor de MSDP es un *RP*, y el *RP* tiene una entrada (*, G) para el grupo en el *SA*, indica que hay un receptor interesado, entonces el *RP* crea el estado (S, G) conectando al receptor con la fuente a través del camino más corto. Los datos encapsulados son desencapsulados y enviados hacia abajo al árbol compartido de ese *RP*.

Cuando el último *router*, el más cercano al receptor, recibe el paquete IP Multicast, puede conectar el camino más corto hacia la fuente al árbol de distribución. El *keepalive* de MSDP envía periódicamente mensajes *SAs* que incluyen todas las fuentes en el dominio del *RP*. La figura muestra cómo los datos fluirían entre una fuente en un dominio "A" a un receptor en un dominio "E".

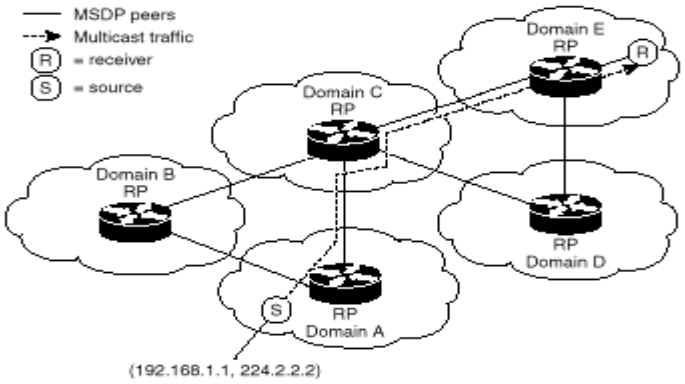


Figura 1.10.2.1.- Funcionamiento de MSDP entre RPs (Ref.:4).

[4] Tomado Referencia Bibliográfica 3.

MSDP se desarrolló para procesos *peers* entre ISPs, ya que los ISPs no deseaban confiar en un *RP* mantenido por otro ISP compitiendo para proporcionar servicio a sus clientes. MSDP permite a cada ISP tener su propio *RP* local, y a la vez remitir y recibir los paquetes IP Multicast desde y hacia Internet.

1.10.3. BORDER GATEWAY MULTICAST PROTOCOL, BGMP.

La idea básica de este protocolo es que por cada grupo IP Multicast que se encuentre activo sobre Internet, se utilizará un único árbol de distribución *Shared Tree Bidireccional* que expanda todos los dominios que contengan receptores y fuentes de un grupo. Uno de estos dominios, *AS*, se lo designará como raíz y cumplirá las funciones de la raíz en un árbol de distribución *Shared Tree Bidireccional*. El resto de los dominios, *ASs*, enviarán mensajes de *Join* y *Prune* hacia el dominio raíz, de la misma forma que se realiza en el protocolo PIM-SM.

Como sólo un dominio funcionara como raíz, deberá contarse con algún mecanismo para determinar el dominio raíz del árbol.

1.11. ANYCAST RP.

El protocolo de red IP Multicast se despliega como un componente íntegro en aplicaciones conectadas a través de una red de computadoras *misión-críticas*. Estas aplicaciones deben ser robustas y escalables para entregar la fiabilidad que demanda el usuario.

Anycast RP proporciona balanceo de carga y redundancia de ruteo basado en PIM-SM. *Anycast RP* permite que dos o más *RPs* compartan la carga para la registración de la fuente, y la posibilidad de actuar como r*outer*s de *Backup* uno del otro. MSDP es el protocolo que hace que *Anycast RP* sea posible de implementar.

Anycast RP es una aplicación de MSDP. Originalmente fue desarrollado para el ruteo de aplicaciones IP Multicast entre dominios. La utilización de MSDP en *Anycast RP* proporciona redundancia y balanceo de carga. Por ejemplo, en una red corporativa se utiliza *Anycast RP* para configurar una red PIM-SM tolerante a fallas dentro de un sólo dominio IP Multicast.

En *Anycast RP*, dos o más *RPs* son configurados con la misma dirección IP sobre una interfase *Loopback*. La dirección IP *Loopback RP Anycast* debe configurarse con una mascara de 32-bit. Todos los *router*s *dowstream* deben configurarse de modo que conozcan que la dirección *IP Loopback Anycast* del *RP* es la dirección IP de su *RP* local. El protocolo de ruteo automáticamente seleccionará el *RP* topologicamente más cercano para cada fuente y receptor. Asumiendo que las fuentes se localizan uniformemente alrededor de la red, un número igual de

fuentes se registrarán con cada *RP*. Es decir, en el proceso de registro las fuentes se compartirán igualmente en todos los *RPs* de la red.

Ya que una fuente puede registrarse con un *RP* y los receptores pueden registrarse a un *RP* diferente, se necesita un método para que los *RPs* puedan intercambiar información sobre las fuentes activas. Este intercambio de información se hace con MSDP.

En *Anycast RP*, todos los *RPs* se configuran para ser pares de MSDP uno de otro. Cuando una fuente se registra con un *RP*, un mensaje de *SA* se enviará al resto de los *RPs*, que les informa que hay una fuente activa para un grupo IP Multicast particular. El resultado es que cada *RP* sabrá sobre las fuentes activas en el área de los otros *RPs*. Si cualquiera de los *RPs* fallara, el protocolo de ruteo IP convergería, y uno de los *RPs* se volvería el *RP* activo en más de un área. Las nuevas fuentes se registrarían con el *RP Backup*. Los receptores se unirían al nuevo *RP* y se mantendría la conectividad.

Cabe notar que el *RP* normalmente se necesita sólo para empezar nuevas sesiones entre fuentes y receptores. Debido a que los *RPs* comparten el árbol de ruteo, las fuentes y receptores pueden establecer un flujo de datos de IP Multicast directamente. Si un flujo de datos IP Multicast estuviese ya establecido directamente entre una fuente y un receptor, entonces una falla de un *RP* no afectará esa sesión. *Anycast RP* asegura que nuevas sesiones entre fuentes y receptores puedan establecerse en cualquier momento.

1.12. CICLO DE VIDA DE UN GRUPO IP MULTICAST.

Una arquitectura de red que tiene como objetivo proveer soporte a comunicaciones Multicast se ve sobrecargada con la tarea de administrar las sesiones Multicast de una manera transparente hacia los usuarios, lo cual impone ciertos requisitos en la implementación de este tipo de redes.

Las fases o pasos más relevantes durante el ciclo de vida de una sesión IP Multicast se pueden resumir como:

- Creación del grupo o sesión Multicast.
- Construcción del árbol de distribución.
- Transmisión de datos.
- Deshabilitación de la sesión Multicast.

En la siguiente figura se muestran las distintas fases de una sesión IP Multicast.

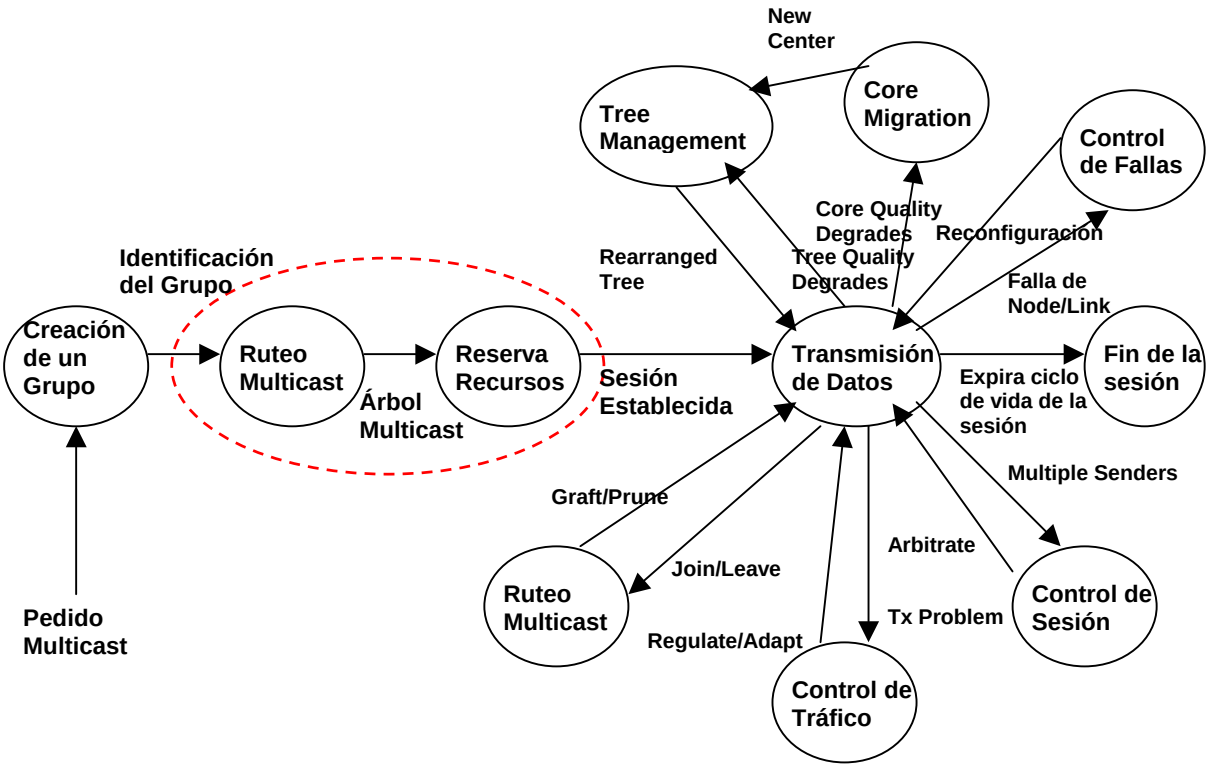


Figura 1.12.1.- Ciclo de Vida de una Sesión Multicast (Ref.:5).

Creación del Grupo o Sesión Multicast.

El primer paso en la iniciación de una sesión Multicast es asignar una única dirección al grupo Multicast tal que los datos de un grupo no colisionen con los de otro. Tanto los grupos (o sesiones) como las direcciones Multicast tienen un tiempo de vida asociado.

Las direcciones de los grupos pueden ser estáticas o dinámicas, dependiendo de si fueron asignadas en forma permanente a un determinado grupo o si son asignadas a diferentes grupos en distintos instantes de tiempo. La forma más habitual de asignar direcciones a los grupos es asignar direcciones estáticas a grupos Multicast permanentes, y direcciones dinámicas a grupos transitorios.

Construcción del Árbol de Distribución.

Una vez que el grupo fue creado, se construye el árbol de distribución. La determinación de la ruta del tráfico Multicast es una tarea relacionada con la construcción del árbol de

distribución. Se pueden mencionar tres características que hacen a la definición del mecanismo utilizado para la construcción del árbol Multicast:

- Las fuentes necesitan transmitir un único paquete a través del árbol hacia los receptores.

- La estructura del árbol debe permitir transmisiones en paralelo hacia varios receptores.

- La estructura del árbol debe minimizar la replicación de los paquetes de datos.

La determinación del árbol de distribución óptimo para un grupo Multicast estático se logra utilizando el modelo de *Steiner*.

Transmisión de Datos.

Durante la transmisión de datos, se pueden presentar los siguientes eventos:

Membresía Dinámica:

Debido a que las membresías de los grupos Multicast pueden tener características dinámicas, la red debe ser capaz de poder realizar un seguimiento de las membresías de los grupos a lo largo de su ciclo de vida. Esto es necesario para poder comenzar con el *forwarding* de los datos hacia los nuevos miembros de un grupo, y para detener el *forwarding* hacia los receptores que ya no forman parte del grupo Multicast. Esta tarea, según el protocolo de red utilizado, puede ser realizada en forma:

- *Flooding* (inundación).

- Centralizada.

- Distribuida.

Cambios Dinámicos en la Red:

Si durante el ciclo de vida de una sesión, un nodo o enlace de red falla, el servicio puede llegar a verse interrumpido. Para evitar esto, se requieren de mecanismos para detectar

[5] Tomado Referencia Bibliográfica 2.

estas fallas, y que reconfiguren el árbol de distribución de modo que la transmisión ni la recepción de los datos se vea afectada.

Si el protocolo de ruteo Multicast utilizado basa su confiabilidad sobre un protocolo Unicast, su comportamiento ante fallas dependerá del protocolo Unicast determinado.

Si en cambio, el protocolo de ruteo Multicast es independiente de los protocolos Unicast, este deberá implementar sus propios mecanismos de restablecimiento ante fallas.

Problemas de Transmisión:

Pueden presentarse problemas donde el receptor no llega a procesar el flujo de información que recibe, para lo cual es necesario implementar mecanismos de control de flujo; o pueden presentarse errores en la transmisión de los paquetes de datos, donde se necesitaría de un mecanismo para el control de errores.

Competencia entre Fuentes Multicast:

En casos donde múltiples fuentes de tráfico Multicast comparten el mismo árbol de distribución para la transmisión de los paquetes de datos, se puede presentar una situación de competencia de recursos entre las fuentes. Esto resultará en la perdida de datos debido al desborde de los *buffers* (*buffer overflow*), ocasionando problemas de transmisión. Para solucionar este problema, se requerirá de la implementación de un mecanismo que arbitre la transmisión entre las distintas fuentes.

Deshabilitación de la Sesión Multicast.

En algún momento, cuando el tiempo de vida de la sesión expira, la fuente iniciará los procedimientos para dar de baja la sesión. Esto involucrará la liberación de los recursos reservados para la sesión a lo largo de los enlaces que componen el árbol de distribución Multicast, y el “borrado” de las entradas en la tabla de ruteo correspondiente. Finalmente se libera la dirección IP Multicast asociada al grupo Multicast.

2. MULTIPROTOCOL LABEL SWITCHING, MPLS.

La principal característica de MPLS es la de brindar el soporte de requerimientos de calidad de servicio, utilizando una técnica relativamente simple en comparación con los métodos y mecanismos de calidad de servicio que podemos tener en IP Nativo, como pueden ser *Diff Serv* o *RSVP*.

Como se presentará y analizará más adelante, la utilización de MPLS junto con IP Multicast nos permitirá brindar soluciones adecuadas a las necesidades que presentan las nuevas aplicaciones de *Real Time* Interactivas.

En el análisis que se desarrollará en el punto 6 de la presente Tesis, tomaremos algunos de los conceptos que se mostrarán en el presente capítulo.

2.1. ARQUITECTURA.

La arquitectura definida en MPLS utiliza asignación de etiquetas en sentido *downstream* para tráfico IP Unicast. Además soporta asignación de etiquetas bajo demanda en sentido *downstream* y asignación no solicitada. Esto permite que las etiquetas sean globalmente únicas por nodo o por interfase.

Las etiquetas poseen una gran granularidad, lo cual puede traer problemas cuando se producen diferencias de granularidad entre *LSRs*, *Label Switching Routers*, adyacentes.

Para soportar una estructura jerárquica de asignación de etiquetas emplea el mecanismo *LIFO*, *last-in-first-out*. De esta manera, la decisión de *forwarding* se realiza sobre la primera de las etiquetas de la pila.

Para la selección de camino, o *path*, se proponen dos mecanismos:

1. *Hop by Hop*.

2. *Explicit Routing*.

En el primer caso, el *next hop* es designado utilizando los resultados obtenidos de los protocolos de ruteo convencionales. En el segundo caso, una ruta explícita, *explicit route*, es especificada completamente por la fuente. Todos los *LSRs* son capaces de *"forwardear"* paquetes utilizando este mecanismo, pero no tienen la capacidad de originarlos.

La arquitectura de MPLS no requiere la utilización del mecanismo *Control-Driven* para la asignación de etiquetas, aun cuando este método es el más utilizado en estas redes.

No se define un mecanismo de encapsulación para datos etiquetados, pero se permiten dos opciones:

1. Utilizar una encapsulación específicamente desarrollada para MPLS.

2. Utilizar "espacios disponibles" en los *Headers* de la capa de datos o de red.

2.2. COMPONENTES FUNCIONALES DE RUTEO EN LA CAPA DE RED.

Para realizar el *forwarding* de un paquete IP, se tienen en cuenta dos fuentes de información:

1. La tabla de ruteo almacenada en los *router*s.

2. La información transportada en el propio paquete IP.

Luego, además de la componente de *forwarding* tenemos la componente de Control, la cual se encarga de la construcción y mantenimiento de la tabla de ruteo, o tabla de *forwarding*.

Cada uno de los *router*s de la red implementa ambas funciones, Control y *Forwarding.*

La componente de control consiste en uno o más protocolos de ruteo que proveen intercambio de información de ruteo entre los *router*s de la red, además de los algoritmos que los *router*s utilizan para convertir esta información en la tabla de *forwarding*. OSPF, BGP y PIM son ejemplos de este tipo de protocolos.

La componente de *Forwarding* consiste en un conjunto de algoritmos que los *router*s utilizan para tomar la decisión de *forwarding* de un paquete IP.

2.2.1. COMPONENTE DE FORWARDING.

Desde el punto de vista del *forwarding*, los paquetes IP dentro de cada conjunto son tratados por los *router*s de la misma manera, aun cuando los paquetes dentro de un conjunto difieran el uno del otro con respecto a la información del encabezado de la capa de red. Cada

uno de estos conjuntos se definen como *Forwarding Equivalence Classes, FECs*. Un conjunto de paquetes IP Multicast con la misma dirección de red de fuente y destino es un ejemplo de *FEC*.

Una característica importante de los *FECs* es su granularidad de *fowarding*, un *FEC* puede incluir todos los paquetes cuya dirección de red de destino coincide con un determinado prefijo. Además, un *FEC* podría incluir sólo los paquetes pertenecientes a una aplicación particular siendo ejecutada entre dos computadoras, incluyendo así sólo los paquetes IP con el mismo par de direcciones fuente y destino y los ports TCP/UDP utilizados.

La componente de control es responsable de la distribución consistente de la información de ruteo utilizada por los *router*s para construir las tablas de *forwarding*. También es responsable de la consistencia de los procedimientos utilizados por los *router*s para la construcción de las tablas de *forwarding*.

La componente de *forwarding* es responsable de mantener la consistencia de los procedimientos para extraer la información de los paquetes IP, además de mantener una forma consistente de utilizar esta información para encontrar una entrada apropiada en la tabla de *forwarding*, lo cual resulta en un mapeo de los paquetes IP dentro de los *FECs* a través de los múltiples *router*s de una red.

El algoritmo utilizado por la componente de *forwarding* en *Label Switching*, MPLS, para tomar la decisión de *forwarding* de un paquete IP utiliza dos fuentes de información:

- La tabla de *forwarding* almacenada en los *LSR*s.
- La etiqueta transportada por el paquete IP.

La tabla de *forwarding* almacenada en un *LSR* consiste en una secuencia de entradas, donde cada una esta formada por una etiqueta de entrada y una o más sub-entradas, donde cada sub-entrada consiste en una etiqueta de salida, una interfase de salida, y la dirección IP del siguiente *next hop*. En el caso de tráfico IP Multicast, se tendrá más de una sub-entrada. Esta tabla se encuentra indexada por el valor contenido en la etiqueta de entrada. Esta tabla puede incluir información relacionada con los recursos que utilizará el paquete IP, como colas de salida, calidad de servicio, etc.

Etiqueta de entrada	Primer Sub-Entrada	Segunda Sub-Entrada
Etiqueta de entrada	Etiqueta de Salida Interfase e Salida Dirección IP del Next Hop	Etiqueta de Salida Interfase e Salida Dirección IP del Next Hop

Figura 2.2.1.1.- Ejemplo de Tabla de Forwarding.

El algoritmo de *forwarding* se basa en el *swaping* de etiquetas. Cuando un *LSR* recibe un paquete, extrae la etiqueta y utiliza esta información como índice en la tabla de *forwarding*. Una vez que encuentra la entrada indexada por la etiqueta, para cada sub-entrada, el *router* reemplaza la etiqueta en el paquete con la etiqueta de salida de la sub-entrada y envía el paquete por la interfase de salida especificada en la sub-entrada al *next hop* especificado en la misma. Si la sub-entrada no especifica un *next hop*, sino una cola de salida, el *router* situará el paquete en dicha cola. En el caso que el *LSR* mantenga una tabla de *forwarding* por cada interfase, luego de recibir el paquete IP, utilizará la interfase en la cual ha recibido el paquete para seleccionar la tabla de *forwarding* correspondiente.

Una propiedad importante de este algoritmo es que con sólo un acceso a memoria puede obtener toda la información necesaria para realizar el envío del paquete IP. Otra característica importante es que utiliza sólo un algoritmo para el *forwarding* de paquetes, *Label Switching*, soportando así un amplio rango de funcionalidades de ruteo.

La componente de *forwarding* no se limita a un único protocolo de capa de red, haciéndola así multiprotocolo. Además, es capaz de ser soportada sobre cualquier tecnología de capa de física (o capa de enlace).

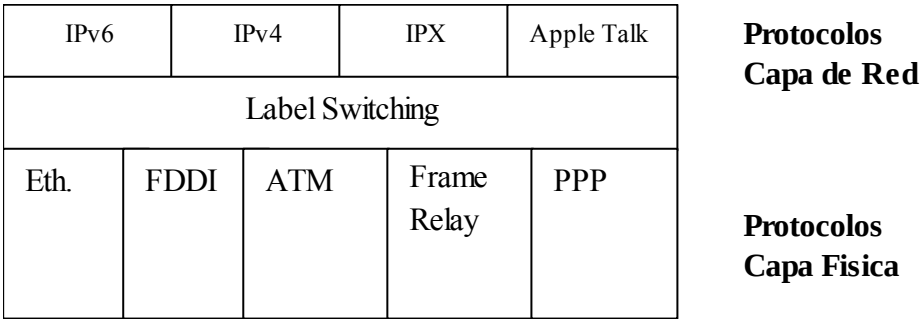


Figura 2.2.1.2.- Label.Switching

2.2.1.1. TRANSPORTE DE ETIQUETAS.

Ciertas tecnologías de capa de enlace, como ATM y Frame Relay, pueden transportar la información de la etiqueta MPLS dentro del *Header* de capa de enlace. En el caso de ATM puede ser transportada como parte de la información de VPI/VCI. En otras tecnologías donde no es posible transportar la información de etiquetas dentro del *Header* de la capa de enlace, se transporta esta información dentro de un "*shim*", que es insertado entre los *Headers* de la capa de red y la capa de enlace.

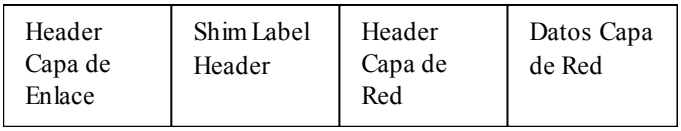


Figura 2.2.1.1.1.- Shim Label Header.

2.2.2. COMPONENTE DE CONTROL.

La componente de control es responsable de:

- Distribuir la información de ruteo entre los *LSR*.
- Los algoritmos utilizados por los *LSR* para convertir la información de ruteo en la tabla de *forwarding.*

Incluye todos los protocolos de ruteo utilizados para el ruteo convencional de paquetes IP (como por ejemplo: BGP, OSPF, PIM, etc). Estos protocolos les brindan a los *LSRs* el mapeo entre *FECs* y direcciones IP de los *Next Hops*.

El procedimiento para la generación de las tablas de *forwarding* es el siguiente:

1. Creación de enlaces, *bindings*, entre las etiquetas y los *FECs*.
2. Informar al resto de los *LSR* de los *bindings* creados.
3. Utilizar la información de los puntos anteriores para construir y mantener actualizada la tabla de *forwarding*.

Protocolos de Capa de Red	Creación de Enlaces, Bindings, entre Etiquetas y FECs	Distribución de Información de "Bindings" a otros LSRs
Mantenimiento de la Tabla de "Forwarding"		

Figura 2.2.2.1.- Componente de Control.

Existen dos tipos de *bindings* de etiquetas, un *binding* local y otro remoto. En el local, el *LSR* asigna etiquetas localmente. En el remoto, el *LSR* recibe información del *binding* de etiquetas de otro *LSR*, que se corresponde con el *binding* de etiqueta generado localmente en este último.

La componente de control utiliza ambos tipos de *bindings* para generar la tabla de *forwarding* con las etiquetas de entrada y salida. En el caso que el *binding* local sea utilizado para generar etiquetas de entrada, se lo denomina *Downstream Label Binding*. En este caso los paquetes de datos fluyen en dirección opuesta al flujo de información del *binding* de etiqueta. En el caso que el *binding* local sea utilizado para generar etiquetas de salida, es denominado *Upstream Label Binding*.

Los *LSRs* mantienen un conjunto de etiquetas libres de asignación las cuales las utilizan para generar los *bindings*.

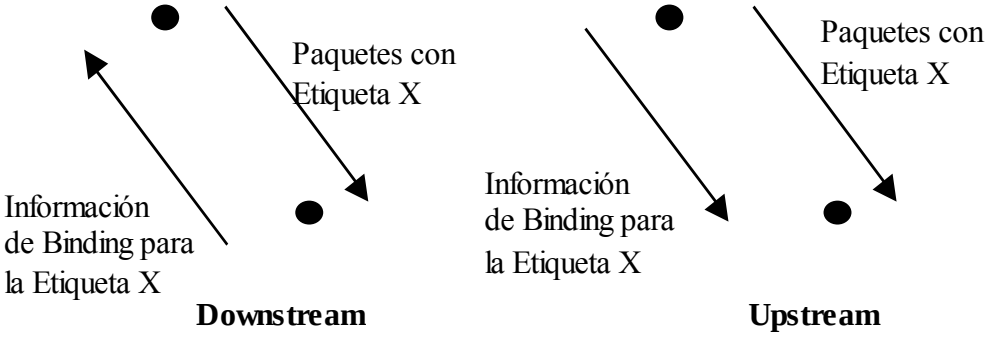


Figura 2.2.2.2.- Bindings Upstream y Downstream.

Un *LSR* "crea" o "destruye" un *binding*, o asociación, entre una etiqueta y un *FEC*, como resultado de un evento en particular. Dicho evento puede ser generado por un paquete que necesita ser *"forwardeado"* o por información de control que tiene que ser procesada por el *LSR*. Cuando el evento es generado por un paquete de datos, se lo denomina *Data-Driven Label Binding*, y cuando es generado por una información de control se lo denomina *Control-Driven Label Binding*. La elección entre uno y otro tendrá efecto sobre la performance del *LSR*.

Bajo condiciones ideales, un *LSR* puede *"forwardear"* datos a la velocidad determinada por la tecnología asociada a la interfase de salida, sin importar cual de los dos métodos se este utilizando. Se entiende por condición ideal, los casos donde la longitud de los paquetes de datos puede considerarse como infinita.

2.2.2.1. DISTRIBUCIÓN DE LA INFORMACIÓN DE LABEL BINDING.

La distribución entre la asociación, *binding*, de una etiqueta local y un *FEC* puede ser realizada de las siguientes formas:

Piggyback on Top of Routing Protocols.

Este esquema es sólo posible cuando se utiliza *Control-Driven Label Binding*. Realiza la distribución de los *bindings* de etiquetas en forma consistente con la información de ruteo. Esto simplifica la operación del sistema, ya que elimina la necesidad de contar con un protocolo dedicado a distribuir información de *bindings* de etiquetas. Sin embargo, la información distribuida por un protocolo de ruteo puede no ser adecuada para la distribución de la información de *binding* de etiqueta, sólo es posible en los casos donde la distribución de información del protocolo de ruteo permite el mapeo entre *FEC* y *Next Hop* para *Piggybacking*. Por esta razón, los protocolos de ruteo que utilizan el algoritmo *Link-State*, como OSPF, no son adecuados para este esquema. No así para protocolos como BGP y PIM. Aun cuando la información de ruteo distribuida por el protocolo de ruteo pueda ser adecuada para la distribución de información de *Label Binding*, la extensión del mismo para transportar esta información puede no ser factible en todos los casos. Esto es así ya que dicha extensión del protocolo puede involucrar cambios en el formato del mensaje utilizado por el mismo.

Otra consideración a tener en cuenta es la posibilidad que una etiqueta, la cual se extrae realizando el *"piggybacking"* en el mensaje del protocolo de ruteo, pueda ser recibida por un dispositivo de red que no entienda de etiquetas. Por lo cual esto debe ser prevenido, o esta acción debe ser realizada de manera tal que estos dispositivos ignoren la información de *binding* de etiqueta recibida.

Protocolo de Distribución de Etiquetas.

Realiza la distribución de la información de *bindin"* de etiqueta mediante la utilización de un protocolo separado. Esta implementación puede provocar la situación en la cual

un *LSR* tenga información de *binding* de etiqueta, asociación entre una etiqueta y un *FEC*, pero no posea información de ruteo, asociación entre un *FEC* y un *Next Hop*, necesaria para utilizar la información de *binding* de etiqueta, o viceversa.

Otra desventaja de este método es la introducción de un nuevo protocolo dentro del sistema de red, lo cual aumenta la complejidad del mismo.

2.2.2.2. MULTICAST FORWARDING.

El *forwarding* de tráfico Multicast utilizando *Label Switching* requiere de ciertas consideraciones sobre la componente de control.

IP Multicast utiliza *Spanning Tree* para realizar el *forwarding* de paquetes, donde un árbol puede ser asociado con una combinación de una fuente particular de tráfico IP Multicast y grupos de receptores, o con un grupo particular de receptores, *Shared Trees*.

Para proveer un *forwarding* consistente del tráfico IP Multicast utilizando *Label Switching*, cuando un *LSR* recibe un paquete de datos IP Multicast, debe ser capaz de determinar en forma no ambigua un árbol de distribución en particular el cual se utilizará para el *forwarding* del tráfico. Para esto, la única información provista por un paquete al *LSR* es:

- La etiqueta transportada por el paquete.
- La interfase por la cual el paquete fue recibido.

Por tales motivos, es necesario que el *LSR* mantenga una tabla de *forwarding* por cada una de sus interfases.

En el caso de tecnologías de capa física con capacidad Multicast nativa, como el caso de enlaces Ethernet, *Label Switching* debe ser capaz de utilizar dicha capacidad. Por lo tanto, los *LSR* conectados a un enlace multiacceso, y que son parte de un árbol de distribución IP Multicast, deben "acordar" un grupo de etiquetas a ser utilizadas en dicho árbol de distribución. Para esto, la componente de control deberá incluir:

1. Procedimientos para elegir un *LSR* en particular dentro del grupo que será responsable de la creación de los *bindings* de etiquetas.
2. Procedimientos para la distribución de la información de *binding* de etiqueta al resto de los *LSRs* del grupo.

Cuando un *LSR* conectado a un enlace multiacceso, como Ethernet, recibe un paquete IP Multicast, debe identificar el árbol de distribución, para lo cual deberá identificar el *Next Hop* previo que envió el paquete. Para esto, dos *LSRs* conectados a un enlace multiacceso no pueden utilizar una combinación de la misma etiqueta y de la misma interfase para crear los *bindings* de etiquetas para diferentes árboles de distribución. Una forma de lograr esto, es particionar en los *LSR* un conjunto de etiquetas que serán utilizadas para paquetes IP Multicast por un conjunto de *LSRs* conectados a un enlace multiacceso, y asignarle a cada *LSR* un subconjunto de estas etiquetas. Cada *LSR* utilizará este subconjunto como uno de etiquetas asociadas a la interfase que lo conecta con la red multiacceso.

2.3. LABEL SWITCHED PATH, LSP.

Para el establecimiento de los *LSP*, MPLS utiliza dos mecanismos de control:

1. *Ordered Control*.

2. *Independent Control*.

En el primer caso, la asignación de etiquetas se realiza de forma ordenada desde una punta a la otra del *LSP*. El establecimiento del mismo puede ser generado por el Ingreso, *Head*, o egreso, *Tail*, del *LSP*. Un *LSR* puede distinguir si es un *Tail* para un dado *FEC*, si el *Next Hop* para este *FEC* no es un *LSR*. Entonces, asignará una etiqueta al *FEC*, *binding* de etiqueta, y publicará la asignación a su *LSR* adyacente. Cualquier adyacencia que tenga al *Tail LSR* como *Next Hop* para dicho *FEC*, le asignará una etiqueta a dicho *FEC* y lo publicará a su adyacencia, y así sucesivamente. Así, la asignación de etiquetas sigue un procedimiento ordenado de egreso a ingreso.

En el segundo caso, el *"switcheo"* de etiquetas se realice sobre la base de protocolos de ruteo *destination-base;* cada *LSR* deberá tomar una decisión independiente para asignar una etiqueta a un dado *FEC*, y luego publicar esta asignación a su adyacencia. El establecimiento del *LSP* estará afectado por la convergencia del protocolo de ruteo utilizado.

El *Ordered Control* ayuda a la prevención de Loops. Es útil en redes que se encuentren migrando de un ruteo IP convencional a MPLS.

Con *Independent Control*, cada *LSR* realiza una elección propia acerca de cómo particionar el conjunto de los posibles paquetes de datos dentro de los *FECs*. Si el *LSR* adyacente toma una decisión diferente a cerca de los *FEC* que utilizará, no será posible establecer un *LSP* para dichos *FECs*.

En *Ordered Control*, la elección de *FECs* puede ser realizada en el *LSR* que inicia el *LSP*. Así todos los *LSR* utilizarán el mismo *FEC*. El *LSR* deberá determinar el *Next Hop* para un dado *FEC*, de manera de poder determinar si el *binding* proviene del *Next Hop* correcto. Una desventaja es el tiempo que se requiere para el establecimiento de un *LSP*. Requiere que los *bindings* sean propagados a través de todos los *LSR* antes que el *LSP* sea establecido. Durante este período algunos paquetes pueden ser descartados incrementando la carga del procesador en los *LSR*. Esto no sucede en *Independent Control*, ya que un *LSR* puede establecer y publicar *bindings* de etiqueta en cualquier momento, sin tener que esperar la propagación de los mensajes.

Otra característica importante de *Independent Control* se basa en el hecho que los *LSRs* tienen la capacidad de almacenar el *binding* de etiqueta de los *LSRs* que eran adyacentes al momento de realizar la propagación de las mismas, lo cual le permite establecer *LSPs* en forma cuasi instantánea cuando se producen cambios en el ruteo. Esto redunda en una más rápida convergencia que la que se tiene con *Ordered Control*.

2.4. PREVENCIÓN Y DETECCIÓN DE LOOPS.

Los paquetes MPLS poseen un campo *TTL* al igual que los paquetes IP, el cual es utilizado para descartar paquetes que hayan caído bajo un Loop transitorio. No obstante, con ciertas tecnologías de red este campo no esta disponible, como el caso de MPLS sobre ATM.

Una forma de mitigar el efecto de los Loops es mediante la utilización de *Buffers*. Esto es útil en el caso de utilizar MPLS sobre ATM, ya que ATM tiene la capacidad de limitar la capacidad de los *Buffers* para un dado VC, lo cual disminuye el efecto producido por los paquetes que entraron en un Loop transitorio. Esta técnica de utilización de *Buffers* también es valida para los casos de Loops no transitorios, los cuales pueden ocurrir por errores de configuración.

Otra forma de controlar la formación de Loops en tecnologías que no soporten *TTL* es mediante la cuenta de *Hops*, pero este mecanismo pude no ser suficiente para controlar el efecto adverso de los Loops transitorios.

Un mecanismo utilizado por MPLS para realizar la detección de Loops es el denominado *Path Vectors*. Un *Path Vector* es una lista de *LSRs* por los cuales han pasado mensajes de *Label Requets* o *Label Mapping*. Si un Loop de ruteo causa que un mensaje de *Label Request* o *Label Mapping* viaje en un Loop, eventualmente un *LSR* “verá” su dirección en el mensaje de *Label Request* o *Label Mapping*, y así detectará un estado de Loop.

El mecanismo denominado *Colored Threads*, es utilizado por MPLS para la prevención de Loops. Este requiere de la utilización de *Ordered Control* para el establecimiento de los *LSPs*. El establecimiento del *LSP* se realiza mediante la extensión de un único flujo, o *Thread*, coloreado en sentido ingreso a egreso del *LSP*. Si el Loop del flujo vuelve sobre si mismo, un nodo detectará un color que él ya había "visto" antes, y así concluirá que se trata de un Loop. En este punto, se interrumpe el establecimiento del *LSP* hasta que desaparezca el Loop. Este mecanismo es tan robusto como el método de *Path Vector* para la prevención de Loops en el establecimiento de *LSPs*, pero requiere mucha menor transmisión y almacenamiento de información en cada *Hop*.

2.5. ENCAPSULACIÓN.

La encapsulación de la etiqueta presenta dos alternativas, dependiendo de la tecnología utilizada:

1. *Shim Header*: este mecanismo se adopta para todas aquellas tecnologías que no permiten transportar la información de Etiqueta en el *Header* del paquete de capa física. Es decir, se utiliza esta metodología para todas las tecnologías, excepto para ATM y Frame Relay.

2. *Header* del *PDU*, *Paquet Data Unit*, de capa física.

Etiqueta (20 Bits)	Exp (3 Bits)	Stack (1 Bit)	TTL (8 Bits)

Exp: campo reservado.
Stack: indica que se ha alcanzado la parte inferior de la pila de etiquetas.
TTL: indica el tiempo de vida de un paquete.

Figura 2.5.1.- Formato de una Entrada en la Pila de Etiquetas.

El *TTL* se obtiene a partir del *TTL* definido en el paquete IP, y es decrementado en 1 por cada *LSR*. De esta manera, un paquete IP que ingresa en una red MPLS egresará de la misma con el *TTL* decrementado en 1 por cada *LSR* que haya tenido que pasar. Una alternativa a este procedimiento es decrementar el *TTL* en 1 únicamente cuando el paquete IP egrese de la red MPLS. De esta manera un *LSP* se verá, desde el punto de vista de IP, como un sólo *Hop*.

Es posible que los paquetes etiquetados requieran de fragmentación, como en el caso de paquetes IP, ya que es posible que un paquete requiera la asignación de más de una etiqueta, por lo cual puede requerirse la fragmentación del mismo. Para estos casos, la

fragmentación es realizada sobre el datagrama IP, y luego se especifica la misma utilizando el bit de *Stack* y la información de etiqueta.

Existen etiquetas reservadas para los casos de fragmentación, estas son:

1. 0: *IPv4 Explicit Null*.
2. 1: *Router Alert*.
3. 2: *IPv6 Explicit Null*.
4. 3: *Implicit Null*.

La etiqueta *Explicit Null* es utilizada en los casos donde se necesita encapsulación de etiqueta, pero no se necesita una etiqueta valida. Cuando se utiliza esta etiqueta, deberá ser la única en la pila de etiquetas.

La mayoría de las encapsulaciones utilizadas en Capa 2, como PPP, Ethernet, etc., contienen un campo que especifica el protocolo de capa 3.

La etiqueta *Router Alert* es utilizada para informar al *router* que el paquete necesita ser tratado de forma más amplia que simplemente realizar un *forwarding* del mismo. Cuando un *LSR* recibe un paquete con esta etiqueta, toma la primer etiqueta de la pila de etiquetas y *"forwardea"* el mismo utilizando esta información de etiqueta en caso que el paquete deba ser *"forwardeado"*.

La etiqueta *Implicit Null* no es un valor valido de etiqueta que puede aparecer en el *Header* de un paquete transmitido, esta reservada para ser utilizada por el protocolo de distribución de etiquetas.

2.6. LABEL DISTRIBUTION PROTOCOL, LDP.

Como se vio anteriormente, es preferible realizar la distribución de etiquetas mediante el mecanismo de *piggybacking* utilizando protocolos de red como BGP, PIM y RSVP.

LDP se basa en la unión del mecanismo *Control-Driven,* TCP, y protocolos ARIS. Sus principales características son:

1. Provee a los *LSRs* de un mecanismo de "*discovery*", que permite a los *LSR peers* descubrirse el uno al otro y establecer comunicación entre ellos.

2. Define cuatro clases de mensajes:

   a. Mensajes *Discovery*

   b. Mensajes *Adyacency*: permiten la inicialización, *keepalive*, y cierre de sesiones entre LSRs.

   c. Mensajes *Label Advertisement*: permiten la publicación de *bindings* de etiquetas, pedidos, y liberaciones.

   d. Mensajes *Notification*: permiten enviar información de aviso de problemas y señalizar errores.

3. Se aplica sobre TCP, lo cual le provee confiabilidad en la entrega de la información.

4. Es fácilmente expandible, utiliza mensajes especificados como un grupo de *TLV*, *Type Length Value*, de objetos codificados.

La codificación *TLV* implica que cada objeto posee un campo *Type* que especifica que clase de objeto es, ej: *binding* de etiqueta, un campo *Length* que especifica la longitud del objeto, y un campo *Value* que depende del campo *Type*.

2.6.1. LSR NEIGHBOR DISCOVERY.

El protocolo LDP Neighbor Discovery se implementa sobre UDP. Un *LSR* envía periódicamente, en forma Multicast, mensajes de *Hello* utilizando un port UDP bien conocido, *Well- Known-Port*, así todos los *LSR* "escuchan" sobre este port UDP por mensajes de *Hello*, aprendiendo de esta manera todos los *LSR* con los cuales tendrá conexiones.

Cuando un *LSR* aprende la dirección de otro *LSR* mediante este mecanismo, establece una conexión TCP con este, estableciéndose así una sesión LDP entre ambos. La sesión LDP es bidireccional.

Un mecanismo de *Discovery* adicional le permite a un *LSR* "descubrir" otro *LSR* aun cuando no estén directamente conectados por una sub-red. En este caso, un *LSR* envía

periódicamente mensajes de *Hello Unicast*, sobre un port UDP bien conocido, a una dirección IP especifica, la cual debe "aprender" por algún otro medio, ej: por configuración. El *LSR* receptor de este mensaje puede responder con un mensaje de *Hello Unicast* al *LSR* que lo origino, estableciéndose luego una sesión TCP entre ambos y la sesión LDP. Este mecanismo es útil para el caso de utilizar *Traffic Engineering* sobre un *LSP* entre dos *LSR*.

2.6.2. TRANSPORTE CONFIABLE.

La necesidad de contar con un transporte confiable surge de que si un *binding* de etiqueta o el pedido de un *binding* no son entregados en forma satisfactoria, el tráfico no podrá ser *"switcheado"* utilizando *Label Switching*, por lo cual debería ser tratado utilizando el proceso de control o descartado. Además, en ciertos casos es necesario recibir los mensajes en orden. Ambas características son logradas utilizando TCP como protocolo de transporte.

TCP le provee a LDP una "paquetización" eficiente de mensajes de capaz superiores dentro de datagramas IP, el *piggybacking* de mensajes de *ACK* en paquetes de datos, y el control de flujo, lo cual es una característica importante para un protocolo de control como lo es LDP.

No obstante, TCP provee de un mecanismo de control de congestión que puede no ser necesario para un protocolo de control *neighbor-to-neighbor*, lo mismo con el control estricto de secuencia en la entrega paquetes.

2.6.3. MENSAJES LDP.

Existen las siguientes clases de mensajes LDP:

1. Inicializacion.
2. *Keepalive*.
3. *Label Mapping*.
4. *Label Witrawal*.
5. *Label Release*.
6. *Label Request*.

7. *Label Request Abort*.

Mensajes de Inicializacion.

Los mensajes de Inicializacion son enviados al comienzo de una sesión LDP, para permitir que dos *LSRs* acuerden los parámetros y opciones de la sesión. Estos mensajes incluyen:

a. *Label Allocation Mode*.

b. *Timers*.

c. Información sobre el rango de etiquetas a ser utilizado entre dos *LSRs*.

Ambos *LSRs* pueden enviar mensajes de Inicializacion y responder con un *Keepalive* si los parámetros son aceptados. Si alguno de los parámetros no es aceptado, el *LSR* responde con un mensaje de *Error Notification*, y la inicializacion de la sesión es terminada.

Mensajes Keepalive.

Son enviados periódicamente por los *LSRs* en ausencia de algún otro mensaje de modo de asegurar que cada uno de los *peers* LDP "sabe" que el otro peer se encuentra funcionando correctamente. En ausencia de mensajes *Keepalive*, o de algún otro mensaje LDP dentro de un intervalo de tiempo, un *LSR* concluye que el *LSR peer*, o que la conexión entre ambos, no se encuentra activa, y termina la sesión LDP.

Mensajes Label Mapping.

Forman una parte fundamental dentro del mecanismo de distribución de etiquetas. Se utilizan para la publicación de información de *binding* entre un *FEC* y una etiqueta.

Mensajes Label Withdrawal.

Son utilizados para "remover" información de *binding* entre un *FEC* y una etiqueta que fuera previamente publicada.

Mensajes Label Release / Label Request / Label Request Abort.

Los mensajes *Label Release* son utilizados por un *LSR* que previamente recibió un mapeo de etiqueta y que ya no necesita de tal mapeo. Esto sucede cuando un *LSR* encuentra que el *Next Hop* para dicho *binding* no es el *LSR* publicado. Para esto el *LSR* se debe encontrar operando en el modo *Conservative Label Retention*.

Un *LSR* puede encontrarse operando en modo de asignación de etiqueta *Unsolicited Downstream* o *Upstream-on-demand*. En este modo un *LSR* pide por un *binding* de etiqueta a su *LSR peer*, en sentido *downstream* o *upstream* dependiendo del modo en que se encuentre operando, utilizando mensajes *Label Request*.

Si un mensaje *Label Request* necesita ser rechazado antes de ser satisfecho, por ejemplo, si el *next hop* para el *FEC* en cuestión ha cambiado, el *LSR peer* aborta el pedido enviando un mensaje *Label Request Abort*.

2.6.4. MODOS DE DISTRIBUCIÓN DE ETIQUETAS.

Los modos de distribución de etiquetas son:

1. *Unsolicited Downstream*.
2. *Unsolicited Upstream*.
3. *Downstream-on-Demand Label Assignment*.
4. *Upstream-on-Demand Label Assignment*.
5. *Ordered LSP Control*.
6. *Independent LSP Control*.
7. *Liberal Label Retention*.
8. *Conservative Label Retention*.

Todos estos tipos de modos de distribución son negociados al inicio de una sesión LDP.

Cuando un *LSR* opera en modo *Conservative Label Retention* sólo almacena los *bindings* entre *FEC* y etiquetas que necesitan en un cierto instante. Toda otra información de *binding* es desechada.

Cuando un *LSR* opera en modo *Liberal Label Retention* almacena todos los *bindings* entre *FEC* y etiqueta que le hayan sido publicados. Este modo de operación permite tener un mejor tiempo de respuesta ante cambios de ruteo, lo cual es importante en dispositivos que almacenan en hardware información de *binding* de etiquetas para luego realizar el *forwarding* de paquetes, como el caso de ATM-LSRs.

2.7. DISTRIBUCIÓN DE ETIQUETAS UTILIZANDO BGP.

Una extensión de BGP-4 permite utilizar a BGP como mecanismo para la distribución de etiquetas en redes MPLS. Estas extensiones definidas en BGP-4 permiten definir *Address Families*, por lo cual para MPLS se define una nueva *Address Family*, definiéndose no sólo el direccionamiento de red sino también una o más etiquetas.

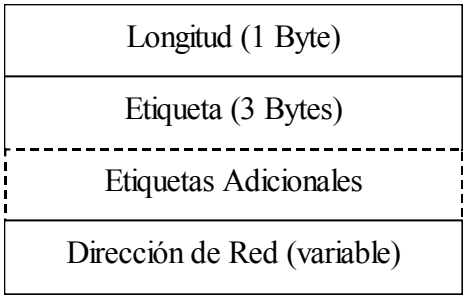


Figura 2.7.1.- Codificación de Etiqueta en BGP-4

El campo Longitud expresa la longitud, en bits, de la etiqueta más la dirección de red. Cada etiqueta es codificada utilizando 3 bytes, donde los últimos 20 bits expresan la etiqueta, el primer bit cumple la misma función que el bit de pila en la encapsulación de *stack*, los restantes 3 bits son setedos en 0.

De esta manera, un dispositivo de red que utilice BGP para la publicación de rutas, puede también realizar la publicación de etiquetas, o pilas de etiquetas, para los paquetes MPLS que utilicen una dada ruta en BGP.

2.8. MULTICAST.

La utilización de MPLS para el transporte de tráfico tipo IP Multicast tiene beneficios sobre la *performance*, y facilita el transporte sobre redes ATM.

Los paquetes IP Multicast utilizan la misma codificación que los paquetes IP Unicast, con la excepción que al utilizar *Label Stack Encoding* el protocolo de capa 1, como por ejemplo, Ethernet o PPP, utiliza un identificador de protocolo de capa 3, como por ejemplo, *ethertype* o *PPP protocol ID*, para indicar que el paquete MPLS etiquetado es transportado dentro del *frame* de capa 2. Sin embargo, se utilizan identificadores diferentes para paquetes IP Unicast e IP Multicast, lo cual tiene ciertas ventajas. Primero, esto hace que los paquetes IP Multicast sean fácilmente reconocidos sin tener que examinar la etiqueta utilizada. Segundo, permite que los paquetes IP Unicast e IP Multicast utilicen espacios de etiquetas diferentes, esto a su vez permite definir una *LFIB*, *Label Forwarding Information Base*, por interfase.

Los Loops presentan un caso particular cuando se trata con tráfico IP Multicast, ya que los paquetes IP Multicast que entran en un Loop pueden ser replicados debido al *forwarding* Multicast, lo cual conduce a incrementar el tráfico no deseado en la red hasta que desaparezca la condición que provoco el Loop. Esto puede ser mitigado con la utilización del campo *TTL* definido en los paquetes MPLS.

El soporte de transmitir tráfico IP Multicast en redes IP MPLS se complica debido al hecho de la variedad de protocolos de ruteo utilizados en IP Multicast. Diferentes protocolos de ruteo IP Multicast pueden generar diferentes estados de *forwarding* que necesitan ser tratados diferentemente al momento de establecer los *LSPs*.

Algunos protocolos de ruteo IP Multicast, como PIM-SM, generan dos estados para el *forwarding* de paquetes, estos son *Shared Tree* y *Source-Specific Tree*. Un *Shared Tree* permite el transporte de paquetes IP Multicast desde cualquier fuente hacia los receptores, mientras que un *Source-Specific Tree* transporta paquetes IP Multicast desde una fuente hacia un grupo de receptores. Los receptores deben estar lógicamente conectados al *Shared Tree* para poder recibir el tráfico generado por las fuentes IP Multicast, o pueden unirse a un *Source-Specific Tree* de modo de optimizar el trayecto entre la fuente y los receptores. Esto puede crear dificultades en un ambiente MPLS cuando un *LSP* Multicast es establecido para ambos casos, *Shared Tree* y *Source-Specific Tree*. De esta forma, una fuente debe enviar paquetes IP Multicast en ambos árboles de distribución, por lo cual un receptor lógicamente conectado en ambos árboles de distribución recibirá paquetes IP Multicast duplicados. Una solución podría ser utilizar MPLS sólo para el árbol de distribución *Source-Specific* y *forwarding* IP convencional para *Shared Trees*.

Otra dificultad que se plantea al utilizar IP Multicast sobre MPLS es la utilización del método de *pyggybacking* sobre los mensajes del protocolo de control, o utilizar un protocolo separado, como LDP, para realizar la distribución de etiquetas. Algunos protocolos como PIM soportan la utilización del método de *pyggybacking*, pero otros como DVMRP no lo soportan.

3. TÉCNICAS DE COMPRESIÓN DE VÍDEO.

La mayor parte de las nuevas aplicaciones de transmisión de vídeo exigen la necesidad de desarrollar estándares de compresión de vídeo internacionales, de manera que permitan el dialogo entre las distintas aplicaciones. Muchas de estas aplicaciones, como vídeo conferencia o juegos en red, presentan características *Real Time* Interactivas, donde juegan un papel muy importante los requerimientos de calidad de servicio como el *delay*, el *jitter*, la pérdida de paquetes y el ancho de banda. Por esto es deseable el contar con una estructura de red que permita la distribución de estas aplicaciones a los receptores basadas en IP Multicast con soporte de requerimientos de calidad de servicio, lo cual nos permitirá tener una opmimización del ancho de banda disponible en los enlaces de la red, además de satisfacer las necesidad de calidad impuestas por cada aplicación en particular.

Debido al gran ancho de banda requerido por estas aplicaciones, y en función de lograr una mejor utilización del ancho de banda, distintos organismo y entes estandarizadores internacionales han recomendado esquemas de compresión de vídeo para varias aplicaciones, como por ejemplo:

- Vídeo Conferencia, ITU H.261 [6], ITU H.263.
- *Storage Media*, MPEG-1 y MPEG-2.
- Televisión Digital (DTV) con MPEG-2.
- Codificación de Vídeo Basada en Objetos MPEG-4.

Los protocolos de red para transmisión de flujos de vídeo se diseñan con el objeto de brindar una transmisión fiable y *on-time* entre los clientes y los servidores. Estos protocolos proveen además direccionamiento de red, transporte, y control de sesión.

Como caso de estudio, a continuación se analizarán la recomendación ITU-T H.323, normas ISO y protocolos de red dedicados a la compresión y transmisión de aplicaciones de vídeo sobre redes IP.

3.1. RECOMENDACIÓN ITU-T H.323.

La ITU ha desarrollado la especificación para Comunicaciones Multimedias basada en paquetes, H323.

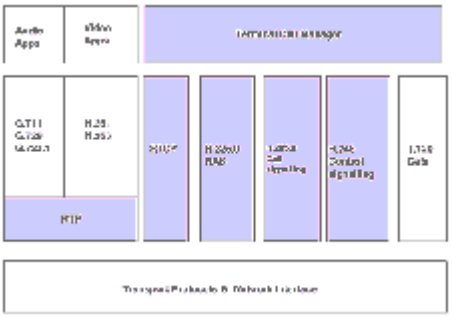


Figura 3.1.1.- Arquitectura Recomendación H.323 (Ref.: 6)

H.323, como se muestra en la figura anterior, es un término "paraguas", ya que hace referencia a otras recomendaciones que incluyen H.225, paquetización y sincronización, H.245, control, H.261 y H.263, *codecs* de vídeo, G.711, G.722, G.728, G.729, y G.723, *codecs* de audio, y la T.120 serie de protocolos de comunicaciones multimedias.

3.2. NORMAS ISO MPEG-X

ISO ha desarrollado las normas MPEG-1, MPEG-2, MPEG-4 y MPEG-7. Cada una de estas normas realmente es un juego de especificaciones para los diferentes aspectos de la compresión y codificación de vídeo, codificación de audio, múltiplexación de vídeo, etc. MPEG-4 es la primera norma que se basa en la codificación de vídeo y audio basada en objetos. MPEG-4 incluye la definición de audio y vídeo óptimo, que codifican la textura, y la representación en 2D o 3D.

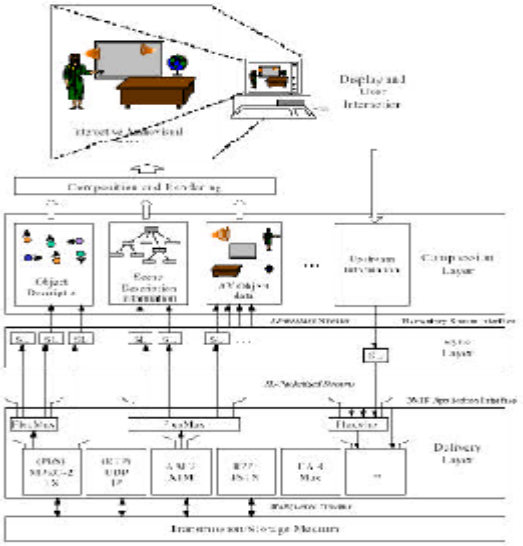


Figura 3.2.1.- Estructura Sistema MPEG-4 (Ref.:7)

[6] Tomado Referencia Bibliográfica 4.
[7] Tomado Referencia Bibliográfica 4.

El sistema MPEG-4 esta dividido en tres capas:

- Capa de condensación.
- Capa de sincronización.
- Capa de entrega, *Delivery*.

La capa de condensación realiza todos los medios de comunicación convenientes, codificando y decodificando de y desde flujos de vídeo. La figura anterior explica la manera en que una escena audiovisual en MPEG-4 está compuesta de objetos individuales, por un árbol de descripción de escena; los objetos visuales que corresponden al orador y la voz son tratados juntos formando un *compound media object*, el cual contiene los componentes de audio y vídeo del orador.

La capa de sincronización maneja flujos elementales y la sincronización de los mismos.

La capa de entrega, *Delivery*, asegura transparente acceso al contenido, independientemente de las tecnologías de entrega de paquetes, como *Real-Time Transport Protocol*, RTP, MPEG-2, H.223 o ATM.

En términos de la estructura de compresión de vídeo, pueden entenderse a los codificadores de vídeo de MEG-4 como una generalización de H.263. MPEG-4 utiliza la idea de *video objects* los cuales corresponden a entidades en el flujo de bits, que el usuario puede manipulear y modificar.

El esquema de compresión de vídeo de MPEG-4 tiene una estructura muy similar a MPEG-1/2 y H.26X. Sin embargo, la contribución más importante de MPEG-4 está en que proporciona muchas nuevas herramientas que apoyan nuevas funcionalidades. De estas nuevas funcionalidades, es obvio que la aplicación más importante de MPEG-4 se encuentra en el ambiente multimedia conectando una red de computadoras. Debe notarse que MPEG-4 soporta todas las funcionalidades proporcionadas por MPEG-1 y MPEG-2.

Las principales ventajas de MPEG-4 se pueden resumir en:

- **Interactividad Basada en Contenido (Content-Based Interactivity).**

Codificación y representación de objetos de vídeo, en lugar de los marcos de vídeos. Estas son dos de las más importantes nuevas funcionalidades, definidas en MPEG-4 que

pueden habilitar aplicaciones basadas en contenidos, *content-based applications*.

- **Compresión Eficiente.**

La eficacia de compresión ha sido el enfoque principal de MPEG-1 y MPEG-2, lo cual condujo a aplicaciones como televisión digital, DTV, y DVD. La mejora en la eficiencia de la codificación de los flujos de bits, y la posibilidad de multiplexar distintos flujos, le da a MPEG-4 una mejor aceptación.

- **Acceso Universal.**

La robustez en ambientes propensos a errores, le permite a MPEG-4 la codificación de contenido sobre un amplio rango de medios de transmisión, como redes móviles e Internet. Además, la escalabilidad temporal, espacial y granular, y la escalabilidad de objetos, le permiten al usuario final decidir dónde usar recursos incluyendo ancho de banda y recursos computacionales.

3.3. PROTOCOLOS PARA LA TRANSMISIÓN REAL-TIME MULTIMEDIA.

El objeto de estos protocolos es la transmisión fiable y *based-time* entre clientes y servidores de vídeo.

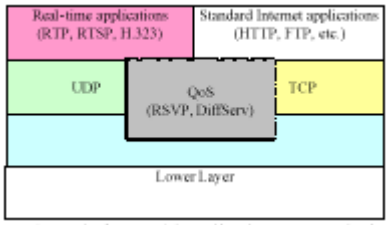


Figura 3.3.1.- Stack de Protocolos para Transmisión Multimedia sobre Internet (Ref.: 8)

Se los puede clasificar en cuatro categorías:

- Capa de Red, *Network*.
- Transporte.
- Señalización, *Signaling*.

[8] Tomado Referencia Bibliográfica 3.

- Control de Sesión, *Session Control*.

Capa de Red.

La Capa de Red de Internet está compuesta por el Protocolo de Internet, IP, por el Protocolo UDP, TCP y otros protocolos asociados con QoS. Aunque TCP asegura comunicaciones libres de errores, no es aplicable para transmitir aplicaciones multimedia *time-critical*, ya que el retardo debido a los mecanismos de retransmisión pueden resultar inadmisibles. UDP, que no ofrece mecanismos de recuperación de errores, es más conveniente para transmitir flujos de multimedia en tiempo real.

Transporte.

El transporte de datos en tiempo real es realizado por el protocolo RTP y controlado por el protocolo RTCP. RTP normalmente se basa sobre UDP/IP. RTP proporciona definiciones de formato de paquete básicos, para permitirle a UDP realizar las comunicaciones en tiempo real.
Pero RTP no define mecanismos de control o algoritmos. RTP es a menudo integrado en los mecanismos de procesamiento de una aplicación, en lugar de ser considerado como una capa separada. Como otros protocolos, RTP tiene un protocolo de control, RTCP. RTCP no transfiere datos. En cambio, transfiere información de control relacionada a un flujo de datos.

Señalización (Signaling).

H.225 y HTTP pueden utilizarse como protocolos de señalización. La señalización constituye el envío de anuncios sobre una sesión multimedia a los participantes de la comunicación, o el envío de "invitación" a ciertos participantes de modo que se unan a la sesión. HTTP puede utilizarse para anunciar sesiones utilizando un formato *bulletin board*. En el sistema H.323, H.225 puede usarse para establecer una conexión entre dos H.323 *endpoints*.

Control de Sesión (Session Control).

Los protocolos de control de sesión más útiles incluyen RTSP y H.245. El control de la sesión define el mensaje y procedimientos para controlar la entrega de los datos multimedia durante una sesión establecida. Por ejemplo, RTSP soporta ordenes VCR como adelantar, rebobinar, pausa, stop, etc. Se basa sobre TCP o UDP. En el sistema H.323, H.245 puede intercambiar mensajes de control extremo-a-extremo que gobiernan el funcionamiento de los H.323 *endpoint*.

4. IP MULTICAST SOBRE MPLS.

Para los casos de transmisión de aplicaciones en Tiempo Real Interactivas donde el poder controlar y definir parámetros de calidad de servicio, como ancho de banda o *delay*, se torna una funcionalidad muy importante, el tranporte de IP Multicast sobre MPLS resulta una alternativa muy atractiva. Como veremos en el desarrollo del presente capítulo, y en el capítulo 5 de la presente Tesis, hoy existen ciertas dificultades en la implementación de este tipo de redes y soluciones, si bien se plantearán distintas alternativas para poder solucionarlas y así obtener los beneficios que nos brindan las tecnologías planteadas.

4.1. PRINCIPALES CARACTERÍSTICAS DE IP MULTICAST SOBRE REDES MPLS.

En redes IP MPLS, las rutas se generan a partir de protocolos de ruteo de capa tres, luego estas rutas pueden ser "mapeadas" en caminos, o *paths*, de capa dos para mejorar la performance de la red.

En el caso de IP Multicast, la solución óptima, es decir, mínimo costo para interconectar un número N de nodos, implica la utilización de árboles de distribución. No existe un protocolo IP Multicast capaz de generar árboles de distribución con el mínimo costo, y diferentes protocolos generan diferentes árboles de distribución.

Las limitaciones más serias presentadas por MPLS son:

1. Espacio de Etiquetas Limitado: el número de bits disponible para la especificación de una etiqueta puede ser insuficiente en algunos escenarios, lo cual limita la cantidad de *LSPs* que pueden ser establecidos.

2. *Merging*: algunas tecnologías de capa 2, no soportan conexiones multipunto a punto a punto y o multipunto a multipunto, lo cual disminuye las posibilidades de *merging* de *LSPs*.

3. *TTL*: algunas tecnologías de capa 2, no soportan la implementacion de *TTL*, lo cual implica, en algunos casos, el no contar con mecanismos de prevención de Loops.

Estas limitaciones pueden tener impacto y deben ser tenidas en cuenta.

Al implementar soluciones de IP Multicast, sobre redes IP MPLS se deben tener en cuenta los siguientes aspectos o características basadas en las necesidades y características de los protocolos IP Multicast:

1. Agregación.
2. *Flood* y *Prune*.
3. Árboles de Distribución Multicast.
4. Coexistencia de diferentes árboles de distribución.
5. Árboles de distribución Bidireccionales y Unidireccionales.
6. Encapsulación de los paquetes Multicast dentro de paquetes etiquetados MPLS.
7. Prevención de Loops.

Agregación.

La granularidad de los flujos de datos en IP Multicast esta dada por (*,G), para árboles de distribución *Shared Tree*, y por (S,G), para árboles de distribución *Source Tree*. Lo cual implica poder realizar agregación de árboles de distribución IP Multicast con diferentes fuentes de tráfico y receptores sobre un mismo *LSP*.

Flood y Prune.

Para el establecimiento de un árbol de distribución IP Multicast algunos protocolos de ruteo, como PIM-DM y DVMRP, “inundan”, *flood*, la red con la transmisión de mensajes Multicast. Las ramas de estos árboles de distribución pueden ser desactivadas, *prune*, por nodos que no “quieren” recibir más tráfico IP Multicast perteneciente a un dado grupo. Estos procesos son repetidos en forma periódica.

Esto implica la necesidad de poder contar con el “mapeo” en forma dinámica de árboles de distribución de capa 3 punto a multipunto, en un *LSP* de capa 2 punto a multipunto, en forma eficiente en lo referente a *Overhead* y tiempos de establecimiento. Esto produce una elevada utilización de etiquetas MPLS a través de la red, lo cual es una desventaja teniendo en cuenta el espacio reducido de etiquetas presentado para algunas tecnologías de capa 2, como ATM y Frame Relay.

En caso de utilizar el mecanismo *Data-Driven* los *router*s sólo generarán estados en sus tablas de ruteo IP Multicast, para un cierto grupo, cuando reciban datos destinados a un grupo en particular. Estos estados pueden ser "borrados" de sus tablas de ruteo cuando el *router* detecta un cierto tiempo de inactividad. Para esto, es necesario contar con un mecanismo que permita minimizar el tiempo entre el arribo de paquetes IP Multicast, y el establecimiento de un *LSP*. Además, como "creación" y "borrado" de rutas creadas por el protocolo IP Multicast de capa 3 pueden ser disparadas en cada nodo de la red por condiciones de tráfico, el *LSP* asociado con una ruta debe poder ser creado y desactivado teniendo en cuenta condiciones de tráfico IP Multicast. Finalmente, si un *LSR* no soporta el *forwarding* de tráfico utilizando el mecanismo de *Data-Driven*, requerirá que el *LSR* en sentido *Upstream* tome la decisión de crear el *LSP* correspondiente,y la publicación de información de etiquetas utilizando los mecanismos *Upstream Unsolicited* o *Downstream on Demand*.

Árboles de Distribución Multicast.

La ventaja de utilizar árboles de distribución *Shared Trees*, cuando se utiliza *Label Switching*, es que estos consumen menor cantidad de etiquetas que los *Source Trees*. En el primer caso se requiere una etiqueta por grupo IP Multicast, mientras que el segundo caso requiere la utilización de una etiqueta por fuente y por grupo IP Multicast. Además, se suma el hecho que el "mapeo" de un *Shared Tree*, de capa 3, en caminos de capa 2, implica el establecimiento de *LSPs* multipunto a multipunto.

Coexistencia de Diferentes Árboles de Distribución.

Algunos protocolos de ruteo IP Multicast, como PIM-SM, soportan ambos tipos de árboles de distribución, por lo cual un *router* puede necesitar mantener estados para ambos árboles de distribución para un mismo grupo IP Multicast. Para poder contar con estas características, dentro de redes IP MPLS, podemos tener los siguientes casos:

a. Terminar los *LSPs* y realizar el *forwarding* del tráfico de un grupo determinado en capa 3. Sin embargo, el tener que regresar a capa 3, puede ser evitado en caso que una entrada en la tabla de ruteo del tipo (S,G), sin una interfase de salida asociada, sea agregada en la tabla de ruteo IP Multicast, *MRT*. La desventaja de esto es que se tendría tráfico duplicado durante un cierto instante de tiempo.

b. Asignar etiquetas específicas por fuentes en los nodos donde se tiene *Shared Tree*. De esta manera, múltiples etiquetas serían asociadas con una entrada (*,G), correspondiendo una etiqueta por fuente activa. Como un nodo en un *Shared Tree*

sólo reconoce a una fuente como activa cuando recibe tráfico de dicha fuente, el *LSP* no puede ser establecido hasta tanto no se cumpla con esta condición, por lo cual se necesita contar con un mecanismo rápido para el establecimiento del *LSP*.

c. Sólo realizar el *switching* MPLS para el caso de árboles de distribución *Source Tree*, y realizar el *forwarding* del tráfico de los árboles de distribución *Shared Tree* utilizando protocolos de ruteo IP. Con esto se asume que los árboles *Shared Tree* sólo serán utilizados por los receptores de un grupo IP Multicast para descubrir las fuentes de tráfico.

d. Un *LSR* que tenga un estado (S,G), con una interfase de salida no nula, publicará una etiqueta para el estado (S,G) al *LSR* en sentido *Upstream*, y así esta etiqueta será propagada en sentido *Upstream* por los *LSR* hasta el *RP*. De esta manera, un *LSP* dedicado es creado para el tráfico con destino (S,G), desde el *RP* hacia el *LSR*. En el último *LSR*, el (S,G) *LSP* es "mapeado", o *merging*, sobre un (*,G) *LSP*, para la interfase de salida apropiada. Esto asegura que los paquetes IP Multicast con destino (S,G), siendo transportados sobre el *Shared Tree*, no "pasen" a través de un *LSR* el cual haya "desactivado" la fuente *S*.

Árboles de distribución Bidireccionales y Unidireccionales.

Los árboles de distribución Bidireccionales tienen la desventaja de crear un gran número de puntos de unión, *merging points*, en cada uno de los N nodos del mismo. Lo cual complica el establecimiento de los *LSPs*.

Los árboles de distribución Unidireccionales "tunelizan" la transmisión de los paquetes de datos IP Multicast hacia el nodo raíz del árbol, *DR*, utilizando un único punto de unión, *merging point*. De esta manera el tráfico IP Multicast, a través de un *LSP*, circula en forma unidireccional entre las fuentes y los receptores, lo cual facilita el establecimiento y funcionamiento del mismo.

Encapsulación de los Paquetes IP Multicast dentro de Paquetes Etiquetados MPLS.

Las fuentes de tráfico IP Multicast, dentro de árboles de distribución *Shared Trees Unidireccionales*, y fuentes no-miembros de un grupo IP Multicast, en árboles de distribución *Shared Trees Bidireccionales*, encapsulan la transmisión de los datos hacia el nodo raíz. Esta encapsulación, y la posterior desencapsulación en el nodo raíz, son realizadas a través de procesos de capa 3. Por lo cual, un camino, o *path*, en capa 2,

nunca puede ser "mapeado" dentro de un *end-to-end LSP*. Es decir, el tráfico no podrá ser *"forwardeado"* utilizando las facilidades de MPLS en capa 2, sobre la interfase del *DR* que envía los paquetes, encapsulación.

En caso que el *LSR* soporte *forwarding* a nivel de capa 2 y capa 3, el tráfico definido por (S,G), en el *DR*, puede ser *"forwardeado"* a nivel de capa 2 en todas las interfases de salida, salvo en la interfase registrada.

Si el nodo raíz decide "unir", o registrar, a la fuente, un *end-to-end* (S,G) *LSP* puede ser establecido.

Prevención de Loops.

Contar con mecanismos para la prevención y detección de Loops, en el caso de tráfico IP Multicast, resulta un factor importante ya que, cuando un paquete IP Multicast es "atrapado" en un Loop, copias del paquete pueden ser emitidas desde el Loop si existen ramas de un árbol de distribución dentro del Loop.

4.2. FORWARDING EN CAPA 2 Y CAPA 3 EN UN NODO.

Debido al hecho que el tráfico IP Multicast puede ser *"forwardeado"* por una o más interfases de salida, el mismo puede ser *"forwardeado"* en capa 2 para alguna de las ramas del árbol de distribución, y en capa 3 para otras ramas del mismo. Esto requiere la capacidad de poder dividir un árbol de distribución en ramas, sobre las cuales el tráfico puede ser *"forwardeado"* en capa 2 o en capa 3.

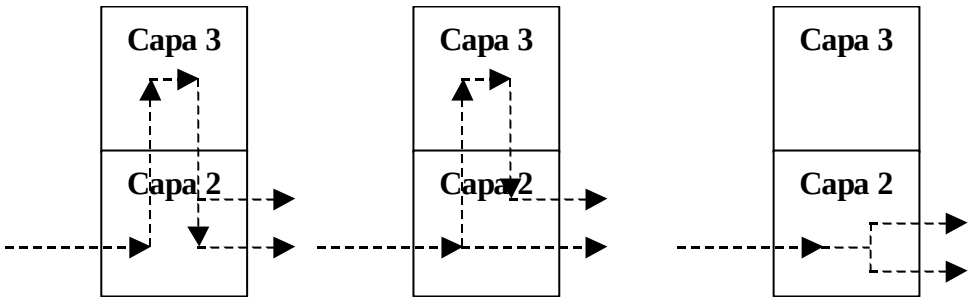


Figura 4.2.1.- Forwarding Multicast en Capa 2 y Capa 3.

El *forwarding* en capa 3 debe tener en cuenta el no "*forwardear*" tráfico sobre las ramas del árbol en las que ha *"forwardeado"* tráfico en capa 2. Esto puede ser logrado agregando un bit de información extra en las tablas de ruteo IP Multicast.

El *forwarding* de paquetes en capa 3 produce una carga de procesamiento en los *router*s MPLS menor que la producida en el *forwarding* de capa 3 en un *router* nativo, sólo de capa 3.

La posibilidad de poder contar con *forwarding* de capa 3 y capa 2 en forma mixta, permite la utilización de umbrales que “disparan” la distribución de etiquetas en los modos de distribución *Downstream Unsolicited* o *Upstream on Demnand*.

4.3. CREACIÓN DE LSPs PARA TRÁFICO IP MULTICAST.

La creación de *LSPs* para flujos de datos IP Multicast puede ser generada por diferentes eventos, los cuales pueden ser “mapeados” sobre los siguientes mecanismos:

a. *Request Driven*: se basa en la intercepción de los mensajes de control recibidos o enviados.

b. *Topology Driven*: se basa en el ”mapeo” de árboles de distribución IP Multicast en capa 3 a árboles de distribución en capa 2. Este proceso es realizado aun cuando no se tenga tráfico en dicho árbol de distribución.

c. *Traffic Driven:* se basa en el ”mapeo” de árboles de distribución IP Multicast en capa 3 a árboles de distribución en capa 2. Este proceso es generado cuando se recibe tráfico IP Multicast sobre el árbol de distribución en capa 3.

4.3.1. REQUEST DRIVEN.

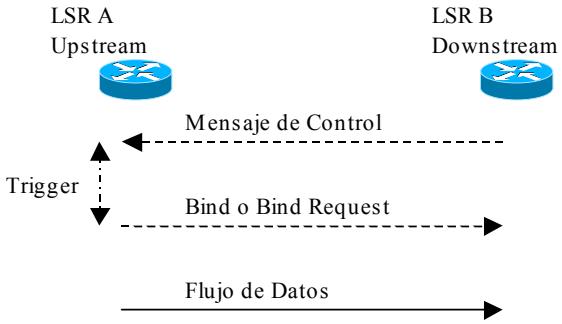


Figura 4.3.1.1.- Creación de LSPs Mensajes de Control Recibidos.

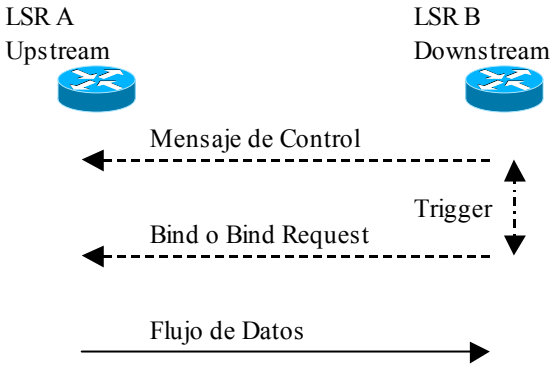


Figura 4.3.1.2.- Creación de LSPs Mensajes de Control Enviados.

Para el caso de tráfico IP Multicast, podemos tener básicamente dos tipos de mensajes de control:

4.3.1.1. MENSAJES DE CONTROL DE RUTEO MULTICAST.

En principio, este tipo de mecanismo puede ser utilizado por protocolos de ruteo IP Multicast, los cuales utilizan señalización explícita, como los mensajes *Join* en PIN.SM y CBT. DVMRP y PIM-DM pueden utilizar este mecanismo pero realizando una modificación sobre los mismos.

Los mensajes de ruteo IP Multicast pueden generar estados por *Hardware*, como CBT, o por *Software*, como PIM-SM. La generación de estados por *Software* implica una mayor utilización de procesamiento en los *routers LSRs*.

Los *Triggers* basados en mensajes generados por los protocolos de ruteo IP Multicast tienen la desventaja que los cálculos generados por los algoritmos de ruteo para generar la tabla de ruteo son repetidos por el módulo MPLS. Ya que en el primer caso se calculan los árboles de distribución a nivel de capa 3, y en el segundo caso se generan los árboles de distribución a nivel de capa 2. Este mecanismo se vuelve más complejo cuando los protocolos de ruteo IP Multicast generan árboles (*,G) y (S,G).

4.3.1.2. MENSAJES DE CONTROL MULTICAST RESOURCE RESERVATION.

Utiliza mensajes RSVP para la generación de los *LSPs*. Una fuente de tráfico IP Multicast enviará mensajes RSVP *Path* al grupo de receptores en sentido *Dowstream* a través del árbol de distribución, luego los receptores contestarán enviando mensajes RSVP *Resv*. Este mecanismo cuenta con la escalabilidad que le brinda RSVP, es decir:

a. Los mensajes RSVP *Resv* soportan la facilidad de *merging*.

b. Los mensajes RSVP *Resv* son sólo enviados sobre la primera rama del árbol de distribución que realiza una reservación de recursos.

4.3.2. TOPOLOGY DRIVEN.

La tabla de ruteo IP Multicast es generada y actualizada por los protocolos de ruteo IP Multicast. El módulo MPLS en los *LSR* mapeará la topología generada por esta, en capa 3, en *LSPs* punto a multipunto en capa 2.

La desventaja de este mecanismo es que se consumen etiquetas MPLS aun cuando no se tenga transmisión de flujos de datos IP Multicast.

4.3.3. TRAFFIC DRIVEN.

Sólo se construyen *LSPs* para árboles de distribución IP Multicast que transporten tráfico, lo cual reduce la utilización de etiquetas MPLS.

Si los *LSRs* no soportan realizar un *forwarding* combinado entre capa 2 y capa 3, este mecanismo para la generación de *LSPs* requerirá de un método de distribución de etiquetas en el cual la información de etiquetas es solicitada por el *LSR* de *Upstream*.

MPLS Multicast es independiente de los protocolos de ruteo IP Multicast en si.

4.4. PIGGY-BACKING.

Brinda un mecanismo alternativo a enviar dos mensajes separados. Así, la información de etiqueta puede ser obtenida realizando un *piggy-backing* sobre los mensajes de control. Podemos tener dos alternativas:

a. Mensajes de Ruteo IP Multicast: los protocolos PIM-SM y CBT poseen mensajes explícitos de *Join*, los cuales permiten transportar la información de etiqueta mapeada sobre los mismos mensajes. Para otros tipos de protocolos de ruteo IP Multicast será necesario realizar una extensión de los mismos.

b. Mensajes RSVP *Resv*: se utiliza una extensión de objeto de modo de permitir el mapeo de la información de etiqueta sobre este tipo de mensajes.

Este método presenta las siguientes ventajas y desventajas:

Ventajas.

a. Se tiene una sincronización entre la distribución de etiquetas y la distribución de rutas IP Multicast. Lo cual permite minimizar el intervalo entre el establecimiento de la ruta IP Multicast y el mapeo de la etiqueta MPLS para una dada ruta.

b. La cantidad de mensajes de control necesarios para realizar la publicación de etiquetas, más allá de las necesarias para soportar protocolos de ruteo IP Multicast, es cero.

Desventajas.

a. Para protocolos de ruteo IP Multicast de características *Dense Mode*, como PIM-DM, no se tiene la posibilidad de realizar *piggy-backing* sobre mensajes de control.

b. No es posible tener co-existencia de árboles de distribución IP Multicast *Source* y *Shared Tree*.

c. Requiere la extensión de algunos protocolos de ruteo IP Multicast para soportar la funcionalidad de *piggy-backing*, lo cual sucede en caso de tener que soportar múltiples protocolos de ruteo IP Multicast.

d. Implica la utilización del modo de distribución de etiquetas *Downstream Unsolicited*, lo cual excluye la posibilidad de utilizar alguno de los métodos de *Trigger* según se vio en puntos anteriores.

e. LDP utiliza TCP para asegurar una comunicación confiable entre *LSRs*. La utilización de *piggy-backing* a menudo reemplaza esta comunicación confiable con mensajes periódicos *soft-states*. Esto incrementa la cantidad de mensajes de control de tráfico.

f. Si se requiere la utilización de un mecanismo VCID la notificación, *inband*, puede ser realizada enviando el *binding* de LDP a través del nuevo VC, requiriéndose sólo un mensaje. Este método no se puede combinar con la funcionalidad de *piggy-backing*, ya que la información de ruteo es enviada antes que el VC sea establecido.

4.5. EXPLICIT ROUTING.

Explicit Routing, para el caso de tráfico IP Unicast, implica el poder "sobreescribir" la tabla de ruteo IP Unicast utilizando *LSPs*.

Para el caso de tráfico IP Multicast, implicaría la posibilidad de "sobreescribir" el árbol de distribución establecido por el protocolo de ruteo Multicast utilizando la información de otro *LSP*. Así, el protocolo de ruteo *shortest path* se vuelve obsoleto y puede ser reemplazado por los mensajes de publicación de información de etiqueta que siguen una ruta explícita.

4.6. DIFFSERV.

Esta funcionalidad permite tener una mayor granularidad en los flujos de datos IP Multicast, requiriendo la utilización de un nuevo código *DSCP*, *DiffServ Codepoint*, para la designación de los árboles de distribución IP Multicast. Así, una fuente de tráfico IP Multicast puede establecer uno o más árboles de distribución con diferentes *DSCPs*.

De esta manera, la asignación de los árboles de distribución IP Multicast serán (S,G,DSCP) o (*,G, DSCP). Estos pueden ser "mapeados" fácilmente sobre *LSPs* MPLS utilizando el método de *Traffic Driven Trigger*. Así, se pueden generar *LSPs* con diferentes atributos. Estos *LSPs* utilizarán la misma ruta IP Multicast, ya que la construcción de los mismos no toma en cuenta la información de DCSP.

4.7. REDES MULTIACCESO.

En el caso de IP MPLS Multicast sobre redes multiacceso, como Ethernet, los *LSRs* que requieran unirse a un grupo IP Multicast deben estar siempre "listos" para aceptar la etiqueta que fue asignada a un dado grupo *LSP* Multicast. Esto puede lograrse de las siguientes formas:

a. Cada *LSR* conectado a la red Multiacceso almacena todas las etiquetas publicadas sobre el enlace multiacceso, aun cuando no haya recibido un mensaje de *Join* para un determinado grupo IP Multicast. Si un nuevo *LSR* es conectado a la red multiacceso, debe recibir la información de las etiquetas asignadas de algún otro *LSR* conectado al enlace, y en caso de utilizar publicación de etiquetas mediante el método *Soft State*, puede esperar un cierto tiempo hasta que sea capaz de almacenar etiquetas por él mismo.

b. Cada *LSR* obtiene su propio rango del cual realizar el almacenamiento y asignación de etiquetas. Así, si un nuevo *LSR* se agrega sobre el enlace multiacceso, el rango de etiquetas deberá ser negociado con el resto de los *LSR* conectados.

c. Para cada enlace multiacceso, un *LSR* puede ser elegido para ser responsable del almacenamiento y asignación de etiquetas. Así, cuando un *LSR* necesita realizar la asignación de una etiqueta, obtiene esta consultando al *LSR* designado para tal fin.

En el caso que un flujo de datos IP Multicast tenga más de un *LSR Downstream*, los cuales requieran utilizar la misma etiqueta MPLS, podemos tener dos soluciones:

a. Realizar la publicación de la información de etiqueta en modo Multicast hacia todos los *LSR*s conectados sobre el enlace compartido.

b. Un *LSR* enviará la información de etiqueta en modalidad Unicast, utilizando un protocolo confiable, TCP, hacia uno o todos los *LSRs* sobre el enlace compartido que requieran esta información.

Debido a que el establecimiento de *LSPs* considera si se tiene un tiempo de vida, *lifetime*, suficiente, y teniendo en cuenta que el número de *LSRs* sobre un enlace multiacceso es limitado, la opción ***b*** presenta condiciones más ventajosas.

Otro punto a tener en cuenta para el caso de enlaces multiaccesos es la decisión entre utilizar asignación de etiquetas *Upstream* o *Downstream*. Para tráfico IP Multicast, la asignación de etiquetas en sentido *Upstream* es más simple, ya que puede haber un sólo nodo *Upstream* que por enlace pertenezca a un árbol de distribución IP Multicast. Este nodo asignará una única etiqueta para un dado *FEC*. Con la asignación en modo *Downstream* puede haber múltiples nodos para un árbol determinado en un enlace multiacceso, cada nodo puede proponer una etiqueta diferente para un dado *FEC*, lo cual requerirá de la utilización de algún proceso determinado para garantizar una única etiqueta MPLS para un dado *FEC* sobre un enlace multiacceso.

Una vez que una etiqueta fue asignada es posible que el nodo, o *LSR*, que realizó la asignación de la misma “abandone” el enlace multiacceso. Para el caso de asignación *Downstream*, esto puede suceder cuando el *LSR* que asignó la etiqueta deje de pertenecer al grupo Multicast. En el caso de asignación de etiquetas utilizando la metodología *Upstream*, puede suceder cuando el *LSR Upstream* cambie debido a un cambio en la topología.

4.8. MECANISMOS DE CONTROL PARA LA DISTRIBUCIÓN DE ETIQUETAS.

En el caso del Control de Distribución de Etiquetas Independiente, cada *LSR* puede tomar la iniciativa de realizar un mapeo de etiqueta MPLS. En el control Ordenado, un *LSR* sólo puede mapear una etiqueta cuando este ya ha recibido una etiqueta proveniente del *next-hop*.

Si se utiliza el método de *Requets Driven*, con el mecanismo de Control Independiente, este se comporta como un método de Control Ordenado, en el cual el pedido de etiquetas es realizado por el *LSR* en sentido *Upstream*.

4.9. CONSIDERACIONES DE SEGURIDAD.

En general la utilización de IP MPLS Multicast no brinda mayores características de seguridad de las que se tiene en IP Multicast o MPLS por separado.

Cuando el árbol de distribución IP Multicast es determinado por alguno de los protocolos de ruteo IP Multicast existentes, los mecanismos de seguridad son los brindados por el protocolo de ruteo en si.

4.10. DISTRIBUCIÓN DE ETIQUETAS EN PIM-SM.

PIM-SM permite a los receptores IP Multicast unirse a un árbol de distribución *Shared Tree*, de la forma (*,G), para un dado grupo IP Multicast G, con un *RP*, *Redezvous Point*, como raíz del árbol de distribución, o unirse a un árbol de distribución *Shortest Path*, de la forma (S,G). De esta forma, un receptor IP Multicast puede recibir tráfico Multicast de una fuente S a través de un árbol de distribución (S,G), o de otras fuentes de tráfico IP Multicast, a través de un árbol (*,G). Cabe notar, que algunos miembros de un grupo G, pueden recibir tráfico Multicast a través del árbol (*,G), mientras que otros miembros del mismo grupo recibirán tráfico IP Multicast a través del árbol de distribución (S,G). Así, el *DR*, *Designated Router*, deberá ser capaz de *"forwardear"* tráfico IP Multicast en ambos tipos de árboles de distribución.

En MPLS surge el problema cuando un nodo *LSR* en un árbol (*,G) necesita *"forwardear"* tráfico IP Multicast en forma diferenciada dependiendo del tipo de árbol de distribución.

Consideremos el siguiente caso:

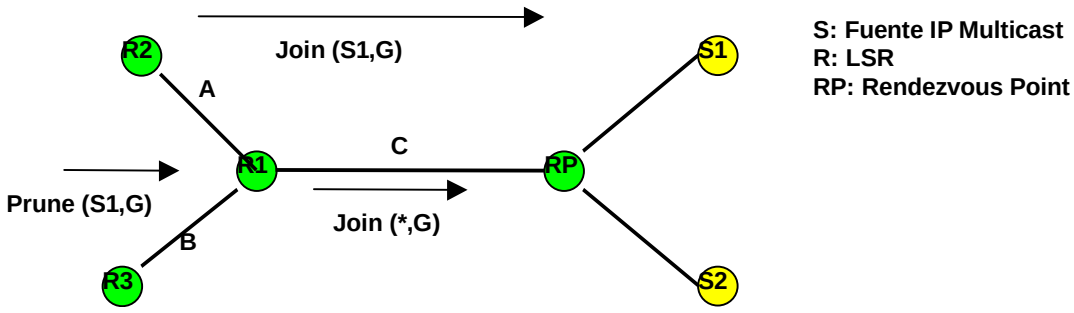


Figura 4.10.1.- Ejemplo PIM-SM en MPLS.

Supongamos que el *LSR* R1 no esta interesado en recibir tráfico proveniente de la fuente S1 a través del árbol de distribución (*,G), ya que se ha unido al árbol (S1,G). Por lo tanto, enviará un mensaje *Prune* (S1,G). De esta forma, R1 *"forwardeara"* tráfico IP Multicast sobre la interfase B, mientras que para la fuente S2, *"forwardeara"* tráfico IP Multicast sobre la interfase A y B. Para esto, en MPLS, no es posible asignar una etiqueta común para todo el tráfico entrante a de R1 sobre la interfase C. Para el tráfico proveniente de S1, sobre la interfase C, se le debe asignar una etiqueta distinta que el proveniente de S2.

La tabla de Ruteo en R1 será:

(*,G)	iif={C}	oif={A,B}
(S1,G)	iif={C}	oif={B}

Así, para el caso de utilizar PIM-SM en redes IP MPLS, el utilizar el método *Topology Driven* para la asignación de etiquetas MPLS producirá un incorrecto *"forwardeo"* de tráfico.

Un *LSR* que soporta IP Multicast envía mensajes PIM *Join/Prune* como resultado de los *Host* que se unen a grupos IP Multicast. Estos mensajes son enviados en sentido *Upstream* a los *LSRs* adyacentes hacia el *RP*, para el caso de un *Shared Tree* (*,G), o hacia una fuente de tráfico IP Multicast, para el caso de un *Source Tree* (S,G). Las etiquetas MPLS son distribuidas asociándolas con las direcciones de la lista *Join* o *Prune*.

Si un *LSR* se une a un grupo IP Multicast en un árbol de distribución *Shared Tree*, enviará mensajes *Join/Prune* en sentido *Upstream*, los cuales contendrán la dirección del grupo IP Multicast y una lista de *Join*. La lista *Join* contendrá un elemento, el cual tendrá la dirección del *RP*, además tendrá una etiqueta MPLS. Esta etiqueta será utilizada por el *LSR Upstream* para enviar tráfico IP Multicast en sentido *Downstream* a través del árbol IP Multicast. Esta etiqueta especifica la ruta, en sentido *Downstream*, desde el *LSR* a través del árbol IP Multicast.

Si el *LSR* en sentido *Downstream* se une al *Shared Tree* para un determinado grupo G, y el *LSR Upstream* tiene en su tabla el estado (S,G) para una fuente S, el *LSR Upstream* deberá unir su lista (*,G), de la interfase de salida, al estado (S,G). Esto se debe realizar debido a que se debe asegurar que el *LSR Downstream* reciba los paquetes IP Multicast enviados desde S al grupo G. En este caso, cuando el *LSR Upstream* reciba un paquete (S,G), este lo *"forwardeara"* hacia el *LSR Downstream*, utilizando la misma etiqueta que el *LSR Downstream* asigno para los paquetes (*,G). En caso que el *LSR Upstream* sea un *ATM-LSR*, esto no podrá ser realizado, debido a que no soporta conexiones multipunto-multipunto.

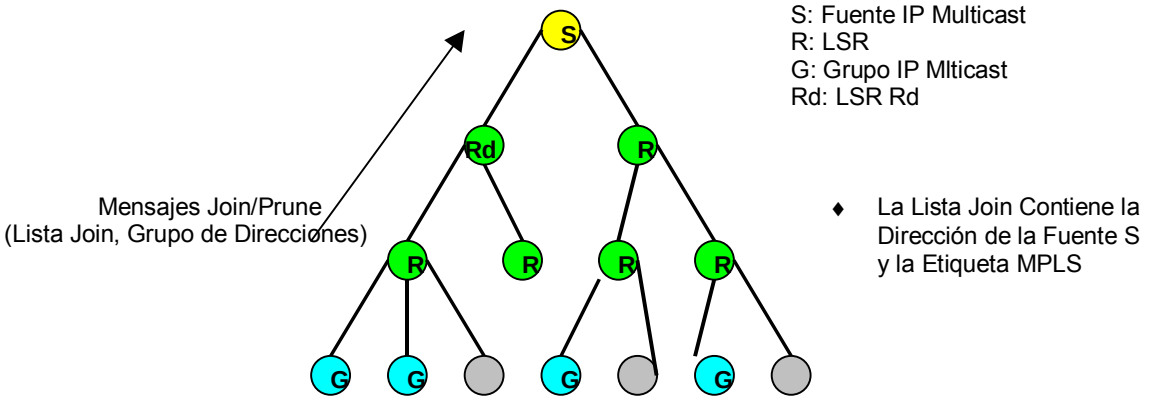


Figura 4.10.2.- Ejemplo de Piggybacking para un LSR Rd que se une a un Árbol Source Tree (S,G).

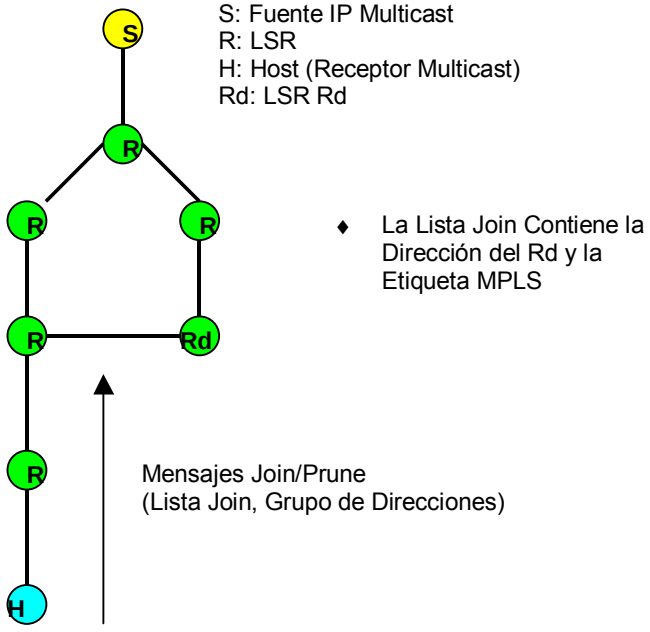


Figura 4.10.3.- Ejemplo de Piggybacking para un LSR Rd que se une a un Árbol Shared Tree (*,G).

PIM-SM *Shortest Path Tree* puede verse como un equivalente a PIM-DM, una etiqueta es asignada utilizando el método *hop-by-hop Traffic Driven* para cada entrada (S,G) de la tabla de ruteo. Una manera de solucionar el problema de la co-existencia de árboles (S,G) y (*,G), sin necesidad de utilizar *forwarding* a nivel de capa 3, nivel IP, es asignando etiquetas especificas en nodos intermedios del árbol de distribución Multicast. De esta forma, múltiples etiquetas serán asociadas con una entrada (*,G), correspondiendo una etiqeta por cada nodo activo del árbol de distribución Multicast. La forma de asignar una etiqueta de forma univoca por fuente (*,G) desde un *binding* de etiqueta desde un grupo (S,G), sería introduciendo un *FEC* (G,S), el cual representaría paquetes IP desde la fuente S, los cuales serán "*forwardeados*" en el árbol de distribución Multicast (*,G). De esta manera, PIM-SM soportaría la asignación de etiquetas por fuente.

4.11. PROTOCOLOS IP MULTICAST DENSE-MODE EN IP MPLS.

Para entender el problema, comenzaremos considerando un ejemplo de implementación del método de control *Topology Driven* utilizando tráfico IP Unicast, para lo cual planteamos la siguiente situación:

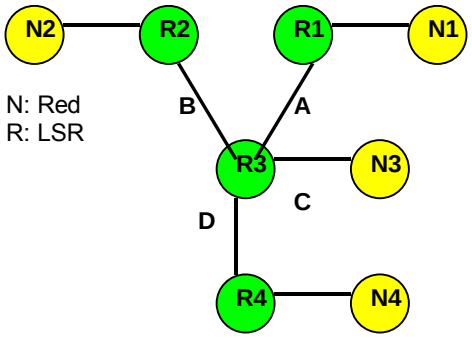


Figura 4.11.1.- Ejemplo de IP Unicast utilizando Método de Control Topology Driven.

Supongamos que R3 soporta agregación de etiquetas MPLS, con lo cual la tabla de *switching* de etiquetas en este *router* sería:

Next Hop	*Interfase de Entrada*	*Etiqueta de Entrada*	*Interfase de Salida*	*Etiqueta de Salida*
R2	A	E1	B	E2
R2	C	E3	B	E2
R2	D	E3	B	E2

Tabla 4.11.1.- Tabla de Switching de Etiquetas en R3.

Las características del *forwarding* MPLS para IP Unicast son:

1. Los *Updates* de los protocolos de ruteo "disparan" la creación o “destrucción” de *bindings* de etiquetas.

2. Los *bindings* de etiquetas son publicados utilizando un *Label Distribution Protocol*, LDP, dedicado. Esto sucede antes que se reciba algún dato sobre el port correspondiente, de esta manera todos los paquetes son *"forwardeados"* en capa 2.

Supongamos ahora que un grupo IP Multicast G tiene miembros en las redes N1 y N4, y que existe una fuente IP Multicast en la red N1. Si se utiliza PIM-DM, R3 recibirá los paquetes IP Multicast con destino al grupo G sobre la interfase A, y los *"forwardeara"* sobre las interfases B, C y D, creando la siguiente entrada en su tabla de ruteo:

(S1,G) iif={A} oif={B,C,D} prune={}.

Los paquetes seran *"forwardeados"* en capa 3, ya que no se ha asignado ninguna etiqueta al estado (S1,G). Luego, recibirá un mensaje *Prune* sobre la interfase C, con la cual la entrada en la tabla de ruteo se modificará de la siguiente forma:

(S1,G) iif={A} oif={B,D} prune={C}.

La interfase C será agregada nuevamente a la lista de interfases de salida luego que expire el *Timer* asociado al mensaje *Prune*. Cabe notar lo siguiente:

i. No existe entrada en la tabla de ruteo de R3 correspondiente al estado (S1,G) con anterioridad a recibir paquetes de datos desde la fuente IP Multicast S1.

ii. No existe la posibilidad de realizar agregación de entradas de ruteo IP Multicast al utilizar un protocolo *Dense Mode*.

iii. Una dada entrada en la tabla de ruteo cambia dinámicamente debido a las desafectaciones periódicas de ramas del árbol de distribución y o al arribo de nuevos miembros a los grupos IP Multicast.

iv. Todos los paquetes de datos IP Multicast son *"forwardeados"* en capa 3, hasta que se le asigne una etiqueta MPLS al estado (S1,G) en la tabla de ruteo.

Los ítems ***i*** y ***iii*** permiten concluir que la asignación de etiquetas para el caso que se utilicen protocolos IP Multicast *Dense Mode*, necesita ser del tipo *hop-by-hop* utilizando el método de control *Topology Driven*. Además, de ***ii*** se puede ver que cada entrada (S,G) necesita de la asignación de etiquetas MPLS de entrada y salida separadas. Así, cuando el primer paquete desde la fuente S, con destino al grupo G, es recibido por un *LSR*, el *forwarding* realizado por IP Multicast implementa el chequeo *RFP* y crea la entrada del estado (S,G) en la tabla de ruteo Multicast. Una vez que esta entrada es creada, se activa el procedimiento para asignar un *binding* de etiqueta al estado (S,G). Hasta tanto esta etiqueta sea asignada, el *forwarding* de los paquetes Multicast se realiza en capa 3.

El arribo de mensajes PIM *Grant* (S,G) requiere el agregar una rama de salida al *LSP* existente.

5. AGGREGATED MULTICAST TREE.

La distribución de tráfico IP Multicast presenta un inconveniente en cuanto a la escalabilidad producida por el número concurrente de grupos Multicast activos, debido a que esto implica que los elementos de red, *router*s, mantengan un estado actualizado de *forwarding* para cada árbol de distribución Multicast, donde la cantidad de entradas en la tabla de *forwarding* crece con el número de grupos activos.

IP Multicast utiliza una estructura en la cual los paquetes de datos son duplicados en los nodos "*fork"* y "*forwardeados*" sólo una vez sobre cada enlace. Así, posee una buena escalabilidad para soportar una gran cantidad de grupos IP Multicast. No obstante, esta estructura de árbol requiere que todos los nodos de la red mantengan información de *forwarding* por grupo Multicast activo, y aun por grupo-fuente activa, información que crece linealmente con el número de grupos Multicast activos. Esto implica la necesidad de contar con un incremento en la memoria de los nodos de red y un incremento en el *delay* del *forwarding* de paquetes.

Para el caso del tráfico IP Unicast, la escalabilidad se mejoró con la sumarización de direcciones IP más la distribución de direcciones IP en forma jerárquica. Este mecanismo no puede ser utilizado para el caso de tráfico IP Multicast, debido a que una dirección IP Multicast se corresponde con un grupo lógico y no transporta información sobre la localización de los miembros del grupo dentro de la red.

Existen varios trabajos de investigación en este campo cuyos objetivos son mejorar la escalabilidad del *forwarding* del tráfico IP Multicast. Algunos proyectos enfocan el problema de escalabilidad reduciendo los estados de *forwarding* utilizando la técnica de *Tunneling*, otros lo enfocan presentando un estado de agregación de *forwarding*. *Thaler* y *Handley*, analizaron la escalabilidad mediante la agregación de estados de *forwarding* utilizando un filtro de entrada-salida para el *forwarding* del tráfico IP Multicast. *Radoslavov*, propone la utilización de algoritmos para agregar los estados de *forwarding.* Ambos trabajos, presentan como objetivo agregar el estado de las tablas de ruteo IP Multicast luego que estas han sido asignadas a los grupos de distribución Multicast. Existen otras arquitecturas las cuales tienen como objetivo eliminar completamente el estado de *forwarding* IP Multicast en los nodos, *router*s, de la red, utilizando una técnica denominada *Network-Transparent Multicast*, la cual deja esta complejidad para los nodos extremos o terminales, *end-points*.

Aiguo Fei, *Jun-Hong Cui*, *Mario Gerla* y *Michalis Faloutsos*, proponen un esquema de forzar a múltiples grupos IP Multicast a utilizar un árbol de distribución compartido. Esta propuesta se basa en mejorar el problema de escalabilidad a través de un esquema que reduce los estados de *forwarding* Multicast forzando a los grupos Multicast compartir un único árbol de

disribución, el cual es llamado *Aggregated Tree*. De esta manera, se reduce el número de árboles de distribución en el *Core* de la red, así como también la cantidad de estados de *forwarding* Multicast. Una desventaja de este mecanismo puede presentarse en el consumo de ancho de banda adicional para el *forwarding* del tráfico IP Multicast hacia nodos de red no pertenecientes al grupo IP Multicast que utilizan el árbol de distribución *Aggregated Tree*.

A continuación se analizará y desarrollará la última de las propuestas, la cual luego será presentada en el capítulo 6, en conjunto con MPLS para brindar un esquema que permite el soporte de calidad de servicio sobre fujo de datos Multicast.

5.1 ESQUEMA DE DISTRIBUCIÓN IP MULTICAST.

Según el árbol de distribución de tráfico IP Multicast utilizado, se puede dividir el *forwarding* del tráfico Multicast en dos grupos, para el caso de utilizar protocolos Multicats Intra-Domain:

a. *Source Specific Tree*: DVMRP, PIM-DM y MOSPF.

b. *Shared Tree*: CBT, PIM-SM y BIDIR-PM.

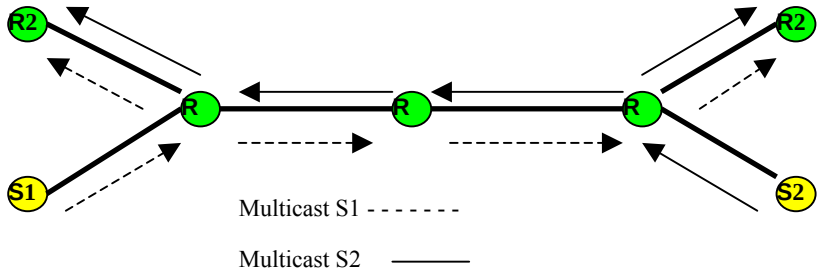


Figura 5.1.1.- Esquema Source Specific Tree.

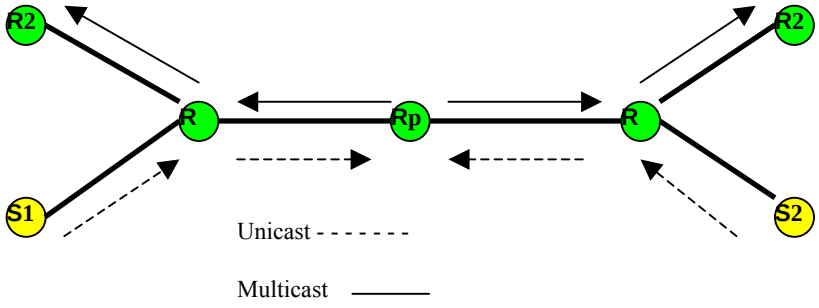


Figura 5.1.2.- Esquema Shared Tree Unidireccional.

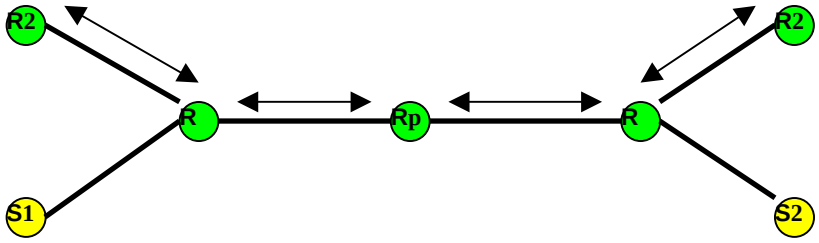


Figura 5.1.3.- Esquema Shared Tree Bi-direccional.

En el caso del esquema *Source Specific Tree*, IP Multicast construye un árbol de distribución dedicado para cada fuente Multicast. Así, cada fuente utiliza su propio árbol de distribución para entregar el tráfico a los receptores de un grupo Multicast.

En el caso del esquema *Shared Tree*, los árboles de distribución IP Multicast se construyen basados en los grupos Multicast; de esta manera, todas las fuentes de un grupo utilizan un mismo árbol de distribución para entregar el tráfico IP Multicast a los receptores del grupo. Es decir, múltiples fuentes de un mismo grupo Multicast comparten un único árbol de distribución. Este esquema permite tener árboles de distribución Unidireccionales, PIM-SM, o Bi-Direccionales, CBT y BIDIR-PIM.

En comparación con los dos esquemas anteriores, el esquema *Aggregated Tree* eleva el establecimiento y manejo del *Tree-Sharing* al nivel Inter-Grupo, donde los múltiples grupos IP Multicast son "forzados" a compartir un único árbol de distribución. Un *Aggregated Tree* puede ser del tipo *Source Specific* o *Shared Tree,* a la vez que un *Shared Tree* puede ser Unidireccional o Bi-Direccional. En este esquema, los *router*s de *Core* sólo necesitan mantener los estados por árbol *Aggegated Tree* en lugar de mantener los estados IP Multicast por Grupo, reduciendo así el número de árboles de distribución IP Multicast y el número de estados de *forwarding* IP Multicast.

Aggregated Multicast puede ser utilizado para diferentes modelos de servicio IP Multicast los cuales empleen una estructura de árbol.

5.2 AGGREGATED IP MULTICAST.

Para mostrar el funcionamiento de esta metodología veremos el siguiente ejemplo:

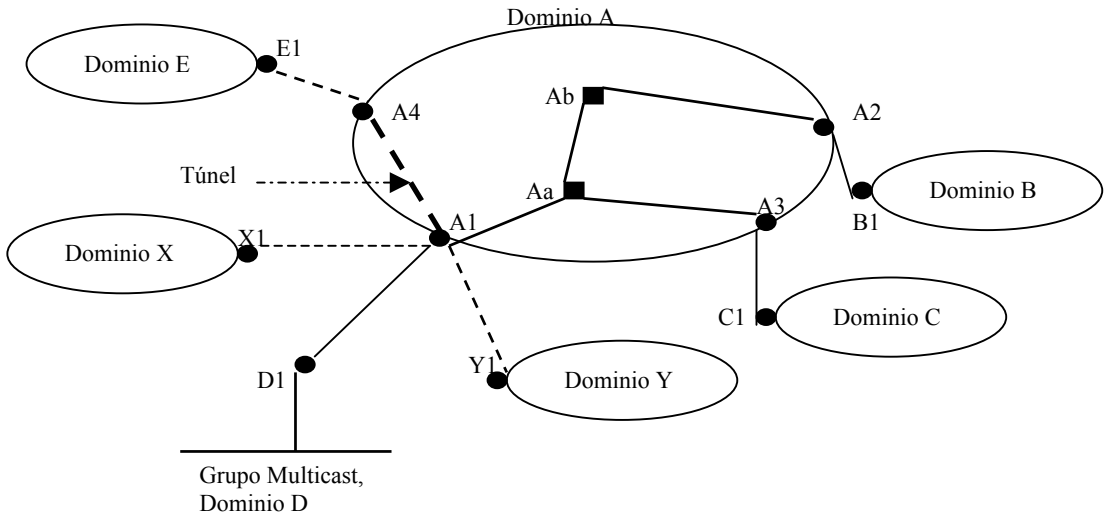


Figura 5.2.1.- Ejemplo Aggregated Multicast.

En la figura 5.2.1 se muestra un ejemplo de una red con jerarquía Inter-Dominio. En esta, el Dominio "A" representa un Dominio Regional o Nacional, perteneciente por ejemplo al *backbone* de un ISP. Los Dominios "D", "X2" e "Y" representan redes de clientes del Dominio "A"; y el Dominio "E" es la red de un cliente del Dominio "A" en otra localidad. Los Dominios "B" y "C" pueden ser otras redes de clientes o Dominios pertenecientes a *Backbones* de otros IPSs *peers* con el Dominio "A". El Dominio "A" "*forwardea*" sesiones originadas en el Dominio "D" y tiene los miembros del grupo en los Dominios "B" y "C". Los *router*s D1, A1, A2, A3, B1 y C1 forman el árbol de distribución IP Multicast a nivel de Inter-Dominio, mientras que A1, A2, A3, Aa y Ab forman un Sub-Árbol de distribución Inter-Dominio dentro del Dominio "A". Consideremos una segunda sesión IP Multicast que se origina en el Dominio "D" y también tiene sus miembros en los Dominios "B" y "C". De esta manera, un Sub-Árbol de distribución con exactamente el mismo conjunto de nodos será establecido para "*forwardear* " su tráfico dentro del Dominio "A". Ahora, si hubiese una tercera sesión IP Multicast, esta generaría un Dominio "X" y también tendría sus miembros en los Dominios "B" y "C", luego el *router* X1, en lugar de D1, será utilizado, pero el Sub-Árbol dentro del dominio "A" involucrará el mismo conjunto de nodos: A1, A2, A3,Aa y Ab.

En nuestro ejemplo, para un dado grupo IP, los *router*s A1, A2 y A3, los denominaremos Nodos Terminales, son aquellos donde el tráfico IP Multicast "entra" o "sale" de un Dominio. Los *router*s Aa y Ab los denominaremos Nodos de Transito, son aquellos *router*s internos dentro de un Dominio.

En el caso de utilizar IP Multicast tradicional en nuestro ejemplo, todos los nodos dentro del Dominio “A” deben mantener información de los estado de *forwarding* de IP Multicast en forma separada para cada uno de los grupos en forma individual, aun cuando sus árboles de distribución Multicast posean la misma forma. Sin embargo, utilizando la metodología *Aggregated Tree*, se establecerá un árbol de distribución pre-definido que utilizará los nodos A1, A2 y A3, utilizando un único grupo de direcciones IP Multicast dentro del Dominio “A”. De esta manera, en nuestro ejemplo, tres grupos utilizarán este único árbol de distribución IP Multicast. Como definición se puede decir que, un *Aggregated Tree* “cubrirá” a un determinado Grupo Multicast, si todos los nodos que conforman el grupo Multicast son nodos del árbol *Aggregated Tree*. Los datos de un grupo especifico serán encapsulados en el nodo terminal de entrada utilizando la dirección IP Multicast del *Aggregated Tree*. Estos serán luego distribuidos a través del *Aggregated Tree* y desencapsulados en el nodo terminal de salida para luego ser distribuidos a los miembros del grupo. De esta manera, los nodos de transito Aa y Ab sólo deberán mantener un única entrada de *forwarding* para el árbol *Aggregated Tree*, sin importar cuantos grupos estén utilizando este árbol de distribución.

De esta manera, la utilización de la metodología *Aggregated Tree* reduce el requerimiento de estados de *forwarding* IP Multicast a ser mantenidos por los *router*s. Los nodos de transito no necesitan mantener estados de *forwarding* para cada grupo individual; en cambio, estos sólo deben mantener los estados de *forwarding* de un reducido número *de Aggregated Trees*. También se reduce la carga de procesamiento vinculada al establecimiento de árboles de distribución Multicast.

Para la implementación de la metodología *Aggregated Tree* se presentan dos requisitos básicos:

a. El grupo de direciones IP Multicast debe ser preservado de alguna manera y pueden ser recobradas en los nodos extremos de salida para determinar como y donde “*forwardear* “ los futuros paquetes de datos Multicast.

b. Utilizar alguna clase de identificación para el *Aggregated Tree*, la cual debe ser transportada a través de la red. Los nodos de transito deberán realizar el *forwarding* de paquetes basados en esta información.

Para lograr los requerimientos señalados en los puntos ***a*** y ***b*** se puede utilizar encapsulación a nivel de capa 3, nivel IP. Esta implementación agregará complejidad y mayor procesamiento en los nodos terminales del *Aggregated Tree*. Otra posibilidad es la utilizar MPLS, mediante el cual se pueden identificar los *Aggregated Tree* mediante la utilización de etiquetas MPLS.

Para el manejo de *Aggregated Trees* y la relación entre los grupos IP Multicast y los *Agregated Trees*, se define la utilización de una entidad central denominada *Tree Manager*. Este último tiene la capacidad y la información para el establecimiento de *Aggegated Trees* en la red. Este recolecta los mensajes *Group Join Inter-Domain* recibidos de de los *router*s de borde, y asigna los grupos Multicast a los *Aggregated Trees* correspondientes. Una vez que determina que *Aggregated Tree* utilizar para cada grupo, el *Tree Manager* puede asignar los estados Multicast correspondientes a los nodos terminales involucrados, o distribuir etiquetas MPLS. Para la construcción de *Aggregated Trees* dentro de un dominio se puede utilizar un protocolo Multicast existente, como PIM-SM, o utilizar señalización IP MPLS, lo cual permite el establecimiento de árboles pre-calculados.

El conjunto de árboles *Aggregated Trees* puede ser establecido basado en la determinación del patrón de tráfico, realizado a través de mediciones. También pueden ser establecidos, cambiados o removidos en forma dinámica basados en el monitoreo del tráfico Multicast.

Otra característica para destacar de este mecanismo es el de facilitar el manejo de QoS para tráfico IP Multicast. Esto es debido a que puede realizar una pre-asignación de un *Aggregated Tree* basada en el requerimiento de ancho de banda o recursos, ya sea en forma estática o dinámica. Esto aplica bien al tráfico IP Multicast debido a su naturaleza intrinseca por-flujo.

Entre las ventajas de esta metodología podemos citar:

a. Los nodos, o *router*s, de transito no necesitan mantener información del estado de *forwarding* multicast para cada grupo individual.

b. Los nodos de *Core* sólo mantienen información de estados Multicast para los *Aggregated Trees* establecidos, lo cual mejora la escalabilidad de IP Multicast con respecto al número de sesiones establecidas.

c. Reduce el *overhead* de control.

Entre las desventajas de esta metodología podemos citar:

a. El depender de un control centralizado, *Centralized Tree Manager*, puede conducir a que el mismo se encuentre sobrecargado y a tener un único punto de falla en la red.

b. *Membership Dynamics* , el problema puede suceder en el caso que se agregue un nuevo nodo, o *router*, y este no sea parte de un *Aggregated Tree* ya establecido, o cuando un nodo deje de formar parte de un *Aggregated Tree* establecido, lo cual puede causar una sobrecarga del ancho de banda si los nodos que siguen perteneciendo al *Aggregated Tree* no tienen capacidad para manejar el ancho de banda requerido. Sin embargo, esta desventaja es menos probable en los nodos de *Core*, donde se establecen los *Aggregated Tree*.

c. *Aggregated Tree Matching*, esto puede ocurrir cuando no existe un *Matching* perfecto o cuando no existe un *Aggregated Tree* para "cubrir" un grupo. Un *Aggregated Tree Matching* se considera que es Perfecto, para un grupo, si todos sus nodos son nodos terminales para el grupo IP Multicast, con lo cual el tráfico no se enviará a los nodos que no lo necesiten recibir.

5.2.1. GROUP-TREE MATCHING.

Cuando se establece o activa un grupo IP Multicast, un *Aggregated Tree* se debería asignar al grupo siguiendo algunas reglas. En caso que este establecido un grupo *de Aggregated Trees*, sólo abrá que seleccionar el árbol de agregación con el mínimo costo el cual pueda "cubrir" los miembros del grupo. En caso de asignación dinámica, los *Aggregated Trees* se establecen por demanda, para lo cual es necesario un método de *Matching* más elaborado y complicado.

En caso de requerirse el *Matching* de un grupo IP Multicast a un *Aggregated Tree*, T, podemos tener cuatro casos fundamentales:

a. El *Aggregated Tree* T "cubre" los miembros del grupo G y los nodos de salida del árbol son nodos terminales del grupo G, esto es denominado *Perfect Match*.

b. El *Aggregated Tree* T "cubre" los miembros del grupo G, pero algunos de los nodos de salida no son nodos terminales para el grupo G, esto es denominado *Pure-Leaky Match*.

c. El *Aggregated Tree* T no "cubre" los miembros del grupo G y todos los nodos de salida del árbol son nodos terminales del grupo G, esto es denominado *Pure-Incomplete Match*.

d. El *Aggregated Tree* T no "cubre" los miembros del grupo G y alguno de los nodos de salida del árbol no son nodos terminales del grupo G, esto es denominado *Incomplete Leaky Match*.

Por ejemplo, en la figura 5.2.1, el *Aggregated Tree* To, con nodos A1, A2, A3, Aa y Ab, es un *Perfect Match* para el grupo IP Multicast Go, cuyos miembros son D1, B1 y D1. Sin embargo, si el *Aggregated Tree*, To, se utilizará también para el grupo G1, con miembros D1 y B1, este sería un *Pure-Leaky Match*, ya que el tráfico hacia G1 será entregado también al nodo A3, y será descartado ya que A3 no posee estados de *forwarding* IP Multicast para este grupo. En el caso del grupo G2, con miembros D1, B1, C1 y E1, el *Aggregated Tree* To será *Pure Incomplete Match*, y para el caso del grupo G3, con miembros D1, B1 y E1, el *Aggregated Tree* To, será *Incomplete Leaky Match*.

Una desventaja del *Leaky Match*, es que utiliza parte del ancho de banda para entregar, o "*forwardear*" datos a nodos que no son miembros del grupo Multicast. No obstante, esto puede ser inevitable, ya que generalmente no es posible establecer *Aggregated Trees* para todas las combinaciones de grupos posibles. Sin embargo*, Leaky Match* mejora la función de compartir los árboles de distribución en los casos *Inter-Group*.

En el caso de *Incomplete Match*, se tienen dos maneras de establecer un árbol para un grupo Multicast. Una forma es construir un *Aggregated Tree* más "grande" moviendo el grupo entero hacia este nuevo árbol de distribución. Otra forma, es extender el *Aggregated Tree* siendo utilizado por el grupo Multicast. Esta última opción implica un mayor *overhead*, ya que todos los grupos, los cuales forman parte del *Aggregated Tree*, deberán realizar los ajustes correspondientes. Una alternativa es utilizar *Tunneling*. En el ejemplo de la figura 5.2.1, supongamos que E1, del dominio "E", se une al grupo Go. Con lo cual, en lugar de extender la dimensión del árbol To, se establece un túnel entre el *router* de borde A4 y el *router* A1. Esta solución combina Multicast Inter-Dominio *Aggregated Tree* y *Tunneling*, preservando escalabilidad en los *router*s de *Core*. Se puede ver también que en este caso se tendrá un *Incomplete Leaky Match*.

5.2.2 ARQUITECTURA ASSM, AGGREGATED SOURCE SPECIFIC MULTICAST.

En ASSM, *Aggregated Source Specific Multicast*, la arquitectura típica es aquella en donde el tráfico IP Multicast es distribuido entre una fuente Multicast y varios *subscribers* en diferentes redes de ISP, *Internet Service Providers*, las cuales están conectadas a través de redes WAN, o TDs, *Transit Domains*. En ASSM, *Aggregated Source Specific Multicast*, se implementará un dominio de transito, de modo de reducir el número de estados de *forwarding* Multicast, este es denominado *MAD*, *Multicast Aggregation Domain*. *Aggregated Multicast* es implementado dentro de un *MAD* a través de canales Multicast agregados en los *router*s de borde de entrada, y desagregados en los *router*s de borde de salida. De esta manera, el protocolo de ruteo Multicast utilizado por un ISP será independiente del utilizado en el *MAD*, en el cual se utilizará PIM-SSM.

Dentro de un *MAD* se agregan múltiples sesiones IP Multicast dentro de un *Aggregated Tree*. Las direcciones IP Multicast utilizadas dentro de un *MAD* son transparentes e independientes al resto de Internet. Se utiliza un esquema de traslación de direcciones IP. Este esquema no puede cambiar la dirección de canal Multicast dentro del *Header* del paquete, debido a que el *router* de borde de salida no podría distinguir entre los múltiples canales o sesiones Multicast, y así distribuir los paquetes a sus destinos. Para solucionar este problema se puede utilizar encapsulación a nivel IP, lo cual introducirá algún *overhead* debido al procesamiento adicional y consumo de ancho de banda. Otra posible solución es la utilización de MPLS, lo cual reduciría el tiempo de procesamiento y el consumo de ancho de banda.

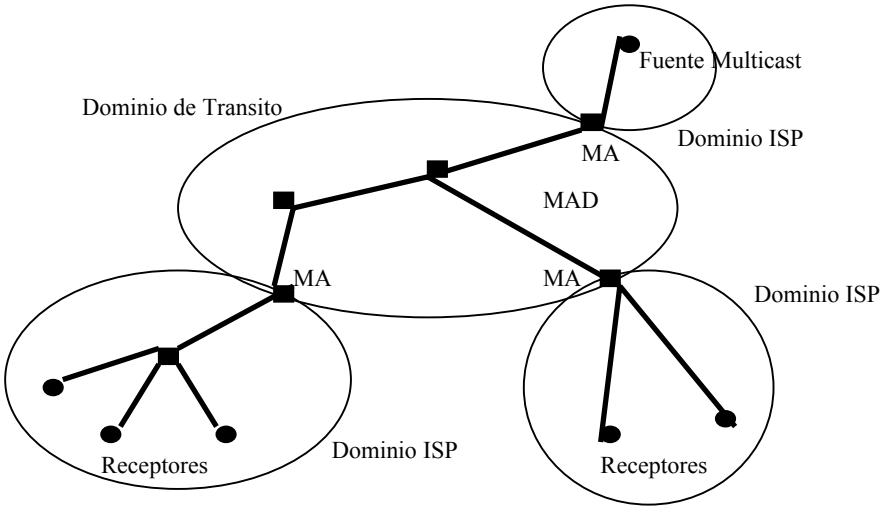


Figura 5.2.2.1.- Arquitectura ASSM.

En los *router*s de borde *MA*, *Multcast Aggregation Routers*, se realiza la encapsulación/desencapsulación de los paquetes IP Multicast. Los *MA* con tráfico entrante al Dominio de Transito se denominan *Source MA*, y *Destination MA* a aquellos con tráfico saliente.

Para las redes de los ISPs, los *router*s *MA* se comportan como un *router* Multicast convencional. Dentro del Dominio *MAD*, los *router*s *MA* cumplen las funciones de Aggregación Multicast. Es decir, cuando un *router Source MA* recibe un paquete Multicast, este realizará la traslación de dirección IP Multicast, encapsulando la misma y enviando el paquete por un dado árbol de distribución *Aggregated Tree*; luego, en el *Destination MA* se realizará el proceso de desencapsulación, y se distribuirá el paquete Multicast en la red del ISP o Grupo Multicast.

5.2.3 PROTOCOLO ASSM, AGGREGATED SOURCE SPECIFIC MULTICAST.

El prtocolo *ASSM*, *Aggregated Source Specific Multicast*, se basa en mensajes al nivel de capa de transporte, entre los *router*s *MA*, el cual es transparente a los *router*s en el *Core* del

dominio *MAD*. Los *router*s de *Core* en el *MAD* sólo deben procesar los mensajes de ruteo del protocolo PIM-SSM y transmitir los datos.

Los *router*s *MA* tienen la función de realizar la agregación de los componentes Multicast, "mapeo" de grupos y árboles de distribución, cambios en los miembros de un grupo IP Multicast, etc.

Para poder distinguir entre los diferentes tipos de mensajes, se agrega un prefijo adicional en los mensajes del protocolo de ruteo. En el caso del protocolo PIM-SSM, los mensajes de subscripción y des-subscripción se denominan *M-Join* y *M-Leave*. En ASSM, se definen los mensajes *A-Join*, *A-ACK*, *A-Leave* y *A-Move*.

5.2.3.1. SOURCE MA ROUTERS.

Dentro del Dominio *MAD* se define el protocolo PIM-SSM, con lo cual cada *Aggregated Tree* tendrá sólo un *router Source MA* y múltiples *router*s *Destination MA*. El *router Source MA* deberá mantener información sobre: *Group Membership*, nodos de los árboles *Aggregated Trees* e información del "mapeo" entre árboles de distribución y grupos Multicast. Para determinar que árbol de distribución utilizar para cada grupo se utiliza un algoritmo denominado *Group-Tree Matching.*

5.2.3.2. SOURCE ADVERTISEMENT.

Una fuente IP Multicast se anuncia, o "declara", en Internet enviando *mensajes Global Multicast Directory Service* y *SDR Advertising*, conteniendo la dirección de la fuente y del grupo destino de los paquetes IP Multicast, (S,G). Cualquier miembro que quiera recibir el tráfico Multicast deberá enviar mensajes *Querry* al *Directory Service* de modo de obtener la dirección IP Multicast de la fuente.

5.2.3.3. PROCESO DE SUSCRIPSIÓN DE UN HOST LOCAL A UN CANAL MULTICAST.

En la siguiente figura se muestra el proceso de Subscripción.

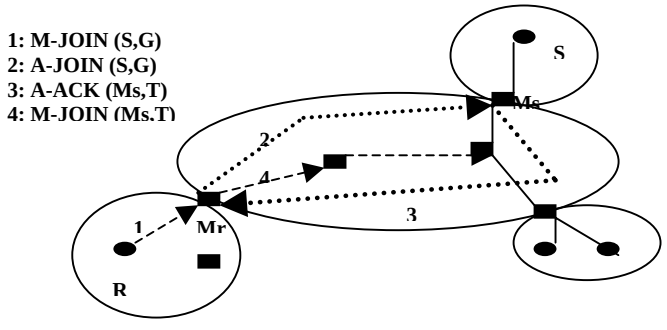


Figura 5.2.3.3.1.- Esquematización del Proceso de Subscripción del Receptor R al grupo (S,G).

En la figura 5.2.3.3.1, cuando un receptor, R, quiere unirse a un grupo Multicast (S,G), envia un mensaje *SSM M-Join* hacia la fuente S. Si el grupo (S,G) está distribuido dentro del dominio del ISP, entonces el protocolo de ruteo Multicast interceptará el pedido de Subscripción y agregará al receptor dentro de un árbol de distribución IP Multicast. De lo contrario, el mensaje llegará al *router Mr*, el cual verificará si el grupo (S,G) posee algún *Aggregated Tree* asignado. Para esto, el *Mr* enviará un mensaje *A-Join (S,G)*, a nivel de la capa de transporte, hacia la fuente S. Este mensaje será tomado por el *router Ms*, el cual chequeará si tiene un estado de *forwarding* Multicast para el grupo (S,G). Si lo tiene, verificará el *Aggregated Tree* asociado al grupo (S,G), (Ms,To). Luego ejecutará el algoritmo *Group-Tree Matching*, y *Ms* obtendrá, en este ejemplo, que existe un *Aggregated Tree* que se ajusta mejor para este caso, (Ms,T). Con lo cual, el resto de los receptores, o *subscribers*, del grupo (S,G), se migrarán a este nuevo *Aggregated Tree*. Luego, el *Ms* responderá enviando un mensaje *A-Ack (Ms,T)* hacia *Mr*. Si *Mr* ya se encuentra subscripto al *Aggregated Tree* (Ms,T) no se realizarán más operaciones. De lo contrario, *Mr* se unirá al *Aggregated Tree* enviando un mensaje *M-Join (Ms,T)*.

En el caso que *Ms* no tenga el estado de *forwarding* para (S,G) asignará una dirección IP Multicast para el *Aggregated Tree*, (Ms,T), para el grupo (S,G), luego responderá a *Mr* con un mensaje *A-Ack (Ms,T)*. Al recibir el *A-Ack (Ms,T)*, *Mr* establecerá el *Aggregated Tree* (Ms,T) enviando un mensaje *M-Join (Ms,T)*. Dentro del Dominio del ISP, *Ms* deberá subscribir el grupo (S,G) utilizando un mensaje *SSM M-Join (S,G)* hacia la fuente S.

5.2.3.4. DES-SUBSCRIPCIÓN DE HOST LOCALES.

En la siguiente figura se muestra el proceso de Des-Subscripción.

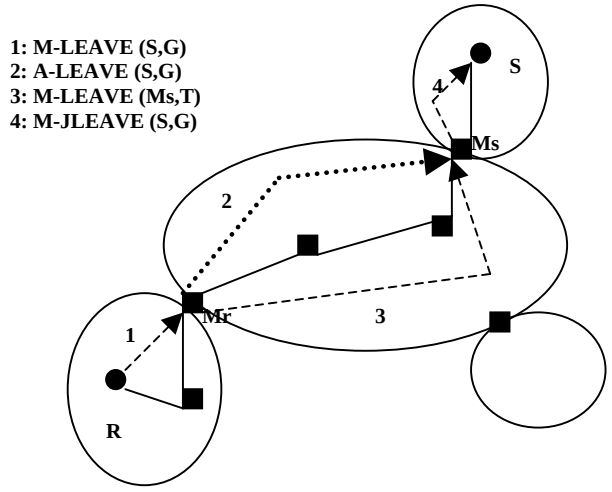


Figura 5.2.3.4.1.- Esquematización del Proceso de Des-Subscripción del Receptor R al grupo (S,G).

Cuando un *router Mr* recibe un mensaje de Des-Subscripción *M-Leave (S,G)* desde un receptor R, este chequeará si quedan miembros en el grupo (S,G). Si no quedan más miembros en este grupo, *Mr* enviará un mensaje *A-Leave (S,G)* hacia *Ms* y borrará de su base de información el estado Multicast asociado al grupo (S,G) y el "mapeo" entre (S,G) y el *Aggregated Tree* (Ms,T). *Mr* enviará un mensaje de Des-Subscripción *M-Leave (Ms,T)*.

El mensaje *A-Leave (S,G)* causará un cambio de árbol de distribución.

Los *router*s *Source MA* realizan el seguimiento de los miembros de los grupos. Cuando detectan que el número de miembros de un grupo es 0, estos Des-Subscribirán el grupo (S,G) del dominio del ISP.

5.2.3.5. CAMBIO DE AGGREGATED TREES.

La operación de cambio o intercambio de *Aggregated Tree* es realizada por un *Source MA* con el objeto de mejorar el grado de agregación. La misma puede ser generada por mensajes *Join* y *Leaving* originados por un *host* local.

Para minimizar el envio de mensajes adicionales, en ASSM, el cambio o intercambio de árbol de *Aggregated Tree* se realiza simplemente realizando la des-subscripción y subscripción de un miembro de un árbol de distribución a otro. Teniendo en cuenta una posible pérdida de información durante el procedimiento de cambio o intercambio de árbol de distribución, primero se realiza la subscripción al nuevo *Aggregated Tree* y luego la des-subscripción *del Aggregated Tree* actual.

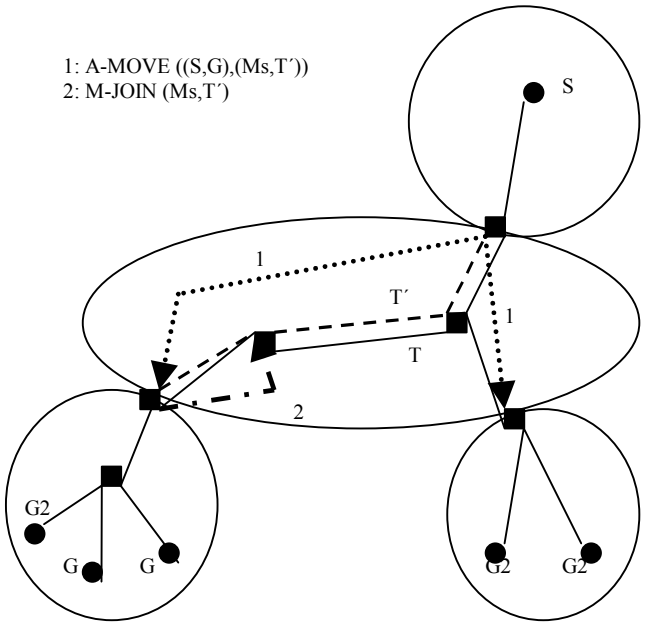


Figura 5.2.3.5.1.- Ejemplo de Operación de Cambio o Intercambio de Arbol de Distribución.

Para realizar el cambio de un grupo IP Multicast (S,G) de un *Aggregated Tree* (Ms,T) a otro *Aggregated Tree* (Ms,T´), se define un mensaje *A-Move ((S,G), (Ms, T´))*. De esta forma, el *Router Ms* enviará este mensaje a través del *Aggregated Tree* (Ms,T) hacia todos los receptores notificando del cambio.

En la figura 5.2.3.5.1 se muestra el procedimiento de un cambio de *Aggregated Tree*. En este se ve que cada *router Destination MA* que recibe un mensaje *A-Move ((S,G),(Ms,T´))* puede requerir:

a. Subscribirse al árbol (Ms,T´), si no se encuentra en el árbol de distribución *Aggregated Tree*.

b. Des-subscribirse del árbol (Ms,T), si no existen receptores en ninguno de los grupos IP Multicast.

Además, el *router Ms* deberá actualizar su mapeo de interno de Grupos-Árboles de distribución. De esta manera, el grupo (S,G) será agregado al árbol de distribución (Ms,T´).

5.2.3.6 PROCEDIMIENTO DE MATCHING GRUPO IP MULTICAST- ÁRBOL DE DISTRIBUCIÓN.

En ASSM, los grupos IP Multicast son agregados a un árbol de distribución *Aggregated Tree* basados en la ejecución de un algoritmo de *Matching* entre Grupo IP Multivast y Árbol de Distribución, también se la conoce como función de *Overhead*, la cual se aplica sobre cada *Aggregated Tree*. Este algoritmo será ejecutado a partir de la recepción de mensajes *A-Join* y *A-Leave*.

Para implementar la metodología de *Aggregated Tree* en una red IP Multicast, se presentan dos grandes temas a tener en cuenta:

a. Cuales son los *Agrregated Trees* que deben ser establecidos.

b. Que árbol de distribución *Aggregated Tree*, ya establecido, debe ser utilizado para un dado grupo IP Multicast.

Dada una red con *n* nodos de borde, que pueden ser nodos terminales para un grupo IP Multicast, la cantidad de combinaciones de grupos Multicast se puede considerar como 2^n, con lo cual se ve que la misma crece exponencialmente con la cantidad de nodos de borde.

Para una red razonablemente grande, no resultará practico plantear un esquema de pre-establecer *Aggregated Trees* para todos los grupos Multicast posibles, ya que la cantidad de árboles de distribución *Aggregated Trees* pre-establecidos puede ser mayor que la cantidad de grupos IP Multicast activos. Por esto, se debería pre-establecer sólo un sub-conjunto de árboles de distribución *Aggregated Tree*, y luego establecer árboles de distribución dinámicamente.

5.2.3.6.1 FUNCIÓN OVERHEAD.

Definimos una red como un Gráfico Indirecto G (V,E). A cada *router* de borde de la red, denotado por el par (i,j) se le asignará un costo:

$$c_{ij} = c_{ji}$$

los cuales representan el costo de transportar una unidad de datos desde el nodo *i* al nodo *j*.

Dado un árbol de distribución *T*, el costo total para distribuir una unidad de datos por el árbol de distribución *T* será:

$$C(T) = \sum_{(i,j)\in T} c_{ij}, \text{link } (i,j) \in T$$

Suponiendo que cada enlace tiene un costo de 1, el costo del árbol de distribución será:

$$C(T) = |T| - 1$$

donde: | T| representa el número de nodos del árbol *T*.

Dado un grupo Multicast G y un árbol de distribución *T*, se define que el árbol de distribusión *T* “cubre” al grupo Multicast G, si todos los miembros de G son nodos del árbol de distribución *T*. Si un grupo G es “cubierto” por un árbol de distribución *T*, todos los paquetes de datos distribuidos a través del árbol alcanzarán a todos los miembros del grupo G, suponiendo un árbol de distribución bi-direccional.

Consideremos una red en la cual se utiliza un algoritmo de ruteo *A* para establecer árboles de distribución IP Multicast. Dado un grupo Multicast G, definimos al árbol IP Multicast para dicho grupo como:

$$T_A(G)$$

Ahora, si consideramos que el grupo Multicast G es "cubierto" por un *Aggregated Tree*, definimos el *Aggregation Overhead* como:

$$\Delta(T,G) = C(T(G)) - C(T_A(G))$$

La ecuación anterior refleja el ancho de banda utilizado si se utiliza un árbol de distribución *Aggregated Tree*, T(G), para transportar los datos al grupo Multicast G, en lugar de utilizar un árbol de distribución IP Multicast convencional $T_A(G)$.

Si definimos la cantidad de datos transmitidos como D_G, el ancho de banda utilizado puede cuantificarse como $D_G.\Delta(T,G)$. Cabe destacar que $T_A(G)$ puede no ser necesariamente el árbol de distribución de mínimo costo, y que el árbol de distribución *Aggregated Tree* T(G) puede ser más eficiente que $T_A(G)$, es posible que $\Delta(T,G)$ resulte negativa.

A continuación definiremos la función *Overhead* para los sigiuentes casos:

a. Árboles de Distrbución Pre-Establecidos en forma estática.

b. Árboles de Distribución establecidos en forma dinámica.

Árboles de Distrbución Pre-Establecidos en forma Estática.

Consideremos:

a. Una red modelizada por un gráfico indirecto G(V,E).

b. Un árbol de distribución con costo C(T).

c. Un conjunto N de grupos Multicast.

El objetivo es considerar un conjunto *n* de árboles de distribución, cada uno de ellos cubriendo un conjunto diferente de nodos, y un *Matching* de un grupo G con un árbol *Aggregated Tree* T(G), tal que cada grupo G es "cubierto" por un *Aggregated Tree* T(G), con la finalidad de lograr minimizar el *overhead* de la agregación total. Así se pre-establecerán *Aggregated Trees* en forma estática basados en mediciones de tráfico.

En casos reales, diferentes grupos pueden requerir diferentes anchos de banda y tener tiempos de vida diferentes; además, pueden transmitir cantidades de datos diferentes.

Para un grupo Multicast G, el cual transmite una cantidad de datos D_G, el *overhead* de Agregación Multicast sería:

$$D_G . \Delta(T, G)$$

Sin embargo, si D_G es independiente del tamaño del grupo Multicast *g* y de la cantidad de miembros del mismo, estadísticamente el efecto final sería el mismo, si se considera a todos los grupos como si todos entregarían la misma cantidad de datos. Por lo cual, en estas condiciones, el *overhead* de agregación sería:

$$\sum_G \Delta(T, G)$$

Luego, podemos definir el *Overhead* Promedio como:

$$\delta_A = \frac{\sum_G \Delta(t, G)}{\sum_G C(T_A(G))} = \frac{\sum_G C(T(G))}{\sum_G C(T_A(G))} - 1$$

Árboles de Distribución Establecidos en forma Dinámica.

En este caso, en lugar de considerar un conjunto de grupos Multicast estáticos, se considera que ellos se activan y desactivan en forma dinámica. Por lo cual, en este caso, el objetivo es generar y mantener un procedimiento para el establecimiento de *Aggregated Trees* en forma dinámica, junto con la función de *Matching* entre árboles de distribución y grupos Multicast cada vez que se active y desactive un grupo, minimizando el *Overhead* Promedio de Agregación.

Si se establece un limite superior en cuanto a la cantidad de *Aggregated Trees* que pueden estar activos en forma simultánea, el procedimiento utilizado para realizar el *Matching* en *Aggregated Trees* establecidos en forma dinámica, puede ser utilizado también para el *Matching* de *Aggregated Trees* establecidos en forma estática. De esta forma, el conjunto de *Aggregated Trees* para los grupos estáticos se establecerá uno por uno y se utilizará el procedimiento de *Matching* dinámico para el establecimiento de *Aggregated Trees* para los grupos que se generan en forma dinámica.

En caso de grupos establecidos en forma estática, la cantidad de grupos Multicast es finita, la cual consideraremos igual a N. Para el caso de grupos establecidos en forma dinámica, se considerará que se tiene una cantidad N que representa el promedio de

grupos activos en forma simultánea. Si *n* es la cantidad máxima de *Aggregated Trees* activos en forma simultánea,

a. Cuando n $\rightarrow$ N, se tendrá un mínimo en el *Overhead* Promedio de Agregación; es decir, $\delta_A \rightarrow 0$.

b. Si n = N, el *Overhead* de Agregación Promedio será cero; es decir, $\delta_A = 0$.

5.2.3.6.2 DYNAMIC TREE SHARING WITH AGGREGATION OVERHEAD THRESHOLD CONTROL.

A continuación se presentará una solución para el caso del establecimiento de *Aggregated Trees* en forma dinámica, y el caso de compartir un mismo árbol de distribución para más un grupo Multicast. Para esto, se definirá una función que representará el *Overhead* de Agregación promedio controlado estadísticamente bajo un dado umbral de *Overhead*, o *Treshold*.

Se definen:

a. MTS, *Multicast Tree Se*t, como el conjunto de grupos de árboles de distribución establecidos actualmente en la red.

b. G(T), el conjunto de grupos activos actualmente en la red, "cubiertos" por el *Aggregated Tree* T, siendo $T \in MTS$.

Ambas variables, G(T) y MTS, son función del tiempo, aun cuando la variable tiempo no este explícitamente indicada en la ecuaciones.

Luego, el *Overhead* de Agregación Promedio se puede definir como:

$$\overset{\approx}{\delta}(T) = \frac{\sum_{GenG(T)} \Delta(T,G)}{\sum_{GenG(T)} C(T_A(G))} = \frac{|G(T)|.C(T)}{\sum_{GenG(T)} C(T_A(G))} - 1$$

la ecuación anterior cambia, o se actualiza, con cada cambio en el tiempo de G(T).

$|G(T)|$ representa la cantidad de grupos "cubiertos" por T. Por lo cual, en un instante t, el *Overhead* de Agregación Promedio para todos los grupos $\delta_A(t)$, en el instante t, puede ser calculado a partir de la ecuación $\tilde{\delta}(T,t)$. Luego, si el *Aggregated Tree* T se utiliza para "cubrir" el grupo G, el *Overhead* de Agregación Promedio se define como:

$$\delta(T,G) = \frac{C(T) - C(T_A(G))}{C(T_A(G))}$$

Sea $\tilde{\delta}'(t,G)$ el *Overhead* de Agregación Promedio si el grupo Multicast G es "cubierto" por el *Aggregated Tree T*, entonces se tiene:

$$\tilde{\delta}'(T,G) = \frac{(|G(T)|+1).C(T)}{\sum_{G_\alpha en\{G(T),G\}} C(T_A(G_\alpha))} - 1 = \frac{(|G(T)+1|).C(T)}{\frac{|G(T).C(T)|}{1+\tilde{\delta}(T)} + C(T_A(G))} - 1$$

Cuando una nueva sesión IP Multicast continúa con un grupo Multicast G, tenemos básicamente tres opciones para este nuevo grupo:

a. Un *Aggregated Tree* "cubre" el grupo G, y es utilizado para distribuir los paquetes de esta nueva sesión Multicast.

b. Un *Aggregated Tree* existente puede extenderse para cubrir este nuevo grupo IP Multicast.

c. Se establece un nuevo *Aggregated Tree* para este nuevo grupo.

Consideremos a b_t como el *Overhead* Umbral de Ancho de Banda. El objetivo es controlar, estadísticamente, el porcentaje total de *Overhead* para que resulte:

$\tilde{\delta}(T) < b_t$ para cada $T \in MTS$

o que se tenga: $\delta_A < b_t$

A continuación se mostrará el procedimiento para utilizar las dos suposiciones mencionadas anteriormente:

a. Calcular el árbol de distribución utilizando un algoritmo de IP Multicast Nativo, $T_A(G)$

b. Para cada árbol de distribución *T* en MTS, si *T* "cubre" al grupo G, se calcula $\tilde{\tilde{\delta}}'(T,G)$; de no ser así, se calcula un árbol de distribución extendido T^e que "cubra" el grupo Multicast G y luego se calcula $\tilde{\tilde{\delta}}'(T^e,G)$, si resulta $\tilde{\tilde{\delta}}'(T,G) < b_t$ o $\tilde{\tilde{\delta}}'(T^e,G) < b_t$, entonces T o T^e será considerado candidato para cubrir al grupo Multicast G.

c. Entre todos los candidatos, se tomará uno tal que cumpla que C(T) o $C(T^e) + |C(T)|.(C(T^e) - C(T))$ resulte un mínimo, el cual lo denominamos T_m; luego T_m será utilizado para "cubrir" el grupo Multicast G, finalmente se actualizará MTS quedando $G(T_m)$ y $\tilde{\tilde{\delta}}(T_m)$.

d. En caso de no encontrarse ningún árbol de distribución "candidato" en el paso b, $T_A(G)$ será utilizado para "cubrir" al grupo Multicast G y se agregará a MTS, resultado $G(T_A(G))$ y $\tilde{\tilde{\delta}}(T_A(G))$.

Como cada grupo IP Multicast tiene un tiempo de vida limitado, esto presenta un procedimiento simple que se puede aplicar para los casos donde se presente la baja de grupos Multicast. Cuando en un grupo G no queden más receptores, este grupo será quitado de G(T), donde T es el árbol de distribución utilizado para "cubrir" a G. Luego, si G(T) no tiene miembros, el árbol de distribución *T* será quitado de MTS.

5.2.3.6.3 MÉTRICAS DE PERFORMANCE.

Se utilizan para cuantificar la efectividad de un método de agregación.

Sea N(t), la cantidad de grupos Multicast activos en la red en el instante t, y sea M(t) la cantidad de árboles de distribución Multicast; luego, el Grado de Agregación se define como:

$$AD(t) = \frac{N(t)}{M(t)}$$

El *Overhead* de Agregación Promedio se define como:

$$\delta_A(t) = \frac{\sum_G C(T(G))}{\sum_G C(T_A(G))} - 1 = \frac{\sum_{T \in MTS} |G(T)|.C(T)}{\frac{|G(T)|.C(T)}{1 + \tilde{\delta}(T)}} - 1$$

donde: δ_A representa el ancho de banda extra utilizado para transportar tráfico IP Multicast utilizando *Shared Aggregated Trees*.

En general, se puede asumir que un *router* necesita una entrada de ruteo por cada dirección IP Multicast en su tabla de *forwarding*. En el caso del ruteo en Multicast IP convencional, el número total de entradas en la tabla de *forwarding* es igual a la cantidad de nodos |*T*| en el árbol de distribución Multcast, o sub-árbol de distribución dentro de un dominio Multicast. Cuando se utiliza *Aggregated IP Multicast*, existen dos tipos de estados en la tabla de *forwarding* Multicast: entradas para los *Shared Aggregated Trees* y entradas *Group-Specific* en los nodos terminales. El número de entradas para un *Aggregated Tree T* será igual a la cantidad de nodos del árbol de distribución |*T*|, y las mismas pueden ser compartidas por todos los grupos Multicast que utilizan el árbol de distribución *T*. El número de entradas *Group-Specific* para un grupo Multicast será igual a la cantidad de nodos terminales, ya que sólo estos nodos necesitan estados *Group-Specific*. Además, en *Aggregated Multicast* se introducen los conceptos de:

- Estados Irreducibles.
- Estados Reducibles.

Todos los nodos terminales requieren de la información de estado necesaria para determinar como "*forwardear*" los paquetes Multicast recibidos.

Dado un conjunto de grupos Multicast ϑ, si cada grupo G es "cubierto" por un árbol de distribución $T_A(G)$; luego, la cantidad de entradas en la tabla de estados será:

$$N_A = \sum_{G \in \vartheta} |T_A(G)|$$

Por lo cual, si el mismo conjunto de grupos Multicast es "cubierto" utilizando un conjunto de *Aggregated Tree* MTS, el número total de entradas en la tabla de estados será:

$$N_T = \sum_{T \in MTS} |T| + \sum_{G \in \vartheta} |G|$$

donde |*T*| representa la cantidad de nodos en el árbol de distribución *T*, y |G| representa el tamaño del grupo G.

El primer termino de la ecuación anterior da la cantidad de entradas que deben ser mantenidas por los *Aggregated Trees*, mientras el segundo termino denota la cantidad de entradas que los nodos terminales de entrada y salida de un grupo Multicast deben mantener de modo de tener la información necesaria para realizar el *forwarding* de paquetes IP Multicast. Finalmente, la Reducción de Estados total se puede definir como:

$$r_{as} = 1 - \frac{N_T}{N_A} = 1 - \frac{\sum_{T \in MTS} |T| + \sum_{G \in \vartheta} |G|}{\sum_{G \in \vartheta} |T_A(G)|}$$

Sin embargo, una mejor métrica para reflejar la reducción de estados se logra al utilizar *Group Aggregation* con Reducción de Estados Reducibles, representada por la siguiente ecuación:

$$r_{rs} = 1 - \frac{\sum_{T \in MTS} |T|}{\sum_{G \in \vartheta} (|T_A(G)| - |G|)}$$

Otra métrica útil es la denominada *Relación Hit*, la cual se define como:

$$HR(t) = \frac{\text{número de grupos "cubiertos" por árboles de distribución existentes}}{\text{número de grupos totales}} = \frac{N_h(t)}{N_t(t)}$$

Cuanto mayor sea HR(t), menor será la necesidad de establecer nuevos árboles de distribución para “cubrir” nuevos grupos IP Multicast.

De forma similar se define la *Relación Extend* como:

$$ER(t) = \frac{\text{número de grupos "cubiertos" por árboles de distribución extendidos}}{\text{número de grupos totales}} = \frac{N_e(t)}{N_t(t)}$$

El costo de extender un árbol de distribución para “cubrir” un nuevo grupo Multicast, se espera que sea menor que el de establecer un nuevo árbol. Luego, será menor el porcentaje de grupos de distribución que requieren el establecimiento de nuevos árboles de distribución.

6. ANÁLISIS DE TÉCNICAS Y MECANISMOS DE QoS SOBRE REDES IP MULTICAST.

Durante el desarrollo de este capítulo se analizarán distintos protocolos, mecanismos y métodos que nos permiten provisionar parámetros de calidad de servicio sobre fujos de datos en redes IP Multicast.

Se comienza con la presentación de los tópicos de QoS que se deben tener en cuenta al momento de transportar aplicaciones en Tiempo Real Interteractivas, que requieren de la provisión y control de los requerimientos de QoS impuestos por estas. Luego, se analizarán distintas propuestas que permiten la transmisión de tráfico IP Multicast teniendo en cuenta los requerimientos de calidad de servicio impuestos por las aplicaciones que generan los flujos de datos.

6.1. TÓPICOS QoS EN IP MULTICAST.

Con el advenimiento de nuevas aplicaciones que requieren tener en cuenta las condiciones de *delay*, *jitter*, ancho de banda y pérdida de paquetes, el soporte de QoS sobre las redes IP Multicast se torno un tópico a tener en cuenta, y sobre el cual se han realizado varios desarrollos e investigaciones.

Si bien la utilización del protocolo RSVP permite reservar recursos para un árbol de distribución Multicast, no brinda mecanismos que permitan determinar el mejor camino, este queda en manos de los protocolos de ruteo IP Multicast.

Los protocolos Multicast pueden ser clasificados en dos grandes grupos:

- *Source-Based:* estos utilizan los mecanismos de *Shorted Path Tree*, *SPT*, para determinar el árbol de distribución entre las fuentes y los receptores Multicast. Cada rama del árbol de distribución resulta el camino más corto entre la fuente y cada miembro del grupo. En general, este camino resulta ser el de menor costo en cuanto a saltos, por lo cual presentan las mejores características en cuanto al *delay*. Sin embargo, presentan problemas de escalabilidad en el caso de redes de gran tamaño. Los protocolos IP Multicast DVMRP, PIM-DM y MOSPF entran en esta categoría.

- *Center-Based:* utilizan los mecanismos de *shared-tree* para determinar el árbol de distribución entre las fuentes y los receptores Multicast. Estos protocolos son adecuados en los casos que se tengan redes de gran tamaño y los grupos Multicast

se encuentren dispersos en puntos disímiles de la red. En general brindan buenas características de ancho de banda. Un ejemplo de este tipo de protocolos es CBT.

En la siguiente figura se presenta un resumen de las técnicas de red para poder brindar QoS en redes IP Multicast.

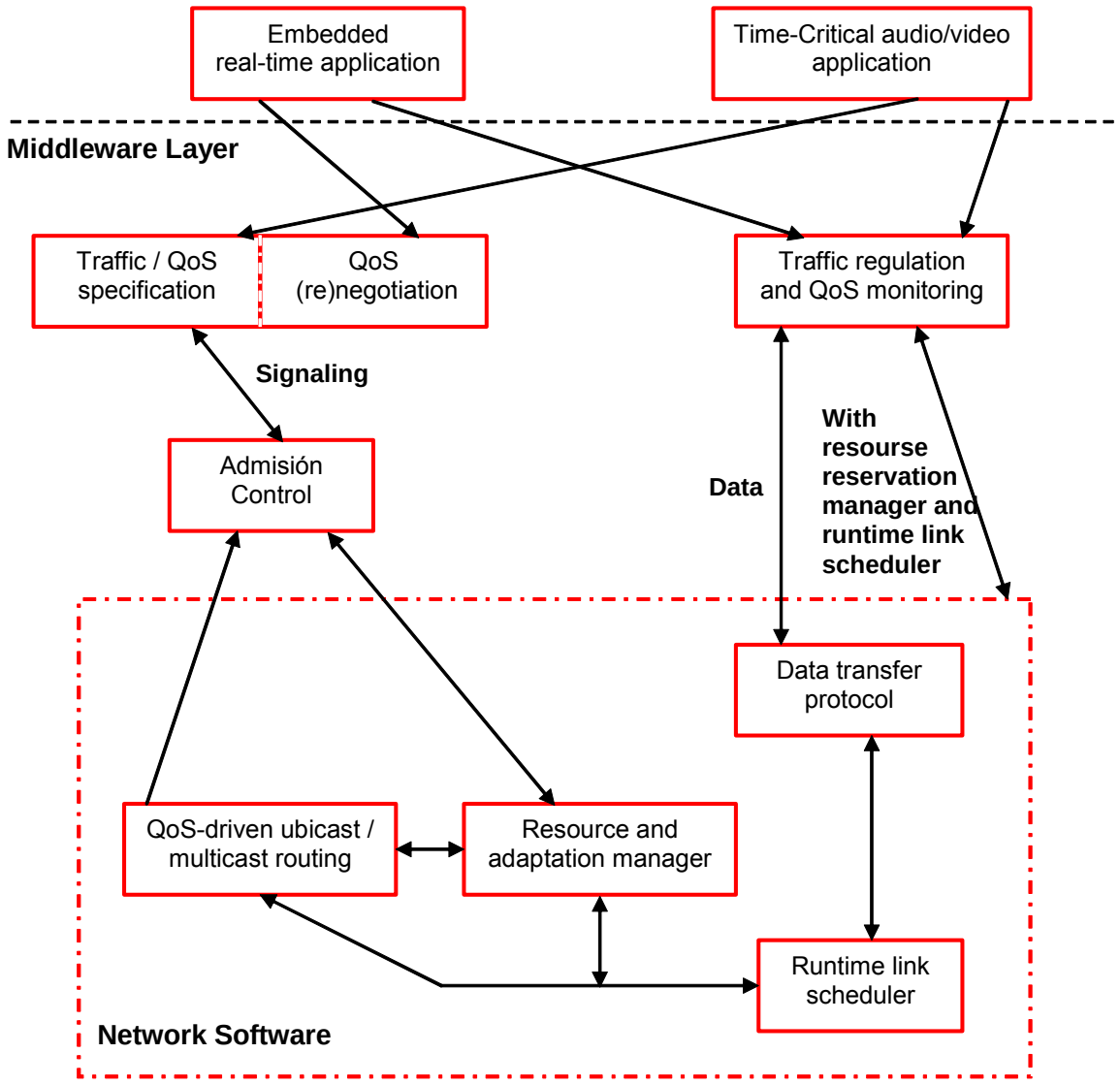


Figura 6.1.1.- Técnicas de Red para la Provisión de QoS (Ref:9).

La calidad de servicio, QoS, en los árboles de distribución Multicast, no se ve sólo afectada por los protocolos de ruteo utilizados, sino que también se deben tener en cuenta:

- Ruteo en Base a Condiciones de QoS.

[9] Tomado Referencia Bibliografica 34

- Reconfiguración de los Árboles de Distribución.
- Migraciones de Árboles de Distribución.

Ruteo en Base a Condiciones de QoS:

Un árbol de distribución Multicast es construido dinámicamente en función de los miembros que se unen y dejan el grupo. Así el problema de contar con un ruteo dinámico que tenga en cuenta los requerimientos de QoS se puede presentar como:

- Dado un nuevo miembro M_N , encontrar un camino, *path*, desde M_N hacia el nodo de un árbol que satisfaga las condiciones de QoS del nuevo miembro.

Las condiciones de QoS pueden clasificarse en dos grandes grupos:

- Requerimientos de Enlaces: ej. Ancho de banda.
- Requerimientos de Camino, o *path*: ej. *delay*, *jitter*.

De lo anterior se puede concluir que un nuevo miembro puede presentar condiciones de QoS que pertenezcan a uno o a ambos de los grupos nombrados anteriormente. Por lo cual, para cumplir con estos requisitos, un protocolo de ruteo Multicast que tenga en cuenta condiciones de QoS debería cumplir con:

- Mejorar la probabilidad de éxito de los mensajes de *Join*.
- Mejorar los costos de los caminos de los árboles de distribución.
- Minimizar el tiempo requerido por un miembro para unirse a un grupo.
- Escalabilidad.

Basados en el mecanismo utilizado para conectar un nuevo miembro a un árbol de distribución, los protocolos de ruteo IP Multicast se pueden clasificar como:

- *Single-Path Routing, SPR*.
- *Multiple-Path Routing, MPR*.

La mayoría de los protocolos SPR presentan características de *best-effort*.

Protocolos	Overhead	Caminos Candidatos	Seleción del Camino teniendo en cuenta Requerimientos de QoS
SPR	Bajo	Uno	No
MPR	Alto	Multiples	No

Tabla 6.1.1.- Comparación entre Protocolos SPR y MPR (Ref: 10).

El *overhead* en los protocolos SPR es bajo ya que sólo realiza la búsqueda de un único camino. Sin embargo, si el camino más corto no tiene los recursos necesarios para brindar los requerimientos de QoS, el protocolo no podrá encontrar un camino que cumpla con estos requisitos de calidad de servicio.

En los protocolos MPR, la búsqueda de caminos tiene en cuenta múltiples caminos candidatos para incrementar las chances de encontrar una rama que cumpla con los requisitos de calidad de servicio impuesto por un nuevo miembro. Sin embargo, esto produce *overheads* elevados en redes de gran tamaño.

Reconfiguración de los Árboles de Distribución.

En el caso de sesiones Multicast dinámicas es importante asegurar que los mecanismos por los cuales los miembros de un grupo se unen y dejan el mismo, *join/leave*, no afecte a la sesión Multicast, y que el árbol de distribución, luego de los mecanismos de *join/leave*, se mantenga dentro de los parámetros óptimos, de modo que continúe brindando los requisitos de QoS impuestos por los receptores del mismo.

Un método para soportar los mecanismos dinámicos de *join/leave* es el reconstruir el árbol de distribución cada vez que un miembro deje o se una a una sesión Multicast. Esto puede involucrar la migración de nodos existentes en el árbol de distribución hacia nuevos nodos, lo cual puede provocar impactos en las sesiones IP Multicast no tolerables para los servicios que se brindan a través de estas.

Otro método que se podría utilizar para soportar los mecanismos dinámicos de *join/leave* es cambiando la estructura del árbol de distribución, en forma incremental, utilizando mecanismos de *graft/prune*. Una desventaja de este esquema es que la calidad del árbol de distribución, como podría ser el costo del mismo, se puede ver deteriorada a lo largo del tiempo como consecuencia de los cambios incrementales.

De lo anterior se ve que un algoritmo de ruteo Multicast debería tener en cuenta dos objetivos básicos, y a la vez contradictorios en algún punto:

- Reducción de costos.
- Minimizar efectos sobre los servicios.

Migraciones de Árboles de Distribución.

La selección de los nodos de *Core* presenta un punto importante a tener en cuenta debido a que la distribución de los mismos tendrá influencia sobre los parámetros de QoS como costo y *delay*.

El mantenimiento de un árbol de distribución Multicast, junto con sus requerimientos de QoS, implicará el tener que realizar una selección *online* de un nuevo *Core,* lo cual redundará en la construcción *online* de un nuevo árbol de distribución basado en la arquitectura del nuevo *Core*. De lo anterior se deduce que se tendrán migraciones de los miembros de los grupos Multicast de un árbol de distribución a otro.

La migración del *Core* también puede ser provocada como parte de un esquema de recuperación ante fallas.

6.1.1. PROBLEMAS DE RUTEO MULTICAST.

En este punto se presentarán los problemas intrinsecos de algoritmos y protocolos de ruteo IP Multicast para poder brindar condiciones de QoS impuestas por las aplicaciones.

Como se vio en puntos anteriores en este trabajo de Tesis, una red puede ser representada como un gráfico $G = (V,E)$, donde V representa el conjunto de los nodos de red y E el conjunto de los enlaces que interconectan los nodos de la red. Bajo este esquema tenemos que:

- $|V|$ representa la cantidad de nodos en la red.
- $|E|$ representa la cantidad de enlaces en la red.

Asociados con cada enlace, tenemos los parámetros que los caracterizan. Por ejemplo, se puede definir la función *delay* como:

[10] Tomado referencia Bibliografica 35.

$d: E \rightarrow R^{+}$, la cual asigna un número positivo a cada enlace en la red.

Así, el valor de $d(i)$ nos brindará una medida del *delay* que experimentan los paquetes de datos a través del enlace $I \in E$, y tiene en cuenta el *delay* de las colas en los nodos, el tiempo de transmisión y el tiempo de propagación.

De forma análoga a lo anterior se puede definir la función ancho de banda, que nos brindará el ancho de banda disponible en los enlaces de la red. Estos parámetros son denominados genéricamente como *Link State*, y son almacenados en cada nodo de la red.

Luego, tenemos los parámetros que nos caracterizan los nodos de la red, los cuales se pueden tener por cada una de las interfases de un nodo, o *router*, de la red. Estos son denominados genéricamente como *Node State*. Al conjunto de parámetros de red y de enlaces, se lo denomina *Network State*.

Definimos a $M \subset V$ como el conjunto de nodos de red que forman parte de un grupo Multicast *M*, donde $v \in M$.

Consideremos que los paquetes originados en un nodo v_S tienen que ser entregados a un conjunto de nodos receptores, representadops por $M - \{v_S\}$. Dependiendo del rol de cada nodo dentro de un grupo IP Multicast, este puede ser fuente, receptor o ambos. Por lo cual, un árbol de distribución T se puede pensar como un sub-gráfico de G, que vincula todos los nodos del grupo M. Por lo cual, se puede representar al árbol de distribución T como:

$P_T(v_S, v_d)$

que representa el camino, o *path*, desde un nodo fuente v_S hacia un nodo receptor v_d.

En base a lo anterior, dado un grupo Multicast M y un conjunto probable de objetivos de optimización, representados por la función O, el ruteo IP Multicast se puede ver como un proceso destinado a construir, basado en la topología de red y los parámetros de red, un árbol de distribución T que optimiza la función de objetivos O. Esto se ve representado en la siguiente figura:

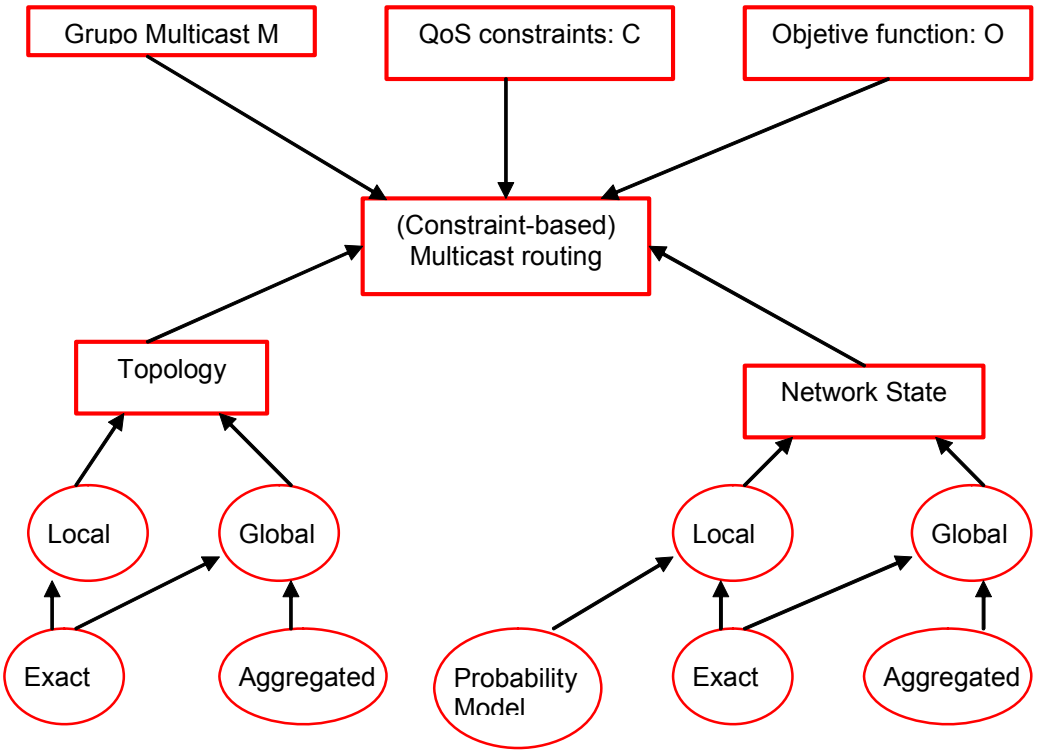


Figura 6.1.1.1.- Componentes de Ruteo Multicast (Ref: 11).

En el caso de los protocolos de ruteo Multicast *constraint-based*, se tiene un conjunto de requerimientos C dados por:

- *Delay end-to-end*.
- *Jitter*.
- Ancho de banda mínimo.
- Probabilidad de pérdida de paquetes.

Estos requerimientos pueden darse en forma individual o un sub-conjunto de ellos. De modo que el árbol de distribución deberá brindar los requerimientos de calidad de servicio impuestos por las aplicaciones.

De acuerdo a las condiciones de calidad de servicio que se requieran, los árboles de distribución se pueden clasificar genéricamente como:

Additive Tree Constraint:

En estos, para cualquier camino, o *path*, dado por:

$P_T\ (u,v) = (u,i,j,.....,k,v)$

se considera que el requerimiento, *constraint*, es aditivo si:

$$m\,(u,v) = m\,(u,i) + m(i,j) + ... + m\,(k,v)$$

Por ejemplo, el *delay end-to-end*, $d(u,v)$, desde el nodo *u* al nodo *v* es aditivo, y es igual a la suma de los *delays* individuales $d(i,j)$ a lo largo del camino $P_T\ (u,v)$.

Multiplicative Tree Constraint:

Se considera que el árbol presenta requerimientos multiplicativos si:

$$m\,(u,v) = m\,(u,i) \times m(i,j) \times ..... \times m\,(k,v)$$

Por ejemplo, la probabilidad $1 - P_L(u,v)$, para un paquete de alcanzar el nodo *v* desde el nodo *u* a lo largo del camino $P_T\ (u,v)$, es multiplicativo e igual al producto de las métricas individuales $1 - P_L\,(i,j)$ a lo largo del camino $P_T\ (u,v)$.

Concave Tree Constraint:

Se considera que el árbol presenta requerimientos cóncavos si:

$$m\,(u,v) = \min\left[m\,(u,i)\,,\, m(i,j)\,,.....,\, m\,(k,v)\right]$$

Por ejemplo, el ancho de banda $b(u,v)$ disponible a lo largo del camino, entre los nodos *u* y *v*, es cóncavo e igual al ancho de banda mínimo entre los enlaces que componen el camino $P_T\ (u,v)$.

Así, de acuerdo a lo tratado anteriormente, se puede ver que los problemas de ruteo en los protocolos IP Multicast, con soporte de QoS, se pueden clasificar, en forma genérica, como:

1. ***Link-Constrained*:** proporcionan el manejo de un requerimiento de enlace en la construcción de los árboles de distribución, ej.: de ancho de banda.

[11] Tomado Referencia Bibliografica 34

2. ***Multiple Link-Constrained*:** proporcionan el manejo de uno o más requerimientos de enlace en la construcción de los árboles de distribución, ej.: de ancho de banda y *buffering*.

3. ***Tree-Constrained***: proporcionan el manejo de un requerimiento en la construcción de los árboles de distribución, ej.: *delay* de ruteo.

4. ***Multiple Tree-Constrained:*** proporcionan el manejo de uno o más requerimientos en la construcción de los árboles de distribución, ej.: *delay* de ruteo y *jitter*.

5. ***Link and Tree-Constrained:*** proporcionan el manejo de un requerimiento de enlace y uno de árbol de distribución en la construcción de los árboles de distribución, ej.: *delay* y ancho de banda.

6. ***Link Optimization:*** proporcionan el soporte de una función de optimización para localizar un árbol de distribución óptimo, ej.: maximización del ancho de banda de los enlaces que forman el árbol de distribución.

7. ***Link- Constrained Link Optimization:*** proporcionan el manejo de un requerimiento de enlace, y además una función de optimización de enlace es utilizada para localizar un árbol de distribución óptimo, ej.: requerimiento de ancho de banda más la optimización de la utilización de los *buffers*.

8. ***Tree Optimization:*** utilizan una función de optimización para localizar el árbol de distribución óptimo, ej.: minimización del costo total del árbol de distribución.

9. ***Link- Constrained Tree Optimization:*** proporcionan el manejo de un requerimiento de enlace, y además una función de optimización para localizar el árbol de distribución óptimo, ej.: requerimiento de ancho de banda más la minimización del costo total del árbol de distribución.

10. ***Tree Constrained Link Optimization:*** proporcionan el manejo de un requerimiento de árbol de distribución, y además una función de optimización de enlace es utilizada para localizar un árbol de distribución óptimo, ej.: *delay* y optimización del ancho de banda.

11. ***Tree Constrained Tree Optimization:*** proporcionan el manejo de un requerimiento de árbol de distribución, y además una función de optimización para localizar el árbol de

distribución óptimo, ej.: *delay* más la minimización del costo total del árbol de distribución.

12. ***Link and Tree Constrained Tree Optimization:*** proporcionan el manejo de un requerimiento de árbol de distribución más un requerimiento de enlace, y además una función de optimización para localizar el árbol de distribución óptimo, ej.: *delay*, ancho de banda más la minimización del costo total del árbol de distribución.

6.1.2. RECUPERACIÓN ANTE FALLAS EN ESQUEMAS MULTICAST CON SOPORTE DE QoS.

Teniendo en cuenta los requerimientos de las nuevas aplicaciones sobre IP Multicast, se hace necesario que los protocolos de ruteo Multicast brinden mecanismos de recuperación ante fallas. Estos mecanismos deben tener en cuenta varias consideraciones de diseño, como por ejemplo:

- Escalabilidad.
- Rápida recuperación.
- Minimizar el *overhead* del protocolo Multicast.
- Minimizar los impactos sobre los servicios.

Los mecanismos de recuperación ante fallas los podemos clasificar en dos grandes grupos:

- *Protection-Based:* emplean mecanismos dedicados de protección, como por ejemplo redundancia de caminos. Estos esquemas son adecuados para servicios en tiempo real, en los cuales la recepción de cada paquete de datos es crucial.
- *Restoration-Based*: en estos mecanismos, al detectarse la ocurrencia de una falla, se intentará re-enrutar el tráfico Multicast a través de enlaces o nodos distintos, con impacto mínimo sobre el servicio.

En servicios Multicast Multimedia, el problema de contar con mecanismos de recuperación ante fallas es bastante complicado, ya que la reservación de recursos y el comportamiento dinámico de las sesiones y de los grupos interactuan con la reconfiguración de la red. La utilización de un esquema *protection-based* no es apropiado para este tipo de

servicios, ya que resulta no escalable debido a la cantidad de caminos redundantes necesarios para asegurar el servicio ante una falla.

En el caso de protocolos de ruteo Multicast de características *Core-Based*, estos tienen un punto de falla en el *Core*. Si este falla, todas las sesiones Multicast establecidas serán afectadas. Por lo cual, estos protocolos deberían contar con mecanismos de recuperación de fallas, los cuales tengan en cuenta los aspectos que se muestran en la siguiente figura:

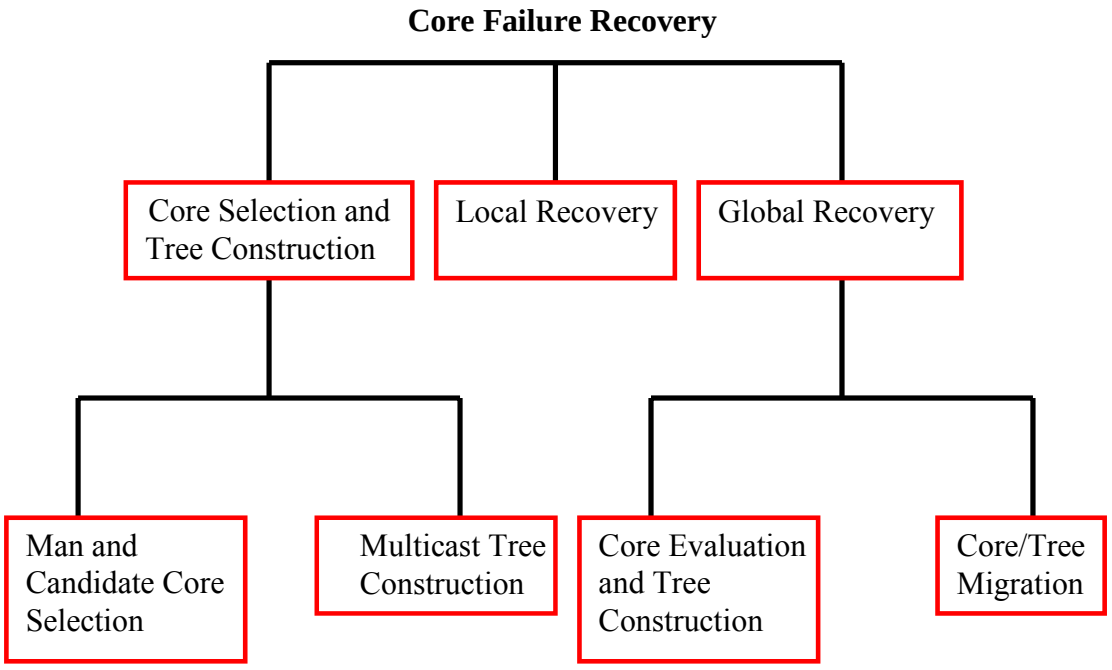


Figura 6.1.2.1.- Aspectos Necesarios en un Mecanismo de Recuperación ante Fallas (Ref: 12).

Como se ve en la figura, estos mecanismos presentan tres aspectos principales:

- *Core Selection and Tree Construction:* se seleccionará un nuevo *Core* de una lista de nodos candidatos, considerando un árbol de distribución tal que minimice el impacto sobre el servicio.

- *Local Recovery:* cuando ocurre una falla en un nodo de *Core* o en un enlace, la recuperación de la falla puede tener lugar en una escala local.

- *Glabal recovery:* en algunos casos donde el nivel de la falla es importante, puede ser necesario recuperar en forma global los grupos Multicast. Por lo cual, se elegirá un nuevo *Core* y se migrarán los árboles de distribución, sesiones y grupos.

[12] Tomado Referencia Bibliografica 33.

6.2. PROTOCOLOS, MÉTODOS Y ESQUEMAS MULTICAST CON SOPORTE DE REQUERIMIENTOS DE QOS.

A continuación se analizarán los distintos protocolos, métodos y esquemas que permiten brindar calidad de servicio sobre redes IP Multicast:

- AQoSM.
- QoSMIC.
- QMRP.
- SoMR.
- RIMQoS

Estos nos proporcionarán la posibilidad de controlar y administrar los requisitos de calidad de servicio que nos imponen las nuevas aplicaciones. Como caso de comparación de los protocolos y mecanismos mencionados anteriormente se tomarán los requerimientos impuestos por las aplicaciones *Real Time* Interactivas de vídeo conferencia, desarrollado en el capítulo 3 de la Tesis. En éstas consideraremos como parámetros de QoS a ser controlados: el ancho de banda, el *jitter*, el *delay* y la pérdida de paquetes.

6.2.1. ANÁLISIS DEL MÉTODO AQoSM.

Como se vio en el capítulo 5, el objetivo del *Aggregated Multicast Tree* es el de reducir los estados de *forwarding* IP Multicast, no teniendo en cuenta consideraciones de QoS como *Delay*, Ancho de Banda, etc. No obstante, proporciona una buena herramienta para simplificar la administración del tráfico y provisión de requerimientos de QoS. El método AQoSM se basa en *Aggregated Multicast Tree* para proporcionar QoS sobre flujos de datos IP Multicast.

Para comenzar, se considerará un único dominio MPLS, un dominio de backbone, el cual soporta Diff-Serv, y una aplicación de vídeo conferencia a ser transmitida a través de este.

Como se propone en *Aggregated Multicast Tree*, la inteligencia de la red se encuentra en una entidad lógica denominada *Tree Manager*, el cual se establecerá en un nodo centralizado en la arquitectura de red. Este nodo contará con información sobre:

a. Topología de la red.

b. Disponibilidad de recursos.

c. Grupos Multicast.

d. Requerimientos de QoS de los grupos Multicast.

El *Tree Manager* presenta conectividad con los nodos de red, tanto con los nodos de borde, *ER*, como con los nodos del core, *CR*. Esta compuesto por diferentes módulos lógicos de servicio:

a. *Control de Admisión*: mantiene información sobre los ancho de banda disponibles en los enlaces.

b. *Matching de grupos-árboles de distribución*: mantiene información de los grupos activos y árboles establecidos, además de la tabla que mapea grupos con árboles de distribución. Se encarga de realizar el "*matching*" de grupos con árboles de distribución nuevos o establecidos.

c. *Ruteo*: dentro de sus funciones podemos encontrar:

   i. "Dialogo" con los *router*s de la red para obtener información sobre la topología de red.
   ii. Establecimiento y desactivación de árboles de distribución.

d. *Políticas de control*: ayuda a mantener una política de control de red.

Consideremos una situación en la cual se presenta un pedido para el establecimiento de un nuevo grupo IP Multicast, el cual recibirá los flujos de datos generados por la aplicación de vídeo conferencia.

Al iniciar la red, antes de tomar algún pedido de inclusión de grupos Multicast o establecimiento de árboles de distribución, el *Tree Manager* "interroga" a los *router*s de la red sobre la siguiente información:

a. Estado y disponibilidad de enlaces, *links*.

b. Topología de red.

c. Ancho de banda disponible en los enlaces.

d. Grupos y árboles Multicast establecidos.

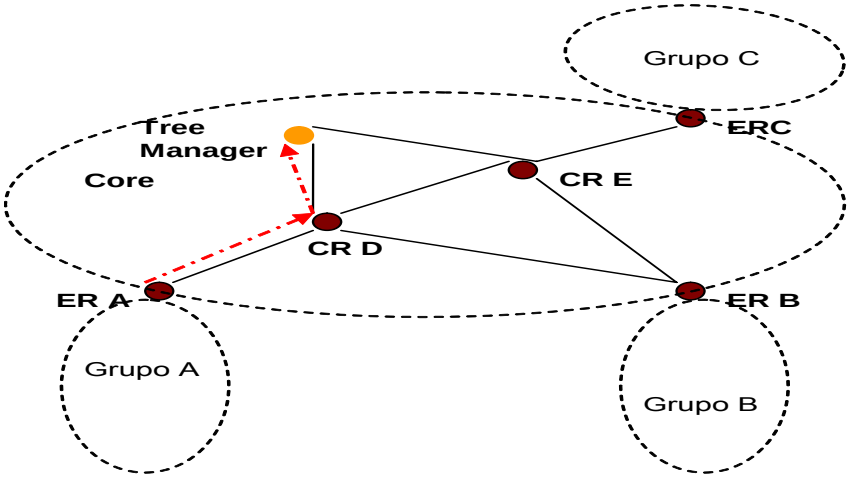


Figura 6.2.1.1.- Pedido de Inclusión de un Nuevo Grupo Multicast.

Como se ve en la figura anterior, el *ER A* envía un mensaje al *Tree Manager* indicando el establecimiento de un nuevo grupo IP Multicast. Entonces, el *Tree Manager* primeramente intentará mapear el grupo dentro de un árbol de distribución establecido previamente. Si no encuentra un árbol de distribución, el *Tree Manager* calculará un nuevo árbol de acuerdo con los requerimientos de QoS del grupo y de sus miembros. Luego, mediante el módulo de admisión verificará si el ancho de banda disponible es adecuado para los requerimientos del grupo. En caso que de la verificación resulte que el ancho de banda disponible es insuficiente, el pedido de establecimiento de grupo será rechazado. Si el ancho de banda resultara adecuado, se establecerá un árbol de distribución MPLS a través de la utilización del protocolo LDP.

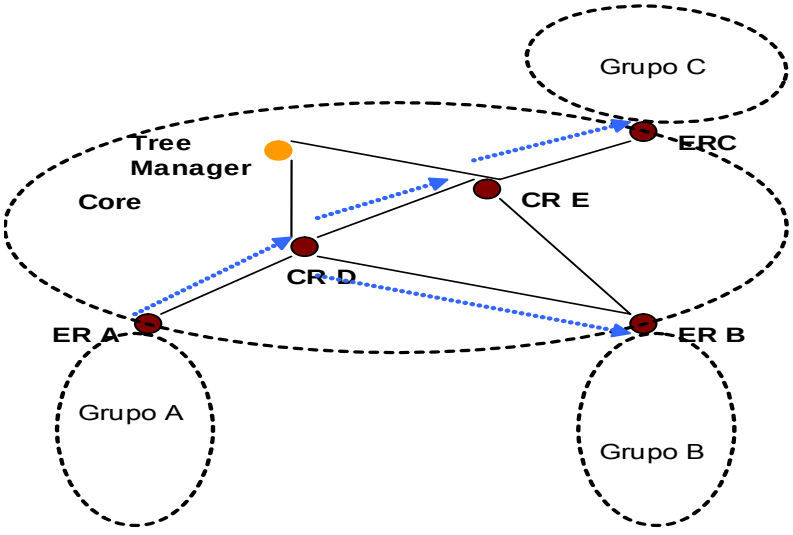


Figura 6.2.1.2.- Envío de Paquetes sobre Árbol Multicast.

Una vez que el *Tree Manager* establece el árbol de distribución, propaga la información de *Group-tree Matching* a los *router* de borde, *ER*, dentro del grupo Multicast. Los *router*s de borde de entrada, *Source ER*, encapsulan, clasifican y etiquetan los paquetes de datos

pertenecientes al grupo y los "*forwardean*" al resto de los routres dentro del árbol de distribución. Luego, los *routers* de borde receptores, *Receiver ER*, realizan las operaciones de desencapsulación y *forwarding* de los paquetes a los miembros del grupo conectados a estos.

Ahora se presentará un detalle de los pasos descriptos en el mecanismo anterior.

- Para la recolección de la información del estado de los enlaces, *link-state*, se propone utilizar un protocolo de ruteo Unicast *link-state* como OSPF o BGP. Esto nos dará un menor *overhead* y la utilización de protocolos conocidos.

- Para el envío de información de *Group Memenbership*, desde los *router*s de borde hacia el *Tree Manager*, se propone utilizar el mecanismo de *piggybacking* sobre los paquetes del protocolo de ruteo *link-state*.

- Para las funciones de control de admisión, se propone la utilización de métodos de medición.

- Para el establecimiento de un árbol de distribución, en los casos que el *Tree Manager* no encuentre un árbol de distribución previamente establecido que cumpla con los requerimientos del grupo Multicast, se propone la utilización de protocolos del ruteo Multicast como PIM-SM, para calcular el nuevo árbol de distribución, lo cual establecerá un árbol de distribución Multicast bi-direccional. El establecimiento de árboles bi-direccionales permite que una mayor cantidad de grupos puedan compartir un árbol de distribución, redundando en una disminución de estados de *forwarding* Multicast y menor cantidad de *Aggregated Tree*.

- Para el *matching* de un grupo a un árbol de distribución, el *Tree Manager* necesita almacenar tablas que contengan información de los árboles de distribución establecidos, grupos Multicast activos y entradas de *matching group-tree*. Para esto, se propone la utilización de un mecanismo similar al descripto para *Aggregated Tree*, el cual se describe a continuación:

  Una red puede ser modelizada como un gráfico unidirecto representado por G (V,E); donde, a cada borde (i,j) se le asigna un costo positivo $c_{ij} = c_{ji}$, el cual representará el costo de transportar una unidad de dato desde el nodo, o *router*, *i* hacia el *j* (o viceversa).
  Dado un árbol de distribución Multicast *T*, el costo total de ditribuir una unidad de datos sobre este será:

$$C(T) = \sum_{(i,j)\in T} c_{ij}$$

Si consideramos que cada enlace tiene un costo de 1, el costo del árbol de distribución *T* será:

$C(T) = |T| - 1$, donde $|T|$ representa la cantidad de nodos en *T*.

Consideremos que la cantidad de árboles de distribución Multicast establecidos en la red esta representada por MTS.

Consideremos que un árbol de distribución Multicast Nativo (calculado utilizando PIM-SM), el cual satisface los requerimientos de membresía y QoS de un grupo Multicast *g*, como: $T_A(g)$; mientras que el *Aggregated Tree* utilizado por el grupo *g* esta representado por $T(g)$. Puede presentarse el caso donde $T(g)$ no posea un *Perfect Match* con el grupo Multicast *g*, ya que por ejemplo, algunos de los nodos, o *router*s,del *Aggregated Tree* $T(g)$ no sean nodos miembros del grupo *g*, con lo cual los paquetes Multicast llegarán a receptores que no pertenecen al grupo, con lo cual estos serán descartados. Esto generará un ancho de banda no utilizado eficientemente para la transmisión, *bandwidth overhead*.

Supongamos que un *Aggregated Tree* T_0 es utilizado por los grupos Multicast g_i, con $1 \le i \le n$, donde cada grupo posee un árbol de distribución Multicast Nativo representado por $T_A(g_i)$, luego, el *overhead* de ancho de banda promedio para T_0 se puede definir como:

$$\delta_A(T_0) = \frac{\sum_{i=1}^{n} B(g_i).(C(T_0) - C(T_A(g_i)))}{\sum_{i=1}^{n} B(g_i).C(T_A(g_i))} = \frac{C(T_0).\sum_{i=1}^{n} B(g_i)}{\sum_{i=1}^{n} B(g_i).C(T_A(g_i))} - 1$$

donde $B(g)$ representa el ancho de banda requerido por el grupo *g.*

Cuando el *Tree Manager* detecta un nuevo grupo Multicast *g*, actualiza la tabla de grupos Multicast y ejecuta el siguiente algoritmo de "*matcheo*":

1.- Calcula un árbol de distribución Multicast Nativo $T_A(g)$, para el grupo *g*, basado en la membresía Multicast, requerimiento de ancho de banda

del grupo, y ancho de banda disponible en los enlaces. En caso que no se cumpla con el requerimiento de ancho de banda del grupo, el pedido de establecimiento será desechado.

2.- Para cada árbol de distribución *T* en MTS, si *T* "cubre" los nodos necesarios para el grupo *g*, y se cuenta con ancho de banda suficiente en los enlaces que componen las ramas del árbol de distribución, el *Tree Manager* calculará $\delta_A(T)$. Si $\delta_A(T) < b_t$, el árbol de distribución*T* es considerado como un candidato para ser asignado al grupo *g*.

3.- Si el *Tree Manger* tiene más de un árbol de distribución que pueda ser asignado al grupo *g*, seleccionará aquel que satisfaga la condicón de *C (T)* mínimo, denominando con T_{min} al árbol de distribución que satisfaga la condición anterior, y se asignará este árbol al grupo *g*, para lo cual el *Tree Manager* actualizará las tablas de grupos Multicast y *group-tree matching*.

4.- Si en el paso descripto en el punto 2, el *Tree Manager* no encuentra ningún árbol de distribución que pueda ser considerado como candidato para ser asignado al grupo *g*, se asiganará el árbol $T_A(g)$, árbol de distribución Multicast Nativo, al grupo *g*. Luego el *Tree Manager* incluirá a $T_A(g)$ en MTS, y actualizará las tablas de grupos Multicast y *group-tree matching*. Obviamente, que de no ser posible establecer $T_A(g)$, con las condiciones descriptas en el punto 1, el pedido de establecimiento será rechazado.

En el caso que varíe el requerimiento de ancho de banda del grupo *g*, el *Tree Manager* verificará si el árbol de distribución asignado a este cumple con este cambio, de no ser así, ejecutará el procedimiento descripto anteriormente.

- Para el establecimiento del árbol de distribución bi-direccional en MPLS se propone la utilización de un esquema centralizado, donde el *Tree Manager* generará todas las etiquetas MPLS y las distribuirá a los *router*s de borde, *ER*. Cuando el árbol queda inactivo, el *Tree Manager* envía a los *router*s de entrada del *Aggregated Tree*, mensajes de liberación de etiquetas, y actualiza las entradas en la tabla de árboles de distribución y *tree-matching*.

6.2.1.1. PERFORMANCE DE AQoSM Y COMPARACIÓN CON IP MULTICAST NATIVO.

6.2.1.1.1. DESCRIPCIÓN DE LAS COSIDERACIONES Y MÉTODOS A SER UTILIZADOS EN LA COMPARACIÓN DE AQoSM CON IP MULTICAST NATIVO.

Para verificar la *performance* del método AQoSM definiremos una serie de parámetros que nos permitirán verificar y demostrar la misma:

- **NMT (Números de Árboles de Distribución MPLS):** se considera como la cantidad promedio de árboles MPLS que mantiene el *Tree Manager*. Da una medida del *overhead* que produce el tener que mantener árboles Multicast. Cuanto mayor sea este número, mayor será el procesamiento necesario en los *router*s de borde, *ER*, y en el *Tree Manager*.

- **NLFE (Número de Entradas de Etiquetas de *Forwarding*):** se considera como la cantidad promedio de entradas de *forwarding* , correspondientes a etiquetas MPLS, tanto en los *router*s de borde, *ER*, como en los *router* de *core*, *CR*. Da una dimensión del requerimiento de memoria y procesamiento de *forwarding* en los *router*s de la red.
  Cuanto menor sea este indicador, menor será la cantidad de memoria requerida y se tendrá un *forwarding* más rápido de las etiquetas MPLS.

- **RRR (Tasa de Rechazo de Pedido de Establecimiento de un grupo Multicast):**

$RRR(t) = \frac{N_R(t)}{N_A(t)}$, donde:

$N_A(t)$: cantidad de pedidos de establecimiento de grupos Multicast, en un período t, después de alcanzado el estado de estabilidad.

$N_R(t)$: cantidad de pedidos de establecimiento de grupos Multicast rechazados.

- **TSR (Tasa de Establecimiento de Árboles de Distribución):**

$TSR = \frac{N_A(t) - N_M(t) - N_R(t)}{N_A(t)}$, donde:

$N_M(t)$: representa la cantidad de pedidos de establecimiento de grupos Multicast, en un período t, que pueden ser asignados, "*matcheados*" a árboles de distribución existentes.

Consideremos la siguiente estructura de red:

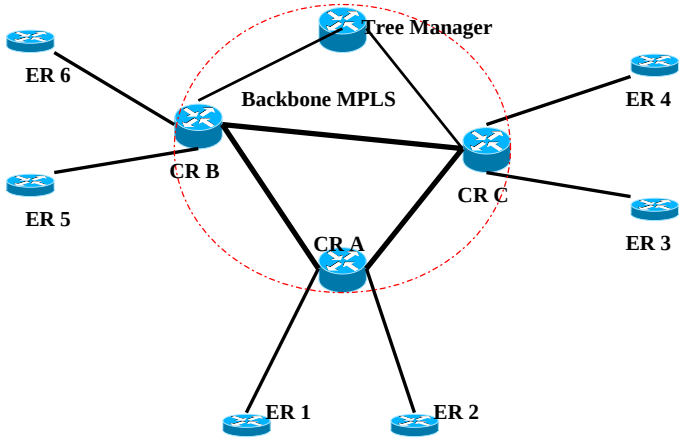


Figura 6.2.1.1.1.1.- Esquema de Red Propuesto.

La *performance* de la agregación de árboles de distribución Multicast se ve afectada por la distribución de los miembros de un grupo Multicast dentro de la red.

Para la generación y distribución de los miembros de un grupo Multicast se considerará:

a. Que los miembros de los grupos Multicast y los nodos no están correlacionados (esto refleja la distribución de los miembros de grupos Multicast de muchas aplicaciones reales como Juegos en Internet).

b. Distribución de los miembros dentro de un grupo Multicast.

c. Correlación entre grupos Multicast.

d. Distinto peso a los nodos Multicast según su "participación".

e. Correlación entre nodos.

f. Distribución del tamaño de los grupos Multicast.

Así, a cada nodo de la red se le asigna un peso que representa la probabilidad de que el nodo sea parte de un grupo Multicast.

A partir de las consideraciones anteriores se desarrollará un modelo que nos permitira generar grupos Multicast, teniendo en cuenta el "peso" de los nodos o *router*s de la red:

Modelo para la Generación de Grupos Multicast.

1. Consideremos dos nodos o *router*s los cuales denominaremos *i* y *j*, donde el peso del nodo i esta dado por $w(i)$ y el del nodo j por $w(j)$.

2. Consideraremos $0 \langle w(i)$ y $w(j) \leq 1$.

3. $N(i)$: cantidad de grupos que tienen al nodo *i* como miembro.

4. $N(j)$: cantidad de grupos que tienen al nodo *j* como miembro.

5. De lo anterior se tiene, que en promedio, $\frac{N(i)}{N(j)} = \frac{w(i)}{w(j)}$.

6. Consideraremos que la cantidad de nodos en la red es igual a *N* y que los nodos están numerados de 1 a *N*.

7. Consideraremos que para cada nodo *i*, $1 \leq i \leq N$, y que $0 \leq w(i) \leq 1$

8. Luego, tomando en cuenta lo presentado en los puntos 1 a 7, se tiene el siguiente algoritmo que permitirá simular la creación de los grupos Multicat:

```
for i = 1 to N do
        generate a random number between 0 and 1, let it be p
        if p < w(i) then
                add i as a group member
        end if
end for
```

Basados en el modelo descripto anteriormente, y en la arquitectura de red considerada, a los nodos de *Backbone* o *Core*, los CR_i, les asignaremos un peso de 0. Al resto de los nodos de la red, ER_i, les asignaremos un peso de 0,2 o 0,8 , dependiendo del tráfico en tiempo real del

nodo de *core* asociado. El fundamento de esta consideración se basa en el hecho que, en un *router*, cuanto mayor sea el tráfico que pasa a través de este, mayor sera su participación en el proceso de comunicación de la red, lo cual le da mayor probabilidad de unirse a un grupo Multicast.

Con respecto a los anchos de banda de los enlaces en la red, consideraremos que los enlaces entre los nodos de borde y los nodos de *core* es infinito, y en los enlaces entre los nodos de *core* tomaremos anchos de banda comerciales (STM-16).

Para la generación de las sesiones Multicast, consideraremos que estas arriban siguiendo una distribución de Poisson, con tasa de arrivos λ.

Consideraremos que el tiempo de vida de una sesión cumple con una distribución exponencial con promedio μ.

En el estado de estabilidad, el número promedio de sesiones estará dado por:

$$\overline{N} = \lambda \, . \, \mu$$

Definamos ahora tres tipos de grupos Multicast:

i. De bajo ancho de banda: 10 kbps.
ii. De ancho de banda medio: 100 kbps.
iii. De gran ancho de banda: 1 Mbps.

Supongamos que los requerimientos de ancho de banda de los grupos que son generados cumplen con las siguientes consideraciones:

i. 50% de los grupos presentan requerimiento de ancho de banda bajo.
ii. 30% de los grupos presentan requerimiento de ancho de banda medio.
iii. 20% de los grupos presentan requerimiento de ancho de banda alto.

Consideremos que el estado de estabilidad se alcanza después de $T = 10\,\mu$.

6.2.1.1.2. COMPARACIÓN DE AQoSM E IP MULTICAST NATIVO.

A continuación compararemos la *preformance* obtenida si utilizaramos AQoSM versus la utilización de IP Multicast Nativo sobre MPLS (PIM-SM/CBT MPLS), suponiendo la transmisión de una aplicación de vídeo conferencia. Para esto nos basaremos en los desarrollos y consideraciones realizadas en el punto anterior.

En Multicast Nativo, un árbol de distribución en MPLS es construido utilizando el protocolo PIM-SM/CBT para cada uno de los grupos Multicast.

Para el caso de AQoSM consideraremos que los árboles generados son bi-direccionales.

Para ambos casos, tomaremos que cada miembro de un grupo puede ser fuente o receptor de tráfico Multicast.

Una vez establecida una sesión Multicast, su *router* de *core*, o *RP*, es seleccionado aleatoriamente entre los tres *router*s de *core* del esquema de red propuesto.

Para el caso de AQoSM el "mapeo" o asignación de un grupo a un árbol de distribución es realizado utilizando el algoritmo descripto en el punto 6.1, donde el protocolo de ruteo será PIM-SM/CBT, al igual que para el caso de IP Multicast Nativo.

En ambos casos, AQoSM y Multicast Nativo (PIM-SM/CBT), si el árbol de distribución calculado a partir del *router* de *core* asignado no cumple con los reuquerimientos del grupo, se seleccionará un nuevo *RP* entre el resto de los nodos candidatos, hasta que se encuentre un árbol que satisfaga los requerimientos del grupo Multicast, y en caso de no poder encontrarse un árbol que cumpla con los requerimientos del grupo, el grupo será rechazado. De esta manera se logrará un mejor balanceo de carga en la red.

Para el caso de AQoSM consideraremos que el umbral de *overhead* de ancho de banda, b_{th} varia entre 0 y 0,3 (con pasos de 0,1). De aquí, gráficando estas condiciones, se tiene:

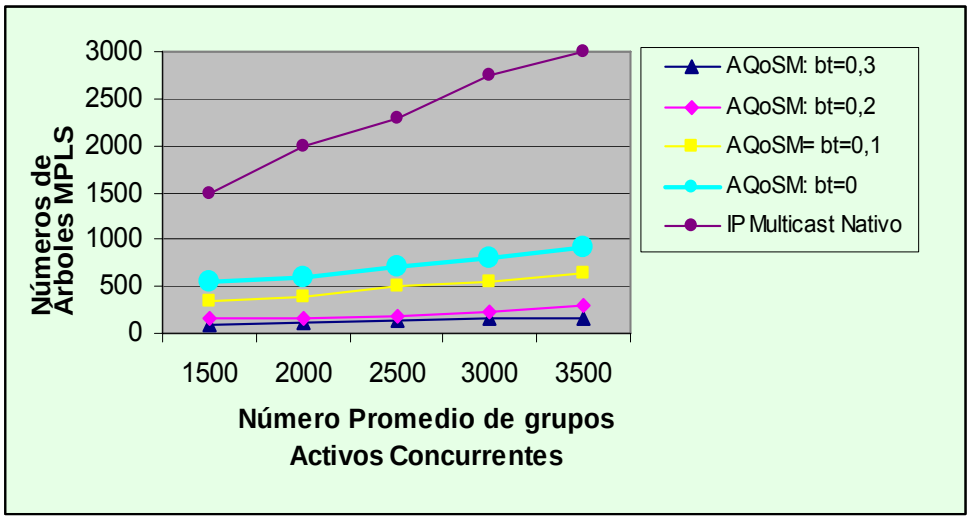


Figura 6.2.1.1.2.1.- Gráfico Número de Árboles MPLS vs Número Promedio de Grupos Activos Concurrentes.

En el gráfico anterior se muestra la relación entre el Número de Árboles MPLS con el número promedio de Grupos Concurrentes Activos, tanto para el caso de AQoSM e IP Multicast Nativo. Las curvas representan esta relación en función del valor del umbral de *overhead* de ancho de banda (b_{th}), el cual se hizo variar entre 0 y 0.3, con pasos de 0.1.

Del gráfico se puede concluir que AQoSM escala con el promedio de grupos IP Multicast activos; en cambio, para el caso de IP Multicast Nativo (PIM-SM/CBT), la cantidad de árboles MPLS crece casi linealmente con la cantidad de grupos activos. En AQoSM, cuando la cantidad de grupos activos tiende a aumentar, la catidad de árboles MPLS aumenta, aunque en mucha menor proporción que para el caso de IP Nativo, lo cual implica menor *overhead* originado por el mantenimiento de los árboles distribucuión.

También se ve en el gráfico anterior que para el caso de AQoSM, cuanto mayor es la cantidad de grupos Multicast activos, mayor cantidad de grupos pueden compartir un árbol MPLS.

6.2.1.2. CONCLUSIONES.

- ⇨ Una característica importe AQoSM es la de separar la entidad lógica de un grupo de la de un árbol de distribución. Así, varios grupos pueden ser multiplexados sobre un único árbol mediante la etiquetación de los paquetes de datos a través de MPLS.

- ⇨ Un grupo Multicast puede ser asignado a grupos de distribución distintos durante su tiempo de vida, lo cual se puede realizar en una forma rápida y eficiente.

- Permite optimizar la utilización de los recursos de red, permitiendo realizar reparto de carga, *load balacing*, y poder adaptarse a los cambios en los requerimientos impuestos por las aplicaciones y grupos Multicast.

- Comparándolo con un esquema Multicast Nativo, o tradicional, como puede ser PIM-SM/CBT:

    i. Se reducen las entradas en la tabla de *forwarding* de etiquetas MPLS.

    ii. Se reduce la sobrecarga, *overhead*, relacionada con el establecimiento y mantenimiento de los árboles de distribución al aumentar la cantidad de grupos Multicast activos.

    iii. Permite el soporte de una mayor cantidad de receptores.

    iv. Se utilizan *LSPs* MPLS entre los nodos ramas del árbol, lo cual permite reducir la cantidad de estados de *forwarding* y así mejorar la escalabilidad que presenta un esquema Multicast Nativo. Este último requiere que los *routers*, o nodos, mantengan un estado de *forwarding* por cada grupo Multicast activo.

    v. Sólo se tienen estados Multicast en los *brandching routers*, o nodos pertenecientes a las ramas del árbol de distribución, y el resto de los *router*s no necesitan almacenar información sobre los estados de distribución Multicast.

    vi. Entre dos *brandching routers*, o nodos pertenecientes a las ramas del árbol de distribución, se utilizan protocolos Unicast, lo cual aporta a la reducción de los estados de distribución Multicast.

    vii. El establecimiento de *LSPs* entre *brandching router*s, o nodos pertenecientes a las ramas del árbol de distribución, permite reducir los recursos de memoria en los *router*s y evita la necesidad de utilizar técnicas de encapsulación entre estos.

- Una ventaja que trae este mecanismo es la posibilidad que los grupos se intercambien entre distintos árboles de distribución en forma rápida y eficiente. Así, se reduce el costo de establecimiento de cada grupo. Una razón por la cual un grupo

puede requerir un cambio de árbol de distribución es por razones de QoS como *delay* o ancho de banda. Esto resulta muy importante para el caso de las aplicaciones de vídeo conferencia, o aplicaciones en Tiempo Real Interactivas.

⇨ Se reduce la carga de procesamiento y memoria en los *router*s, o nodos, del *core*.

⇨ Los *router*s de borde tienen la mayor parte de la inteligencia de red, lo cual permite que los *routers* de *core* se dediquen a su función central que es el *forwarding* de los paquetes de datos.

6.2.2. ANÁLISIS DEL PROTOCOLO QoSMIC.

Las principales características de este protocolo son:

- Se presenta como un protocolo de ruteo Multicast para Internet, el cual provee caminos, *paths*, sensibles a las condiciones de QoS en una manera escalable, eficiente y flexible.

- Identifica múltiples caminos y selecciona aquel que cumpla con las condiciones de QoS requeridas. Así, un nuevo miembro que se une a un grupo Multicast, puede seleccionar el camino que mejor se adapte a sus necesidades de calidad de servicio.

- Dos de sus características más importantes son su flexibilidad y adaptabilidad.

- La información sobre QoS depende de la información que suministren los nodos, tanto de aquellos que forman parte del árbol de distribución como de los que no. En el caso en que los nodos no provean información de calidad de servicio, el protocolo recurrirá a un procedimiento que basa la estimación de los parámetros de QoS basado en la cuenta de saltos para los diferentes caminos.

- Soporta funcionalidades *user-driven*; es decir, los usuarios pueden desconectarse y reconectarse del árbol de distribución cuando las condiciones de QoS se vuelvan inaceptables para sus requerimientos.

- No provee garantías sobre las condiciones de calidad de servicio de los caminos, o ramas del árbol de distribución, ni tampoco garantiza condiciones globales sobre los distintos caminos del árbol de distribución.

6.2.2.1. DESCRIPCIÓN DEL PROTOCOLO.

QoSMIC implementa una solución de ruteo heurística, la cual intenta encontrar distintos caminos hacia los nodos en las adyacencias del nuevo nodo. No implementa un nodo que cumpla con las funciones de *core* de la red, de esta forma el árbol de distribución se encuentra siempre "cerca" de los miembros activos del grupo Multicast, lo cual le permite a QoSMIC el poder brindar una mejor *performance* extremo a extremo y el poder manejar una mayor cantidad de usuarios.

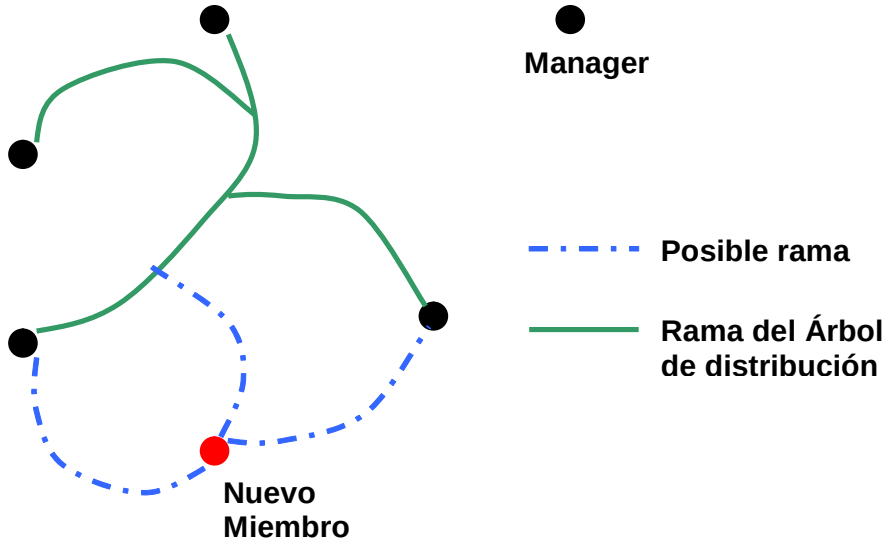


Figura 6.2.2.1.1.-Procedimiento QoSMIC de Identificación de Rama del Árbol Multicast (Ref.:13).

Para el análisis supondremos el caso de un árbol de distribución IP Multicast, sobre el cual una fuente esta transmitiendo un aplicación de vídeo conferencia a los miembros del grupo, y se presenta el caso de un nuevo miembro que quiere pasar a formar parte de este grupo Multicast.

QoSMIC construye un árbol de distribución utilizando una metodología *receiver-driven*, la cual esta compuesta fundamentalmente por tres fases:

Search Phase:

Se identifican los nodos, que forman parte de un árbol de distribución existente, que sean posibles candidatos a formar parte de las ramas del árbol de distribución. Esta fase es llevada a cabo a través de dos mecanismos: *Local Search* y *Multicast Tree Search*.

[13] Tomado Referencia Bibliografica 37

- *Local Search*: el nuevo nodo intenta alcanzar el árbol de distribución a través del procedimiento *Broadcast* limitado a sus nodos adyacentes.

- *Multicast Tree Search:* los *router*s interiores del árbol de distribución ejecutan un algoritmo distribuido para seleccionar los candidatos.

Bidding Phase:

Los nodos seleccionados como candidatos se presentan como tales al nuevo miembro, para lo cual envían mensajes *Bid*. Estos mensajes transportan información sobre las características de calidad de servicio del camino entre el nodo candidato y el nuevo nodo miembro.

Select Phase:

El nuevo nodo selecciona el camino que mejor cumpla con sus requerimientos de calidad de servicio.

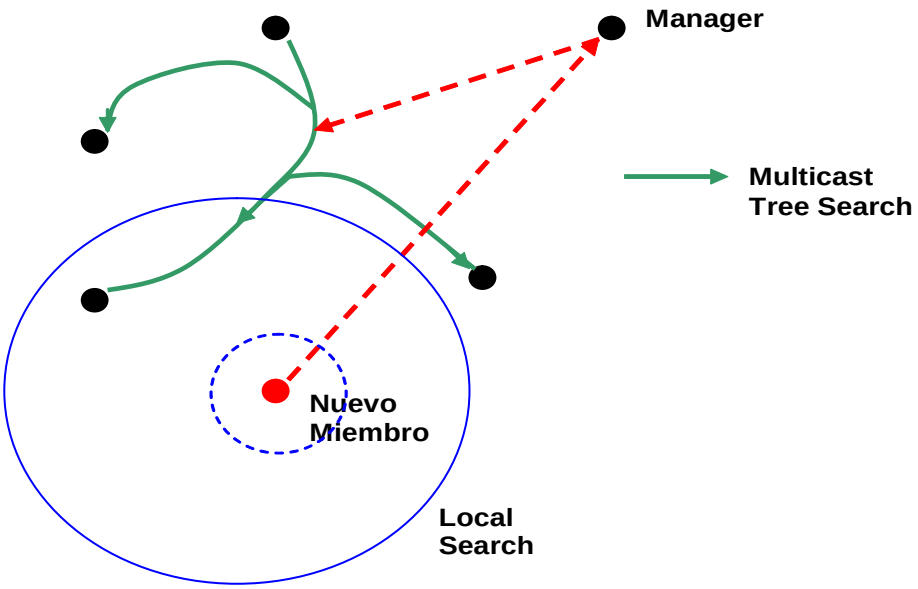


Figura 6.2.2.1.2.- Procedimientos QoSMIC Local Search y Multicast Tree Search (Ref.:14).

En QoSMIC un receptor puede cambiar de un árbol de distribución *Shared-Tree* a un árbol *Source-Based* para mejorar las condiciones de QoS o evitar congestiones. Las fases descriptas anteriormente aplican para ambos tipos de árboles de distribución.

6.2.2.2. DESCRIPCIÓN DE MESAJES Y MECANISMOS QoSMIC.

En este punto se presentarán los mecanismos y mensajes utilizados por QoSMIC. En las siguientes tablas se brinda un resumen las diferentes funciones desempeñadas por los nodos bajo este protocolo y los mensajes utilizados.

Nodo	Funciones
Manager	Supervisa el grupo y el funcionamiento de los procedimientos, no necesariamente participa en la distribución de los datos a través del árbol de distribución.
Candidato	Considerado como posible punto de unión ante el pedido de un nuevo miembro.
Designado	Se encuentra conectado a un conjunto de usuraios. Se encarga de recibir los pedidos de los usuarios e iniciar las busquedas que puedan ser necesarias.
Nuevo	Nodo al cual esta conectado un nuevo miembro o nuvo usuario.
Destination	Nodo que posee miembros activos.
In-Tree	Nodos que forman parte del árbol de distribución.
Intermediate	Nodos que forman parte del árbol de distribución y que no son nodos Source ni Destination.
Border	Nodo que conecta multiples dominios.

Tabla 6.2.2.2.1.- Funciones de los Nodos bajo el Protocolo QoSMIC (Ref.:15).

Mensaje	Descripción
Bid-Req	Busqueda local, *Local Search* , de un nuevo nodo miembro por nodos *In-Tree*
Bid	Los nodos candidates se identifican ellos mismos como tales ante el nuevo nodo miembro, transporta información de la calidad de servicio de los enlaces a través de los cuales ha "viajado".
M-Join	El nuevo nodo miembro se conecta con el nodo *Manager* , quien iniciará el proceso *Multicast Tree Search.*
Bid-Order	El nodo *Manager* indica a los nodos *In-Tree* que envíen mensajes *Bid* .
Join	El nuevo nodo miembro envía un mensaje *Join* a través del camino seleccionado, para establecer/actualizr el estado de ruteo.
Prune	Se utilizan para desactivar ramas del árbol de distribución.

Tabla 6.2.2.2.2.- Descripción de los Mensajes en QoSMIC (Ref.:16).

Cada nodo mantiene información sobre el ruteo Multicast, lo cual les permite "*forwardear*" cada paquete sobre el enlace apropiado. Cada enlace, el cual es parte de un árbol

[14] Tomado Referencia Bibliografica 37
[15] Tomado Referencia Bibliografica 37.

de distribución, tiene una entrada en la tabla Multicast. Así, si un paquete de entrada concuerda con una entrada en la tabla de ruteo Multicast, para un determinado enlace o interfase de salida en un nodo, este será "*forwardeado*" a través de dicho enlace o interfase. Las entradas en la tabla Multicast pueden corresponder tanto a un *Shared Tree* como a un *Source-Based Tree*.

6.2.2.3. MECANISMOS DE BUSQUEDA DE CANDIDATOS, SEARCH PHASE.

Como se presento en el punto 6.2.2.1, el protocolo QoSMIC tiene básicamente dos mecanismos para la búsqueda de candidatos:

- *Local Search.*
- *Multicast Tree Search.*

A continuación se describen dichos mecanismos de búsqueda:

Local Search:

a. El nuevo nodo miembro envía mensajes *Bid-Req*, en modo *flood*, a sus nodos adyacentes, o vecinos. El *Reversed Path Multicasting* es controlado mediante el campo *TTL*.

b. Cada nodo *In-Tree* que recibe un mensaje *Bid-Req* se transforma en un nodo candidato, y envía un mensaje *Bid*, en modo Unicast, hacia el nuevo nodo miembro. Este mensaje *Bid* colectará información sobre la calidad de servicio de los enlaces por los cuales atraviesa, llevando esta información hacia el nuevo nodo miembro. El nodo candidato considerará al nuevo nodo miembro como un dependiente tentativo, y no podrá dejar el árbol de distribución al menos que expire el estado de dependencia tentativa con el nuevo nodo miembro.

c. El nuevo nodo miembro recibirá todos los mensajes *Bid* de sus nodos adyacentes. El procedimiento terminará en forma no satisfactoria si no recibe respuestas de sus nodos adyacentes luego de expirado un *timer* establecido para dicho porposito. De lo contrario se entrará en la fase de establecimiento de la conexión.

Multicast Tree Search:

[16] Tomado Referencia Bibliografica 37.

a. El nuevo nodo miembro contacta al nodo *Manager* y este indica a algunos nodos *In-Tree* que se propongan como nodos candidatos.

b. El nuevo nodo miembro envía un mensaje *M-Join* hacia el nodo *Manager* del grupo Multicast.

c. El nodo *Manager* establece una sesión *bidding* enviando mensajes *Bid-Order*. Así, un grupo de los nodos que reciben este mensaje se proponen como candidatos.

d. Los nodos candidatos enviarán mensajes *Bid*, en forma Unicast, hacia el nuevo nodo miembro. Estos mensajes, al igual que en el mecanismo de búsqueda *Local Search*, colectarán información sobre la calidad de servicio de los enlaces por los cuales atraviesan.

Loop-Free Routing:

Para ambos mecanismos de búsqueda descriptos antreriormente, si un nodo *In-Tree* recibe mensajes *Bid* para un mismo árbol de distribución, el nodo toma el rol de candidato, desechando el mensaje *Bid* original e iniciando un nuevo mensaje *Bid*. Este procedmiento es denominado *Take-Over*, y garantiza árboles de distribución libres de Loops.

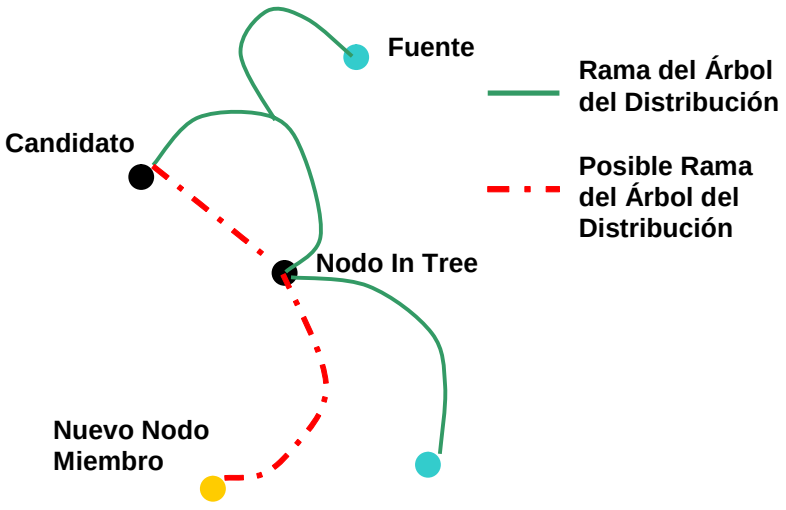


Figura 6.2.2.3.1.- Procedimiento de Take-Over (Ref.: 17).

6.2.2.4. ESTABLECIMIENTO DE LA CONEXIÓN.

[17] Tomado Referencia Bibliografica 37.

Este procedimiento comienza luego que finaliza la fase de *bidding* y se compone de los siguientes pasos:

a. El nuevo nodo miembro selecciona el mejor candidato de acuerdo a sus necesidades de calidad de servicio, basado en la información recibida a través de los mensajes *Bid*.

b. El nuevo nodo candidato enviará un mensaje *Join* hacia el nodo candidato que haya enviado el mensaje *Bid* con mejores condiciones de QoS, el cual recorrerá el camino seguido por el mensaje *Bid* pero en dirección contraria, y establecerá un estado de ruteo a lo largo del camino.

c. Cuando el nodo candidato seleccionado recibe el mensaje *Join* del nuevo nodo miembro, este comienza a enviar los paquetes de datos a través de la rama del árbol establecida.

d. Si un nodo *In-Tree* recibe un mensaje *Join*, este inicializa el procedimiento de *Take-Over*, termina el mensaje y comienza a "*forwardear*" datos sobre el nuevo camino establecido.

6.2.2.5. DEJAR UN GRUPO.

Un nodo Designado recibe un pedido de liberación a través del mismo protocolo por el cual comunico el mensaje de *Join*. El pedido de liberación puede ser para un grupo completo o para una fuente de un grupo. Cada vez que un nodo sensa un cambio en la membresía de un grupo, este quitará el enlace del árbol de distribución, y chequeará si este se transforma en un nodo terminal, o *leaf*, del árbol de distribución. Si es así, enviará un mensaje *Prune* hacia el árbol de distribución, y quitará el estado correspondiente a dicho nodo de su tabla de ruteo Multicast, de esta forma dejará de ser un nodo *In-Tree* del árbol de distribución.

6.2.2.6. SELECCIÓN DEL CANDIDATO EN MULTICAST TREE SEARCH.

Durante el procedimiento de búsqueda *Multicast Tree Search* el nodo *Manager* debe seleccionar un grupo de nodos *In-Tree* como candidatos. Esto lo puede realizar en forma centralizada o distribuida.

Selección Centralizada:

El nodo *Manager* elegirá los candidatos. Consideremos dos casos:

a. Si el nodo *Manager* tiene información global sobre la topología de la red, la selección del candidato puede realizarse en forma eficiente con mínimo *overhead*. Esto puede suceder si se esta utilizando un protocolo del tipo *Link State* o si el nodo *Manager* tiene acceso a una base de datos centralizada con información de ruteo.

b. Cuando QoSMIC se comporta como un protocolo *core-based*, forzando que el árbol de distribución sea ruteado hacia un único candidato.

Selección Distribuida:

En la ausencia de información centralizada, cada nodo *In-Tree* tendrá que decidir si se transforma en un nodo candidato utilizando un mecanismo distribuido. A continuación se presentan tres mecanismos distribuidos que pueden ser implementados en forma conjunta o separados.

a. *Directividad*: los nodos distantes son desalentados de transformarse en candidatos. Los mensajes *Bid-Order* llevan información sobre la mínima distancia entre el árbol de distribución y el nuevo nodo miembro. Así, un nodo con una distancia mayor a esta distancia mínima no se transformará en candidato, y si esta distancia excede un umbral, el mensaje *Bid-Order* no será ya "*forwardeado*".

b. *Mínimo Local:* en ausencia de información de ruteo global, los nodos localmente óptimos son seleccionados como candidatos. A medida que el mensaje Bid-Order viaja a lo largo de la rama de un árbol de distribución, la distancia al nuevo nodo miembro, de los últimos dos nodos, es incluida en el mensaje *Bid-Order*. Si el nodo $i+1$ identifica que el nodo *i* se encuentra "*más cercano*" al destino que los nodos $i+1$ e $i-1$, éste enviará un mensaje hacia el nodo *i* , el cual se transformará en el nodo candidato.

c. *Elección Fraccional:* en este mecanismo, se pueden elegir $\frac{1}{n}$ candidatos.

La elección de los diferentes mecanismos dependerá de la topología de red y del comportamiento del tráfico. Si se cuenta con información global sobre la topología de la red, el mecanismo de *Manager Selection* es la mejor opción, ya que reduce considerablemente el *overhead* de control. En caso de no contar con dicha información,

una buena opción sería la de implementar, en forma conjunta, los mecanismos *Directividad* y *Elección Fraccional*.

6.2.2.7. CONCLUSIONES.

- ⇨ Permite que los árboles de distribución *Shared Tree* igualen la *performance end-to-end* de los árboles de distribución *Source-Based Tree*.
- ⇨ El *overhead* es razonable y controlable.
- ⇨ El soporte de aplicaciones en Tiempo Real Interactivas no es tan eficiente como en el caso de AQoSM, ya que no provee garantías sobre la calidad de servicio de los enlaces o ramas del árbol de distribución.

6.2.3. ANÁLISIS DEL PROTOCOLO QMRP.

QMRP se plantea como un protocolo de ruteo IP Multicast que tiene en cuenta aspectos de QoS y que presenta una mayor escalabilidad que IP Multicast Nativo. Proporciona el soporte de requerimientos de QoS no aditivos, como ancho de banda y espacio en *buffers.*

Las características más importantes de este protocolo de ruteo se pueden resumir en:

- ♦ *Escalabilidad:* lo logra reduciendo el *overhead* en la construcción de los árboles de distribución. Intercambia entre ruteo *single-path* y ruteo *multiple-path* de acuerdo con las condiciones de la red, y sólo agrega, en forma incremental, caminos adicionales dentro de los procesos de búsqueda.
- ♦ *Requerimientos de QoS:* minimiza el *overhead* y maximiza la posibilidad de éxito, tal que los procesos de ruteo tienen la capacidad de manejar requerimientos de QoS.
- ♦ *Eficiencia:* detecta múltiples ramas de un árbol para un nuevo miembro, seleccionando la mejor de ellas según los requerimientos este último.
- ♦ *Robustes:* no se basa en un elemento extra de control, el proceso de ruteo es enteramente descentralizado.
- ♦ *Operatividad:* puede operar sobre protocolos Unicast ya existentes en la red.
- ♦ *Respuesta:* presenta un mecanismo que permite detectar la terminación del proceso de ruteo, ya sea por una falla o que haya terminado en forma satisfactoria.

- *Loop Free:* el protocolo construye árboles de distribución libres de Loops de ruteo.

A continuación se presentará el modelo de red utilizado y una descripción del protocolo

6.2.3.1. MODELO DE RED.

La red es modelizada como un conjunto de nodos interconectados por un conjunto de enlaces *full-duplex* asimétricos, teniendo en cuenta las siguientes consideraciones:

1. Existe un protocolo de ruteo Unicast el cual puede entregar un mensaje desde un nodo hacia cualquier otro nodo de la red.

2. Cada nodo mantiene un estado local actualizado, incluyendo la disponibilidad de recursos en cada interfase de red.

3. Los nodos no poseen información sobre ningún estado de red global ni del estado de ningún otro nodo.

4. Si se utiliza un árbol de distribución *shared-tree*, un nuevo miembro tendrá la capacidad de mapear una dirección de grupo IP Multicast, consultando al nodo de *core* del árbol de distribución a demanda, posiblemente a través de una sesión de *query/response*.

6.2.3.2. DESCRIPCIÓN DEL PROTOCOLO.

El protocolo comienza con la elección de un único árbol de distribución, pero en caso de ser necesario, puede expandir la búsqueda separándola en uno o más puntos de manera controlada. Estos puntos son seleccionados según las condiciones de la red.

Puede trabajar como un protocolo *intra-domain* o *inter-domain*, pudiéndose utilizar para construir tanto *shared-trees* como *sender-based trees*.

A continuación se presentará una descripción del funcionamiento del protocolo:

Consideraremos el caso de un nuevo miembro que se une a un árbol Multicast en el cual una fuente se encuentra transmitiendo los contenidos de una aplicación de vídeo conferencia entre los miembros del mismo.

Cuando un nuevo miembro se une a un grupo IP Multicast, este obtiene la dirección IP del *core* del árbol de distribución a través de una sesión *inquiring directory*. Luego, inicia el proceso de ruteo enviando un mensaje de *Request* hacia el nodo de *core* a través de un camino Unicast. Se definen dos modos de búsqueda:

- *Single Path Mode*.
- *Multiple Path Mode*.

El proceso de ruteo comienza con el modo *Single Path*, intentando buscar sólo el camino de ruteo Unicast recorrido por el mensaje *Request*.

El mensaje *Request* presenta las siguientes funciones:

- Transporta los requerimientos de calidad de servicio.
- Chequea la disponibilidad de recursos de cada nodo intermedio, sólo prosigue cuando el nodo cumple con las condiciones de QoS requeridas.

Si todos los nodos intermedios tienen recursos, QMRP se comporta de forma similar al protocolo PIM-SM, y encuentra una nueva rama del árbol de distribución recorriendo un único camino.

Si alguno de los nodos intermedios no cumple con los requerimientos de calidad de serivicio solicitados, QMRP activa el modo *Multiple Path*, enviando un mensaje *NACK* hacia el nodo previo (anterior al nodo que no cumple con los requisitos de QoS). Al recibir el mensaje de *NACK*, este nodo enviará un mensaje *Request* hacia todos sus nodos *"vecinos"*, salvo sobre las interfases sobre las cuales recibió los mensajes de *Request* y *NACK*. Así, cada mensaje *Request* buscará su propio sub-camino. Una vez que se encuentra una rama del árbol factible, se envía un mensaje *ACK* hacia el nuevo miembro.

En la siguiente figura se mostrará con un ejemplo como funciona el protocolo cuando un nuevo miembro, *t*, quiere unirse a un árbol de distribución existente, y uno de los nodos, *j*, no cumple con las condiciones de calidad de servicio:

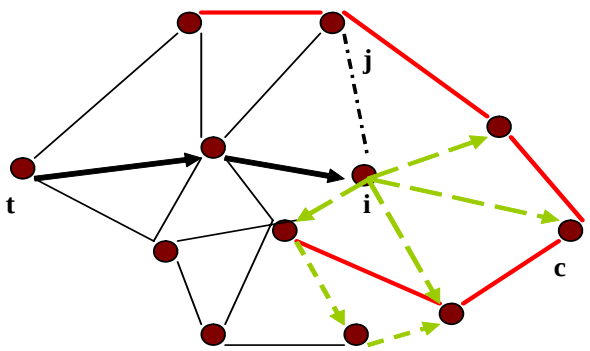


Figura 6.2.3.2.1.- Ejemplo QMRP (Ref: 18).

6.2.3.2.1. ESTADOS DE QMRP.

El protocolo QMRP implementa un mecanismo de cinco estados en cada nodo, donde el comportamiento de cada nodo dependerá del estado en que este se encuentre, pudiendo cambiar su estado luego de recibir un mensaje de control. En la siguiente figura se muestra el comportamiento de un nodo en cada uno de los estados de QMRP, y cuando un nodo cambia de estado. Mientras un nodo incrementa su propia maquina de estados, el comportamiento colectivo de todos los nodos da forma al proceso de ruteo.

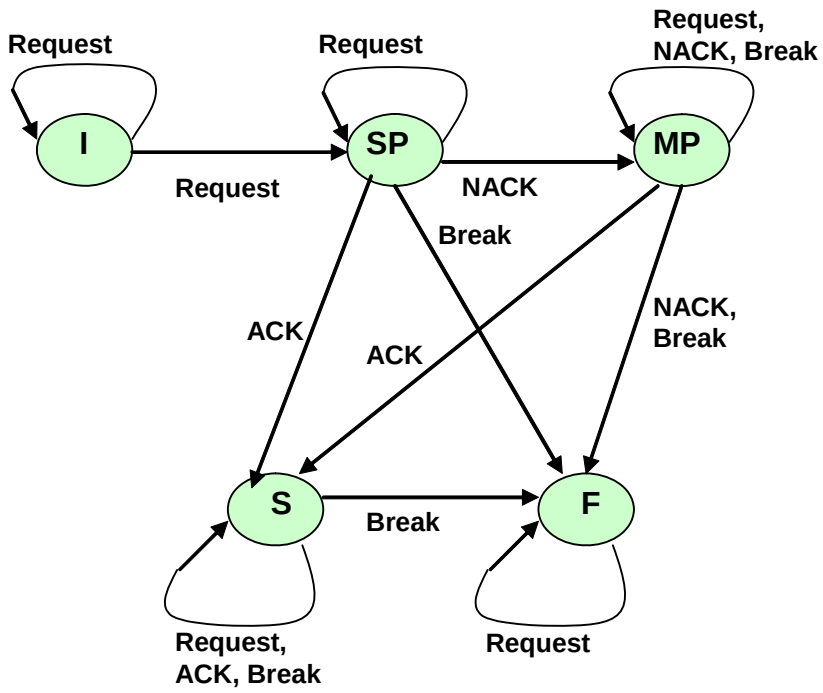


Figura 6.2.3.2.1.1.- Transición de Estados en QMRP (Ref: 19).

Los cinco estados representados en la figura anterior son:

a. ***I***: estado inicial.
b. ***SP***: estado *Single Path*.
c. ***MP***: estado *Multiple Path*.
d. ***F***: estado de falla.
e. ***S***: estado de éxito.

Los mensajes se definen como:

18 Tomado Referencia Bibliografica 35
19 Tomado Referencia Bibliografica 35.

- *Request*: posibilitan el procedimiento de búsqueda.

- *NACK*: detiene la búsqueda del árbol de distribución, y a la vez inicializa el estado *MP* en el nodo receptor, lo cual subsecuentemente amplía la búsqueda.

- *Break:* detiene la búsqueda del árbol de distribución, sin inicializar el estado *MP* en el nodo receptor.

- *ACK:* transforma una rama de la búsqueda del árbol de distribución como una parte del árbol Multicast. También transporta las propiedades de QoS del camino (como: *delay*, ancho de banda, etc).

El protocolo presenta dos árboles de distribución, con sus correspondientes entradas en las tablas de ruteo de los nodos:

a. *Search Tree*: se almacena en forma temporaria como entradas en las tablas de ruteo de los nodos que reciben el mensaje *Request*.

b. *Multicast Tree*: se almacena en las tablas de ruteo Multicast de los nodos que forman el árbol de distribución.

Inicialmente un nuevo miembro se encuentra en el estado *SP*, los nodos del árbol de distribución se encuentran el estado *S*, mientras que el resto de los nodos estarán en el estado *I*.

Supongamos que un nodo *i* recibe un mensaje *Request* a través de su interfase de red *l*, y supongamos que el nodo *j* es el *next hop* del nodo *i*. En la siguiente tabla se muestra el pseudo código con el comportamiento del nodo *i* en cada estado y como cambia de estado.

Estado	Pseudocodigo
Estado Inicial (I)	**switch** (the received message) **case** REQUEST(*id*): **if** (*i* has the required resources) create a routing entry R {in, out, *id*, β} R.in := *l* R.out := { *j*} R.β := SP /*change to the SP state*/ send REQUEST (*id*) to *j* **else** return NACK (*id*) to *l* **otherwise:**

	discard the received message
Single Path (SP)	**switch** (the received message) **case** REQUEST(*id*): return NACK (*id*) to **case** NACK (*id*) : R.β := MP R.out := 0 **for** every network interface *l`* **do** **if** ( *l`* ≠ *l* and *l`* ≠ R.in) R.out := R.out + { *l ´* } send REQUEST (*id*) to *l ´* **case** BREAK (*id*) : R.β := F send BREAK (*id*) to R.in **case** ACK (*id*) : /* join multicast tree, which changes to the S state automatically */ create multicast entry M { G, in, out, *prop*} M.in := *l* M.out := {R.in} M.prop := ACK.prop send ACK (*id*) to R.in **for** every of *R´* {in´, out´, id´, β´} of G **do** M.out := M.out + {*R ´. l ´* } send ACK (*id*) to *R ´.* in´

Multiple Path (MP)	**switch** (the received message)
	case REQUEST (*id*) :
	return NACK (*id*) to *l*
	case NACK (*id*) :
	R.out := R.out - { *l* }
	If (R.out = 0)
	R.β := F
	if (BREAK (*id*) was received previously)
	send BREAK (*id*) to R.in
	else
	send NACK (*id*) to R.in
	case BREAK (*id*) :
	R.out := R.out - { *l* }
	If (R.out = 0)
	R.β := F
	send BREAK (*id*) to R.in
	case ACK (*id*) :
	create a multicast entry M { G, in, out, *prop*}
	M.in := *l*
	M.out := {R.in}
	M.prop := ACK.prop
	send ACK (*id*) to R.in
	for every of *R´* {in´, out´, id´, β´} of G **do**
	M.out := M.out + {*R ´. l ´* }
	send ACK (*id*) to *R ´.* in´

	switch (the received message) **case** REQUEST (*id*) :

Failure (F)	return NACK to **otherwise:** discard the received message
Success (S)	**switch** (the received message) **case** REQUEST (*id*) : return ACK (*id*) to *l* **case** BREAK (*id*) : M.out := M.out - { *l* } **If (M**.out = 0) remove the multicast entry **case** ACK (*id*) : **if** (ACK.prop is better than *M*.prop) send BREAK (*id*) to *M*.in M.in := *l* **else** send BREAK (*id*) to **otherwise;** discard the received message

Tabla 6.2.3.2.1.1.- Pseudo Código de Estados del Protocolo QMRP (Ref: 20).

En las siguientes figuras se muestra en forma gráfica el mecanismo de selección de camino, empleado por el protocolo QMRP, para el caso en que un nuevo miembro requiera unirse a un grupo Multicast cuyo árbol de distribución se encuentra previamente establecido.

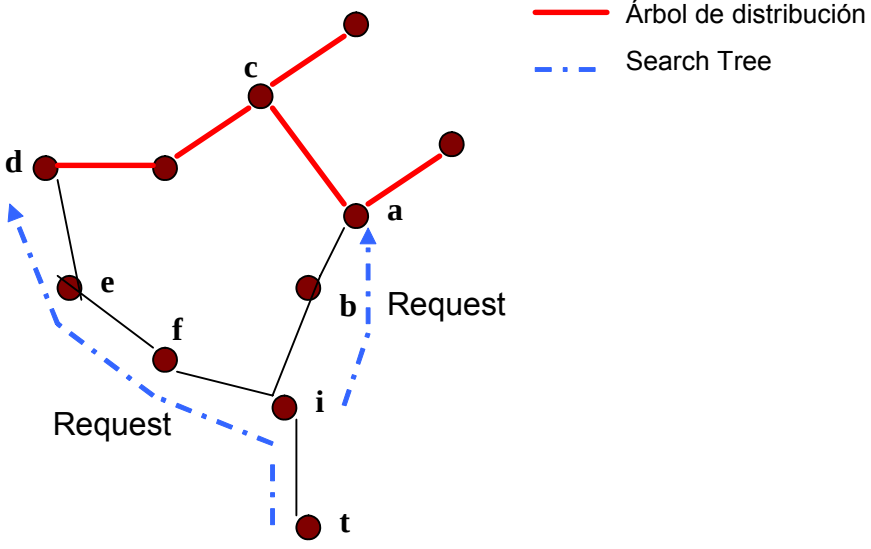


Figura 6.2.3.2.1.2.a.- Selección de Camino en QMRP (Ref: 21) .

En la figura anterior, se considera que ambas ramas del *search tree*, que se tienen luego del nodo *i*, son factibles. Estas alcanzan al árbol de distribución en los nodos *a* y *d* respectivamente. Para el ejemplo se supone que la información transportada en los mensajes

[20] Tomado Referencia Bibliografica 35.
[21] Tomado Referencia Bibliografica 35.

ACK.prop es la cantidad de saltos, *hops*. Supongamos que el mensaje *ACK* proveniente de *d* arriba primero al nodo *i*. Así, como se muestra en la siguiente figura, la rama compuesta por $d \rightarrow e \rightarrow f \rightarrow i \rightarrow t$, se transforma en una nueva rama del ábol de distribución.

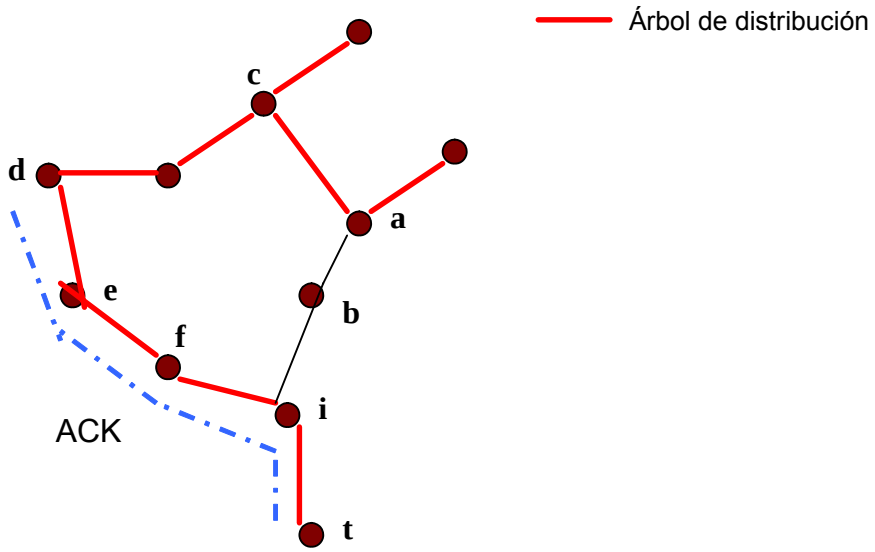


Figura 6.2.3.2.1.2.b.- Selección de Camino en QMRP (Ref: 22).

En la siguiente figura, se muestra el caso en que el *ACK* desde el nodo *a* llega al nodo *i* luego de haberse recibido el *ACK* del nodo *d*. Luego, la rama $a \rightarrow b \rightarrow i$ se vuelve parte del árbol de distribución, y además presenta una mejor métrica que la rama $d \rightarrow e \rightarrow f \rightarrow i \rightarrow t$.

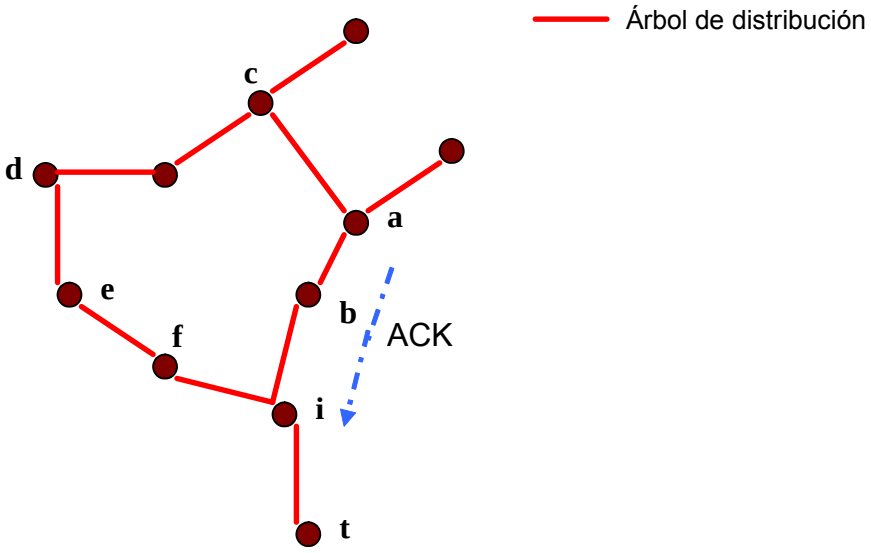


Figura 6.2.3.2.1.2.c.- Selección de Camino en QMRP (Ref: 23).

[22] Tomado Referencia Bibliografica 35.
[23] Tomado Referencia Bibliografica 35

Finalmente, en la siguiente figura se muestra que el nodo *i* envia un mensaje *BREAK* tal que la rama $d \rightarrow e \rightarrow f \rightarrow i$ deja de ser parte del árbol de distribución, quedando la rama $a \rightarrow b \rightarrow i \rightarrow t$, como el camino que une al nuevo miembro *t* con el árbol de distribución.

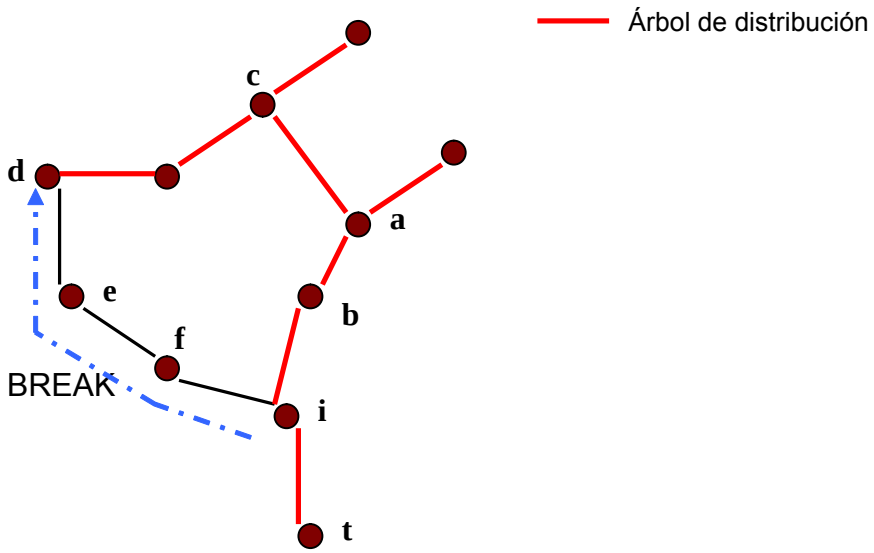


Figura 6.2.3.2.1.2.d.- Selección de Camino en QMRP (Ref: 24).

De lo anterior se puede ver que un *search tree* se forma de manera incremental, donde cada nodo debe poseer los recursos requeridos.

6.2.3.3. QMRP-m.

Se puede observar, a partir de lo presentado en los puntos anteriores, que de no limitar el mecanismo de búsqueda de QMRP este puede conducir a *overhead* elevados, debido a que potencialmente puede ocacionar búsquedas *search tree* muy extensas. Para evitar esto, se definieron los siguientes parámetros:

MBL: Maximum Brandching Level.

Si la cantidad de nodos que entran en el estado *MP* es muy elevada, el *overhead* resultante será muy grande. Por lo cual, si consideramos que entre un nuevo miembro *m* y cada nodo perteneciente a un *search tree*, existen al menos *m* nodos entrando al estado *MP*, el máximo número de nodos permitidos para entrar al estado *MP* se puede definir como:

[24] Tomado Referencia Bibliografica 35

$\sum_{i=0}^{m-1}(d-2)^i = \frac{(d-2)^m - 1}{d-3}$, donde *d* representa el grado máximo del nodo.

A esto se lo denomina QMRP-m, donde *m* representa el *Maximum Branching Level*. En la siguiente figura se presenta un ejemplo de QMRP-2.

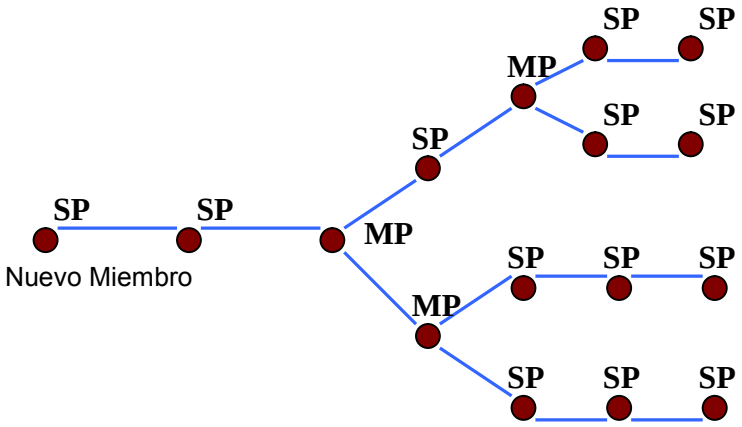


Figura 6.2.3.3.1.- Search Tree en QMRP-2 (Ref.: 25).

El *MBL* puede ser implementado parametrizando la cantidad de mensajes de ruteo a través de un contador.

MBD: Maximum Brandching Degree.

Otro parámetro que se utiliza para limitar el mecanismo de búsqueda de QMRP es el *MDB*, *Maximun Brandching Degree.*

Un nodo que entra en el estado *MP* puede tener un gran número de nodos adyacentes, lo cual puede causar un *overhead* excesivo. Por lo cual el parámetro *MDB* permite especificar el máximo número de mensajes de *Request* que pueden ser enviados por un nodo en el estado *MP*. Así, si el grado máximo de ramificaciones *x* es mayor que el grado del nodo menos dos, el nodo seleccionará en forma aleatoria, o basada en la distancia en saltos al nodo de *core*, una cantidad *x* de interfases de salida desde las cuales no se han recibido mensajes de *Request* o *NACK*, y enviará mensajes *Request* a través de estas interfases. Se debe tener cuidado al setear el valor del *MDB*, ya que un valor elevado producirá un *overhead* considerable.

Así, de lo anterior se define que considerando:

[25] Tomado Referencia Bibliografica 35.

a. $MBL = m$

b. $MDB = x$

se tiene que el máximo número de nodos permitidos para entrar al estado *MP* se define como:

$$\sum_{i=0}^{m-1} x^i = \frac{x^m - 1}{x - 1}$$

6.2.3.4. PERFORMANCE DE QMRP Y COMPARACIÓN CON QoSMIC.

Comenzamos definiendo dos parámetros que se utilizarán para evaluar la *performance* del protocolo:

a. $\text{Tasa de Exito} = \frac{\text{Cantidad de Nuevos Miembros Aceptados}}{\text{Cantidad Total de Mensajes Join Request}}$.

b. $\text{Overhead Promedio de Mensajes} = \frac{\text{Número Total de Mensajes Enviados}}{\text{Número Total de Mensajes Join Request}}$.

El valor máximo del parámetro *Brandching Degree*, en QMRP-m, es de 10. Un nodo con un grado mayor que 12, que entra en el estado *MP*, limitará el envío de mensajes *Request* a sólo 10 de sus nodos adyacentes.

Para el caso de QoSMIC se considerará que los procesos de *Local Search* y *Tree Search* son implementados en forma secuencial, donde el procedimiento *Tree Search* sólo se implementará en caso que falle el primero. Con esto, se reducirá el *overhead* del protocolo, pero se corre el riesgo de que se incremente el *delay*. También se considerará la implementación de los parámetros *directividad*, *mínimo local* y *elección fraccional*.

A continuación se presenta la comparación del protcolo QMRP-m con respecto a QoSMIC:

En la comparación se considera una topología de red, de 600 nodos, basada en *power-law-network*, la cual considera que los grados de los nodos de red en Internet obedecen la siguiente ley logarítmica:

- La mayoría de los nodos tienen grados pequeños, y un pequeño número de nodos tienen un grado elevado.

- Cuando el grado de los nodos disminuye, la cantidad de nodos con ese grado decrece exponencialmente.

En la simulación se considerará que:

a. Se tiene una fuente generando tráfico Multicast a tráves de una aplicación de vídeo conferencia en Tiempo Real en forma Interactiva.

b. Que el árbol de distribución se genera en forma aleatoria.

c. Se toma un nuevo nodo miembro, fuera del árbol de distribución, seleccionado en forma aleatoria.

d. Se toman tres tamaños distintos de árboles de distribución, con 6, 45 y 180 nodos, incluyendo nodos internos que no forman parte del árbol de distribución.

e. El estado de cada enlace directo es generado en forma aleatoria, ya sea que cumplan o no con los requerimientos de calidad de servicio.

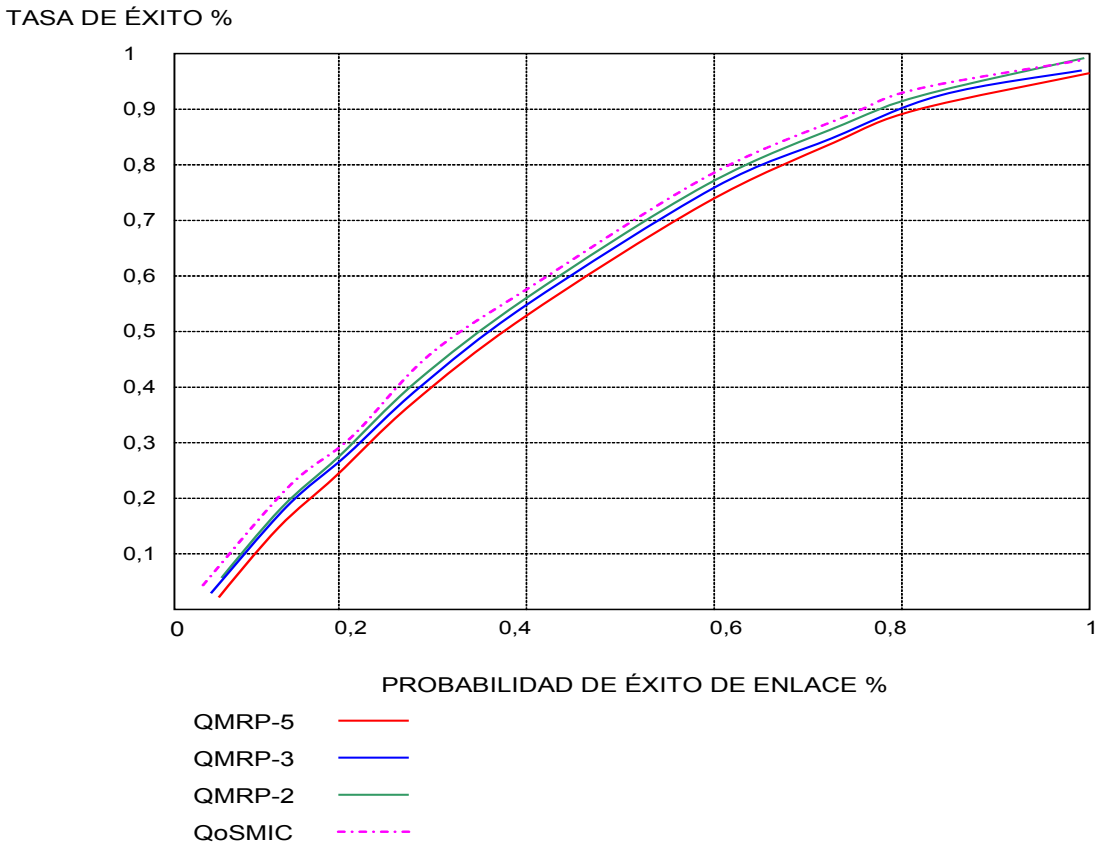
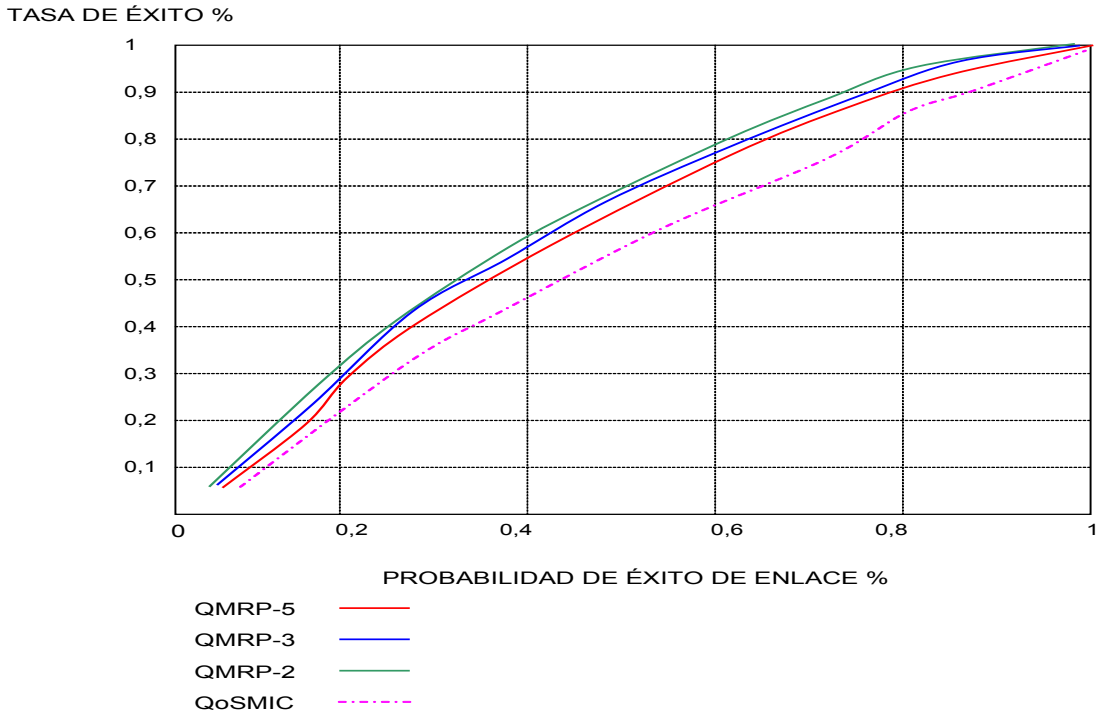


Figura 6.2.3.4.1.a.- Tasas de Éxito considerando un Árbol de Distribución de 180 Nodos (Ref.:26)



26 Tomado Referencia Bibliografica 35.

Figura 6.2.3.4.1.b.- Tasas de Éxito considerando un Árbol de Distribución de 45 Nodos (Ref.:27)

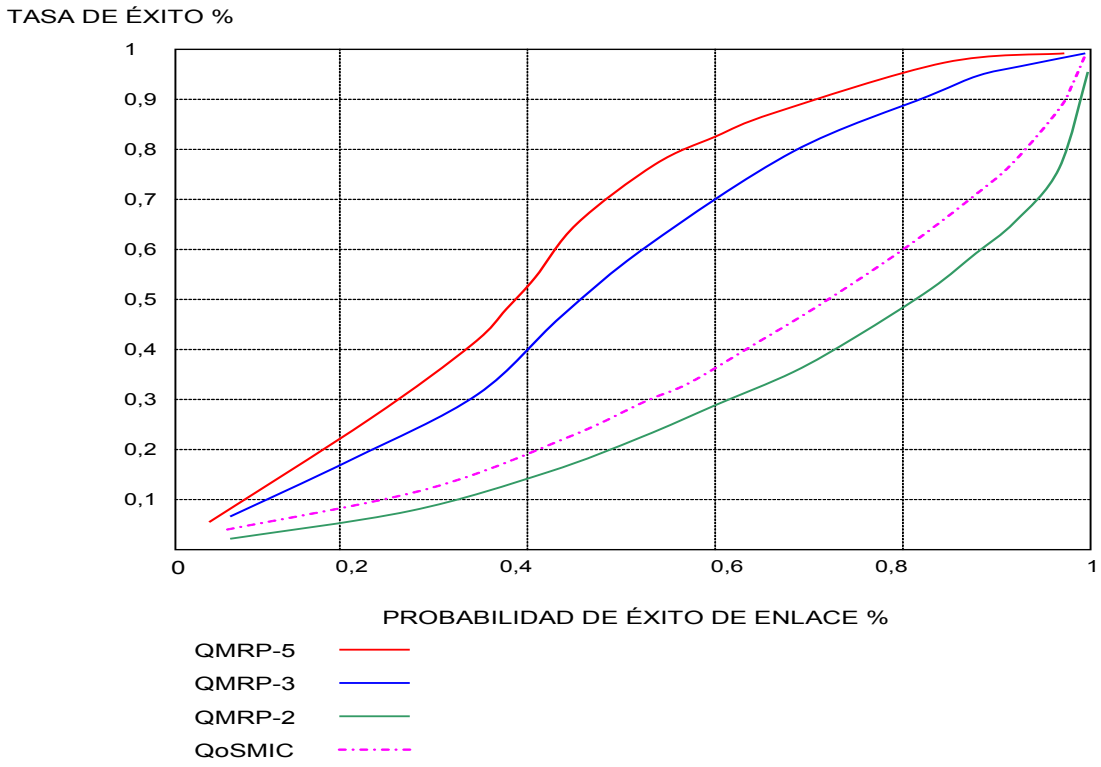


Figura 6.2.3.4.1.c.- Tasas de Éxito considerando un Árbol de Distribución de 6 Nodos (Ref.:28).

En las figuras anteriores puede verse que:

- Las tasas de éxito de las variantes de QMRP-m simuladas son mejores que las que se pueden logran con QoSMIC. Las diferencias son más pronunciadas cuanto menor sea la cantidad de miembros por grupo. Esto se debe a que cuanto menor es el número de nodos en un árbol de distribución Multicast, QoSMIC sólo puede seleccionar un pequeño número de caminos candidatos, lo cual limita su capacidad de encontrar un camino que resulte factible según los requerimientos de calidad de servicio de los nuevos miembros.

- QMRP-m es independiente de la cantidad de nodos de un árbol para encontrar el camino adecuado. La elección del parámetro *MB*L, en QMRP-m, permite tener un balance entre el *overhead* producido por los mensajes y la Tasa de Éxito. Un valor de $m = 2$ presenta la elección más adecuada para lograr este balance.

27 Tomado Referencia Bibliografica 35.
28 Tomado Referencia Bibliografica 35.

En las siguientes figuras se grafica el *overhead* de mensajes con respecto a la probabilidad de éxito para los casos considerados anteriormente.

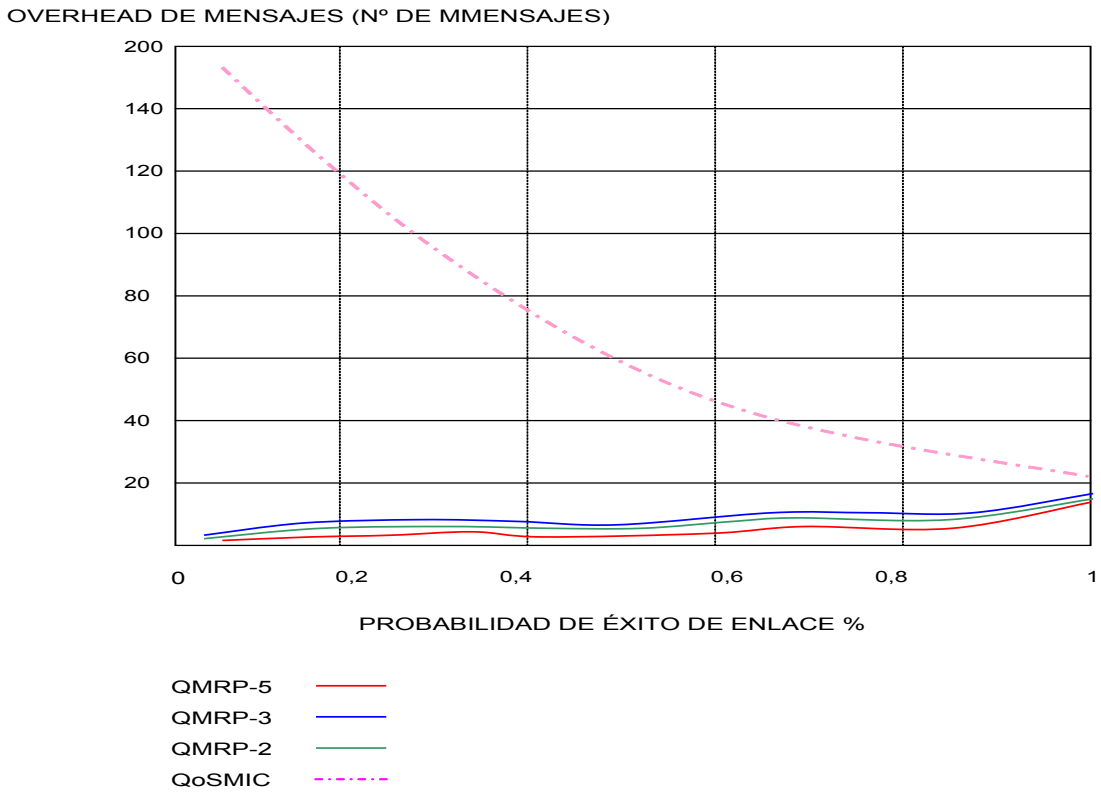


Figura 6.2.3.4.2.a.- Overhead de Mensajes considerando un Árbol de Distribución de 6 Nodos (Ref.:29).

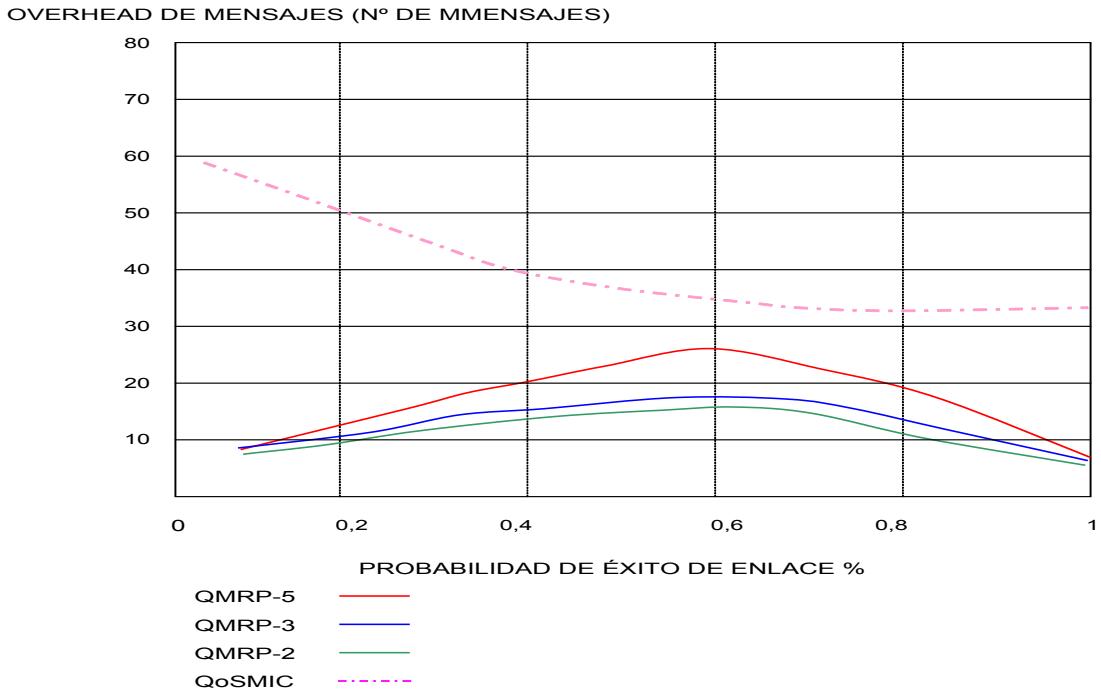


Figura 6.2.3.4.2.b.- Overhead de Mensajes considerando un Árbol de Distribución de 45 Nodos (Ref.:30).

29 Tomado Referencia Bibliografica 35.

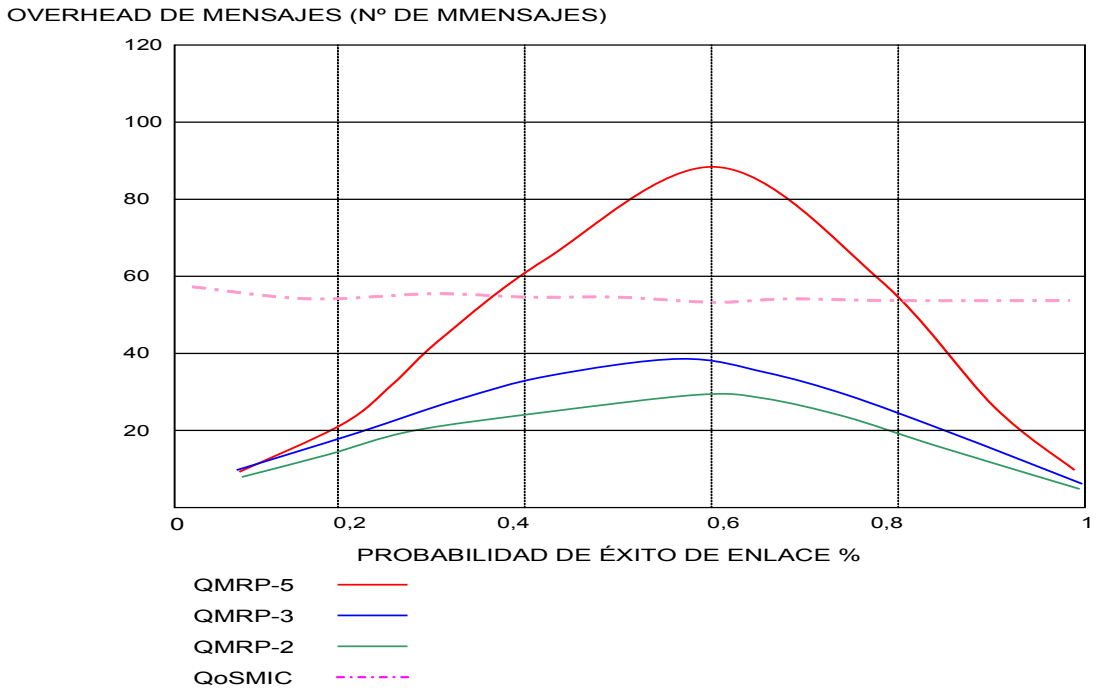


Figura 6.2.3.4.2.c.- Overhead de Mensajes considerando un Árbol de Distribución de 6 Nodos (Ref.:31).

De las figuras anteriores se puede ver que:

- Para QoSMIC el *overhead* es elevado cuanto mayor es la cantidad de miembros en un grupo Multicast, mientras la probabilidad de éxito disminuye con la cantidad de miembros por grupo Multicast.

- QMRP-3 y QMRP-2 presentan mucho menor *overhead* que QoSMIC. Así, el *overhead* dependerá del valor adoptado para el parámetro *MBL*; cuanto mayor sea el *m* seleccionado, mayor será el *overhead*.

6.2.3.5. CONCLUSIONES.

⇨ Presenta un buen rendimiento en término de probabilidad de éxito y bajo *overhead* de mensajes.

30 Tomado Referencia Bibliografica 35.
31 Tomado Referencia Bibliografica 35.

- ⇨ QMRP-m aumenta la tasa de éxito para encontrar una ruta, ya que examina múltiples caminos que pueden ser hasta *m* saltos más largos que el camino más corto (*Shortest Path Tree*).

- ⇨ Cuando existen muchos árboles de distribución en la misma región de la red, presenta el problema denominado *cross-efect*, produciendo una elevada utilización de los elementos de la red.

- ⇨ Presenta un buen balance entre el *overhead* de ruteo y probabilidad de éxito.

- ⇨ Otra desventaja es que almacena estados temporariamente en los *router*s por cada mensaje *Join Request*s. Lo deseable es que los *router*s sólo mantengan información por grupo Multicast, y no información por grupo y por mensajes *join*.

- ⇨ Fue diseñado para aplicaciones con requerimientos de QoS no aditivos como ancho de banda y espacio en *buffer*. No posee mecanismos para manejar requerimientos de QoS aditivos como el *delay*, con lo que no es aconsejable para ser utilizado con aplicaciones de tiempo real interactivas.

6.2.4. ANÁLISIS DEL PROTOCOLO SoMR.

Las características más importantes de este protocolo de ruteo se pueden resumir en:

- ♦ Es un protocolo de ruteo Multicast que permite tener en cuenta condiciones de QoS.

- ♦ Presenta un muy bajo *overhead* de comunicación y no requiere de estados fuera de Multicast para realizar el proceso de ruteo.

- ♦ Logra un balance entre un buen rendimiento del proceso de ruteo y *overhead* mediante una selección cuidadosa del sub-gráfico de red, sobre el cual realiza la búsqueda de un camino que cumpla con los requerimientos de QoS, autorregulando la búsqueda según las condiciones de la red.

- ♦ Minimiza los requerimientos del sistema, dependiendo sólo de los estados locales almacenados en los nodos de la red. El proceso de ruteo es totalmente descentralizado.

A continuación se presentará el modelo de red utilizado y una descripción del protocolo

6.2.4.1. MODELO DE RED.

Tengamos en cuenta las siguientes consideraciones:

a. Existe un protocolo de ruteo Unicast el cual puede entregar un mensaje entre dos nodos interconectados en la red. Cada nodo tiene información de la cantidad de saltos que lo unen con otro nodo a lo largo del camino Unicast.

b. Cada nodo mantiene la información actualizada de su estado.

c. Cualquier nuevo miembro es capaz de "mapear", a demanda, una dirección de grupo Multicast hacia el nodo raíz del árbol de distribución, posiblemente a través de una sesión de *query/response*.

d. Tenemos una fuente Multicast que se encuentra emitiendo tráfico generado por una aplicación de vídeo conferencia en Tiempo Real.

Supongamos que *k* e *i* son dos nodos interiores del árbol de distribución Multicast.

Sea $P_{k,i}$ el camino dentro del árbol de distribución que une los nodos *k* e *i*. El *delay* garantizado en este camino se denomina *in-tree delay*, y esta representado por $\text{delay}(P_{k,i})$.

Sea *T* el conjunto de nodos interiores del árbol de distribución, y *r* el nodo raíz del mismo.

Un árbol de distribución que satisfaga los requerimientos de *delay* cumple con:

$$\forall i \in T, \text{delay}(P_{k,i}) \leq D$$

Se asume que cada enlace (i,j) puede asegurar cierto *delay*, para una dada clase de servicio, CoS, con la cual esta asociada el grupo Multicast. A esto se lo representa por:

$\text{delay}(i,j)$.

6.2.4.2. DESCRIPCIÓN DEL PROTOCOLO.

Consiste en dos fases:

Primera Fase:

La primer fase es similar a *SPR*, *Shortest Path Routing*, en la cual un nuevo miembro, *t*, envía un mensaje *Join* al nodo raíz, *r*, a través un camino Unicast. Este mensaje transporta información sobre el *delay* del camino que atraviesa.

Cuando el mensaje *Join* llega a un nodo interior, *k*, del árbol de distribución, si el *delay* acumulado cumple con los requerimientos, el camino se selecciona como una rama factible del árbol de distribución.

Luego, se envía un mensaje *Construction*, en forma inversa, a lo largo del camino Unicast, para construir una nueva rama del árbol hacia el nuevo miembro del grupo Multicast.

Segunda Fase:

Se activa en caso que en la primera fase el nuevo miembro no pueda unirse al grupo Multicast, ya que no se ha podido encontrar un camino que lo una al árbol de distribución cumpliendo con los requerimientos de QoS.

Por lo cual, en este caso, el mensaje de *Join* seguirá su camino hasta alcanzar al nodo raíz *r*. Cuando el nodo *r* recibe el mensaje de *Join* inicia la segunda fase, la cual emplea un procedimiento de ruteo *Multiple-Path*.

Así, *r* enviará mensajes *Grow* hacia sus nodos vecinos. Estos mensajes *Grow* viajarán a través de caminos Unicast hacia el nuevo miembro. A medida que estos mensajes progresan a lo largo de los caminos, intentarán construir nuevas ramas del árbol.

Cada mensaje *Grow* transporta información sobre el *delay* requerido, *D*, además de la información del *delay* acumulado a lo largo del camino. Por lo tanto, cuando un nodo intermedio, *i*, recibe un mensaje *Grow*, tendrá acceso a la información sobre el *delay* desde *r* a *i*, representado por $\text{delay}(P_{r,i})$.

A continuación se describirá el proceso que se tiene cuando un nodo intermedio *i* recibe un mensaje *Grow*:

Primero, *i* realizará una prueba *EW*, *Early Warning*, para verificar si el *delay* acumulado a lo largo del camino Unicast satisface el *delay* requerido, *D*. Si el resultado de la prueba *EW* resulta satisfactorio, el proceso de ruteo Unicast continúa hacia el nuevo miembro *t*. De no ser así, el nodo *i* iniciará un proceso de ruteo *Multiple-Path*.

Consideremos que *j* es el nodo vecino de *i* a lo largo de un camino Unicast, en el caso en que *i* inicio el proceso de ruteo *Multiple-Path*. La prueba *EW* tiene en cuenta los siguientes parámetros:

a. *D*: *delay* requerido.

b. delay $(P_{r,i})$: *delay* a lo largo de un camino Unicast entre el nodo raíz *r* y el nodo intermedio *i*.

c. delay(i, j) : *delay* que presenta el enlace entre los nodos intermedios *i* y *j*.

d. *l*: longitud del camino Unicast entre el nodo intermedio *i* y el nuevo miembro *t*.

Teniendo en cuenta lo anterior, el proceso *EW* se puede definir como:

if delay(i, j) > (D − delay(Pr,i))/l
then warning
else pass

donde:

a. $D - \text{delay}(P_{r,i})$: da el *delay* permitido para el resto de la construcción del árbol de distribución, en función del *delay* requerido *D*.

b. $\frac{D - \text{delay}(P_{r,i})}{l}$: da la proporción del *delay* restante, o permitido, para cada enlace que compone el camino entre el nodo intermedio *i* y el nuevo miembro *t*.

Si la prueba *EW* se pasa satisfactoriamente, lo cual significa que el camino encontrado es probable que satisfaga las condiciones de QoS requeridas, el nodo intermedio *i* se agregará al enlace (i, j) dentro del árbol de distribución y enviará el mensaje *Grow* hacia su nodo adyacente *j*. Así, si cada nodo intermedio pasa en forma satisfactoria la prueba *EW*, se establece una nueva rama del árbol para el nuevo miembro *t*.

Si, en cambio, la prueba *EW* alerta que el camino puede violar los requerimientos de QoS, el nodo intermedio *i* se transformará en un *brandching point*. Así, mensajes *Grow*

serán enviados hacia un sub-conjunto de nodos adyacentes, *x*, que satisfagan la siguiente prueba de QoS:

if delay(i, x) > D − delay(Pr,i)
then fail
else pass

Si la prueba anterior arroja resultados negativos para cada uno de los nodos adyacentes, se enviará un mesaje *Break*, hacia atrás, de modo que se "corte" la rama del árbol que había sido parcialmente construida. Así, cuando un nodo intermedio *k* recibe un mensaje *Break* de un nodo intermedio *i*, este borrará el enlace (k,i) del árbol Multicast, y en caso que el nodo *k* se transforme en un nodo Terminal del árbol de distribución, y no sea un miembro del grupo Multicast, este se borrará del árbol de distribución Multicast y propagará el mensaje *Break* hacia sus nodos vecinos. Finalmente, cuando el mensaje *Break* llega al nodo raiz *r*, la rama es totalmente borrada del árbol de distribución.

Siempre que un mensaje *Grow* llegue al nuevo miembro *t*, una rama factible será establecida. El nodo *t* puede recibir múltiples mensajes *Grow*, provenientes de diferentes caminos o ramas. En este caso, éste enviará hacia atrás mensajes *Break* para desactivar las ramas que no cumplan con los requerimientos de QoS, para lo cual utilizará la información de calidad de servicio (como *delay*, ancho de banda, etc) que es transportada por los mensajes *Grow*.

Si bien, de lo anterior, se puede ver que un único mensaje de *Join* puede resultar en múltiples ramas temporarias, un punto importante que se debe notar es que no importa la cantidad de mensajes *Join* simultáneos que se puedan tener, cada nodo mantiene

sólo una entrada Multicast por grupo, y no mantiene información de cada mensaje *Join* en particular.

Evitando Loops de Ruteo:

En el proceso de construcción de las ramas del árbol de distribución hacia el nuevo miembro pueden producirse Loops de ruteo.

Consideremos un mensaje *Grow* que construye una rama de un árbol de distribución a lo largo de un camino Unicast *P* hacia un nuevo miembro *t*. Algunos de los enlaces de *P* pueden ya ser parte del árbol de distribución mientras que otros no.

Consideremos que a un mensaje *Grow* que viaja a través de un enlace que forma parte de un árbol de distribución le asignamos un color Azul. Y, a los mensajes que viajan por enlaces que no son parte de un árbol de distribución, le asignamos el color Verde. Sólo estos últimos pueden formar Loops ya que unen nuevos enlaces al árbol de distribución.

Un nodo que envia un mensaje *Grow* puede determinar el "color" del mensaje de la siguiente forma:

- Cuando un nodo *i* envía un mensaje *Grow* a un nodo adyacente *j*, si el nodo *j* es el "padre" o el "hijo" del nodo *i* dentro de un árbol de distribución Multicast, *i* marcará al mensaje con el "color" Azul, de lo contrario lo marcará con "color" Verde.

Basado en el esquema de colores anterior resulta fácil realizar la detección de Loops de ruteo. Cuando un nodo interior recibe un mensaje *Grow Verde* se tendrá la situación en la cual se ha formado un Loop de ruteo. En la siguiente figura se muestra como actua el mecanismo de detección de Loops.

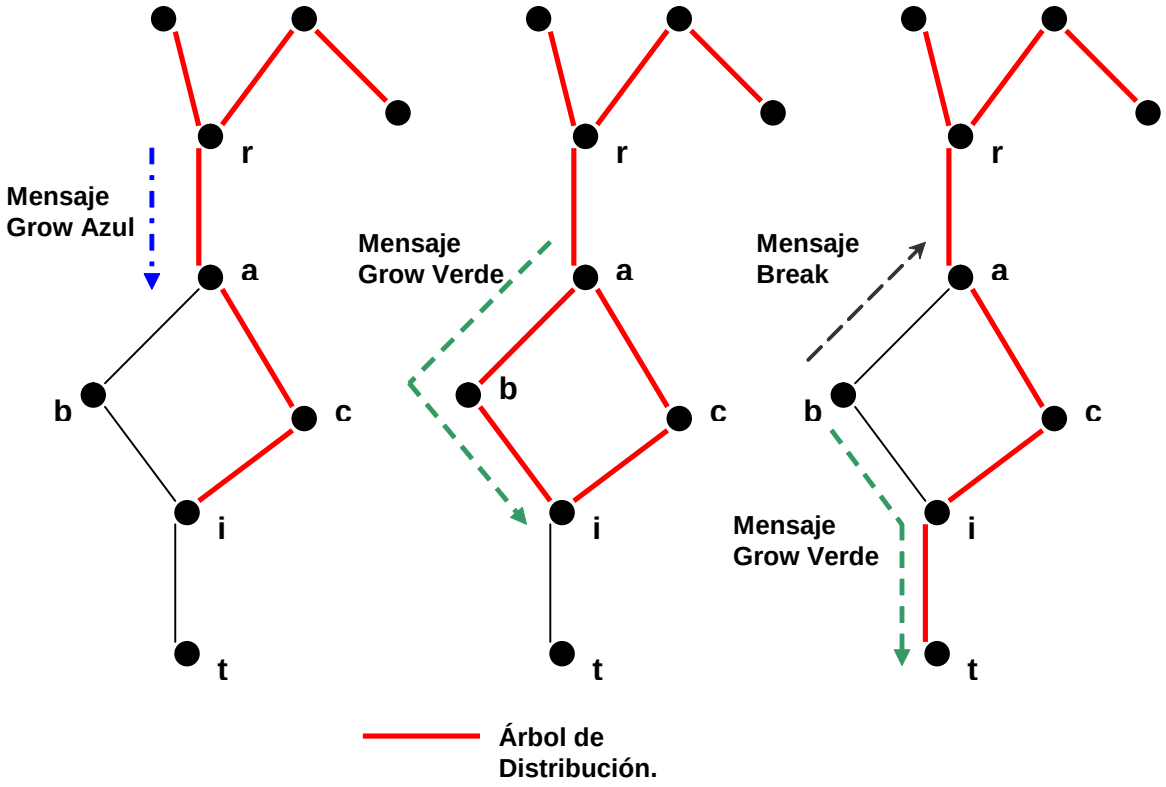


Figura 6.2.4.2.1.- Mecanismo de Detección de Loops (Ref: 32).

Como se ve en la figura anterior, el nodo interior *i* detecta un Loop cuando recibe un mensaje *Grow-Verde* proveniente del nodo *b*. Entonces, se envía un mensaje *Break* hacia atrás para "romper" el Loop de ruteo, mientras el mensaje *Grow* continúa construyendo la rama del árbol hacia el nuevo miembro *t*.

6.2.4.3. SoMR-m

Cada vez que la prueba *EW* genera un mensaje de *warning* en un nodo intermedio, el nodo se convierte en un *brandching point*, con lo cual a patir de este se generarán múltiples ramas. Si no se restringe la cantidad de *brandching points* esto puede aumentar el *overhead* de ruteo. Para controlar la cantidad de estos últimos se definen los siguientes parámetros:

MBL: Maximum Brandching Level.

Si consideramos que entre el nodo raíz *r* y un nodo interior del árbol de distribución existen al menos m *brandching points*, el máximo número de *brandching points* estará dado por:

[32] Tomado Referencia Bibliografica 36.

$\sum_{i=0}^{m-1}(d-1)^i$, donde *d* representa al máximo grado del nodo.

Cuando:

a. $d = 2 \rightarrow \sum_{i=0}^{m-1}(d-1)^i = m$

b. $d \rangle 2 \rightarrow \sum_{i=0}^{m-1}(d-1)^i = \frac{(d-1)^m - 1}{d-2}$

A esto se lo denomina SoMR-m, donde *m* representa el *Maximum Brandching Level*. En la siguiente figura se presenta un ejemplo de SoMR-3:

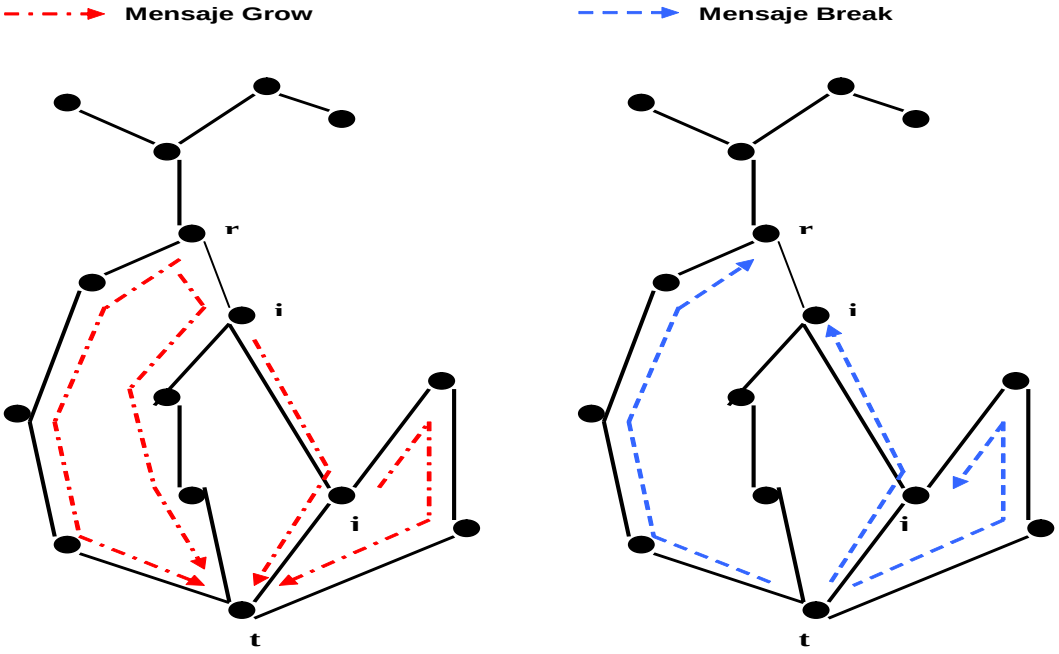


Figura 6.2.4.3.1.- Ejemplo de SoMR – 3 (Ref.: 33).

El *MBL* puede ser implementado parametrizando la cantidad de mensajes de *Grow* a través de un contador.

Directividad:

Puede implementarse disminuyendo las ramas del árbol de distribución que llegan al nuevo miembro *t*.

[33] Tomado Referencia Bibliografica 36.

MBD: Maximum Brandching Degree.

Un *brandching point* puede tener un gran número de nodos adyacentes, lo cual puede causar un elvado *overhead*. Por lo cual un parámetro adicional que se define es el *MDB*, *Maximum Brandching Degree*. Este, especifica la cantidad máxima de mensajes *Grow* que les permiten enviar a los *brandching points*.

Así, si el grado máximo de ramificaciones *x* es menor que el grado del nodo menos uno, el nodo seleccionará en forma aleatoria, o basada en la distancia en saltos al nuevo miembro, una cantidad *x* de interfases de salida desde las cuales no se han recibido mensajes de *Grow*, y enviará mensajes *Grow* a través de estas interfases. Se debe tener cuidado al setear el valor del *MDB*, ya que un valor elevado producirá un *overhead* considerable.

De lo anterior se define que considerando:

c. $MBL = m$

d. $MDB = x$

Por lo cual se tiene que el máximo número de *brandching points* estará dado por:

$$\sum_{i=0}^{m-1} x^i = \frac{x^m - 1}{x - 1}$$

6.2.4.4. PERFORMANCE DE SoMR Y COMPARACIÓN CON QoSMIC.

En el presente punto se comparará la *performance* de SoMR-m con QoSMIC, tomando las siguientes consideraciones:

a. Red de *n* nodos distribuidos uniformemente.

b. Una fuente emitiendo tráfico Multicast generado por una aplicación de vídeo conferencia en Tiempo real.

c. Diámetro de la red 2 *w* saltos.

d. El número de nodos en el *k-neighborhood* de un nodo, N_k, crece siguiendo un ley cuadrática con el valor de *k*, $N_k = \alpha \,.\, k^2$. Por lo cual el diámetro esta dado por: $\alpha \,.\, w^2 = n$.

e. Cuando un protocolo *spanning join* envía mensajes *broadcast* a sus nodos "vecinos", o adyacentes, dentro de un radio de *k* saltos, la cantidad de mensajes enviados estará dada por: $\alpha \,.\, k^2$. Así, en el peor de los casos, el número total de mensajes enviados en forma consecutiva estará dado por:

$$\sum_{k=1}^{w} (\alpha \,.\, k^2) = \alpha \,.\, \frac{w \,.\, (w+1) \,.\, (2 \,.\, w+1)}{6} \approx \frac{n \,.\, (2 \,.\, w+1)}{6} \in O\,(nw)$$

f. El *overhead* de mensajes, producido por el mecanismo de *Local Search* en QoSMIC, dentro de un "vecindario", *neighborhood*, pequeño con un radio constante, se considera como una constante.

g. Sea *T* el tamaño del árbol de distribución Multicast, el peor caso de *overhead* en el mecanismo *Tree Search*, en QoSMIC, estará dado por: $O\,(T)$.

h. Consideremos SoMR-m, con $MBD = x$. El número máximo de *brandching points* estará dado por: $\frac{x^m - 1}{x - 1}$, mientras que el máximo número de ramas estará dado por $x \,.\, \frac{x^m - 1}{x - 1}$; donde tanto *m* como *x* son ambos constantes pequeñas. La longitud de cualquier rama del árbol de distribución estará dada por $O\,(2w)$. Así, en el peor caso, el número total de mensajes enviados estará dado por: $O\left(x \,.\, \frac{x^m - 1}{x - 1} \,.\, 2w \right) = O\,(w)$.

i. Considerando que todos los nodos de la red tienen grado *d*, será $w = O\left(d / \sqrt[2]{n}\right)$.

Al igual que para el caso de QMRP-m, se define:

a. $\text{Tasa de Exito} = \frac{\text{Cantidad de Nuevos Miembros Aceptados}}{\text{Cantidad Total de Mensajes Join Request}}$.

b. $\text{Overhead Promedio de Mensajes} = \frac{\text{Número Total de Mensajes Enviados}}{\text{Número Total de Mensajes Join Request}}$.

Para la simulación se tuvieron en cuenta las siguientes observaciones:

- SoMR:

  - ⇨ El máximo grado de *brandching* en SoMR-3 es 5.
  - ⇨ Un *brandching point* puede enviar al menos 5 mensajes *Grow* a sus "vecinos".

- QoSMIC:

  - ⇨ Los mecanismos de *Local Search* y *Tree Search* se implementan en forma secuencial, donde el segundo sólo se eimplementará si falla el primero.
  - ⇨ Se implementan: Directividad, Mínimo Local y Elección Fraccional.

- Red:

  - ⇨ Topología de red, de 600 nodos, basada en *power-law-network*.
  - ⇨ El 5 % de los enlaces se tomarán, en forma aleatoria, como saturados. El *delay* en estos enlaces se considerará infinito.
  - ⇨ El *delay* de los enlaces no saturados se generará en forma aleatoria, tomando un rango de $[\,0\,,\,200\,]$ unidades de tiempo.
  - ⇨ El nodo raíz del árbol de distribución se seleccionará en forma aleatoria, estableciéndose un requerimiento de *delay* para el árbol de distribución. Luego, los nodos de la red comenzarán a unirse al árbol en forma aleatoria, intentándolo cada uno una vez.

Luego, a partir de lo presentado anteriormente se tienen los siguientes gráficos:

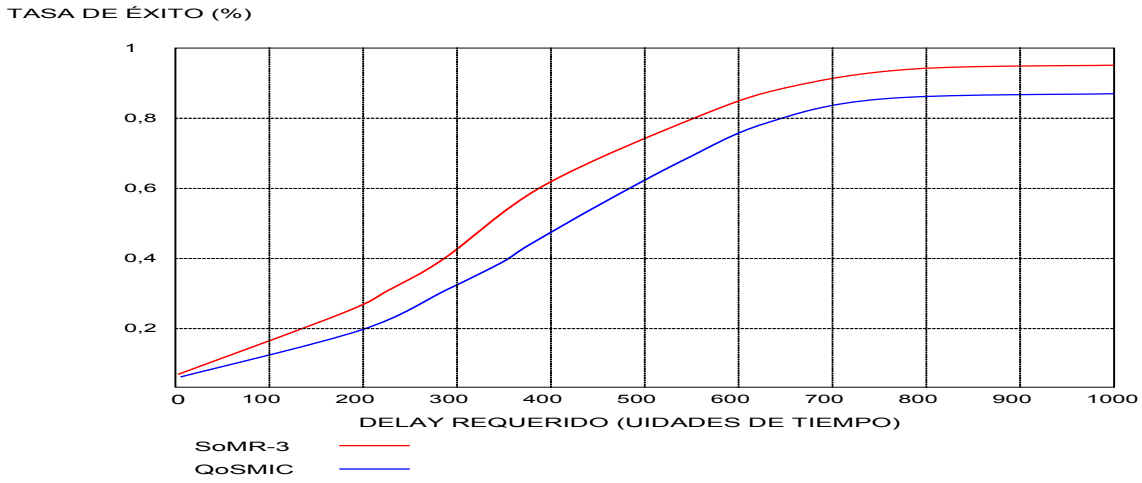


Figura 6.2.4.4.1.a.- Comparación de Performance entre SoMR-3 y QoSMIC (Ref.: 34)

[34] Tomado Referencia Bibliografica 36

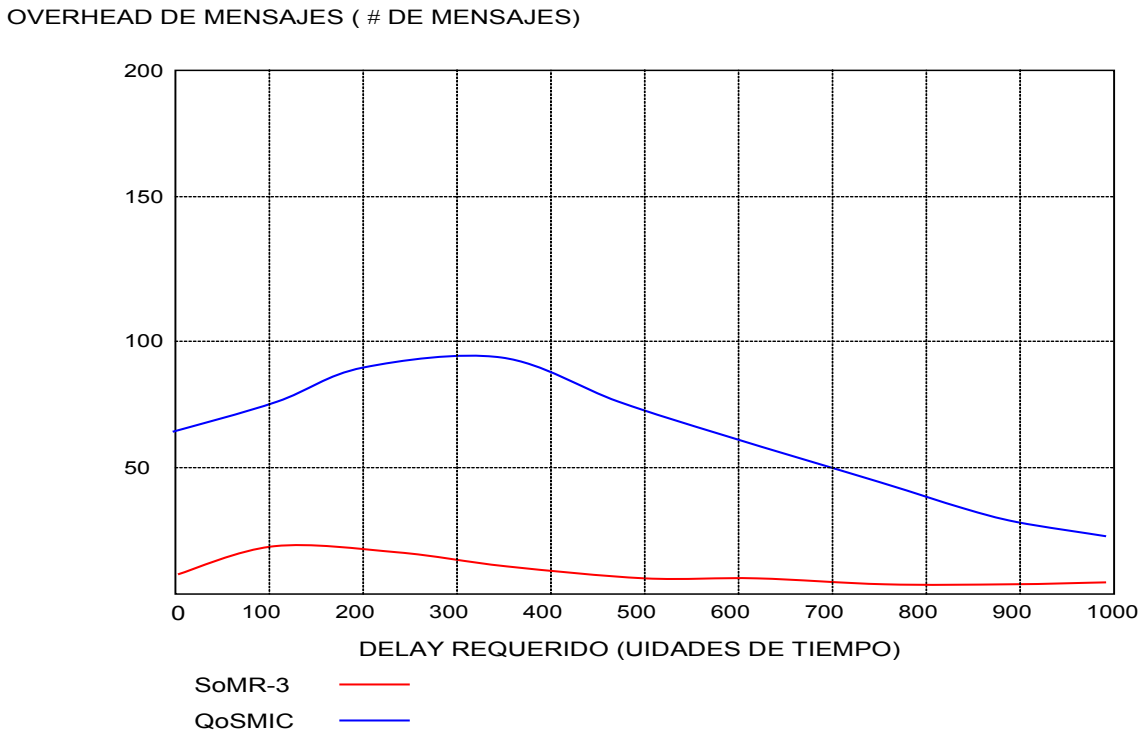


Figura 6.2.4.4.1.b.- Comparación de Performance entre SoMR-3 y QoSMIC (Ref.: 35)

De las figuras anteriores se puede ver que la tasa de éxito de SoMR-3 es mejor que la que se obtiene con QoSMIC.

6.2.4.5. CONCLUSIONES.

- SoMR fue diseñado para aplicaciones con requerimientos de QoS aditivos como el *delay*.
- Elimina el almacenamiento temporario de estados por grupo y por mensaje *Join* en los *router*s.
- No requiere almacenar otro estado extra de ruteo más que el árbol de distribución.
- Elimina el mecanismo de búsqueda del árbol de distribución que presenta QMRP, lo cual simplifica su implementación.
- Emplea un mecanismo denominado *Early-Warning,* el cual le permite tener en cuenta las características aditivas de requerimientos de QoS como el *delay*, por el

[35] Tomado Referencia Bibliografica 36

cual se identifica el punto más apropiado para explorar caminos alternativos que le permiten maximizar la probabilidad de éxito.

⇨ Presenta un buen balance entre el *overhead* de comunicación y la probabilidad de éxito.

⇨ Presenta un menor *overhead* y una mayor probabilidad de éxito que QoSMIC.

6.2.5. ANÁLISIS DEL PROTOCOLO RIMQoS.

Las principales características de este protocolo son:

- Brinda soporte de condiciones de QoS Multicast con esquemas de membresía tanto dinámicos como estáticos.

- Intenta minimizar el costo de un árbol de distribución; para lo cual, permite que un nodo se una a un árbol de distribución a través de un camino de bajo costo.

- Permite cambiar un miembro de una rama del árbol a otra, la cual posea mejores condiciones de calidad de servicio.

- Permite una implementación totalmente distribuida y el soporte de múltiples métricas de QoS y requerimientos.

- Reduce la cantidad de mensajes que se tendrían en un esquema Multicast Nativo, simplificando el procediciendo de *Join* de los nuevos miembros.

- No requiere el conocimiento previo de los miembros existentes.

- El procedimiento de reservación de recursos no requiere la utilización de demasiados recursos en los nodos de la red.

- Fue desarrollado para soportar la implementación de calidad de servicio bajo un esquema Multicast desde una única fuente hacia varios receptores.

Un nodo receptor utilizará un protocolo de ruteo Unicast con soporte de QoS para calcular un camino desde una fuente, o desde un nodo raíz, hacia él mismo, tal que se satisfagan las condiciones de calidad de servicio. El receptor luego enviará un mensaje *Join Request* para unirse al grupo Multicast. Así, el protocolo RIMQoS permite el soporte de calidad

de servicio bajo un esquema de una a muchas aplicaciones, desde una fuente hacia más de un receptor.

6.2.5.1. DESCRIPCIÓN DEL PROTOCOLO.

RIMQoS fue diseñado para trabajar sobre protocolos de ruteo existentes, los cuales soporten QoS y tengan la funcionalidad de pre-calcular los caminos.

A continuación se presentará una breve descripción de los mensajes utilzados por el protocolo:

- Se supone que los nodos almacenan información de *forwarding* Multicast de las interfases de entrada y de salida.

- Cuando un nuevo miembro, receptor, solicita unirse a un grupo, envía un mensaje *Join Request* hacia la fuente, o hacia el nodo raíz del árbol de distribución. Luego, cada interfase en la tabla de ruteo Multicast será etiquetada como *Active*, activa, o *Pending*, pendinte. Así, el tráfico Multicast sólo será "*forwardeado*" por las interfases activas.

- Para la reservación de recursos se utilizan dos tipos de mensajes: *Accept Notification* y *Deny Notification*. Cuando un mensaje *Join Request* es aceptado o no, en algún nodo, un mensaje *Accept/Deny Notification* será enviado hacia el nuevo receptor. Los mensajes *Accept Notification* reservarán los recursos necesarios, cambiando las indicaciones de *Pending* a *Active*, a medida que atraviesan las interfases de los nodos.

- Cuando un miembro deja un grupo, envirá mensajes *Prune* en dirección *upstream* a lo largo de las ramas del árbol de distribución.

- Cuando un miembro se une a un grupo, los "*contratos de servicio*" concernientes a las condiciones de QoS serán efectivos hasta tanto la conexión sea liberada. No se consideran renegociaciones sobre la calidad de servicio.

- Tanto el nodo fuente como los nodos interiores de un árbol de distribución enviarán mensajes QoS *Probing* en forma periódica, o bajo ciertos umbrales, hacia todos o algunos de los nodos receptores. Los nodos intermedios procesarán y modificarán estos mensajes, tal que la información sobre la calidad de servicio del camino entre la fuente y el nodo receptor sea transportada en el mensaje.

6.2.5.1.2. OPERACIÓN INTRA-DOMAIN.

Consideremos una situación *intra-domain* donde la dirección del grupo Multicast es conocida por los potenciales receptores a través de algún tipo de mecanismo.

Supongamos que cada nodo receptor tiene información sobre el estado de todos los enlaces, información *link state*, dentor de su dominio.

Consideremos que el ancho de banda requerido, dado por *B*, y el *delay* máximo permitido, dado por *D*, son requerimientos impuestos por la fuente de tráfico Multicast que se encuentra emitiendo datos de una aplicación de vídeo conferencia en Tiempo Real.

Cuando un nodo recibe un pedido de *Join*, de uno de sus *host* locales, a través por ejemplo de IGMP, si el nodo no se encontraba unido al árbol de distribución, este calculará una ruta desde la fuente hacia él mismo la cual satisfaga, por ejemplo, las condiciones de ancho de banda y *delay*. Esta ruta es generalmente la de mínimo costo. Luego, este nodo enviará un mensaje *RIMQoS Join* hacia el nodo fuente, especificando completamente la ruta.

Cuando un nodo intermedio recibe un mensaje *RIMQoS Join Request*, si este no es parte de árbol de distribución Multicast, creará una nueva entrada en la tabla Multicast (group*/source*, *in-interface*, *out-interface*, *destination*). La interfase *out-interface* es aquella a través de la cual el nodo recibe el mensaje de *request*, y a través de la cual el nodo "*forwardeara*" el pedido basado en la información de ruteo contenida en el mensaje. Luego actualiza en el mensaje *Join Request* la característica de calidad de servicio del enlace por el cual este recibió el pedido y "*forwardea*" el mismo hacia el próximo nodo. Además marcará la entrada creada en la tabla de ruteo como *pending*, pendiente, e iniciará un *timer*. Luego, la entrada será marcada como *active*, activa, cuando se recibe un mensaje *accept*, el cual será "*forwardeado*" en sentido *downstream*. Cuando una entrada de ruteo es marcada como *pinned*, la interfase o enlace requerida es reservada. Si la reservación del recurso no resulta satisfactoria, por ejemplo debido a un cambio en las condiciones de red, el proceso de *route pinning* es abandonado y se enviará un mensaje de *deny*. Cada vez que un mensaje *Join Request* es aceptado, el nodo deberá enviar un mensaje *Probing* junto con un mensaje *accept* para informar a los nuevos nodos de las características de calidad de servicio del camino.

Cuando un nodo recibe un mensaje *Join Request*, y este ya formaba parte de un árbol de distribución Multicast, verificará la información de QoS para el grupo o fuente.

Consideremos el ejemplo presentado en la siguiente figura:

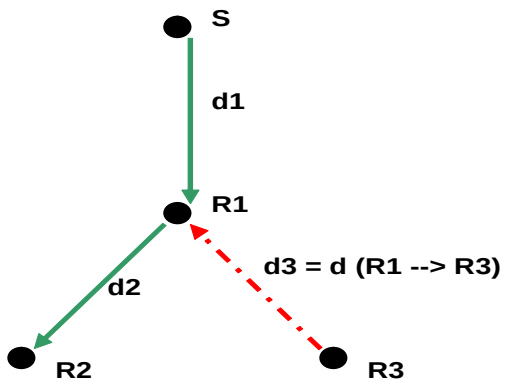


Figura 6.2.5.1.2.1.- Procedimiento de Join (Ref.: 36).

Donde:

- R_1: representa un nodo intermedio, y el *delay* del enlace esta representado por *d1*.
- R_3 : representa a un nuevo nodo miembro.
- d_3 : representa el *delay* acumulado.
- S : representa al nodo fuente.

Si $d_1 + d_3 \leq D$, el nodo R_1 agregará la correspondiente interfase *out-interface* a la entrada existente en la tabla de ruteo Multicast para dicho grupo, y luego "*forwardeara*" un mensaje *accept* si el proceso de reserva de recursos, ancho de banda, resultó satisfactorio; de lo contrario, enviará un mensaje *deny*.

Si, en cambio, $d_1 + d_3 > D$, tendremos dos posible situaciones para este caso:

a. El *next hop* es el nodo adyancente a R_1 en sentido *upstream*. Por lo cual, en este caso, el nodo R_1 sólo podrá agregar la interfase a través de la cual el nodo recibió el mensaje *Join Request* en la entrada correspondiente de la tabla de ruteo Multicast marcándola como *pending*. Luego enviará el mensaje *request* hacia R_0. Este mensaje *Join Request* puede ser aceptado por el nodo R_0 o "*forwardeado*". El

estado de *pending* será cambiado cuando se reciba un mensaje *accept*, o cuando se reciba un mensaje de *deny* o se cumpla con el *Timer* establecido.

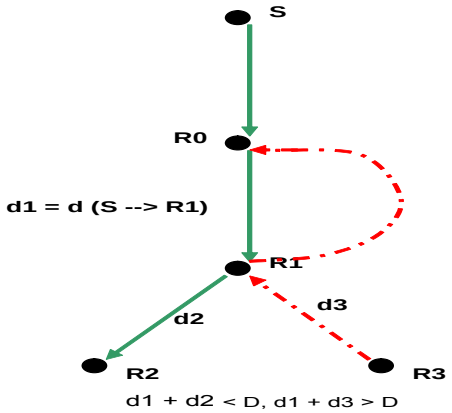


Figura 6.2.5.1.2.2.- Procedimiento de Join, donde el Camino Existente no Cumple con el Delay, caso a. (Ref.: 37).

b. En este caso, el *next hop*, R' no es el nodo adyacente a R_1 en sentido *upstream*.

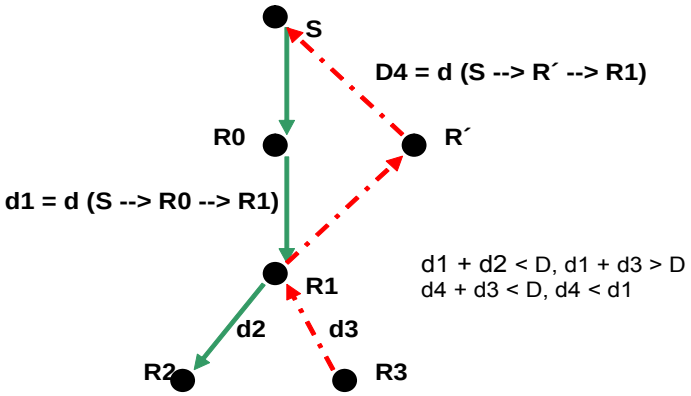


Figura 6.2.5.1.2.3.- Procedimiento de Join, donde el Camino Existente no Cumple con el Delay, caso b. (Ref.: 38).

En este caso, R_1 "*forwadeara*" el mensaje Join Request hacia R' .Además, agregará una entrada en estado pending con R' como nodo adyancente en sentido *upstream*, marcándolo como nodo "preferido" para el grupo o fuente Multicast. Cuando R_1 reciba un mensaje de *accept* de R' , cambiará el estado de la entrada en la tabla de ruteo habilitándola para el *forwarding* de datos, y enviará un mesaje de *accept* a través de ésta. También enviará un mensaje *Prune* a R_0 , pero sólo después de recibir tráfico Multicast proveniente de R' . Esto dará lugar a un intercambio entre la rama antigua y una nueva rama.

[36] Tomado Referencia Bibliografica 38.
[37] Tomado Referencia Bibliografica 38.
[38] Tomado Referencia Bibliografica 38.

Para evitar que se formen Loops de ruteo, un nodo nunca "*forwardea*" mensajes de *Join Request* hacia un nodo que se encuentre "*forwardeando*" tráfico Multicast, y descartará todo mensaje *Join Request* de un nodo del cual este recibiendo tráfico Multicast.

6.2.5.2. RIMQoS GENERALIZADO.

Como se vio en puntos anteriores, la mayoría de las métricas de QoS pueden ser clasificadas en tres categorías:

a. *Transitiva*.
b. *Aditiva*.
c. *Multiplicativa*.

Consideremos una sesión Multicast la cual es generada por una fuente que emite tráfico Multicast de una aplicación de vídeo conferencia, la cual presenta las siguientes condiciones de QoS, de extremo a extremo:

a. El ancho de banda disponible es: $bw \geq BW$.
b. El *delay* es: $d \leq D$.
c. El *jitter* es: $j \leq J$.

Consideramos que en cada nodo se cuenta con información sobre el ancho de banda residual, el *jitter* y el *delay* del enlace. Además, se supone que se utiliza un protocolo de ruteo Unicast capaz de encontrar un camino que cumpla con los requerimientos de calidad de sericio.

Un nuevo nodo receptor enviará un mensaje *Join Request* hacia el nodo fuente a través de un camino *source route*, el cual él mismo calculó. Este camino puede ser de mínimo costo calculado en base a una métrica pre-definida. Todos los nodos intermedios "*forwardearan*" el mensaje de *Join Request* hasta que sea aceptado o rechazado. Cuando un nodo que no sea miembro del árbol de distribución reciba un mensaje *Join Request*, actualizará la información sobre las características de calidad de servicio del enlace sobre el cual recibió el mensaje y lo enviará al siguiente nodo que figura en el mensaje.

Ahora consideremos que un nodo miembro del árbol de distribución reciba un mensaje *request*. Supongamos que el nodo R_1 de la figura 6.2.5.1.2.1 recibe un mensaje *request* del nodo R_3, para unirse a un grupo cuyo nodo fuente es *S*. El *delay* entre S y R_1 será: $d_1 = d(S \rightarrow R_1)$, y el *jitter* será: $j_1 = j(S \rightarrow R_1)$, miemtras que el mensaje *request* transportará el *delay* entre R_1 y R_3, dado por: $d_3 = d(R_1 \rightarrow R_3)$, y el *jitter* será: $j_3 = j(R_1 \rightarrow R_3)$.

Si $d_1 + d_3 \leq D$ y $j_1 + j_3 \leq J$, entonces el mensaje *Join Request* puede ser aceptado por el nodo R_1. Si no se cumple la condición anterior se peden presentar dos posibilidades:

a. El nodo *upstream* R_0 es el *next hop* especificado en el mensaje *Join Request*: Con lo cual, el nodo R_1 "*forwardeara*" el mensaje *Join Request* hacia R_0, y agregará una entrada en la tabla Multicast con estado *pending*, correspondiente a R_3.

b. El *next hop* especificado en el mensaje *Join Request* no es R_0: si $d_1 \geq D - d_3$ y $j_1 \geq J - j_3$, entonces el nodo R_1"*forwardeara*" el mensaje *Join Request* hacia el *next hop* R', tal lo indicado en el mensaje *Join Request*, agregará la entrada correspondiente en la tabla de ruteo Multicast y marcará a R' como nodo preferido en sentidfo *upstream*.

Si al menos uno de los casos anteriores se satisfacen, el nodo R_1 podrá calcular un nuevo camino desde el nodo fuente *S* hacia R_1, el cual cumplirá con:

a. $\min[D - d_3 , d_1]$
b. $\min[J - j_3 , j_1]$

Si tal camino no puede ser encontrado, el nodo R_1 denegará el mensaje de *Join Request*. De lo contrario, R_1 enviará un nuevo mensaje *Join Request* con la información sobre el nuevo camino y sus características de calidad de servicio. Si este es aceptado; entonces, el mensaje original proveniente del nodo R_3 también lo será. Esto se muestra en la siguiente figura.

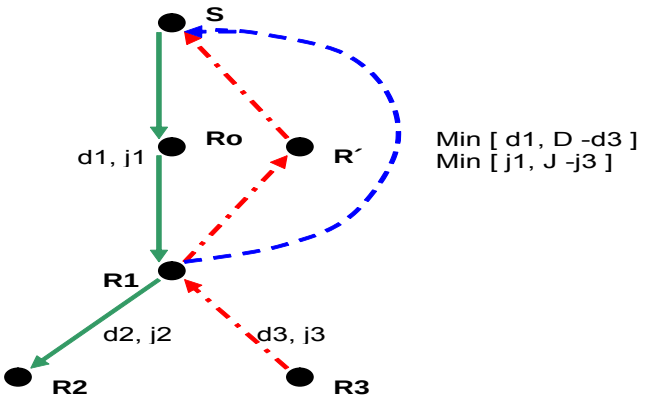


Figura 6.2.5.2.1.- Procedimiento de Cálculo de un Nuevo Camino con Requerimientos de Jitter y Delay (Ref.: 39).

[39] Tomado Referencia Bibliografica 38.

6.2.5.3. EVALUACIÓN DE LA PERFORMANCE DE RIMQoS.

Podemos citar dos parámetros que caracterizan la *performance* del protocolo RIMQoS, y nos permitan realizar una evaluación del mismo:

a. *Complejidad de Cómputo*: parametriza la capacidad de cómputo necesaria para encontrar los caminos y construir el árbol de distribución.

b. *Control del Overhead*: controla el *overhead* producido por los mensajes.

En RIMQoS se delega la mayor parte de la complejidad de cómputo a los nodos receptores. Estos nodos realizan los cálculos necesarios para el ruteo de los mensajes y paquetes de datos, mientras que el resto de los nodos del árbol de distribución sólo realizan el procesamento de los mensajes.

Complejidad de Cómputo:

Si el camino de *Join*, *path join*, es calculado bajo demanda, la complejidad de cómputo dependerá del protocolo Unicast utilizado.

Si los requerimientos de QoS son ancho de banda y *delay*, tendremos opciones de ruteo, bajo condiciones de QoS, determinadas en forma heurística, dadas por $O\left(n \,.\, |E|\right)$, donde *n* representa la cantidad de nodos del árbol de distribución y el $|E|$ representa la cantidad de nodos de borde. En la mayoría de las redes se puede considerar que $|E| = O(n)$, por lo cual podemos representar la complejidad de cómputo como $O(n^2)$. Para un grupo Multicast de *m* miembros, la complejidad de cómputo para la construcción del árbol de distribución estará dada por: $O(m\, n^2)$.

Control del Overhead:

En RIMQoS el proceso mediante el cual un nuevo miembro se une a un grupo requiere de dos mensajes: un mensaje *Join Request* y un mensaje *Join Accept* o *Join Deny*. Así, el establecimiento de un grupo de *m* miembros requerirá de $2 \,.\, m$ mensajes.

Un mensaje *Join Request* será procesado por todos los *h hops*, o nodos, recorridos a lo largo del camino, o *path*, hasta alcanzar al nodo en el cual el mensaje es aceptado o

.

rechazado. Así, el costo del procesamiento de mensajes para el caso de la unión de *m* nuevos nodos miembros sería de $2 . m . h$.

En el caso del protocolo QoSMIC centralizado, considerando que se tiene un nodo raiz que realiza el cómputo del árbol de distribución y que este luego utiliza un algoritmo de señalización para el establecimiento del mismo, el costo de procesamiento de mensajes estaría dado por $m . H_{avg}$; donde H_{avg} es el promedio de la métrica *Hop-Count*.

6.2.5.4. CONCLUSIONES.

- Al computar los receptores la ruta del árbol de distribución, se simplifica la participación de otros nodos de la red, lo cual permite el soporte de una membresía de grupo completamente dinámica.

- Especifica como un nuevo miembro puede ser agregado dentro de un árbol de distribución existente sin afectar la calidad de servicio de los otros miembros del grupo, y con mínimos ajustes sobre el árbol de distribución.

- Construye árboles de ditribución de mínimo costo.

- Combina las fases de establecimiento de la conexión y reservación de recursos.

- Simplifica el procedimiento de envío y recepción de mensajes.

- Soporta cualquier número de métricas de QoS y requerimientos.

- Presenta una baja complejidad de cálculo.

6.2.6. CONCLUSIÓN.

Durante el desarrollo del presente trabajo de Tesis se presentó y profundizo en el análisis de los distintos temas que nos permitieron desarrollar el capítulo 6 de la misma, donde se analizaron algunos protocolos y técnicas que permiten tener en cuenta los requerimientos de calidad de servicio de las aplicaciones Multicast. La importancia de esto radica en la proliferación de las nuevas aplicaciones de datos, las cuales hacen necesario y deseable el poder contar con técnicas y métodos que nos permitan tener una transimisón en modo Multicast con soporte de QoS.

De los protocolos y métodos analizados, basando nuestro análisis y comparación en el caso de tener una fuente Multicast que transmite datos provenientes de una aplicación de vídeo conferencia interactiva en Tiempo Real, el que mejor se ajusta a estas necesidades es el método AQoSM. Como justificación se pueden citar las siguientes conclusiones:

- AQoSM puede implementarse sobre redes MPLS y utilizar las funcionalidades de QoS nativas de MPLS, que le permiten a AQoSM establecer árboles de distribución cuyas ramas son creadas de acuerdo a las especificaciones de calidad de servicio de las aplicaciones.

- Al utilizar la técnica de *Aggregated Tree*, esto le brinda un mayor grado de escalabilidad que el que presenta un esquema Multicast Nativo sobre redes MPLS.

- Nos permite que un grupo Multicast pueda cambiar de árbol de distribución en forma dinámica, rápida y eficiente en el caso que el árbol al cual este conectado deje de cumplir con sus requerimientos de calidad de servicio. Analizándolo desde el punto de vista de nuestro caso, esto puede ocurrir cuando se agreguen nuevos receptores a la sesión, lo cual requerirá de un mayor ancho de banda y que se continúe cumpliendo con las condiciones de *delay* y *jitter*.

- De los métodos analizados es el que presenta el menor *overhead*, lo cual es una característica importante desde el punto de vista de aprovechar al máximo posible el ancho de banda disponible en los enlaces de la red para cumplir con los requerimientos impuestos por las sesiones Multicast.

- Una desventaja es que depende de un nodo centralizado, el *Tree Manager*, del cual dependerá la administración y funcionamiento de la red. Para mejorar este aspecto se podrían implementar esquemas de redundancia, los cuales le darán una mayor robustes.

- Una característica importante que podemos destacar es que delega la inteligencia a los nodos de borde de la red, lo cual permitirá reducir el procesamiento de información innecesario en los nodos del *core*.

- Comparándolo con el protocolo QoSMIC, para el caso de transmisiones de aplicaciones interactivas en Tiempo Real, AQoSM presenta una mejor *performance*, ya que se adapta mejor a los requerimientos de QoS de estas aplicaciones y nos permite tener una combinación de estos sin que signifique un gran aumento en la complejidad de administración del mismo.

- Comparándolo con el protocolo QMRP, AQoSM nos permitirá el soporte de requerimientos de QoS aditivos como lo son el *delay* y el *jitter*, los cuales son muy deseables de poder administrar y controlar en los casos de transmisiones de aplicaciones Multicast de vídeo conferencia.

- Con respecto a SoMR, si bien este nos permite el soporte de requerimiento de QoS aditivos, el método AQoSM se presenta más flexible al momento de tratar con el dinamismo que presenta la membresía de los receptores a los grupos Multicast.

7 GLOSARIO.

ABR	Area Border Router
AFI	Address Family Identifier
AS	Sistema Autonomo
ASSM	Aggregated Source Specific Multicast
ATM	Asynchronous Transfer Mode
BGMP	Border Gateway Multicast Protocol
BOOTP	Bootstrap Protocol
CBT	Core-Based Tree
CoS	Class of Service
DM	Dense Mode
DNS	Domain Name System
DR	Designated Router
DSCP	Diff Serv Code Point
DVMRP	Distance Vector Multicast Routing Protocol
EW	Early Warning
FEC	Forwarding Equivalence Class
IANA	Internet Assigned Numbers Authority
IDMR	Inter-Domain Multicast Routing
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IP	Internet Protocol
ISP	Internet Service Provider
LAN	Local Area Network
LDP	Label Distribution Protocol
LFIB	Label Forwarding Information Base
LIFO	Last In First Out
LSA	Link State Advertisement
LSP	Label Switched Path
LSR	Label Switching Router
MA	Multicast Aggregation Router
MABR	Multicast Area Border Router
MAC	Media Access Control
MAD	Multicast Aggregation Domain
MASC	Multicast Address Set Domain
MBD	Maximun Brandching Degree
MBGP	Multiprotocol Border Gateway Protocol
MBL	Maximun Brandching Level

MBone	Multicast Backbone
MOSPF	Multicast extensions to Open Shortest Path First
MPLS	Multiprotocol Label Switching
MPR	Multiple Path Routing
MRT	Multicast Routing Table
MSDP	Multicast Source Discovery Protocol
MTS	Multicast Tree Set
NIC	Network Interface Card
OSPF	Open Shortest Path First
PDU	Paquet Data Unit
Peer	Par
PIM	Protocol Independent Multicast
PIM-DM	Protocol Independent Multicast - Dense Mode
PIM-SM	Protocol Independent Multicast - Sparse Mode
PIM-SSM	Protocol Independent Multicast Spare Source Mode
QoS	Quality of Service
RFP	Reverse Path Forward
RIP	Routing Information Protocol
RP	Redezvous Point
RPB	Reverse Path *Broadcast*
RPF	Reverse Path Forwarding
RPM	Reverse Path Multicast
RTCP	Real Time Transport Control Protocol
RTP	Real Time Protocol
RTSP	Real Time Session Protocol
SM	Spare Mode
SPR	Single Path Routing
SPT	Shortest Path Tree
ST	Steiner Tree
Sub-AFI	Subsequent Address Family Identifier
TCP	Transmission Control Protocol
TD	Transit Domains
TLV	Type Length Value
TRPB	Truncated Reverse Path *Broadcast*
TTL	Time-To-Live
UDP	User Datagram Protocol
WAN	Wide Area Network

8. BIBLIOGRAFÍA.

1. Anycast RP, Cisco Document.
   "Comprender el efecto del ruteo Aycast en redes Multicast".

2. Managing Group Dynamics and Failures in QoS Multicast. A Striegel y G. Manimaran, Junio 2002.
   "Caractierísticas y tópicos relacionados con las implenentaciones de protocolos IP Multicast, y su impacto sobre el soporte de requerimientos de QoS".

3. Developing IP Multicast Networks, Volumen 1, Cisco Press.
   "Especificaciones sobre los distintos protocolos de ruteo Multicast y su Impacto en redes MPLS".

   Video Over the Internet: Standards and Protocols, Yang Xiaokang, Zhu Ce, Li Zhengguo, Feng Genan and Wu Si, Centre for Signal Processing, Nanyang Technological University".
   "Aportes sobre las características de los sistemas de compresión de vídeo con sus ventajas y desventajas".

4. Cisco White Paper, Internet Protocol (IP) Multicast, copyright 2000.
   "Especificaciones sobre los distintos protocolos de ruteo Multicast y su Impacto en redes MPLS".

5. IP Multicasting: Concepts, Algoritms ando Protocols, Mohammed Banikazemi.
   "Especificaciones sobre los distintos protocolos de ruteo Multicast y su Impacto en redes MPLS".

6. MPLS Technology and Applications; Bruce Davie, Yakov Rekhter; Morgan Kaufmann Publishers.
   "Especificaciones las características de una red IP MPLS, sus elementos, protocolos de , IP Multicast en Redes IP MPLS".

7. RFC 3353; IP Multicast in an MPLS Enviroment, Agosto 2002.
   "Especificaciones sobre la Implementación de IP Multicast en redes IP MPLS, ventajas, desventajas, características a tener en cuenta al implementar este tipo de soluciones de red ".

8. Using PIM to Distribute MPLS Labels for Multicast Routes (Internet Draft); Dino Farinaci, Noviembre 2000.
"Especificaciones sobre la utilización del protocolo IP Multicast PIM para la publicación de rutas en redes IP MPLS".

9. An Effective Solution for Multicast Scalability (Internet Draft); Ali Boudani, Bernard Cousin, Noviembre 2001.
"Características y temas a tener en cuenta en cuanto a la escalabilidad de soluciones de IP Multicast en redes IP MPLS".

10. IP Multicast Support in MPLS Networks (Internet Draft); Arup Acharya, Frederic Griffoul.
*"Características y t*emás *a tener en cuenta al soporte de soluciones de IP Multicast en redes IP MPLS".*

11. RFC 2365, Administratively Scoped IP Multicast; D. Meyer.
"Especificaciones y características técnicas de soluciones basadas en IP Multicast".

12. RFC 2770, GLOP Addressing in 233/8; P. Lothberg, D.Meyer; Febrero 2000.
"Definiciones sobre direccionamiento IP en redes IP Multicast".

13. RFC 1112, *Host* Extensions for IP Multicasting; S. Deering.
"Definiciones sobre direccionamiento IP para Hosts en redes IP Multicast".

14. RFC 2236, Internet Group Management Protocol, Version 2; W. Fenner; Noviembre 1997.
"Especificación y Caraterísticas del protocolo e ruteo IGMP en implementaciones basadas en redes IP Multicast".

15. RFC 1584, Multicast Extensions to OSPF; J.Moy; Marzo 1994.
"Definiciones para soportar transporte de update IP Multicast utilizando el protocolo de ruteo IP OSPF Unicast".

16. RFC 2283, Multiprotocol Extensions for BGP-4; T. Bates, R. Chandra, D. Katz, Y. Rekhter; Febrero 1998.
"Definiciones para soportar transporte de update IP Multicast utilizando el protocolo de ruteo IP BGP Unicast".

17. Forwarding State Reduction for Sparese Mode Multicast Communications. Proceedings of IEEE INFOCOM. J. Tian and G. Neufeld.
"Definiciones del Multicast Aggregated Tree. Base para el desarrollo de la propuesta de la tesis.".

18. Exploiting the bandwidth-memory Tradeff in Multicast State Aggregation. P.I Radoslavov, D. Estrin, y R. Govindan.
"Definiciones del Multicast Aggregated Tree. Base para el desarrollo de la propuesta de la tesis."

19. On the Aggregatability of Multicast Forwarding State. Proceedings of IEEE INFOCOM. D. Thaler y M. Handley.
"Definiciones del Multicast Aggregated Tree. Base para el desarrollo de la propuesta de la tesis."

20. Extending the Internet Multicast Architecture. P.Francis. Yoid. http://www.aciri.org/yoid/docs/index.html
"Definiciones del Multicast Aggregated Tree. Base para el desarrollo de la propuesta de la tesis."

21. A Case for End System Multicast. Proceeding of ACM Sigmetrics. Y. chu S. Rao y H. Zhang.
"Definiciones del Multicast Aggregated Tree. Base para el desarrollo de la propuesta de la tesis."

22. Aggregated Multicast: an Approach to Reduce Multicast State. Aiguo Fei, Junhong Cui, Mario Gerla, Michalis Faloutsos.
"Definiciones del Multicast Aggregated Tree. Base para el desarrollo de la propuesta de la tesis."

23. Aggregated Multicast, A Comparative Study. Jun-Hong Cui, Jinkyu Kim, Dario Maggiorini, Khaled Boussetta y Mario Gerla..
"Definiciones del Multicast Aggregated Tree. Base para el desarrollo de la propuesta de la tesis."

24. Definiciones del Multicast Aggregated Tree. Jun-Hong Cui, Michalis Faloutsos y Mario Gerla.
"Base para el desarrollo de la propuesta de la tesis."

25. Aggregated Multicast with Inter-Group Tree Sharing. Aiguo Fei, Junhong Cui, Mario Gerla y Michalis Faloutsos.
"Definiciones del Multicast Aggregated Tree. Base para el desarrollo de la propuesta de la tesis."

26. Internet Draft: An Effective Solution for Multicast Scalability, The MPLS Multicast Tree (MMT). Ali Boudani, Bernard Cousin y Jean-Marie Bonnin, Octubre 2004.
"Definiciones del Multicast Aggregated Tree. Base para el desarrollo de la propuesta de la tesis."

27. Multicasting in MPLS Domains, Marzo 2003. Baijian Yang, Prasant Mohapatra.
"Estudio sobrela implementación de IP Multicast sobre redes MPLS".

28. Scalable QoS Multicast Provisioning in Diff-Serv-Supported MPLS Networks. Jun-Hong Cui, Jinkyu Kim, Aiguo Fei, Michalis Faloutsos y Mario Gerla.
"Definiciones del Multicast Aggregated Tree. Base para el desarrollo de la propuesta de la tesis."

29. MPLS - Multiprotocol Label Switching Architecture. E. Rosen, A. Viswanathan y R. Callon, Enero 2001.
"Características y temas a tener en cuenta en cuanto a la arquitectura de MPLS".

30. A Protocol to Improve the State Scalability of Source Specific Multicast. Jun-Hong Cui, Dario Maggiorini, Jinkyu Kim, Khaled Boussetta y Mario Gerla.
"Definiciones del Multicast Aggregated Tree"

31. Extensión de los métodos Hop-by-Hop, CR-LDP y RSVP-TE para Multicast IP sobre MPLS. Yezid Donoso Meisel.
"Definiciones y propuestas de implementación de IP Multicast sobre MPLS"

32. A Survey of QoS Multicast Issues. Aaron Stringel y G. Manimaran, IEEE Communicatios Magazine, Junio 2002.
"Especificaciones y definiciones sobre las características de QoS de los protocolos IP Multicast. Base para el desarrollo de la propuesta de la tesis."

33. Multicast Routing and Its QoS Extensión: Problems, Algoritms, and Protocols. Bin Wang y Jennifer C.Huo.
"Especificaciones y definiciones sobre las características de QoS de los protocolos IP Multicast. Base para el desarrollo de la propuesta de la tesis."

34. A QoS-Aware Multicast Routing Protocol. Shigang Chen, Klara Nahrstedt y Yuval Shavitt. *"Especificaciones y definiciones sobre las características de QoS de los protocolos IP Multicast. Base para el desarrollo de la propuesta de la tesis."*

35. A Scalable Distributed QoS Multicast Routing Protocol. Shigang Chen, Yuval Shavitt. *"Especificaciones y definiciones sobre las características de QoS de los protocolos IP Multicast. Base para el desarrollo de la propuesta de la tesis."*

36. QoS-Aware Multicast Routing for the Internet: The Desing and Evaluation of QoSMIC. Shuqian Yan, M. Faloutsos y A. Banerjea. *"Especificaciones y definiciones sobre las características de QoSMIC.Base para el desarrollo de la propuesta de la tesis."*

37. Received-Initiated Multicast with Multiple QoS Constraints. A. Fei y M. Gerla. *"Especificaciones y definiciones sobre las características de QoS de los protocolos IP Multicast. Base para el desarrollo de la propuesta de la tesis."*

38. Multimedia Services over the IP Multicast network, Antonio F. Gómez-Skarmeta, Angel L. Mateo, Pedro M. Ruiz; INFORMATIQUE 3/2001. *"Especificaciones y definiciones sobre las características de Servicios Multimedia sobre protocolos IP Multicast. Base para el desarrollo de la propuesta de la tesis."*